Experimental Evaluation of an Exact Algorithm for the Orthogonal Art Gallery Problem

Marcelo C. Couto, Cid C. de Souza[⋆], and Pedro J. de Rezende[⋆⋆]

Instituto de Computação
Universidade Estadual de Campinas — Campinas, Brazil
couto.marcelo@gmail.com, {cid,rezende}@ic.unicamp.br

Abstract. We consider the Orthogonal Art Gallery problem (OAGP) whose goal is to minimize the number of vertex guards required to watch an art gallery whose boundary is an n-vertex orthogonal polygon P. Here, we explore an exact algorithm for OAGP, which we proposed in [1], that iteratively computes optimal solutions to Set Cover problems (SCPs) corresponding to discretizations of P. While it is known [1] that this procedure converges to an exact solution of the original continuous problem, the number of iterations executed is highly dependent on the way we discretize P. Although the best theoretical bound for convergence is $\Theta(n^3)$ iterations, we show that, in practice, it is achieved after only a few of them, even for random polygons of hundreds of vertices. As each iteration involves the solution of an SCP, the strategy for discretizing P is of paramount importance. In this paper, we carry out an extensive empirical investigation with five alternative discretization strategies to implement the algorithm. A broad range of polygon classes is tested. As a result, we are able to significantly improve the performance of the algorithm, while maintaining low execution times, to the point that we achieve a fivefold increase in polygon size, compared to the literature.

1 Introduction

The classical *Art Gallery Problem* originally posed by Victor Klee in 1973 consists in determining the minimum number of guards sufficient to cover the interior of an n-wall art gallery [2]. Chvátal showed, in what became known as *Chvátal's Art Gallery Theorem*, that $\lfloor n/3 \rfloor$ guards are occasionally necessary and always sufficient to cover a simple polygon with n vertices [3].

Many variants of the art gallery problem have been studied in the literature. In this paper, we study the variation of the classical art gallery problem that deals specifically with orthogonal polygons (edges parallel to the x or y axis) where guards can only be placed on vertices that define the outer boundary of the gallery. This is called the *Orthogonal Art Gallery Problem* (OAGP) and

[⋆] Partially supported by CNPq – Conselho Nacional de Desenvolvimento Científico e Tecnológico – Grants # 307773/2004-3, 472504/2007-0 and FAPESP – Fundação de Amparo à Pesquisa do Estado de São Paulo – Grant # 107/97.

[⋆⋆] Partially supported by CNPq – Grant # 201205/2005-0, 472504/2007-0.

is an important subclass, due to most real life buildings and galleries being orthogonally shaped [4].

The earliest major result concerning this problem, due to Kahn *et al.* [5], states that $\lfloor \frac{n}{4} \rfloor$ guards are occasionally necessary and always sufficient to cover an orthogonal polygon with n vertices. Later, Schuchardt and Hecker proved that minimizing the number of guards in this variation is also NP-hard [6], settling a question that remained open for almost a decade [7].

Several placement algorithms have been proposed in the past, such as Edelsbrunner *et al.* [8] and Sack and Toussaint [7], which deal with the problem of efficiently placing exactly $\lfloor n/4 \rfloor$ guards covering a given orthogonal gallery.

On the other hand, in a recently revised manuscript, based on [9], Ghosh presents an $O(n^4)$ time approximation algorithm for simple polygons yielding solutions within a $\log n$ factor of the optimal. Further approximation results include Eidenbenz [10] who designed algorithms for several variations of terrain guarding problems and Amit *et al.* [11] who analyze heuristics with experimental evidence of good performance in covered area and in the number of guards.

Another approach tackled by Erdem and Sclaroff [12] and Tomás *et al.* in [13] consists of modeling the problem as a discrete combinatorial problem and then solving the corresponding optimization problem. The former discretize the interior of the polygon with a fixed grid, yielding an approximation algorithm and the latter gives empirical analysis of an exact method of successive approximations based on dominance of visibility regions.

Finally, in [1], we presented an exact algorithm to optimally solve the OAGP. In this algorithm, we iteratively discretize and model the problem as a classical *Set Cover problem* (SCP). Besides demonstrating the feasibility of this approach, we showed that, in practice, the number of iterations required to solve instances of up to 200 vertices was very small and that the resulting algorithm turned out to be quite efficient.

Our contribution. Though the number of iterations executed by the exact algorithm we proposed in [1] was shown to be polynomially bounded, its practical performance is much better depending on how the polygon is discretized. This becomes clearer when we notice that at each iteration an instance of SCP, a NP-hard problem, has to be solved at optimality, in our case, by an Integer Programming (IP) solver.

