

Bicriteria approach to the optimal location of surveillance cameras^{*}

Aleksandra Groß and Horst W. Hamacher

*TU Kaiserslautern, Fachbereich Mathematik, Paul-Ehrlich-Straße 14, 67663 Kaiserslautern
Emails: a.gross@mathematik.uni-kl.de, hamacher@mathematik.uni-kl.de*

Abstract: We consider the problem of finding efficient locations of surveillance cameras, where we distinguish between two different problems. In the first, the whole area must be monitored and the number of cameras should be as small as possible. In the second, the goal is to maximize the monitored area for a fixed number of cameras. In both of these problems, restrictions on the ability of the cameras, like limited depth of view or range of vision are taken into account. We present solution approaches for these problems and report on results of their implementations applied to an authentic problem. We also consider a bicriteria problem with two objectives: maximizing the monitored area and minimizing the number of cameras, and solve it for our study case.

Keywords: Bicriteria, Event monitoring, Art gallery problem, Set cover problem, Visibility

Introduction

Surveillance by cameras is one of the main tools in order to increase civil security. The efficiency of their usage is highly depending on their location. In this paper we focus on the camera location problem for public events, where crowds of people gather in one place. The main objectives which organizers have to achieve when locating these cameras is to maximize the area coverage and to minimize the costs (which can be assumed to be proportional to the number of cameras).

In the following we consider two different tasks: monitoring the whole area with a minimum number of cameras and monitoring the largest possible part of the area with a fixed number of cameras. As a starting point we take one problem from computational geometry - the art gallery problem [11]. We consider different solution approaches, developed from the art gallery problem and set cover problem models. Restricting properties of the cameras, such as limited depth of view, limited range of vision, etc. are taken into account. We also go further and consider the bicriteria problem, in which we maximize the monitored area and minimize the number of cameras.

1. Background

The well-known art gallery problem is originally motivated by the real life problem of guarding an art gallery and answering the following question: How many points (guards) in a polygon (art gallery) are needed, so that for every point in the polygon there is at least one guard who can see this point. A point is said to have a guard if a line segment from this point to the guard does not leave the polygon. Clearly, this problem can also be applied to the camera location problem.

There are some variations of the art gallery problem, including those that differ in guards' types. We distinguish the following:

- Vertex guards: the cameras can be placed only at the corner points of the area.
- Perimeter guards: the cameras can be placed everywhere on the boundary of the area.
- Point guards: the cameras can be placed everywhere in the area.

In the case of perimeter guards or point guards, one can also distinguish between the discrete and continuous case. In the discrete case the possible locations for guards are predefined and their number is finite. In the continuous case we have an infinite number of possible locations which consist of topologically connected pieces.

^{*} This work was partially supported by BMBF grant 13N12826.

1.1 Upper bounds

There are different approaches to solving an art gallery problem. One can find for example an upper bound of the minimum number of guards. Fisk proved in [7], that for a simple polygon with n vertices at most $\lfloor \frac{n}{3} \rfloor$ vertex guards are sufficient and sometimes necessary. In order to show that, one must first find a triangulation of the polygon, which is a polygonal decomposition into a set of non-intersecting triangles. Then the triangle vertices must be colored by means of three colors, so that no two adjacent vertices share the same color. Since the dual graph to the triangulation is a tree, in order to find a valid coloring, one can start with a triangle corresponding to one of the leaves in the dual graph and then continue with adjacent triangles. Each color class is a valid guard set. A polygon triangulation can be done in $O(n \log n)$ time and the coloring in $O(n)$ [8], which gives us in total $O(n \log n)$ time complexity.

