Journal of Universal Computer Science, vol. 18, no. 14 (2012), 1906-1932
submitted: 22/12/10, accepted: 13/7/12, appeared: 28/7/12 © *J.UCS*

A Hybrid Metaheuristic Strategy for Covering with Wireless Devices

Antonio L. Bajuelos
(University of Aveiro, Portugal
leslie@ua.pt)

Santiago Canales
(Universidad Pontifícia Comillas de Madrid, Spain
scanales@dmc.icai.upcomillas.es)

Gregorio Hernández
(Universidad Politécnica de Madrid, Spain
gregorio@fi.upm.es)

Mafalda Martins
(University of Aveiro, Portugal
mafalda.martins@ua.pt)

Abstract: In this paper we focus on approximate solutions to solve a new class of Art Gallery Problems inspired by wireless localization. Instead of the usual guards we consider wireless devices whose signal can cross a certain number, k, of walls. These devices are called k-transmitters. We propose an algorithm for constructing the visibility region of a k-transmitter located on a point of a simple polygon. Then we apply a hybrid metaheuristic strategy to tackle the problem of minimizing the number of k-transmitters, located at vertices, that cover a given simple polygon, and compare its performance with two pure metaheuristics. We conclude that the approximate solutions obtained with the hybrid strategy, for 2-transmitters and 4-transmitters, on simple polygons, monotone polygons, orthogonal polygons and monotone orthogonal polygons, are better than the solutions obtained with the pure strategies.

Key Words: Computational Geometry, Art Gallery Problems, Visibility and Coverage Problems, Hybrid Metaheuristics and Approximation Algorithms.
Category: G.1.6, I.2.8

1 Introduction

Geometric optimization problems related to visibility are very interesting given their applicability in several areas such as computer graphics (e.g., [Dobkin, Teller 2004]), pattern recognition (e.g., [O'Rourke, Toussaint 2004]) and robotics (e.g., [Wein et al. 2006]). In Computational Geometry these problems sprung from an Art Gallery Problem posed by Victor Klee in 1973: determining the minimum number of guards that are sufficient to cover the interior of an n-wall art gallery room. Chvátal showed that $\lfloor \frac{n}{3} \rfloor$ guards are occasionally necessary and always sufficient to guard a simple polygon with n vertices [Chvátal 1975]. By simple polygon we mean a region on the plane enclosed by a simple cycle of line segments and its interior is topologically equivalent to a disk. According to the original definition, two points p and q on a simple polygon P are said to be visible if the line segment joining p and q does not cross the exterior of P. Since the publication of Chvátal's result, many researchers have studied several variants of visibility. In 1987, J. O'Rourke [O'Rourke 1987] published the first book devoted to these problems. Other two survey papers were also written on this subject, one in 1992 by T. Shermer [Shermer 1992] and, the second one, in 2000 by J. Urrutia [Urrutia 2000]. Since then, a large number of papers in this area have been published and some important problems have been solved.

The development of the wireless networks paved the way for further research on visibility in a different way. For example, [Aichholzer et al. 2009 (a)] defined a new variant of the original Art Gallery Problem, which arises from a practical, everyday problem: How to place wireless routers in a building in such a way that a computer placed anywhere within the building receives a strong enough signal to guarantee a stable Internet connection? There are two main limitations when trying to connect a computer to a wireless network: its distance to the wireless router and the number of walls that separate it from the router. However, in many buildings, the most significant limiting factor is the number of walls that separate the computer from the wireless router and not the distance between them. The communications in wireless networks, where the signals are blocked by walls, was the motivation that encouraged [Aichholzer et al. 2009 (a)] to study the k-transmitter Art Gallery Problem: Given n, what is the smallest number of k-transmitters that is sufficient to cover any polygon with n vertices? A k-transmitter refers to a wireless router whose signal can cross k walls. It is said that a point y in a simple polygon P is covered or illuminated by a k-transmitter placed on a point $x \in P$ if the line segment $\overline{xy}$ crosses at most k walls (edges) of P. It is easy to observe that (i) for $k = 0$ this problem is reduced to the Art Gallery Problem and (ii) for $k = n$ just one n-transmitter placed on any point is sufficient to cover P (trivial solution). [Aichholzer et al. 2009 (a)] obtained combinatorial bounds for monotone polygons and monotone orthogonal polygons. They proved that every monotone polygon with n vertices can be covered with $\lceil \frac{n}{2k} \rceil$ k-transmitters, and there is a monotone polygon n-vertex polygon that requires at least $\lceil \frac{n}{2k+3} \rceil$ k-transmitters to be covered [1]. These authors also proved that $\lceil \frac{n}{2k+4} \rceil$ k-transmitters are always sufficient and sometimes necessary to cover a monotone orthogonal polygon. If the k-transmitter are restricted to the vertices of P (*vertex k-transmitters*), the implicit assumption is that the transmitter is placed just inside the polygonal region, and so must penetrate one wall to reach the exterior. The results established above remain valid to vertex

[1] personal communication with the authors

k-transmitters. However, the problem for simple polygons and orthogonal polygons remains open. In a more recent paper, [Fabila-Monroy et al. 2009] studied this notion of visibility for other geometric configurations. For example, they studied the problem of determining the number of k-transmitters that is always sufficient to cover any arrangement with n lines on the plane. They also studied the problem of covering simple polygons, orthogonal arrangements of lines and orthogonal polygons using a few transmitters with high power. Among other results, they proved that any simple polygon with n vertices can be covered with a transmitter of power $\lceil \frac{n}{2k+3} \rceil$ and this bound is tight up to an additive constant. While the center of attention in [Aichholzer et al. 2009 (a), Fabila-Monroy et al. 2009] is on finding a small number of high power transmitters, [Ballinger et al. 2010] are focused on lower power transmitters. For instance, they proved that $\frac{n}{6}$ 2-transmitters are sometimes necessary to cover a simple polygon and that to cover the plane in the presence of n disjoint orthogonal line segments, $\lceil \frac{5n+6}{12} \rceil$ 1-transmitters are sufficient and $\lceil \frac{n+1}{4} \rceil$ are sometimes necessary. In [Aichholzer et al. 2009 (b)] the notion of k-convexity is introduced and studied. A polygon P is k-convex if every line segment with endpoints in P crosses at most $2(k-1)$ edges of P. Thus, a 2-convex polygon can be covered by a 2-transmitters placed anywhere in P.

Since all the previous results are combinatorial, in this paper we are interested in algorithmic results regarding simple polygons. The results presented by [Aichholzer et al. 2009 (a)] are combinatorial solutions that solve the problem to most polygons with n vertices. However, not all polygons with n vertices require the established number of k-transmitters and can be covered with less. This reasoning justifies the following algorithmic problem: Given a n-vertex polygon P determine the minimum number of k-transmitters that cover P. Since the problem of finding the minimum number of guards (i.e., 0-transmitters) that cover a given simple polygon is NP-hard ([Aggarwal 1984] and [Lee, Lin 1986]), it is strongly believed that this problem is also NP-hard, both for simple and orthogonal polygons (for $0 < k < n$) [Aichholzer et al. 2009 (a)]. Therefore it makes sense to tackle this problem using approximation algorithms. In general, these approximation methods can be designed specifically to solve a problem (e.g., greedy strategies) or can be based on general metaheuristics (e.g., simulated annealing and genetic algorithms). A metaheuristic is a general algorithmic framework that can be adapted to different optimization problems with relatively few modifications. For a comprehensive survey on metaheuristics see, e.g., [Blum, Rolli 2003, Glover, Kochenberger 2003]. There are several works where approximation algorithms were developed to tackle art gallery problems, for instance [Amit et al. 2010, Eidenbenz et al. 2001, Ghosh 2010, Packer 2008, Tomás et al. 2003]. For the special case of vertex guards [Couto et al. 2011] have recently developed an exact method that is based on a set-cover approach.

[Abellanas et al. 2006] applied metaheuristic techniques for solving some variants of visibility problems. Following these ideas [Bajuelos et al. 2009] and [Martins 2009] obtained good results on the application of metaheuristic techniques. In particular, these authors studied the following problems: calculating the minimum number of vertex-guards that cover a polygon and estimating the maximum hidden vertex set in polygons.

In this paper we study the problem of minimizing the number of vertex k-transmitters that cover a given simple polygon. In the next section we formalize the problem and state some useful results and definitions. In section 3 we describe a new algorithm to construct the region covered by a k-transmitter located on a point of a simple polygon with n edges, for all the possible values of k ($0 < k < n$). In section 4 we discuss approximation methods designed to solve approximately the problem of minimizing the number of k-transmitters: a hybrid approach that uses both the general metaheuristics Genetic Algorithms (GAs) and Simulated Annealing (SA) and two other pure approaches based on the general metaheuristics GAs and SA. In section 5 we describe the choice of the metaheuristics parameters, we present the experimental results obtained with the three methods, for $k = 2$ and $k = 4$, on randomly generated polygons and compare the performance of the hybrid metaheuristic with the other two approximation methods. Furthermore, we also use the least squares method to determine, approximately, the average number of 2-transmitters and 4-transmitters that cover a given simple polygon, obtained from the conducted experiments. Finally, in section 6 we present some conclusions.