In this paper, we conduct a thorough experimental investigation concerning the trade-off between the number and nature of discretizing points and the number of iterations, analyzing the practical viability of each approach. Our test data, available in [14], includes multiple instances for each size of the vertex set, for various classes of orthogonal polygons with up to *a thousand* vertices.

The new experimental results significantly surpassed those we reported in [1]. This is due to the exploration of alternative discretization strategies, which allow us to address difficult instances as well as to handle a fivefold increase in the polygon size compared to the literature, while attaining low execution times.

Organization of the text. In the next section, we explain the basic ideas that support the algorithm. Section 3 is devoted to the description of the algorithm and the alternative strategies to discretize the polygon. Next, in Section 4 we give an account of the set up of the testing environment and present the different classes of instances used. Besides, following the recommendations of Johnson [15], McGeoch and Moret [16], Sanders [17] and Moret [18], we show an extensive experimental analysis of the algorithm implemented with multiple discretization strategies, including the evaluation of multiple measurements. Concluding remarks are drawn in the last section.

2 Basics

In an instance of the OAGP we are given an orthogonal simple polygon P that bounds an art gallery and we are asked to determine the minimum number and an optimal placement of vertex guards in order to keep the whole gallery under surveillance. Vertex guards are assumed to have a range of vision of $360\,^\circ$.

The approach used by the algorithm described in Section 3 transforms the continuous OAGP into a discrete problem which, in turn, can be easily modeled as an instance of the SCP. In fact, for the last two decades, this has been the only known technique for transforming art gallery problems leading to efficient approximation algorithms. Below, we describe in detail the approach used by the algorithm, starting with some basic definitions.

An *n-wall orthogonal art gallery* can be viewed as a planar region whose boundary consists of an orthogonal simple polygon (without holes) P, i.e., one whose n edges are parallel to the x or y axis. The set or vertices of P are denoted by V and a vertex $v \in V$ is called a *reflex vertex* if the internal angle at v is greater then $180\,^\circ$. Whenever no confusion arises, *a point in P* will mean a point either in the interior or on the boundary of P.

Any point y is said to be visible from any other point x if and only if the closed segment joining x and y does not intersect the exterior of P. The set $V(v)$ of all points visible from a vertex $v \in V$ is called the *visibility region of v*. In order to determine $V(v)$, we employ the linear time algorithm proposed by Lee [19] and extended by Joe and Simpson [20,21].

A set of points S is a *guard set* for P if for every point $p \in P$ there exists a point $s \in S$ such that p is visible from s. Hence, a *vertex guard set G* is any subset of vertices such that $\bigcup_{g \in G} V(g) = P$. In other words, a vertex guard set for P gives the positions of stationary guards who can oversee an entire art gallery of boundary P. Thus, the OAGP amounts to finding the smallest subset $G \subset V$ that is a vertex guard set for P.

From the above discussion one can see that the problem of finding the smallest vertex guard set for P can be seen as a *continuous* minimum set cover problem, where every visibility region $V(v), v \in V$ is a set and points $p \in P$ are elements of the set.

Notice that the term *continuous* is used here to denote the fact that there is an infinite number of elements to be covered in the SCP instance, as the points of P in the above definition comprise an infinite set. To cope with this, one can discretize the problem, generating a representative finite number of points in P so that the formulation becomes manageable. We now describe how the solutions to successively refined discrete instances are guaranteed to converge to an optimal solution to the original continuous problem. To this end, consider an arbitrary discretization of P into a finite set of points $D(P)$. An IP formulation of the corresponding SCP instance is shown below.

$$\begin{aligned} z = \min \sum_{j \in V} x_j & \\ \text{s.t.} \sum_{j \in V} a_{ij} x_j \geq 1, & \text{ for all } p_i \in D(P) \qquad (1) \\ x_j \in \{0,1\}, & \text{ for all } j \in V \end{aligned}$$

where the binary variable x_j is set to 1 if and only if vertex j from P is chosen to be in the guard set. Moreover, given a point p_i in $D(P)$ and a vertex j of P, a_{ij} is a binary value which is 1 if and only if $p_i \in V(j)$.