This upper bound for a minimum number of guards is valid only for guards which can see in any direction, i.e. have a 360° range of vision. There are some attempts to solve the problem for guards with range of vision of 180° ([1], [12]). The best known result is presented in [12] and states that the minimum number of point guards with range of vision equal to 180° is, as in the previous case, $\lfloor \frac{n}{3} \rfloor$. The problem is solved by dividing the polygon with *good cuts* into smaller polygons. A cut, that decomposes a polygon with n edges into two polygons with n_1 and n_2 edges, is called good cut, if it satisfies the following condition:

$$\lfloor \frac{n_1}{3} \rfloor + \lfloor \frac{n_2}{3} \rfloor \leq \lfloor \frac{n}{3} \rfloor$$

The authors showed that any polygon with n edges either contains a good cut, or can be monitored with at most $\lfloor \frac{n}{3} \rfloor$ point guards with range of vision of 180° .

1.2 Exact solutions for vertex guard problem

An exact way to solve the problem with vertex guards is to decompose the polygon into visibility cells ([5]) and then solve a set cover problem. A visibility cell in a simple polygon P is a maximally connected subset of P with the property that each point in a cell is seen by the same set of vertices of P . Two points in a polygon see each other if a line segment connecting these point lies in the polygon and does not cross its boundary. In figure 1 an example of a polygonal decomposition into visibility cells is presented. The idea of the decomposition technique is the following: from each potential guard vertex p shoot a ray through every vertex v seen by p and take a line segment from v until the point where the ray hits the boundary of P .

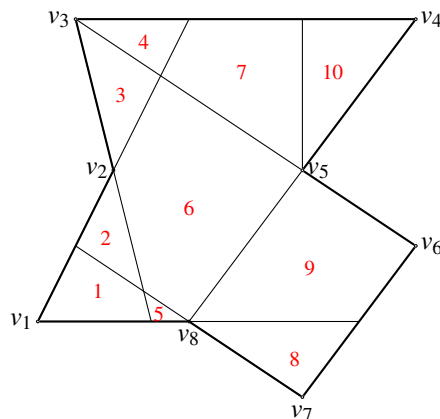


Figure 1. Polygon decomposition into visibility cells.

After this decomposition into visibility cells is performed, a set of visible cells is assigned to each vertex and the problem can be solved as a set cover problem defined as follows: Given a set U of some elements and a set S of n sets whose union equals U , identify the smallest subset of S whose union equals U . Using the polygon of figure 1 with its visibility cells decomposition as example, we can define the set U as the set of all cells, i.e. $U = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ and the following assignment of

visible cells to each vertex as the set S

- v_1 {1, 2, 5, 6, 7, 9, 10} := S_1 ,
- v_2 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} := S_2 ,
- v_3 { 3, 4, 5, 6, 7, 8, 9, 10} := S_3 ,
- v_4 {1, 2, 3, 4, 5, 6, 7, 10} := S_4 ,
- v_5 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} := S_5 ,
- v_6 {1, 2, 3, 5, 6, 8, 9 } := S_6 ,
- v_7 { 2, 3, 4, 6, 7, 8, 9 } := S_7 ,
- v_8 {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} := S_8 .

Obviously it holds $U = \bigcup_{i=1}^8 S_i$. However here we need only one set S_2, S_5 or S_8 in order to cover the whole U . This can be interpreted as follows: only one guard at one of the vertices v_2, v_5 or v_8 is needed in order to monitor the whole polygon.

The set cover problem is known to be NP-hard [9] such that its exact solution may take prohibitively long. It is therefore usually solved by approximate algorithms or heuristics, e.g. the following greedy heuristic: In each iteration a current solution is extended by adding one vertex from which the largest number of not yet monitored cells can be seen. The same approach, consisting of decomposition into visibility cells and solving set cover problem, can also be applied to any discrete problem without restricting only to vertex guards, i.e. when there are beside vertices some other points where the guards can be located.

The determination of the set of a polygon’s visibility cells takes $O(n^3 \log n)$ [5] time and the exact solution of the set cover problem takes $O(m2^n)$, where m is the number of visibility cells. The greedy heuristic runs in $O(mn)$ time [6]. Therefore the exact algorithm for solving the vertex art gallery problem runs in $O(m2^n)$ time, whereas the greedy solution can be found in $O(n^3 \log n + mn)$.