2 Problem Description

A simple polygon P is a region of the plane enclosed by a finite collection of straight line segments forming a simple cycle. Non-adjacent segments do not intersect and two adjacent segments intersect only at their common endpoint. These intersection points are the vertices of P and the line segments are the edges of P. The interior and the boundary of P are denoted by $int(P)$ and $bd(P)$, respectively. This paper only focuses on simple polygons, and therefore we call them just polygons throughout the paper. Every polygon P with n vertices (or n-vertex polygon) is well defined by the sequence of its vertices $v_1, \ldots, v_n$, given in counterclockwise (CCW) order. In this way, it is easy to see that P is located to the left (positive side) of any edge traversed from v_i to v_{i+1}. Orthogonal polygons are those whose edges are parallel to the axes.

As stated above, there are two main limitations when trying to connect a computer to a wireless network: its distance to the wireless router and the number of walls between the computer and the router. In a first approach, we only consider the number of walls that separates the computer from the router. So, first of all, it is necessary to define when a computer located on point $y \in P$ is covered or illuminated by a wireless router placed on a point $x \in P$.

Definition 1. Let P be an n-vertex polygon. A wireless router, located on a point $x \in P$, which transmits a stable signal through at most k edges (walls) of P along a straight line is denoted by *k-transmitter* [Aichholzer et al. 2009 (a)].

Definition 2. Let P be an n-vertex polygon and $k \in \{0, \ldots, n\}$. A point $y \in P$ is *covered* by a k-transmitter placed on a point $x \in P$ if the line segment $\overline{xy}$ crosses at most k edges (walls) of P (see Fig. 1(a)). That is, y is covered by a k-transmitter placed on x if the line segment $\overline{xy}$ intersects the relative interior of the edges of P at most k times.

Definition 3. Let P be an n-vertex polygon. The *k-transmitter visibility region* of a k-transmitter placed on a point $x \in P$ is the set of all points $y \in \mathbb{R}^2$ that are covered by x. This set is denoted by $Vis_k(x, P)$, where x is a k-transmitter[2].

Figure 1(b) illustrates the visibility region of a 2-transmitter. Note that this region can be unbounded in some cases.

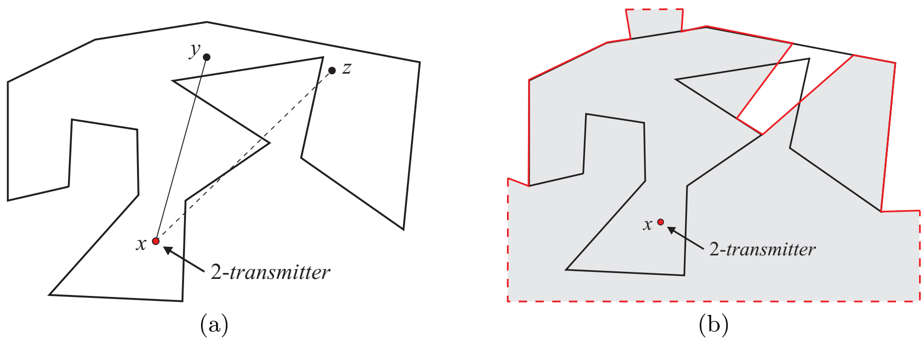


Figure 1: (a) *The 2-transmitter placed on x covers y but it does not cover z*; (b) $Vis_2(x, P)$.

Now, we can state our main problem: *Given an n-vertex polygon P, what is the minimum number of k-transmitters (placed on points of P) that cover P?*

Let P be a polygon with n vertices and $g_{km}(P)$ be the smallest number of k-transmitters that cover P, that is, $g_{km}(P) = \min\{|S| : S \subset P, P \subseteq \bigcup_{x \in S} Vis_k(x, P)\}$. Let $G_{km}(n)$ be the maximum of $g_{km}(P)$ over all polygons with n vertices, i.e., $G_{km}(n) = \max\{g_{km}(P) : P \in \mathcal{P}_n\}$, where $\mathcal{P}_n$ denotes the set of all n-vertex polygons. Thus, $G_{km}(n)$ k-transmitters always suffice to cover any n-vertex polygon and are necessary to cover at least one n-vertex polygon.

[2] For simplification purposes, sometimes the expression"x is a k-transmitter", $x \in P$, is used instead of "a k-transmitter placed on a point x".

So, the combinatorial problem asks for $G_{km}(n)$. As stated before, the first combinatorial results for the problem were obtained by [Aichholzer et al. 2009 (a)], and can be applied to monotone polygons (polygons whose intersection with a vertical line is connected - a point, a line segment or an empty set). However, to the best of our knowledge, there are not any combinatorial results related to non-monotone, simple and orthogonal, polygons. On the other hand, it is strongly believed that the algorithmic problem of finding the minimum number of vertex k-transmitters that cover a given polygon is NP-hard, for $0 < k < n$ [Aichholzer et al. 2009 (a)]. This brings us to a new variant of the k-transmitter Art Gallery problem that we designate by MINIMUM VERTEX k-TRANSMITTER SET PROBLEM, MVkT(P, k), and can be stated as:

MVkT(P, k)
Input: A polygon P with n vertices and a number k $(0 < k < n)$ of walls.
Question:What is the minimum number of vertex k-transmitters that cover P?

This paper proposes a hybrid metaheuristic strategy to tackle this problem and compares its performance with two pure metaheuristics. However, before applying any approximate method, the first step to solve the MVkT(P, k) problem is to construct $Vis_k(x, P)$. For classical visibility there exists a linear time algorithm to calculate the visibility region of $x \in P$, $Vis(x, P)$, [Joe, Simpson 1985]. However, an algorithm to determine the region covered by a k-transmitter located at a point $x \in P$, $Vis_k(x, P)$, is unknown to this date. This means that our first challenge is to develop an algorithm to determine this region.

3 k-Transmitter Visibility Polygon

Let P be a polygon with n vertices $V_P = \{v_0, v_1, \ldots, v_{n-1}\}$ and x a point on P where a k-transmitter is placed. In this section we describe an algorithm to construct the region covered by the k-transmitter placed on x. This region is composed of interior and exterior areas of P. For simplicity reasons, we assume that P is contained in a rectangular box R and the visibility region is constructed inside R. In this way, the region covered by x is always bounded and we call it *k-transmitter visibility polygon* and, abusing a bit of the terminology, we denote it by $Vis_k(x, P)$. A vertex $v_i \in V_P$ is called a *critical vertex* for x if the vertices $v_{i-1} \in V_P$ and $v_{i+1} \in V_P$ are on the same half-plane regarding the ray $\overrightarrow{xv_i}$ (see Fig. 2). Now we will describe the algorithm to construct $Vis_k(x, P)$ for all admissible values for k, that is, $0 < k < n$. Although the following collinearity cases arise: (i) x is collinear with two (or more) critical vertices of P and (ii) x belongs to a straight line containing an edge of P; and we have addressed them in our work. We will not include them in the overall description, because they are *degenerate* cases. In the following we present the main steps of the algorithm:

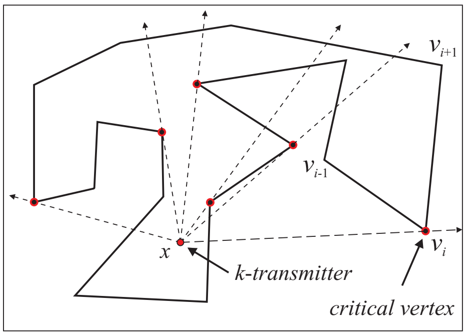


Figure 2: *Critical vertices for x.*

(1) Find the critical vertices and sort these points angularly around x in counter-clockwise order. Draw all the rays with source x passing through the critical vertices of P.

(2) Using the critical vertices, calculate the intersection points between the rays and the edges of P (see Fig. 3(a)). The rays divide each edge of the polygon in one or more segments. Label each segment with the number of edges (walls) crossed by the ray from x to the segment. Repeat this method to label the boundary of the rectangular box (see Fig. 3(b)).

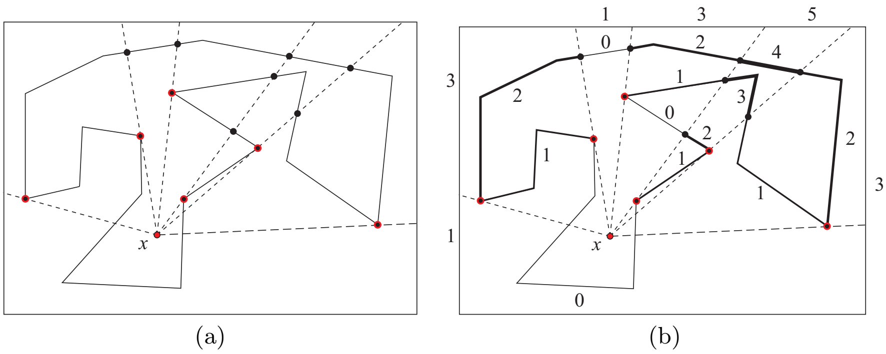


Figure 3: (a) *Rays and intersection points;* (b) *Labeled segments.*

(3) $Vis_k(x, P)$ is constructed by connecting all the segments with the same label k by angularly sweeping the incident rays. For this, take a segment, $\overline{s_1 s_2}$, with a label k, where $s_1 < s_2$ with respect to the angular sorting around x. At point s_1 start drawing the boundary of $Vis_k(x, P)$ in CCW order. Advance along $bd(P)$ (if k is even continue in CCW order, otherwise continue in CW order) until a segment with a different label k' is reached. If $k' > k$ then continue along the ray incident to the starting point of the new segment in

direction to x until a segment with a label k is found. Otherwise if $k' < k$ then move along the same ray but in the opposite direction. Observe that if another segment labeled k does not exist, we will hit the boundary of the rectangular box. In this case, follow the boundary until the first segment with label greater than k is reached. Repeat this procedure until $Vis_k(x, P)$ is finished.