Given a feasible solution x for the IP above, let $Z(x) = \{j \in V \mid x_j = 1\}$. Constraint (1) states that each point $p_i \in D(P)$ is visible from at least one selected guard position in the solution and the objective function minimizes the cardinality z of $Z(x)$. Clearly, as the set $D(P)$ is finite, it may happen that $Z(x)$ does not form a vertex guard set for P. In this case, we must add a new point inside each uncovered region and include these points in $D(P)$. A new SCP instance is then created and the IP is solved again.

We are now able to describe the algorithm proposed in [1]. In the *preprocessing phase*, three procedures are executed. The first one computes the visibility polygons for the points in V. The second one computes the initial discretization $D(P)$ and the third one builds the corresponding IP model. In the *solution phase*, the algorithm iterates as described above, solving SCP instances for the current discretization, until no regions remain uncovered.

We had shown in [1] that an upper bound on the number of iterations is $O(n^4)$. This result was derived from the fact that the edges of the visibility regions induce a subdivision of P which is comprised of $O(n^4)$ faces or *Atomic Visibility Polygons* (AVPs). One point inside an AVP is enough to guarantee that this entire AVP will be covered by the solution to the discretized problem. Whence follows the upper bound on the number of iterations. However, it can be derived from a result by Bose *et al.* [22] that $\Theta(n^3)$ is a tight bound on the number of AVPs, improving the aforementioned worst case convergence result.

Moreover, the actual number of iterations that is required depends on how many uncovered regions can be successively generated. As the cost of each iteration is related to the number of constraints in (1), an interesting trade-off naturally sprouts and leads one to attempt multiple choices of discretization schemes. On the other hand, any method of cleverly choosing the initial points of the discretization will have a corresponding cost in preprocessing time, opening another intriguing time exchange consideration. These questions are precisely what we address next.

In Section 3 we consider several possible discretization schemes which lead to the various performance analysis discussed in Section 4.

3 Discretization Strategies

The key point in the IP approach is to set up instances of the set cover problem that can rapidly be solved while minimizing the number of iterations required to attain an optimal solution to the original art gallery problem, within the least amount of time. However, one must take into account that sophisticated geometric properties used to build more efficient discretizations will generate a corresponding cost in preprocessing, possibly outweighing the benefits. Below, we discuss alternatives for the discretization of P.

Regular Grid. The first discretization strategy considered is to generate a dense grid inside the polygon in the assumption that few iterations might be required. This was the main venue for the experimentations described in [1]. Such grid is built with a step of size equal to the smallest gap in the x- and y-coordinates of the vertices of P. We also include all the vertices in this initial discretization.

As it turns out, for some polygons the number of grid points grows quadratically with the number of vertices, inflating the number of constraints in the formulation of the SCP which increases the time needed to solve each instance.

A summary of the outcome of the use of regular grids for two classes of polygons can be seen in Figure 3 and Table 1.

Induced Grid. Given the perception that reflex vertices are responsible for part of the difficulty of the problem, a natural discretization strategy to be considered is the grid induced by the edge extensions that intersect in the polygon. In this case, we generate fewer constraints than in the previous strategy while attempting to capture more of the intrinsic visibility information of the polygon. One might expect that this could lead to faster to solve instances of set cover while keeping the number of iterations low.

Just Vertices. In one extreme, given that all vertices of the polygon will have to be covered, we consider the rather sparse case where the starting discretization contains just the vertices of the polygon. Initially, this leads to quicker solutions to the set cover problem than the two previous approaches and has the benefit that each additional constraint comes really from "hard to see" regions. Since in this way we avoid any spurious grid points, one might envision that the potentially higher number of iterations could still be compensated by the smaller size of the SCP instances.

Complete Atomic Visibility Polygons. Recall that an AVP of a polygon P is any (convex) face of the subdivision of P induced by the visibility polygons of all its vertices. It then follows that if a guard set G covers the centroid of an AVP, then it must cover the entire AVP. Therefore, if G covers the centroids of every AVP of P, then G must be a guard set for P.

This suggests that we could solve the problem in a single iteration of the algorithm by building an instance of SCP from all these centroids. However, for sizeable instances, this approach would lead to an impractically large instance of up to $O(|V|^3)$ constraints, where V is the set of vertices of P (see [22]).

Nonetheless, as we will see, not all AVPs need to be represented in the set of constraints in order to guarantee a single iteration. Therefore, we do not need to consider this more costly discretization strategy.

Reduced Atomic Visibility Polygons. Following the previous discussion, we now show that we can significantly reduce the number of constraints required to guarantee that the algorithm will find the minimum number of guards necessary to cover P by solving a single instance of the set cover problem.