When it comes to the problem to place a fixed number of guards so that the largest area is monitored, one can again use the greedy heuristic: In each iteration a new guard is placed at a vertex, from where the largest part of the not monitored area is seen. One can also use this approach for solving the previous problem, where the whole area was to be monitored, and get similar results. The disadvantage here is that we have to compute areas of all visibility cells, which increases the running time.

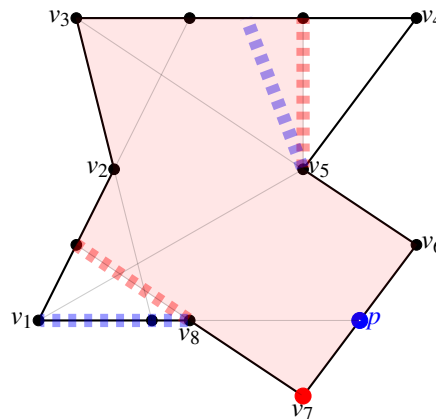


Figure 2. Induced segments on polygon boundary.

Most of the results presented in the literature only deal with discrete problems. To the best of our knowledge, there is just one publication [10] on the optimal placement of the guards solving the perimeter problem for one guard. The idea of the solution is to decompose the polygon boundary into *induced segments*, which are obtained similarly to visibility cells: from each potential guard vertex p shoot a ray through every vertex v seen by p and take the point where the ray hits the boundary of P . These segments have the property, that the visibility polygons for the points on the same segment are closely related. In figure 2 the decomposition of a polygon boundary into induced segments is presented. Consider segment (v_7, p) . One can see a visibility polygon, i.e. a polygon consisting of visible cells, for a vertex v_7 . The authors of [10] showed, that when moving along an induced segment the visibility polygon gains some and loses other triangular areas. To solve a problem for placing a guard on an induced segment one must solve a linear fractional problem, each summand of which corresponds to one of the variable triangles. The algorithm iterates over the segments and then the guard is chosen from the solutions on segments. The running time for the algorithm is $O(n^4 \log b \log n)$ for the approximation to b bits of accuracy.

1.3 Guards as cameras

For the original art gallery problem the following assumptions on guards are made: they have unlimited range of vision (360°), they have unlimited depth of field. These requirements on cameras are rather unrealistic and make the model not suitable for real-life problems.

In [3] another approach with dropped assumption on the depth of view was introduced. The authors also performed the decomposition of the polygon, but now with some predefined grid (see figure 3a). Then an assignment of cells to vertices or some other guard candidates is done. This time a cell is visible for a guard if the segment from the guard to the midpoint of the cell lies in the polygon and if the length of the segment is less than or equal to the given depth of field (figure 3b).

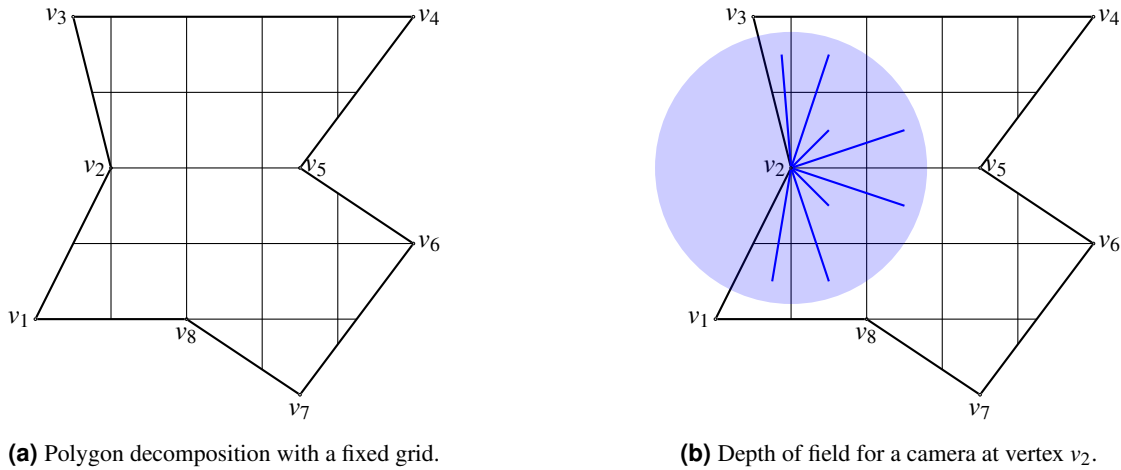


Figure 3. Approach for finding optimal locations for cameras with limited depth of field.