Fig. 4(a) shows how the algorithm connects the segments labeled with 2 in order to build the polygon covered by a 2-transmitter. Fig. 4 (b) illustrates $Vis_2(x, P)$.

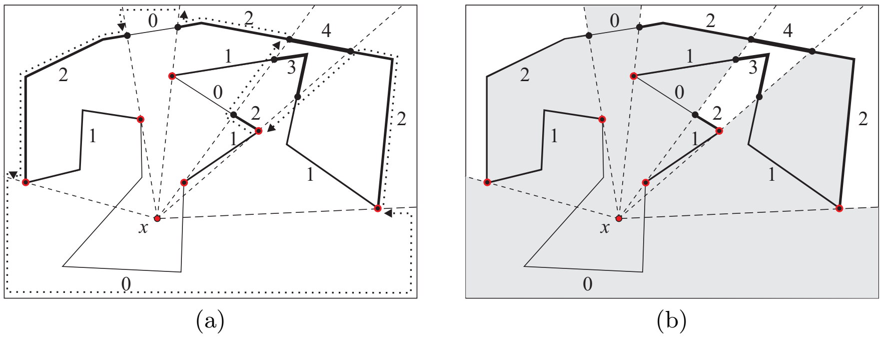


Figure 4: (a) *Method to construct* $Vis_2(x, P)$*;* (b) $Vis_2(x, P)$.

Step (2) needs a more detailed description. The labeling of each segment is done as follows:

1. Label each critical vertex with "+1" or "−1". If v_i is a critical vertex there are four rules to label it (see Fig. 5):

   1.1 if $v_{i-1}v_iv_{i+1}$ is a left-turn and:

   Rule 1: v_{i-1} is on the positive side of the ray $\overrightarrow{xv_i}$ then label v_i with "+1".

   Rule 2: v_{i-1} is on the negative side of the ray $\overrightarrow{xv_i}$ then label v_i with "−1".

   1.2 if $v_{i-1}v_iv_{i+1}$ is a right-turn and:

   Rule 3: v_{i-1} is on the positive side of the ray $\overrightarrow{xv_i}$ then label v_i with "−1".

   Rule 4: v_{i-1} is on the negative side of the ray $\overrightarrow{xv_i}$ then label v_i with "+1".

   Fig. 7(a) illustrates a polygon whose critical vertices labeled according to the previous rules.

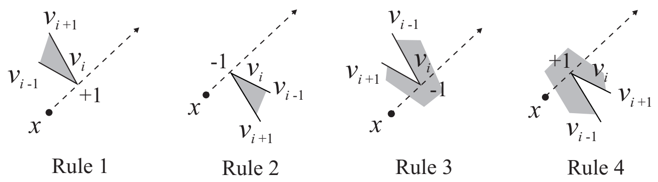


Figure 5: *Rules to label the critical vertices (shaded areas represent $int(P)$).*

2. Label each intersection point q identified in step (2) with "+2" or "−2": Let be $q \in \overline{v_{j-1}v_j}$, where $\overline{v_{j-1}v_j}$ is an edge of P and q can be equal to v_j. If v_{j-1} and v_{i-1} are on the same side regarding the ray $\overrightarrow{xv_i}$ then label q with "−2", else label q with "+2". See Fig. 6 for illustration.

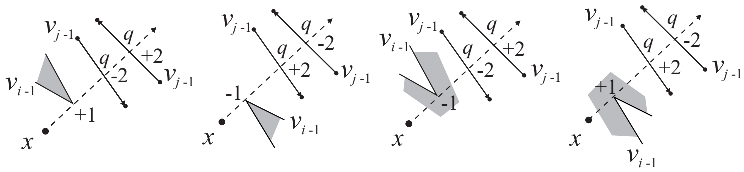


Figure 6: *Rule to label the intersection points.*

Fig. 7(b) illustrates a polygon with the intersection points labeled according to the previous rule.

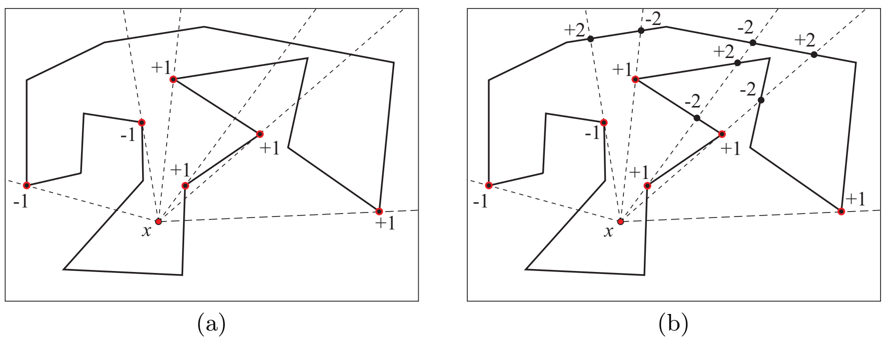
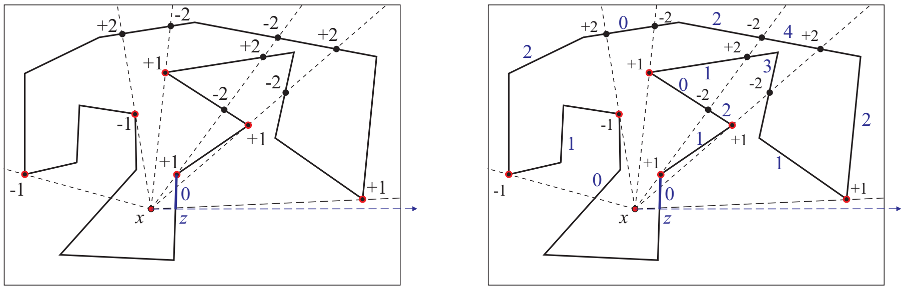


Figure 7: (a) *Labeled critical vertices*; *(b) Labeled intersection points.*

3. Draw the horizontal ray (to the right of x) and detect the first intersection point z with $bd(P)$. Label with 0 the edge to which z belongs (from z to the next vertex/point with a label), see Fig. 8(a). Advance by $bd(P)$ (in CCW direction) until the next labeled vertex/point p is found. Label the built segments adding the label of p to the label of the previous segment. This procedure should be repeated until z is reached. Fig. 8(b) illustrates a

polygon with all the segments labeled.



(a) (b)

Figure 8: (a) *First labeled edge;* (b) *Labeled segments.*

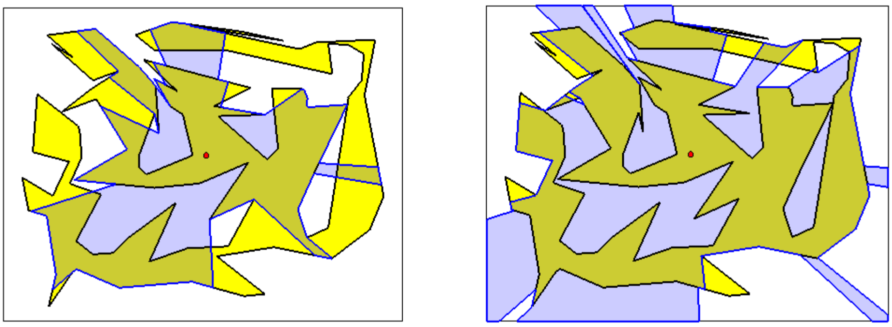
3.1 Algorithm Complexity

The detection of critical vertices is linear, because testing whether a vertex is reflex is done in constant time. Subsequently the management of critical vertices is done in $\mathcal{O}(n \log n)$. In step 2 we have n edges of the polygon that intersect $\mathcal{O}(n)$ rays that connect critical points with x. Therefore, the total number of segments in which these rays divide the edges of the polygon is $\mathcal{O}(n^2)$. Moreover the labeling with +1 or -1 for each critical vertex is done in constant time, and the label +2 or -2 for each intersection point detected in step 1 is also performed in constant time. Thus, as there are $\mathcal{O}(n^2)$ points of intersection, the total cost of labeling is $\mathcal{O}(n^2)$. As there are $\mathcal{O}(n^2)$ segments, the total cost is $\mathcal{O}(n^2)$.