Firstly, given a vertex $v \in V$, an edge of the visibility polygon $V(v)$ is called a *visibility edge* of v. Furthermore, if it is not an edge of P, then it is called a *proper visibility edge* of v. It follows that an AVP is a face in the arrangement of visibility edges, interior to P. Hence, the edges of an AVP are either portions of edges of P or portions of proper visibility edges of vertices of P. An AVP $\mathcal{V}$ is called a *shadow AVP* if it is not visible from any of the vertices whose proper visibility edges spawn $\mathcal{V}$.

Let $G \subset V$ be a partial guard set for P and let U be a maximal connected region not covered by G. Note that U can be partitioned into a collection of AVPs. To see that at least one of these must be a shadow AVP, notice that if one side of a proper visibility edge of, say, vertex v_i that intersects U, is visible from v_i then the opposite side must not be. Hence, by successive partitioning U, at least one shadow AVP is bound to remain.

The Reduced AVP discretization strategy consists of taking all vertices of P plus the centroids of every shadow AVP. Since any guard set that covers all the points of this discretization cannot leave an uncovered region, it follows that no iterations will be required.

It remains to be experimentally analyzed which of these discretizing strategies will bring about the most benefit, timewise. This is done in the next section.

4 Computational Experiments

We now present an experimental evaluation of the several discretization strategies discussed in the previous section. We coded all variants of the algorithm described in earlier sections along with a visibility algorithm from [20]. The implementation was done in `C++`, compiled with `GNU g++ 4.1`, on top of `CGAL 3.2.1`, and used the IP solver `Xpress v17.01.02`. As for hardware, we used a desktop PC featuring a Pentium IV at 3.4 GHz and 1 GB of RAM running `GNU/Linux 2.6.17`.

4.1 Instances

We conducted the tests on a large number of instances downloadable from [14] and grouped into four different classes (see Figure 1). The first two of these classes are composed of n-vertex orthogonal polygons placed on an $n/2 \times n/2$ unit square grid and devoid of collinear edges, as suggested in [23] and the last two are based on a modified version of the von Koch curve (see [24]).

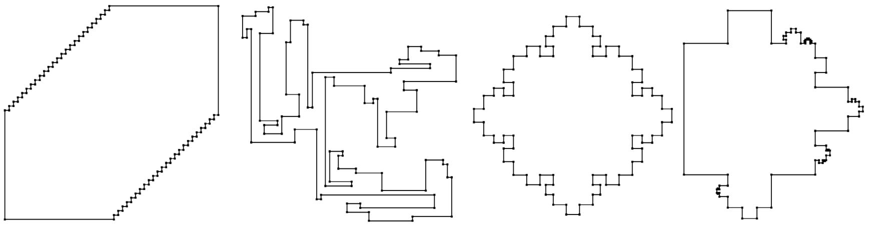

Fig. 1. Sample polygons with 100 vertices: FAT, Random, Complete von Koch and Random von Koch

(1) **FAT:** This class was introduced in [13] as an extreme scenario for the IP approach and also used in [1], where instances with up to 200 vertices were solved to optimality.
(2) **Random:** These are n-vertex randomly generated orthogonal polygons created using the algorithm proposed in [23].
(3) **Complete von Koch (CvK):** These polygons were generated based on a modified version of the von Koch curve. The fractal has a Hausdorff dimension of 1.34 and is generated, starting with a square, by recursively replacing each edge as shown in Figure 2, where $\overline{ar} = \overline{st} = \overline{ub}$ and $\overline{sr} = \overline{tu} = \frac{3}{4}\overline{ar}$.

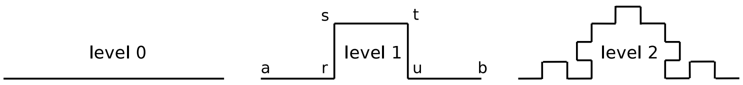

Fig. 2. Levels of modified von Koch polygons

(4) **Random von Koch (RvK):** This class consists of randomized von Koch instances of up to level 4. Starting from a square, each of these instances is generated iteratively until the desired number of vertices is reached. In each iteration, we randomly choose an edge of the current polygon, with level smaller than 4, and decide in a random fashion whether we expand it or not.

The FAT and Random instances were generated for the number of vertices n in the ranges: [20, 200] with step 20, (200, 500] with step 50 and (500, 1000] with step 100. Similar sizes were chosen for the RvK class. The CvK class contains by construction only 3 instances with $n \in \{20, 100, 500\}$.