The idea of limiting the range of vision was presented in [4]. The authors considered the case of cameras with 90° range of vision and calculated the visible cells for the following cameras positions: $0^\circ - 90^\circ$, $45^\circ - 135^\circ$, $90^\circ - 180^\circ$ etc. For our tests, presented in the following section, we used a similar idea. For a vertex angle of 90° and camera range of view of 35° we considered the camera positions $0^\circ - 35^\circ$, $27.5^\circ - 62.5^\circ$ and $55^\circ - 90^\circ$ (figure 4).

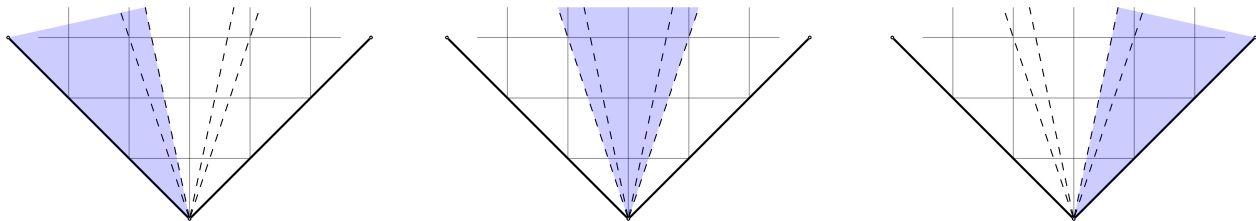


Figure 4. Possible coverage for a guard with 35° range of vision at a corner with an angle of 90° .

2. Bicriteria problem formulation

Now we consider a problem of finding a set of Pareto optimal solutions for the problem with two conflicting objective functions. On one hand the goal is to maximize the monitored area S , on the other hand the amount of cameras N must be minimized due to limited budget. So the problem has the following form:

$$\max S, \quad \min N$$

One of the methods to solve multicriteria optimization problems, in particular bicriteria problems, is to use ε -constraint method [2]. In this method only one of the objectives is optimized and the others become constraints. In our case we transform to constraint the objective which minimizes number of cameras since it always takes only integer values. So we start with solving the problem for 1 camera and then continue until at some point the monitored area can not be maximized. So we have:

$$\begin{aligned} \max \quad & S \\ \text{s.t.} \quad & N \leq \varepsilon \quad \varepsilon = 1, 2, \dots \end{aligned}$$