The construction of the visibility polygon for each fixed value of k (step 3) is done in linear time, as the label k appears at most twice on each ray from x. Moreover, the points with k label on any ray are consecutive or there is only one point between them. To conclude, the construction of all k-visibility polygons is done in $\mathcal{O}(n^2)$ time with $k = 1, 2, ...n - 1$.

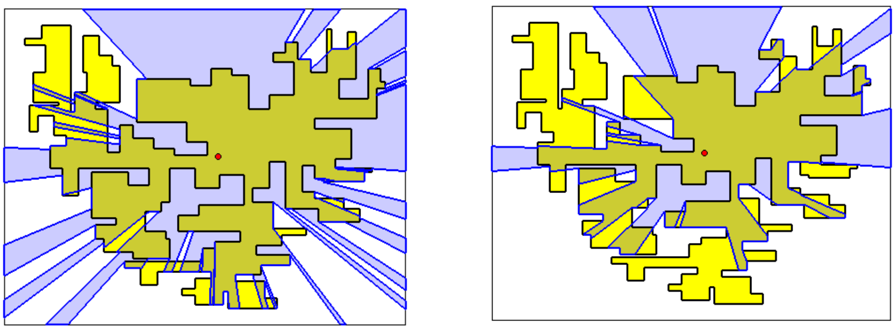
In the implementation of this algorithm we only considered even values of k because, for now, we are only interested in covering the interior of the polygon. However, for odd values the implementation can be done in a similar way. To conclude, figures 9 and 10 show some snapshots of our software.

We can now present the approximation methods that we developed to tackle the Minimum Vertex k-Transmitter Set problem, MVkT(P, k).



(a) (b)

Figure 9: *A 100-vertex simple polygon* P *and:* (a) $Vis_2(x, P)$*;* (b) $Vis_4(x, P)$



(a) (b)

Figure 10: *A 100-vertex orthogonal polygon* P *and:* (a) $Vis_2(x, P)$*;* (b) $Vis_4(x, P)$

4 Approximation Algorithms

Genetic Algorithms (GAs) and the Simulated Annealing (SA) are two classic and general metaheuristics techniques. GAs are population-based search methods and SA is a single-solution search method. Different combinations of these types of metaheuristics, the so called hybrid metaheuristics, have provided powerful search algorithms resulting in successful applications (see e.g. [Blum, Rolli 2008, Talbi 2002]). A well-known way of hybridization is the use of single-solution search methods into population-based techniques [Blum, Rolli 2008]. Indeed, the most successful applications of population-based methods make use of local search procedures (this is explained in 4.3). In this way, we opted to develop a hybridization of these two metaheuristic: we use a GA as a global optimizer and augment its standard genetic operators with a SA strategy. In addition to the hybrid method we also developed two other approximation methods, based on the general metaheuristics GAs and SA, and conducted a performance comparison between the three methods.

In this section we describe our three approximation algorithms to determine a covering vertex k-transmitter set G_{km}, whose cardinality approximates the minimum number of vertex k-transmitters that cover a given polygon P. A set G_{km} of vertices of P is a covering vertex k-transmitter set for P if $P \subset \bigcup_{v \in G_{km}} Vis_k(v, P)$. Each of these methods starts with a pre-processing step to compute and store the k-transmitter visibility polygons of v_i, for all $v_i \in V_P$, using the algorithm described in Sect. 3. After obtaining a covering vertex k-transmitter set for P, G_{km}, with each algorithm, some elements of G_{km} can be redundant. Thus, after running the approximation methods, we iteratively remove those elements in order to refine the obtained solution (post-processing step). In the next subsections we present the three approximation methods. We begin with the methods based on the general metaheuristics, since the hybrid method uses some concepts of the GAs and SA strategies.

4.1 Genetic Algorithm Strategy

Genetic Algorithms (GAs) are population-based search methods that use techniques inspired by evolutionary biology, such as, *chromosomes*, *genes*, *selection* and *crossover*, to solve optimization problems (see, e.g., [Alba 2005]). GAs are implemented as a computer simulation of an optimization problem in which a population of abstract representations of candidate solutions evolves toward better solutions. In the end, the population is composed of "good" solutions and the algorithm outputs the best one. To solve an optimization problem with GAs it is necessary to specify the following components: a genetic representation of the possible solutions, called *individuals*, to the problem (*Encoding*); a way of creating an initial population of possible solutions (*Initial Population*); a function to evaluate the individuals and act as natural selection (*Fitness function*); genetic operators to modify the composition of the solutions (*Selection*, *Crossover* and *Mutation*) and the values of several parameters used by the genetic algorithm (e.g., population size, probability of the genetic operators, population evaluation, population generation, termination condition). For more details on GAs see, for instance, [Glover, Kochenberger 2003]. In the following we describe how these parameters were defined to suit our problem.

Encoding. In our algorithm an individual I is represented by a chain $I = m_0 m_1 \dots m_{n-1}$, where each m_i (called *gene*) represents the vertex $v_i \in V_P$ and its value can be either 0 or 1. If $m_i = 1$ then v_i is a k-transmitter; otherwise v_i is not a k-transmitter.

Population and Initial Population. The population size is given by the number of reflex vertices, r, of the polygon (a vertex of a polygon is a reflex vertex if its internal angle is strictly greater than π). In this way, the input of the problem is associated with the elements of the metaheuristic. [Urrutia 2000] proved that, being P a polygon with r reflex vertices, r guards, placed on the reflex vertices of P, are always sufficient to guard P. It is easy to conclude that this result remains valid if the guards are replaced by k-transmitters. Thus, to create the initial population we consider the set of reflex vertices of P, $R = \{u_0, u_1, \ldots, u_{r-1}\}$. Then each of the r individuals are generated as follows: $\forall i \in \{0, \ldots, r-1\}$, if the polygon is covered by placing a k-transmitter in every vertex of $R \backslash \{u_i\}$, $R \backslash \{u_i\}$ is admitted as an individual of the population; otherwise R is taken as an individual.

Fitness function. For each I, f is defined by $f(I) = \sum_{j=0}^{n-1} m_j$. Our goal is to minimize this function.

Selection. The selection method should choose the best individuals to be reproduced. Since there are many different types of selection, we performed a comparative study taking two common methods into account: the *roulette wheel selection* and the *tournament selection* (see, e.g., [Reeves 2003]). In both methods we chose two individuals to be parents in crossover.

Crossover. Crossover operates on selected genes from parent individuals and creates new individuals (children). As there are many different kinds of crossover, we did a comparative study with four different types of crossover: *single point crossover*, *two-point crossover*, *uniform crossover* and a variant of the single point crossover where the generated children cannot be clones of the parents (see, e.g., [Reeves 2003]). In any crossover method we only generate one child from two parents. Crossover only occurs with a given probability, p_c, decided on the basis of trial and error. We used $p_c = 0.8$, which was experimentally obtained. Note that the child resulting from any of the described crossover methods may not be valid (i.e., it may not correspond to a covering vertex k-transmitter set), in this case the child is not accepted.

Mutation. Our mutation flips each gene from zero to one or vice versa, with a mutation probability p_m. In our case we apply the mutation to the child obtained in the crossover operation, with $p_m = 0.05$ (chosen experimentally). As in the crossover, if the resultant individual is not valid we do not accept it.

Population Generation. As there are many different ways to generate a new population we used a common one (*steady-state reproduction*): we select the worst individual of the population to be deleted replacing it by the child obtained at the mutation.

Population Evaluation. We consider the population evaluation, i.e., the fitness of a population as the minimum value of the fitness function f when applied to all individuals of the population.

Termination Condition. If the fitness of the population remains unchanged for a large number of generations h, we can assume that we are close to optimal and stop the search. In our case, we considered $h = 500$ (chosen experimentally).

4.2 Simulated Annealing Strategy

Simulated Annealing (SA) is a single point search method and it is usually acknowledged as the oldest among the metaheuristics. This strategy is based on an analogy between the physical annealing of solids and combinatorial optimization problems. It is commonly said that SA is one of the first algorithms that has an explicit strategy to escape from locally optimal solutions (see, e.g. [Blum, Rolli 2003]). For that, SA introduces a control parameter T, designated by *temperature*. This parameter should have a high initial value that will decrease during the search process. The search is an iterative process that stops when a *termination condition* is achieved. According to a certain *probability*, control parameter T allows solutions y whose objective function values are worse than the objective function value of the current solution x. Given an optimization problem, it is necessary to adapt it to the SA scheme, which is obtained by specifying the following parameters: (i) Specific Parameters: *solution space*, *objective function*, *neighborhood of each solution* and *initial solution*; (ii) Generic Parameters: *initial temperature* (T_0), *temperature decrement rule*, *number of iterations in each temperature* ($N(T_k)$) and *termination condition*. For more details on SA see, for instance, [Glover, Kochenberger 2003]. These parameters are described in the following.

4.2.1 Specific Parameters

Solution space. The solution space, set S, is the set of all covering vertex k-transmitter sets for P. Thus, S is a finite set and can be represented by $S = \{S_1, S_2, \ldots, S_m\}$, where $S_i = v_0^i v_1^i \ldots v_{n-1}^i$ for $i = 1, \ldots, m$. In this way, each candidate solution S_i is represented by a chain of length n, where v_j^i, with $j \in \{0, \ldots, n-1\}$, represents the vertex $v_j \in P$ and its value is 0 or 1. If $v_j^i = 1$ then the vertex v_j is a k-transmitter; otherwise the vertex v_j is not a k-transmitter.