For our conclusions to be endowed with statistical significance, we had to decide on the sample size (number of instances generated), for each value of n, in the classes Random and RvK. To this end, we ran our algorithm on random instances, while varying the sample size s. We concluded that the variance of the results remains practically unchanged after $s \geq 30$ and, therefore, we decided to generate 30 instances for each value of n. It is worth noting that, up to scaling, only one instance is defined for a given n in the FAT class, hence no decision on sample size is needed in this case.

Thus, in total, our data set is composed of 1833 OAGP instances, having between 20 and 1000 vertices, i.e., our largest instances are five times the largest ones whose optimal solutions are reported in the literature.

4.2 Results

We now discuss the experimental evaluation of the different strategies described in Section 3. All values reported here are average results for 30 instances of each size, or 30 runs of the same instance, for FAT and CvK classes.

The FAT instances were introduced in [13] as an extremal scenario for the IP approach because of the larger number of constraints resulting from regular discretizations of P. Figure 3 displays the amount of time spent by the exact algorithm on the FAT class with each discretization strategy. It can be seen that there is a huge difference between the strategies, though all the discretizations lead to a solution in only one iteration. Notice that, in this case, the Regular and Induced Grids coincide, leading to the same running times. On the other hand, the *Reduced AVP* and *Just Vertices* discretizations are both composed of only the vertices of P, since FAT polygons have no shadow AVPs. Of course, the *Reduced AVP* strategy spent more time on the preprocessing phase, which causes the difference seen in the chart. However, the two strategies can deal with FAT polygons with up to 1000 vertices in reasonable time, going far beyond the results reported earlier for this class which are limited to 200-vertex polygons.

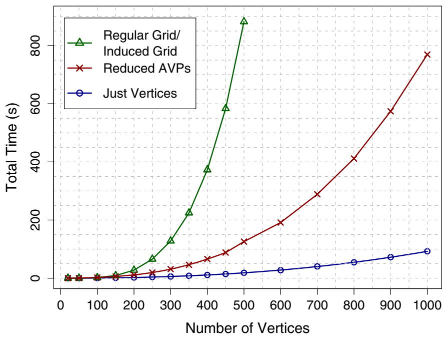

Fig. 3. Total time: FAT polygons

Table 1. Complete von Koch polygons

	Final $\|D(P)\|$			Total Time (s)		
# vertices	20	100	500	20	100	500
Reg. Grid	45	500	6905	0.05	1.57	92.37
Ind. Grid	24	205	1665	0.03	1.41	70.94
Red. AVPs	28	324	5437	0.07	3.14	143.93
Just Vert.	20	107	564	0.04	0.97	29.35

The usage of discretization strategies based on dense grids becomes more discouraging when we analyze the results in Table 1. This table displays the execution time and the size of the discretization of the strategies proposed in Section 3 for the CvK polygons. One can see that for these instances, the *Induced Grid* strategy has a better performance than the *Regular Grid* strategy. The size of the discretization produced by *Regular Grid* grows quadratically in the number of vertices, and thus inflates the number of constraints in the IP formulation increasing considerably the time necessary to optimally solve the SCP instances.

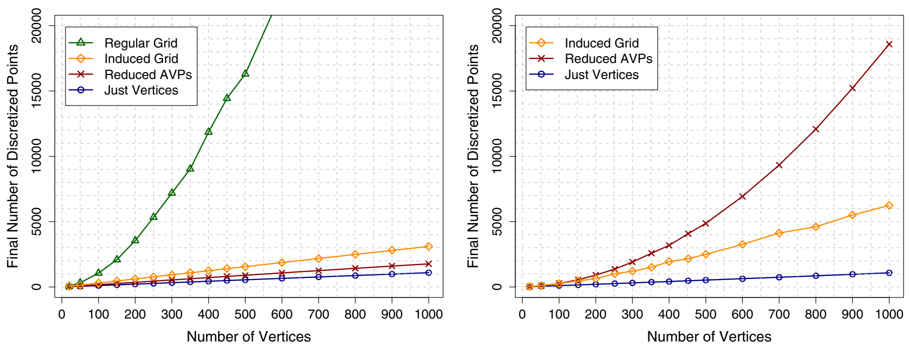

Fig. 4. Final discretization size: (a) Random polygons; (b) Random von Koch polygons

The *Reduced AVP* strategy has a poor behavior for CvK polygons since the number of shadow AVPs increases fast in this case. The *Just vertices* strategy is again the one that spends less time.