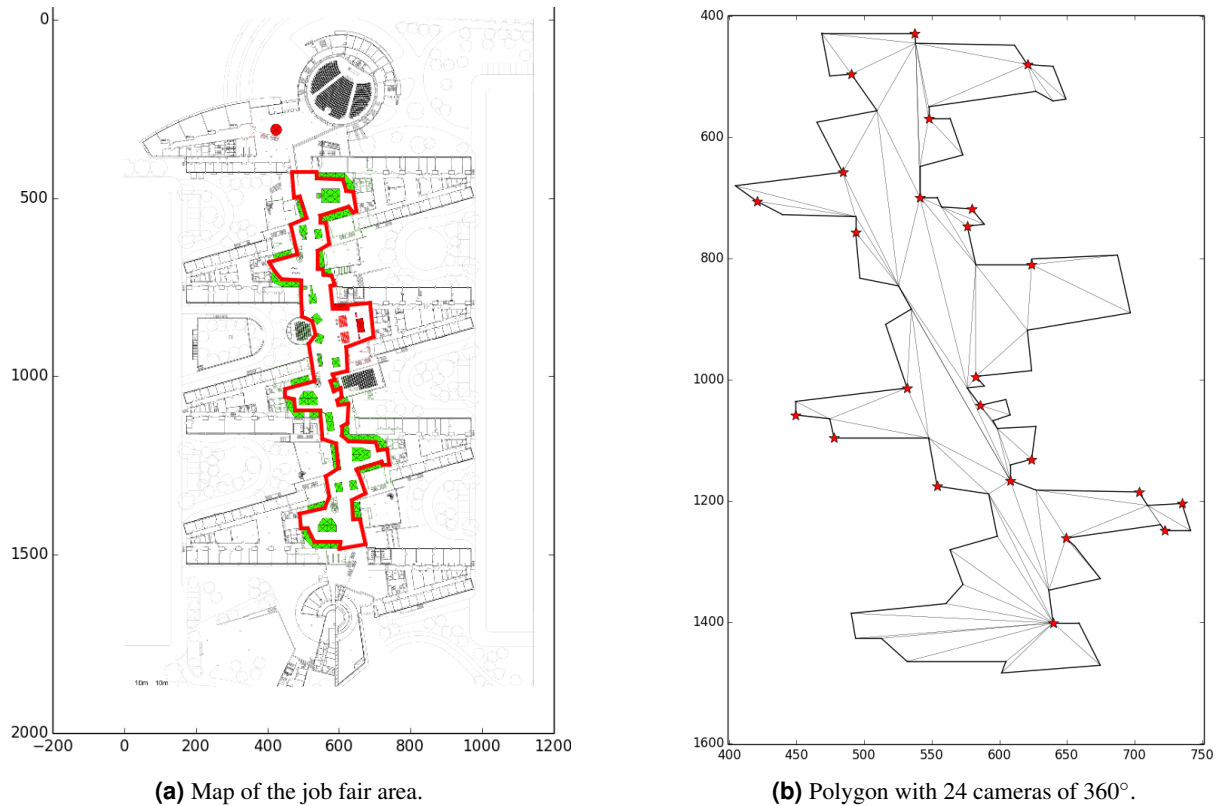


Figure 5. Job fair area.

3. Case study of a job fair

We implemented the presented algorithms and ran them for a real life problem, the job fair in a given area for which the map was available. On this map we fitted a polygon with 85 edges, which is presented in figure 5a.

We start by analyzing the situation for 360° cameras. The upper bound for the minimum number of cameras for monitoring the whole job fair polygon with $n = 85$ edges is $\lfloor \frac{n}{3} \rfloor = 28$. With the help of our implemented algorithm (triangulation and coloring) we obtained however a smaller number of cameras for the job fair: in figure 5b one can see a possible assignment of 24 cameras at the corner points of the area.

A better solution we got after decomposing the polygon into visibility cells and applying the greedy heuristic. In figure 6 one can see the obtained solution with 11 cameras. We also found the optimal solution by solving the NP-hard set cover problem exactly. The algorithm took much more time and the optimal solution contains 10 cameras that monitor the whole polygon.

We also calculated the solution for the case, where the goal is to monitor the largest part of the area with 1, 2 or 3 cameras. The results are the following: if one wants to locate a camera at the best vertex of the polygon then more than half of the area is monitored (see figure 7a). With 3 cameras even 88% of the area can be monitored (figure 7c).

For cameras with 180° range of view, we obtained, for the problem of monitoring the whole area, a solution with exactly $\lfloor \frac{85}{3} \rfloor = 28$ cameras (see figure 8).