Objective function. The objective function $f : S \rightarrow \mathbb{N}$ is defined in a similar way to the fitness function defined for the GA strategy. For each $S_i \in S$, f is defined by $f(S_i) = \sum_{j=0}^{n-1} v_i^j$. As for the GA strategy, our goal is to to minimize this function.

Neighborhood of each solution. According to SA, for each candidate solution $S_i \in S$, an element $S_j \in S$, called neighbor of S_i, must be obtained in order to be analyzed in the next iteration. In our case, to generate a neighbor S_j of $S_i = v_0^i \ldots v_{n-1}^i$ we randomly generate a natural number, uniformly distributed, $t \in [0, n-1]$ and then: (a) if $v_t^i = 1$ then we set v_t^j to 0 and accept this new solution if it is valid, otherwise we discard it; (b) if $v_t^i = 0$, we set v_t^j to 1 and accept this new solution with a certain probability.

Initial solution. The initial solution S_0 is the first covering vertex k-transmitter set to be analyzed. For the initial solution we consider all reflex vertices of P as vertex k-transmitters.

4.2.2 Generic Parameters

Initial temperature (T_0). We performed a comparative study taking into account two different types of T_0: (1) an initial temperature dependent on the number of vertices of the simple polygon P, $T_0 = f(n)$ (we have considered $T_0 = n$ and $T_0 = \frac{n}{4}$) and (2) a constant initial temperature: $T_0 = 500$ (this constant value was chosen on the basis of trial and error).

Temperature decrement rule. The value of the temperature at each iteration k, T_k, is established by a temperature decrement rule. We made an analysis on three different types of rules: (1) $T_{k+1} = \frac{T_0}{1+k}$ (Fast Simulated Annealing (FSA) decrease); (2) $T_{k+1} = \frac{T_0}{e^k}$ (Very Fast Simulated Annealing (VFSA) decrease) and (3) $T_{k+1} = \alpha T_k$, where $0 < \alpha < 1$ (Geometric decrease).

Iterations for each temperature T_k. In our algorithm $N(T_k) = \lceil T_k \rceil$. Note that this choice ensures that there are more iterations while the temperatures are high, when the solutions are still far from optimal.

Termination condition. In theory, the search process should stop when $T_k = 0$. However, much before reaching this value, the probability to accept a move toward a worse solution is practically null. As a result it is possible, in general, to finish the search with a final temperature, T_f, greater than zero without quality loss in the solution. In this sense, the termination condition chosen in our algorithm consists in finishing the search when $T_f \leq 0.005$ or when the last $l = 3000$ consecutive series of temperatures do not achieve a better solution and the percentage of accepted solutions is less than $\varepsilon = 2\%$ (the values of l and ε were chosen experimentally).

4.3 Hybrid Strategy

As stated above, our hybrid metaheuristic consists in using a GA as a global optimizer and augmenting its standard genetic operators with a SA. That is, our hybridization is done by using a single-solution search method into a population-based technique, which usually is a successful combination [Blum, Rolli 2008]. The reason for that becomes clear when the strong points of population-based methods and the single-solution methods are analyzed. There are two main, complementary, forces (concepts) that determine the behavior of a metaheuristic, *diversification* (or *exploration*)and *intensification* (or *exploitation*). Diversification ensures that many and different regions of the search space are "visited", whereas intensification guarantees a carefully and intensively search within those regions, allowing high quality solutions. A good balance between these two goals is important because a search should intensively explore areas of the search space with high quality solutions and move to unexplored areas of the search space when necessary (see, e.g., [Blum, Rolli 2008, Lozano, García-Martínez 2010]). While all metaheuristics are driven by these two forces, some of them have a clear tendency to intensification and others to diversification. It can be said that population-based methods are better in identifying promising areas on vast and complex search spaces, whereas single-solution methods are better in exploiting those promising areas. In other words, population-based metaheuristics are mainly guided by diversification, while intensification guides single-solution methods. The idea of combining these two complementary forces, diversification and intensification, is a good reason to hybridize population-based and single-solution metaheuristics and seek to incorporate the strengths and eliminate weaknesses of both types of methods [Mahfoud, Goldberg 1995].

Although there are many ways to use single-solution methods into population based techniques, to solve the MVkT(P, k) problem we used a SA strategy as a genetic operator of a GA strategy. As the standard genetic operators, this one occurs with a certain probability p_{sa}. In the experimental evaluation we used $p_{sa} = 0.1$ (chosen empirically). Figure 11 illustrates the general scheme of our hybridization. This hybridization allows to observe how a GA behaves on reinforcing the intensification during the search process.

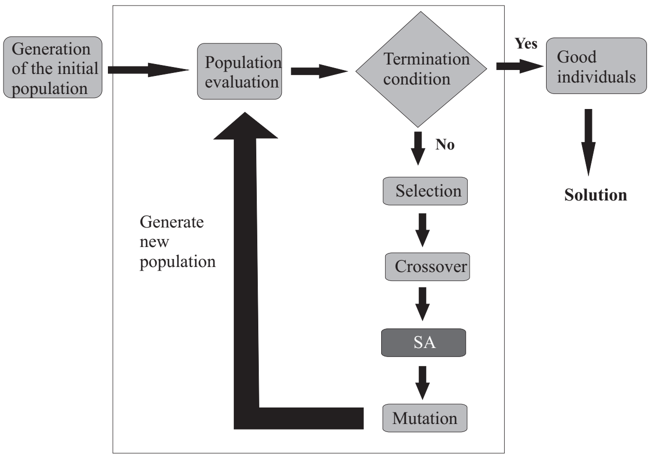


Figure 11: General scheme of the hybrid strategy.

It is important to note that the alternatives concerning the parameters of the metaheuristics (GA and SA) that could be explored is almost endless. With regard to the hybrid heuristics, not only different combinations can be made, but also different parameters can be used. We rely on studies of other problems (e.g. [Martins 2009]), noting that a more exhaustive study in future investigations might improve the obtained results.

5 Experiments and Results

To compare the performance of the hybrid method with the two pure strategies (GA and SA), we implemented the metaheuristics and tested their behaviour over a large set of randomly generated polygons. As said before, it was also necessary to develop and implement a new algorithm to determine $Vis_k(x, P)$, $x \in P$, described in section 3. These implementations were done in C/C++ (for MS Visual Studio 2005) on top of the Computational Geometry Algorithms Library (CGAL 3.2.1). The described methods were tested on a PC featuring a Intel(R) Core (TM)2 CPU 6400 at 2.66 GHz and 1 GB of RAM. Our experiments were done over simple, monotone, orthogonal and monotone orthogonal polygons. The simple polygons were generated using the CGAL function *random_polygon_2*, whose implementation is based on the method of eliminating self-intersections in a polygon by using the so-called "2-opt" moves [Hert et al. 2006], and to generate the monotone polygons we used the algorithm developed by [Snoeyink, Chong 1993]. The orthogonal polygons were generated using the polygon generator developed by Joseph O'Rourke (personal communication 2002) and the monotone orthogonal polygons using the algorithm proposed in [Tomás et al. 2003]. According to this algorithm the polygons are placed on an $\frac{n}{2} \times \frac{n}{2}$ unit square grid and have no collinear edges, and by this reason they are designated by *grid monotone orthogonal polygons*.

Because the simple comparison of two or more values (e.g., averages, medians) might be different according to the statistical distributions, we performed a statistical study to ensure strong statistical results (i.e., determining whether the conclusions are meaningful and not just noise) [Alba 2005]. Therefore, we first applied the Kolmogorov-Smirnof test to check data normality. Since in all performed studies the data sets were not normally distributed, we had to use non-parametric statistical tests, such as the Kruskal-Wallis test. This test ensures statistical difference in the results, with a higher statistical power than the ANOVA test when data are not modeled as a normal population [Waynem 1990]. When at least a data sample was significantly different than the others, multiple comparison tests were used to determine which pairs of results were significantly different and which were not. A significance level of 0.05 was used for all tests.

In the next subsections we present the results and the conclusions of the experiments that were carried out. First, we present how we chose the metaheuristics parameters (subsection 5.1). Next, in subsection 5.2, we analyze and compare the results obtained by the hybrid metaheuristic and the two pure metaheuristics. We performed the computational experiments over sets of simple, monotone, orthogonal and grid monotone orthogonal polygons, each set with 40 polygons of 50, 100, 150 and 200-vertex polygons. This study was made for 2-transmitters and 4-transmitters. Finally, in subsection 5.3, we use the least squares method to determine, approximately, the average number of 2-transmitters and 4-transmitters that cover a given polygon, obtained from the conducted experiments. This was done in order to establish upper bounds, smaller than the theoretical bounds, for the minimum number of vertex k-transmitters that cover an n-vertex polygon.