Figure 4 shows the amount of discretized points necessary for each strategy to achieve the optimal solution of OAGP for Random (in (a)) and RvK (in (b)) polygons. Especially from the Random case, one can see that the *Regular Grid* strategy rapidly becomes impractical due to the huge size of the discretization and, therefore, will no longer be analyzed for other classes of polygons. On the other hand, one can see that the *Reduced AVP* strategy still follows the same behavior of the CvK case for RvK instances, with the discretization size growing fast as the number of vertices of P increases. Nevertheless, this approach is very well-suited for random polygons. The curves corresponding to the *Just Vertices* strategy suggest that the set of vertices of the polygon is a good guess for the initial discretization since few new points are added to it to achieve the optimal solution of an OAGP instance for both classes of instances.

Figure 5 shows the number of iterations each strategy needs to achieve the optimal solution for both classes of random polygons. The chart in (a) displays

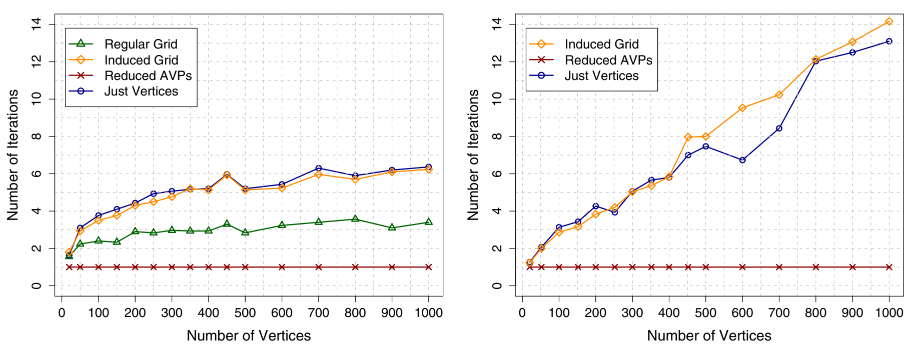

Fig. 5. Number of iterations: (a) Random polygons; (b) Random von Koch polygons

the expected behavior with the number of iterations increasing as the size of the discretizations decrease. Now, relative to the size of the input polygon, the number of iterations remains negligible when compared to the theoretical bound of $\Theta(n^3)$. In chart (b) relative to RvK polygons, the number of iterations increases a bit faster with the instance size but is still small. Somewhat surprisingly, in this case *Induced Grid* iterates slightly more than the *Just vertices* strategy.

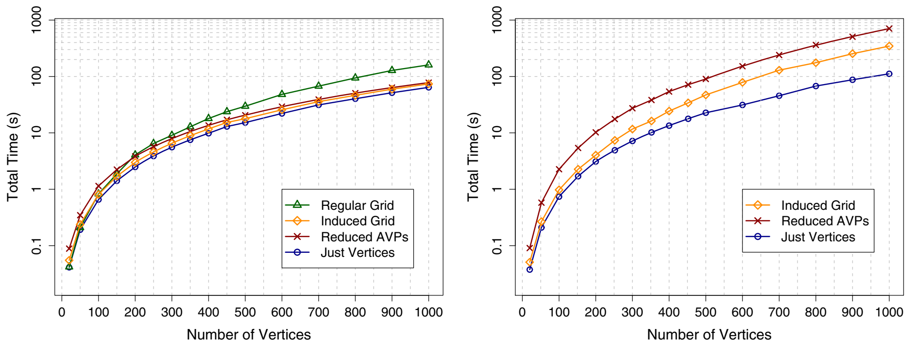

Fig. 6. Total time: (a) Random polygons; (b) Random von Koch polygons

Figure 6 shows the total amount of time, including the preprocessing and processing phases, to solve instances from the random classes. Notice that the curves are plotted in log ×linear format and both charts are in the same scale. One can see that for Random polygons, all the strategies behave similarly except, as expected and explained before, the *Regular Grid*. The tendency of *Just vertices* strategy is very similar in both classes of polygons. This shows that, though we are solving harder instances in the RvK case, the strategy is robust.

We now turn our attention to the time spent by the algorithm in each phase for the discretization strategies. Recall that the preprocessing phase is composed of three procedures. The first one is common to all strategies and computes the visibility polygons. The second one computes the initial discretization and its cost is highly affected by the choice of the strategy to be implemented. The worst case corresponds to the *Reduced AVP* strategy since it requires the computation of all AVPs and the determination of the shadow AVPs and of their centroids. On the other extreme, we have the *Just vertices* strategy where no computation is needed. Finally, in the third procedure of the preprocessing phase one has to build the starting IP model and the time spent in doing so depends on the size of the discretization, which again benefits the *Just vertices* strategy.