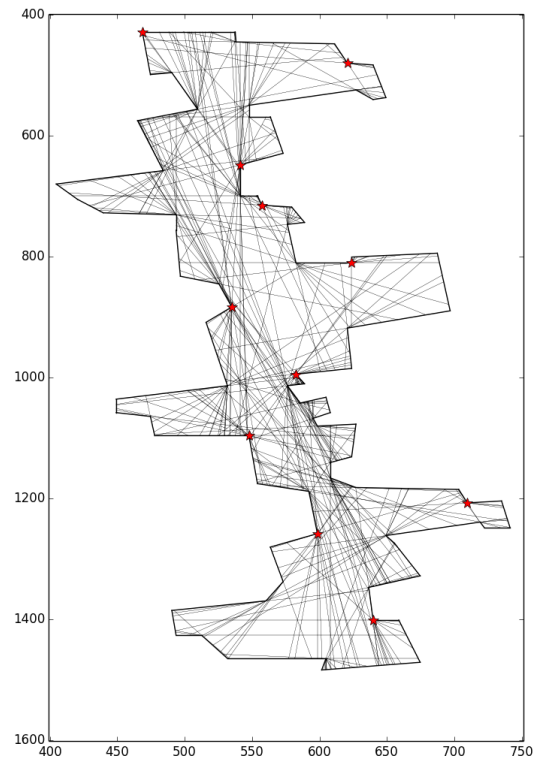
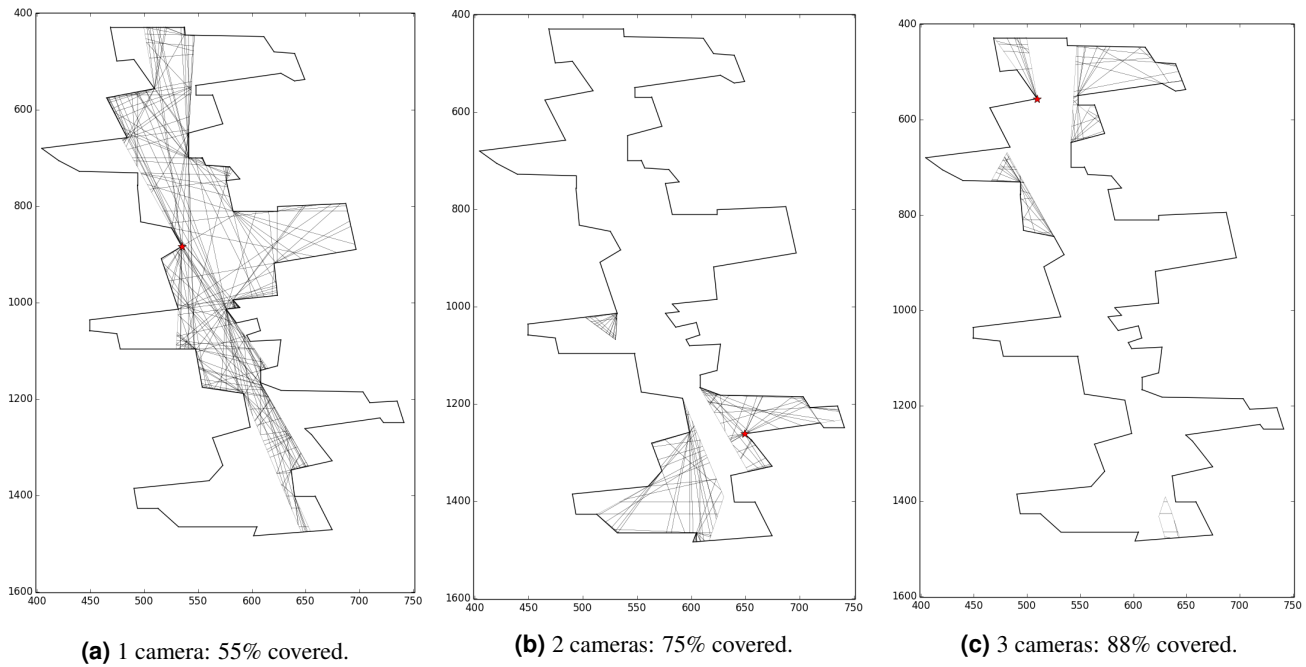


Figure 6. Polygon with 11 cameras of 360° .



(a) 1 camera: 55% covered.

(b) 2 cameras: 75% covered.

(c) 3 cameras: 88% covered.

Figure 7. Monitoring the polygon with 1, 2 and 3 cameras.