5.1 Analysis of the Metaheuristics Parameters

According to subsections 4.1, 4.2 and 4.3, there are various alternatives for some of the GA and SA parameters. Together, as we shall see, they can be combined to yield numerous cases, making it tedious and impractical to perform experiments for the four types of polygons with so many vertices ($n = 50, 100, 150$ and 200) and for the two values of k ($k = 2$ and $k = 4$). Instead, we decided to chose the parameters based on experiments over sets of simple polygons, each one formed by 40 polygons of 30, 50, 70 and 100 vertex polygons, which we believed would give good results and be time efficient. For every set of polygons we determined the average number of vertex 2-transmitters, as well as the average runtime in seconds and the average number of iterations. And as stated above, we also performed a statistical study to compare the obtained results. Since there are too many results, we decided to present only the conclusions of the experiments in the following three subsections.

5.1.1 GA Parameters

According to section 4.1 we have several options for two of the GA parameters: the selection and the crossover operators. The different combinations produce eight cases (see Table 1).

Table 1: *Studied cases for GA.*

Cases	
Case 1	Roulette Wheel Selection and Single Point Crossover
Case 2	Roulette Wheel Selection and Two-Point Crossover
Case 3	Roulette Wheel Selection and Single Uniform Crossover
Case 4	Roulette Wheel Selection and Variant of Single Point Crossover
Case 5	Tournament Selection and Single Point Crossover
Case 6	Tournament Selection and Two-Point Crossover
Case 7	Tournament Selection and Single Uniform Crossover
Case 8	Tournament Selection and Variant of Single Point Crossover

We carried out a statistical study to compare the results obtained by the eight cases (concerning the number of vertex 2-transmitters, the runtime and the number of iterations). This study allowed us to conclude that: (a) regarding the returned number of vertex 2-transmitters, there were no statistically significant differences between the eight cases; (b) concerning the runtime, for $n = 30, 50$ and 70, Case 8 is the fastest one with no statistically significant differences from Case 2, for $n = 100$ Case 8 is the fastest one with statistically significant differences from all the other cases. In this way, the solutions obtained by the eight cases can be considered similar, even though the response time is lower in Case 8. As good solutions result in higher algorithm runtimes, it is appropriate to balance one against the other, so we decided to chose Case 8, which we believe to be the best under these circumstances.

5.1.2 SA parameters

In accordance with subsection 4.2, there are several choices for two of the SA parameters: T_0 and the temperature decrement rule. The different combinations result in nine cases (see Table 2). Once again, we analyzed these nine cases by comparing the number of vertex 2-transmitters, the runtime and the number of iterations. We performed a statistical study to compare the results obtained by them, which allowed us to conclude that concerning the returned number of vertex 2-transmitters, Case 4 is the best, with no statistically significant differences from Case 1 but with significant differences from the other cases. However, regarding these two cases, Case 1 is statistically significantly faster than Case 4.

Table 2: *Studied cases for SA.*

Cases	
Case 1	$T_0 = n$ and $T_{k+1} = \frac{T_0}{1+k}$ (FSA decrease)
Case 2	$T_0 = n$ and $T_{k+1} = \frac{T_0}{e^k}$ (VFSA decrease)
Case 3	$T_0 = n$ and $T_{k+1} = \alpha T_{k-1}$ ($\alpha = 0.9$) (Geometric decrease, $\alpha = 0.9$)
Case 4	$T_0 = 500$ and $T_{k+1} = \frac{T_0}{1+k}$ (FSA decrease)
Case 5	$T_0 = 500$ and $T_{k+1} = \frac{T_0}{e^k}$ (VFSA decrease)
Case 6	$T_0 = 500$ and $T_{k+1} = \alpha T_{k-1}$ (Geometric decrease, $\alpha = 0.9$)
Case 7	$T_0 = \frac{n}{4}$ and $T_{k+1} = \frac{T_0}{1+k}$ (FSA decrease)
Case 8	$T_0 = \frac{n}{4}$ and $T_{k+1} = \frac{T_0}{e^k}$ (VFSA decrease)
Case 9	$T_0 = \frac{n}{4}$ and $T_{k+1} = \alpha T_{k-1}$ (Geometric decrease, $\alpha = 0.9$)

Note that, the best solutions are those obtained in Cases 1 and 4, however, the response time is higher in Case 4. And again, as we want to find a compromise between the goodness of the solution obtained and the algorithm runtime, it seems that Case 1 is the best under these conditions. Therefore, this is the case that we considered as the SA strategy.

5.1.3 Hybrid Strategy Parameters

As stated in subsection 4.3, our hybridization consists of a SA strategy used as a genetic operator of a GA strategy. In this way, we have to chose the parameters for these two strategies. For the GA component of our hybrid metaheuristic we chose to use exactly the same parameters that were selected to our pure GA strategy (that is, Case 8), since it was the GA's case that we considered the best one. Concerning the SA component, we chose to use the parameters of the SA's Case 8. We made this choice because this case is the fastest algorithm and although the returned solution is not statistically significantly better than all the other cases, it is acceptable to work as genetic operator. We think that if we had chosen SA's Case 4 (the elected pure SA strategy) the execution time of our algorithm would not have obtained substantially improved solutions. Nevertheless, as a future work we intend to analyze and experiment other combinations.

5.2 Comparison of the Three Strategies

As stated above, to analyze and compare our three methods, we performed the computational experiments over sets of simple, monotone, orthogonal and grid monotone orthogonal polygons, each set with 40 polygons of 50, 100, 150 and 200-vertex polygons.

5.2.1 Simple and Monotone Polygons

In this section we proceed with the presentation and analysis of the results obtained on simple and monotone polygons. Table 3 presents the results obtained on simple polygons. This table shows the average number of vertex k-transmitters, with $k = 2$ and $k = 4$, (*Solution*), the average runtime in seconds (*Time*) and the average number of iterations (*Iter.*), for the three strategies.

Table 3: *Results obtained with the hybrid, GA and SA strategies on simple polygons for:* (a) $k = 2$ and (b) $k = 4$.

(a) simple polygons, $k = 2$.

n	Hybrid			GA			SA		
	Solution	Time	Iter.	Solution	Time	Iter.	Solution	Time	Iter.
50	2.67	37.35	541.57	2.97	26.75	779.97	3.10	49.97	6208.40
100	4.55	343.50	613.20	5.72	292.62	1559.70	5.32	317.37	10391.00
150	6.65	1091.50	613.70	8.22	1167.90	2887.40	7.67	877.27	14159.00
200	8.35	2664.30	702.75	10.27	2766.30	3995.20	9.70	1729.50	17902.00

(b) simple polygons, $k = 4$.

n	Hybrid			GA			SA		
	Solution	Time	Iter.	Solution	Time	Iter.	Solution	Time	Iter.
50	1.65	30.97	517.67	1.85	24.37	737.27	1.75	48.17	5063.80
100	2.65	417.45	582.17	3.02	457.27	1743.80	2.97	448.85	10669.00
150	3.57	1527.00	572.10	4.25	1876.80	3012.00	3.95	1379.80	14462.00
200	4.37	4207.40	613.30	5.25	4785.10	3929.30	5.15	2974.80	17688.00

As we can notice, although the response time of the hybrid algorithm appears to be greater, the obtained solutions seems to be better than the ones obtained with the two pure strategies (except for $n = 50$, where they appear to be almost equal). However, as stated before, the comparison between the results obtained with the three strategies only makes sense if a statistical study is made to ensure its statistically significance. First of all we studied the results concerning the number of k-transmitters. This study allowed us to conclude that for $n = 50$ there was not a statistically significant difference between the three methods. However, for $n = 100, 150$ and 200 the hybrid method is the best one presenting significant differences from methods SA and GA and the worst method is GA with no significant differences from SA. We also did a statistical analysis regarding the runtime. This study was made in a similar way and it allows us to conclude that the hybrid method is sometimes slower than SA and other times slower than GA. However, we should point out that the differences are not always statistically significant, for example, for $k = 4$ and $n = 100$ no significant differences were found between the three methods; for $k = 4$ and $n = 150$ the fastest method is the SA and it has no significant differences from the hybrid algorithm. In conclusion, for simple polygons, the hybrid method obtains, in general, solutions statistically significantly better than the other two methods, despite being globally slower.

Table 4 shows the results obtained for monotone polygons.

Table 4: *Results obtained with the hybrid, GA and SA strategies on monotone polygons for:* (a) $k = 2$ and (b) $k = 4$.