In Figure 7 one can see that the time spent in the preprocessing phase is in accordance with the discussion above, the *Reduced AVP* strategy being the most time consuming for RvK. What is somehow surprising is that, though we are solving NP-hard problems in the solution phase, the majority of the time consumption refers to the preprocessing phase, which is entirely polynomial. The extraordinary developments in IP solvers together with the fact the SCP instances arising from OAGP are among the easy ones can explain this apparently counter intuitive behavior of the algorithm. Thus, for the *Reduced AVP* strategy to become competitive, a cleverer and faster procedure has to be developed to discard not only shadow AVPs but other ones. Comparing the size of the final discretizations of the different strategies shown earlier there seems to be room for such improvements.

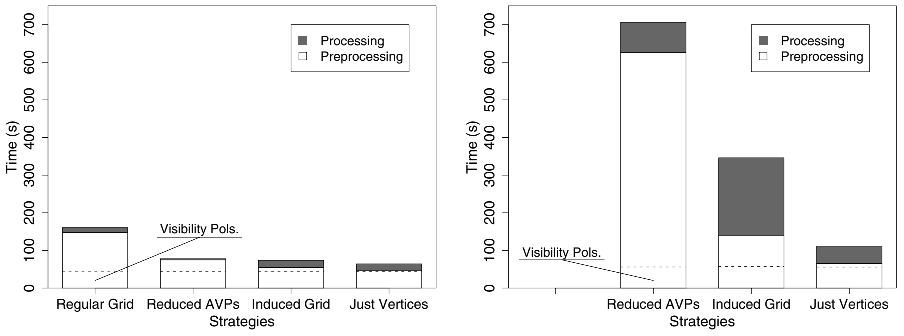

Fig. 7. Execution time for polygons of 1000 vertices: (a) Random polygons; (b) Random von Koch polygons. The lower part of the preprocessing time corresponds to the construction of the visibility polygons.

5 Conclusions and Remarks

In this paper, we conducted an experimental investigation of an exact algorithm for the Orthogonal Art Gallery problem (OAGP) proposed in [1] which relies on the discretization of the interior of the input polygon P and on the modeling of this simplified discrete problem as a Set Cover problem (SCP). The resulting SCP instance is solved to optimality by an IP solver and, if uncovered regions remain, additional constraints are included and the process is repeated. Clearly, the performance of the algorithm depends on the number of such iterations.

This work focused on different strategies to implement the discretization of P. Thorough experimentation was carried out to assess the trade-off between the number of iterations and time spent by the many variants of the algorithm that arise from the alternative discretization methods.

Our conclusion is that this exact algorithm is a viable choice to tackle instances of the OAGP, in light of the fact that the largest ones we solved were five times larger than those reported earlier in the literature.

The apparent advantage of a discretization which ensures an exact solution after a single iteration of the algorithm (like the *Reduced AVP* strategy) did not prove to be effective in practice. This became even clearer when we compared its results with those of the *Just Vertices* strategy which, represents the opposite extreme situation, as, in principle, it starts with the smallest "natural" SCP instance. However, as one can see from Table 2 this strategy leads to a very fast

Table 2. Total Time (seconds): *Just Vertices* strategy

	Polygons Classes			
n	Random	FAT	RvK	CvK
100	0.65	0.58	0.73	0.97
500	15.21	18.47	22.73	29.35
1000	64.13	92.41	111.55	$\nexists$

implementation that takes only few seconds of CPU time to solve OAGP instances with up to 1000 vertices.

The success of this exact algorithm clearly benefits from the extraordinary developments in IP solvers in recent years, which lead to the solution of large instances of SCP in a very small amount of time. Therefore, we believe that the *Reduced AVP* strategy can only become competitive with the *Just Vertices* strategy when the preprocessing time required by the former is significantly reduced. Though we used powerful data structures and packages to perform the necessary geometrical operations, we could not significantly lessen the preprocessing time which, for the largest instances tested here, correspond roughly to the time required by the IP solver to resolve ten instances of SCP.

A promising venue of further investigation lies in trying to identify inexpensive geometric properties that might lead to a set of constraints that capture the essence of the hardness of the problem, such as a significant reduction on the number of AVPs.

References

1. Couto, M.C., de Souza, C.C., de Rezende, P.J.: An exact and efficient algorithm for the orthogonal art gallery problem. In: Proc. of the XX Brazilian Symp. on Comp. Graphics and Image Processing, pp. 87–94. IEEE Computer Society, Los Alamitos (2007)
2. Honsberger, R.: Mathematical Gems II. Number 2 in The Dolciani Mathematical Expositions. Mathematical Association of America (1976)
3. Chvátal, V.: A combinatorial theorem in plane geometry. Journal of Combinatorial Theory Series B 18, 39–41 (1975)
4. Urrutia, J.: Art gallery and illumination problems. In: Sack, J.R., Urrutia, J. (eds.) Handbook of Computational Geometry, pp. 973–1027. North-Holland, Amsterdam (2000)
5. Kahn, J., Klawe, M.M., Kleitman, D.: Traditional galleries require fewer watchmen. SIAM J. Algebraic Discrete Methods 4, 194–206 (1983)
6. Schuchardt, D., Hecker, H.D.: Two NP-hard art-gallery problems for ortho-polygons. Mathematical Logic Quarterly 41, 261–267 (1995)
7. Sack, J.R., Toussaint, G.T.: Guard placement in rectilinear polygons. In: Toussaint, G.T. (ed.) Computational Morphology, pp. 153–175. North-Holland, Amsterdam (1988)
8. Edelsbrunner, H., O'Rourke, J., Welzl, E.: Stationing guards in rectilinear art galleries. Comput. Vision Graph. Image Process. 27, 167–176 (1984)

9. Ghosh, S.K.: Approximation algorithms for art gallery problems. In: Proc. Canadian Inform. Process. Soc. Congress (1987)
10. Eidenbenz, S.: Approximation algorithms for terrain guarding. Inf. Process. Lett. 82(2), 99–105 (2002)
11. Amit, Y., Mitchell, J.S.B., Packer, E.: Locating guards for visibility coverage of polygons. In: Proc. Workshop on Algorithm Eng. and Experiments, pp. 1–15 (2007)
12. Erdem, U.M., Sclaroff, S.: Automated camera layout to satisfy task-specific and floor plan-specific coverage requirements. Comput. Vis. Image Underst. 103(3), 156–169 (2006)
13. Tomás, A.P., Bajuelos, A.L., Marques, F.: On visibility problems in the plane - solving minimum vertex guard problems by successive approximations. In: Proc. of the 9th Int. Symp. on Artificial Intelligence and Mathematics (2006)
14. Couto, M.C., de Souza, C.C., de Rezende, P.J.: OAGPLIB - Orthogonal art gallery problem library, `www.ic.unicamp.br/~cid/Problem-instances/Art-Gallery/`
15. Johnson, D.S.: A theoretician's guide to the experimental analysis of algorithms. In: M.H.G., et al. (eds.) Data Structures, Near Neighbor Searches, and Methodology: Fifth and Sixth DIMACS Implem. Challenges, AMS, Providence, pp. 215–250 (2002)
16. McGeoch, C.C., Moret, B.M.E.: How to present a paper on experimental work with algorithms. SIGACT News 30 (1999)
17. Sanders, P.: Presenting data from experiments in algorithmics, pp. 181–196. Springer, New York (2002)
18. Moret, B.: Towards a discipline of experimental algorithmics. In: Proc. 5th DIMACS Challenge
19. Lee, D.T.: Visibility of a simple polygon. Comput. Vision, Graphics, and Image Process 22, 207–221 (1983)
20. Joe, B., Simpson, R.B.: Visibility of a simple polygon from a point. Report CS-85-38, Dept. Math. Comput. Sci., Drexel Univ., Philadelphia, PA (1985)
21. Joe, B., Simpson, R.B.: Correction to Lee's visibility polygon algorithm. BIT 27, 458–473 (1987)
22. Bose, P., Lubiw, A., Munro, J.I.: Efficient visibility queries in simple polygons. Computational Geometry 23(3), 313–335 (2002)
23. Tomás, A.P., Bajuelos, A.L.: Generating random orthogonal polygons. In: Conejo, R., Urretavizcaya, M., Pérez-de-la-Cruz, J.-L. (eds.) CAEPIA/TTIA 2003. LNCS (LNAI), vol. 3040, pp. 364–373. Springer, Heidelberg (2004)
24. Falconer, K.: Fractal Geometry, Mathematical Foundations and Applications, pp. 120–121. John Wiley & Sons, Chichester (1990)