(a) monotone polygons, $k = 2$.

n	Hybrid			GA			SA		
	Solution	Time	Iter.	Solution	Time	Iter.	Solution	Time	Iter.
50	3.17	26.00	555.70	3.80	13.45	743.15	3.77	37.47	6209.90
100	6.92	189.60	594.90	8.17	122.42	1358.10	8.07	198.25	10496.00
150	10.10	509.62	605.20	11.92	426.42	2501.70	11.62	484.42	14318.00
200	13.37	1184.50	655.12	15.62	934.12	3412.80	15.42	925.45	17836.00

(b) monotone polygons, $k = 4$.

n	Hybrid			GA			SA		
	Solution	Time	Iter.	Solution	Time	Iter.	Solution	Time	Iter.
50	2.15	20.17	536.00	2.55	13.00	731.25	2.45	33.70	6463.50
100	4.00	184.45	562.95	4.82	147.07	1447.80	4.87	209.70	10375.00
150	6.15	515.95	581.45	7.60	469.77	2484.70	7.50	516.30	14186.00
200	7.85	1232.60	588.85	9.70	1080.10	3579.60	9.37	996.92	18149.00

Similarly to what happens with simple polygons, even though the hybrid algorithm seems to take longer it appears to reach improved solutions. Therefore we performed again the statistical study described before. First of all we performed a study concerning the number of k-transmitters, whose results showed that, for n = 50, 100, 150 and 200, the hybrid method is the best one with statistically significant differences from the other two methods and the worst method is GA with no significant differences from SA. Concerning the algorithms' runtime, the statistical analysis allowed us to conclude that on the whole the hybrid method is sometimes slower than SA and other times slower than GA. However, we should point out that in some cases the differences are not statistical significantly different, for example, for $k = 4$ and $n = 100$ the three methods do not present significant differences. Accordingly, the hybrid method obtains solutions significantly better than the other two methods, although it is generally slower.

5.2.2 Orthogonal and Monotone Orthogonal Polygons

In this section we present the analysis of the results obtained for orthogonal and grid monotone orthogonal polygons, and which can be seen in Tables 5 and 6. As in the previous subsection, these tables exhibit the average number of vertex k-transmitters, with $k = 2$ and $k = 4$, (*Solution*), the average runtime in seconds (*Time*) and the average number of iterations (*Iter.*), for the three strategies. Similarly to the cases of simple and monotone simple polygons, it seems that the hybrid algorithm obtains better solutions in spite of an overall slower running time. As previously described, we then carried out a statistical study. This study allowed us to conclude that, concerning the average number of 2-transmitters and 4-transmitters, the best strategy is the hybrid one with statistically significant differences from the SA and GA strategies (except for $k = 4$ and $n = 50$, where a significant difference was not found between the three). Regarding the runtime, although the hybrid strategy is globally slower than SA or GA, in several cases the differences are not statistically significant.

Table 5: *Results obtained with the hybrid, GA and SA strategies on orthogonal polygons for:* (a) $k = 2$ and (b) $k = 4$.

(a) orthogonal polygons, $k = 2$.

n	Hybrid			GA			SA		
	Solution	Time	Iter.	Solution	Time	Iter.	Solution	Time	Iter.
50	2.45	32.25	559.27	2.92	25.27	785.02	2.92	51.22	6400.10
100	4.20	318.17	618.60	5.02	296.62	1607.10	4.82	292.75	10163.00
150	5.95	1091.90	692.70	7.67	1248.80	2930.40	7.20	844.92	13871.00
200	7.95	2541.30	656.45	9.35	2958.30	4052.90	8.92	1748.40	18313.00

(b) orthogonal polygons, $k = 4$

n	Hybrid			GA			SA		
	Solution	Time	Iter.	Solution	Time	Iter.	Solution	Time	Iter.
50	1.35	22.82	555.50	1.65	19.42	821.57	1.67	48.50	4883.30
100	2.22	359.15	594.20	2.95	379.15	1721.60	2.70	370.20	9781.00
150	3.15	1535.00	625.77	3.97	1778.10	2828.80	3.62	1291.90	13926.00
200	3.97	4088.89	649.32	5.05	4674.90	3986.40	4.62	2830.20	18173.00

Table 6: *Results obtained with the hybrid, GA and SA strategies on grid monotone orthogonal polygons for:* (a) $k = 2$ and (b) $k = 4$.

(a) grid monotone orthogonal polygons, $k = 2$.

n	Hybrid			GA			SA		
	Solution	Time	Iter.	Solution	Time	Iter.	Solution	Time	Iter.
50	3.00	27.90	583.10	3.77	13.35	747.75	3.42	35.95	6563.90
100	5.65	212.57	640.67	7.20	163.55	1588.70	6.85	193.40	10302.00
150	8.60	612.90	644.77	11.90	397.87	2325.10	9.67	489.20	14235.00
200	11.25	1391.00	695.65	13.85	1298.60	4128.40	12.95	956.50	18062.00

(b) grid monotone orthogonal polygons, $k = 4$

n	Hybrid			GA			SA		
	Solution	Time	Iter.	Solution	Time	Iter.	Solution	Time	Iter.
50	1.97	22.22	530.27	2.15	17.17	787.22	2.15	31.77	6169.40
100	3.20	212.950	564.92	4.025	194.05	1688.20	3.75	209.65	10160.00
150	4.65	693.85	643.05	5.67	697.37	2822.30	5.67	574.22	14003.00
200	6.00	1468.30	622.40	7.67	1617.10	4059.00	7.27	1130.20	18480.00

5.3 Average number of k-transmitters

Since the hybrid strategy was considered the best, it was therefore used to infer the average of the minimum number of k-transmitters, with $k = 2$ and $k = 4$, that cover a given simple, monotone, orthogonal or grid monotone orthogonal polygon with n vertices. To this end, the referred strategy was applied to sets of simple, monotone, orthogonal or grid monotone orthogonal polygon, each set with 40 polygons of 30, 50, 70, 100, 110, 130, 150 and 200 vertex polygons, respectively. The obtained solutions are presented in Table 7. Then we used the least squares method and we conclude that:

(a) **Simple polygons.** The curve fitted to the data of Table 7(a) (concerning the average number of 2-transmitters) is $f(n) = 0.0383n + 0.7523 \approx \frac{n}{26.10}$, with a correlation factor of 0.9988. The linear function that "best" fits the data presented on Table 7(a) (concerning the average number of 4-transmitters) is $f(n) = 0.0191n + 0.6436 \approx \frac{n}{52.35}$, with a correlation factor of 0.9926.

(b) **Monotone polygons.** The linear function that "best" fits the number of 2-transmitters and vertex 4-transmitters with the number of vertices n of monotone polygons are $f(n) = 0.0662n + 0.1462 \approx \frac{n}{15.10}$, with a correlation factor of 0.9978 and $f(n) = 0.0279n + 0.4162 \approx \frac{n}{35.84}$, with a correlation factor of 0.9973.

(c) **Orthogonal polygons.** The curve fitted to the data of Table 7 (c) (concerning the average number of 2-transmitters) is $f(n) = 0.0365n+0.5844 \approx \frac{n}{27.39}$, with a correlation factor of 0.9988. The linear function that "best" fits the data presented on Table 7(c) (concerning the average number of 4-transmitters) is $f(n) = 0.0174n + 0.5065 \approx \frac{n}{57.47}$, with a correlation factor of 0.9985.

(d) **Grid monotone orthogonal polygons.** The linear functions that "best" fit the number of 2-transmitters and vertex 4-transmitters with the number of vertices n of grid monotone orthogonal polygon are $f(n) = 0.0546n + 0.2614 \approx \frac{n}{18.31}$, with a correlation factor of 0.9987 and $f(n) = 0.0174n + 0.5065 \approx \frac{n}{57.47}$, with a correlation factor of 0.9985.

Table 7: *Average of the minimum number of 2-transmitters and 4-transmitters.*

(a) simple polygons

n	30	50	70	100	110	130	150	200
2-transmitters	1.90	2.67	3.45	4.55	4.87	5.72	6.65	8.35
4-transmitters	1.07	1.65	2.00	2.65	2.82	3.05	3.57	4.37

(b) monotone polygons

n	30	50	70	100	110	130	150	200
2-transmitters	2.32	3.17	4.82	6.92	7.20	8.87	10.10	13.37
4-transmitters	1.75	2.15	3.02	4.00	4.70	5.45	6.15	7.85

(c) orthogonal polygons

n	30	50	70	100	110	130	150	200
2-transmitters	1.67	2.45	3.15	4.20	4.60	5.32	5.59	7.95
4-transmitters	1.00	1.35	1.80	2.22	2.45	2.75	3.15	3.97

(d) grid monotone orthogonal polygons

n	30	50	70	100	110	130	150	200
2-transmitters	1.97	3.00	4.12	5.65	6.22	7.15	8.60	11.25
4-transmitters	1.17	1.97	2.30	3.20	3.50	3.97	4.65	6.00

Table 8 summarizes the observed average number of vertex 2-transmitters and 4-transmitters that cover a given n-vertex polygon (simple, monotone orthogonal and grid monotone orthogonal).

Table 8: *Experimental results for the MVkT(P, k) problem.*

	Polygon P			
	Simple	Monotone Simple	Orthogonal	Grid Monotone Orthogonal
$k = 2$	$\lceil \frac{n}{26} \rceil$	$\lceil \frac{n}{15} \rceil$	$\lceil \frac{n}{27} \rceil$	$\lceil \frac{n}{18} \rceil$
$k = 4$	$\lceil \frac{n}{52} \rceil$	$\lceil \frac{n}{26} \rceil$	$\lceil \frac{n}{57} \rceil$	$\lceil \frac{n}{35} \rceil$

Note that the average number of vertex k-transmitters that cover monotone polygons is greater than the average number of vertex k-transmitters that cover non-monotone polygons. This behavior can be easily explained, since the direction of monotony is unfavorable for signal transmission.

6 Conclusions

In this paper we proposed a new $\mathcal{O}(n^2)$ time algorithm to determine the region covered by a k-transmitter located on a point x of a polygon P with n edges, $Vis_k(x, P)$, for all the possible values of k $(0 < k < n)$. We also developed three approximation strategies to tackle the problem of minimizing the number of k-transmitters, located at vertices, that cover a given simple polygon (MVkT(P, k) problem). One is a hybrid metaheuristic and the other two are based on the general metaheuristics GAs and SA. We compared the performance of the three methods and concluded that hybrid strategy, for 2-transmitters and 4-transmitters, on simple, monotone, orthogonal and grid monotone orthogonal polygons are better than the pure strategies.

Concerning the algorithm to determine $Vis_k(x, P)$ as future research we intend to develop of an algorithm to lower this computational complexity for a fixed value of k. Regarding the approximation algorithms, it is our purpose to try not only different metaheuristics parameters, but also different metaheuristics combinations. We also plan to develop a method that allows to determine the approximation ratio of the algorithms implemented to tackle the MVkT(P, k) problem, because the optimal solution for this problem is unknown. This method will allow to determine the gap between the approximate solutions and the optimal solution.

Acknowledgements

The second and third authors were supported by DIVO MTM2008-05043/MTM, MICINN, Spain and HP2008-0060, Acción Integrada Luso Española, Spain. The first and fourth authors were supported by CIDMA through Programa POCTI, FCT, co-financed by EC fund FEDER and E82/09 Acção Integrada Luso Espanhola, Portugal. The fourth author was supported by a FCT fellowship, grant SFRH/BPD/66431/2009.

References

[Abellanas et al. 2006] Abellanas, M. and Alba, E. and Canales, S. and Hernández, G.: "Solving the Illumination Problem with Heuristics"; Numerical Methods and Applications, Vol. 4310 of Lecture Notes in Computer Science, Springer (2006), 205–213.

[Alba 2005] E. Alba: "Parallel Metaheuristics: A New Class of Algorithms"; Wiley-Interscience Publishers (2005).

[Aggarwal 1984] Aggarwal, A.:"The art gallery theorem: its variations, applications and algorithmic aspects"; Thesis (PhD), The Johns Hopkins University (1984).

[Aichholzer et al. 2009 (a)] Aichholzer, O. and Fabila-Monroy, R. and Flores-Peñaloza, D. and Hackl, T. and Huemer, C. and Urrutia, J. and Vogtenhuber, B.: "Modem Illumination of Monotone Polygons"; Proc. 25^{th} European Workshop on Computational Geometry EuroCG'09, Brussels, Belgium (2009), 167–170.

[Aichholzer et al. 2009 (b)] Aichholzer, O. and Aurenhammer, F. and Hurtado, F. and Ramos, P. and Urrutia J.:"Two-convex polygons". In Proc. 25th European Workshop on Computational Geometry EuroCG '09, Brussels, Belgium, (2009), 117–120.

[Amit et al. 2010] Amit, Y. and Mitchell, J.S.B. and Packer, E.: "Locating Guards for Visibility Coverage of Polygons"; Int. J. Comput. Geometry Appl. 20, 5 (2010), 601–630.

[Bajuelos et al. 2009] Bajuelos, A.L. and Martins, A.M. and Canales, S. and Hernández, G.: "Metaheuristic Approaches for the Minimum Vertex Guard Problem"; The Third IEEE International Conference on Advanced Engineering Computing and Applications in Sciences (ADVCOMP 2009), IEEE Computer Society (2009), 77–82.

[Ballinger et al. 2010] Ballinger, Brad et al: "Coverage with k-Transmitters in the Presence of Obstacles"; COCOA'10, in Proceedings of the 4th International Conference on Combinatorial Optimization and Applications - Volume Part II, Kailua-Kona, Hawaii, USA, Dec. (2010), 1–15.

[Blum, Rolli 2003] Blum, C. and Rolli, A.:"Metaheuristics in combinatorial optimization: Overview and conceptual comparison"; ACM Comput. Surv., 35, 3 (2003), 268–308.

[Blum, Rolli 2008] Blum, C. and Rolli, A.: 'Hybrid Metaheuristics: An Introduction"; In Hybrid Metaheuristics: An Emerging Approach to Optimization, C. Blum, M. Blesa, A. Roli and M. Sampels (eds.), Springer (2008).

[Chvátal 1975] Chvátal, V.:"A combinatorial theorem in plane geometry"; Journal of Combinatorial Theory Series B, 18 (1975), 39–41.

[Couto et al. 2011] Couto, M., de Rezende, P., de Souza, C.: "An Exact Algorithm for Minimizing Vertex Guards on Art Galleries"; International Transactions in Operational Research 18 (4), 425-448 (2011) doi:10.1111/j.1475-3995.2011.00804.x

[Dobkin, Teller 2004] Dobkin, D. and Teller, S.: "Computer Graphics, Handbook of discrete and computational geometry"; CRC Press, Inc., Boca Raton, Florida, USA (2004), 1090–1116.

[Eidenbenz et al. 2001] Eidenbenz, S. and Stamm, C. and Widmayer, P.: "Inapproximability Results for Guarding Polygons and Terrains"; Algorithmica, 31, 1 (2001), 79–113.

[Fabila-Monroy et al. 2009] Fabila-Monroy, R. and Vargas, A.R and Urrutia, J.: "On Modem Illumination Problems"; XIII Spanish Workshop on Computational Geometry, Zaragoza, Spain (2009).

[Ghosh 2010] Ghosh, S.: "Approximation algorithms for art gallery problems in polygons"; Discrete Applied Mathematics, 158, 6 (2010), 718–722.

[Glover, Kochenberger 2003] Glover, F.W. and Kochenberger, G.A.: "Handbook of Metaheuristics"; International Series in Operations Research & Management Science Vol. 114, Springer (2003).

[Hert et al. 2006] Hert, S. and Hoffmann, M. and Kettner, L. and Schnherr, S.: "Geometric Object Generators, CGAL User and Reference Manual 3.2.1"; CGAL Editorial Board (2006).

[Joe, Simpson 1985] Joe, B. and Simpson R.B.: "Visibility of a simple polygon from a point"; Report CS-85-38, Dept. Math. Comput. Sci., Drexel Univ., Philadelphia, PA (1985).

[Lee, Lin 1986] Lee, D. T. and Lin, A. K.: "Computational complexity of art gallery problems"; IEEE Transactions on Information Theory, 32, 2 (1986), 276–282.

[Lozano, García-Martínez 2010] Lozano, M. and García-Martínez, C.:"Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: Overview and progress report"; Computers & Operations Research, 37, 3 (2010), 481–497.

[Mahfoud, Goldberg 1995] Mahfoud, S. W. and Goldberg, D. E.:"Parallel recombinative simulated annealing: a genetic algorithm"; Parallel Comput., 22, 1 (1995), 1–28.

[Martins 2009] Martins, A.M.: "Geometric Optimization on Visibility Problems: Metaheuristic and Exact Solutions"; Thesis (PhD), University of Aveiro (2009).

[O'Rourke 1987] O'Rourke, J.: "Art gallery theorems and algorithms"; Oxford University Press, Inc., New York, NY, USA (1987).

[O'Rourke, Toussaint 2004] O'Rourke, J. and Toussaint, G. T.: "Pattern recognition, Handbook of discrete and computational geometry"; CRC Press, Inc., Boca Raton, Florida, USA (2004), 1135–1162.

[Packer 2008] Packer, E.: "Computing Multiple Watchman Routes"; Proceedings of the 7th international conference on Experimental algorithms, WEA (2008), 114–128.

[Reeves 2003] Reeves, C.R.: "Genetic Algorithms, In: Handbook of Metaheuristic", F. Glover e G.A. Kochenberger (eds). Kluwer, Boston (2003), 55-82.

[Shermer 1992] Shermer, T.C.: "Recent results in art galleries [geometry]"; Proceedings of the IEEE, 80, 9 (1992), 1384–1399.

[Snoeyink, Chong 1993] Snoeyink, J. and Chong, Z.: "Generating random monotone polygons"; Technical Report 93–28, Dept. of Computer Science, University of British Columbia (1993).

[Talbi 2002] Talbi, E.G.: "A Taxonomy of Hybrid Metaheuristics"; Journal of Heuristics, 8, 5 (2002), 541–564

[Tomás et al. 2003] Tomás, A.P. and Bajuelos, A.L. and Marques, F.: "Approximation Algorithms to Minimum Vertex Cover Problems on Polygons and Terrains"; International Conference on Computational Science, Springer-Verlag (2003), 869–878.

[Urrutia 2000] Urrutia, J.:"Art gallery and illumination problems, Handbook of computational geometry"; J.-R. Sack and J. Urrutia (eds.), Elsevier (2000), 973–1027.

[Waynem 1990] Waynem, D. W.:"Applied Non-Parametric Statistics"; PWS-KENT Publishing Company (1990).

[Wein et al. 2006] Wein, R. and Van den Berg, J.P and Halperin, D.: "Planning near-optimal corridors amidst obstacles;"; Proc. 7th International Workshop on the Algorithmic Foundations of Robotics - WAFR 2006, New York, USA (2006).