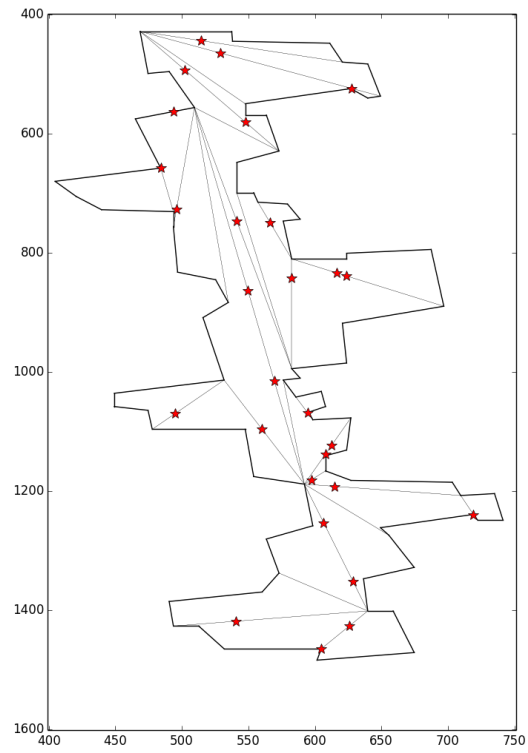
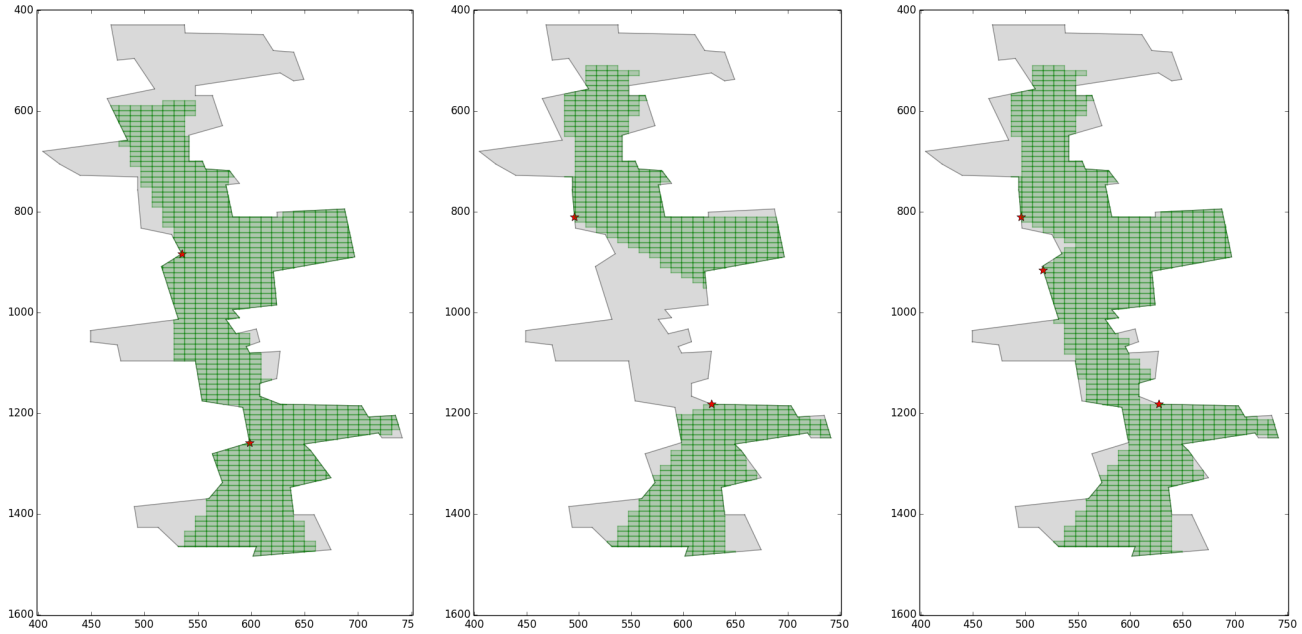


Figure 8. Polygon with 28 cameras of 180° .

For cameras with limited depth of view we solved the problem of covering the most part of the area. With 2 cameras, that have range of vision of 360° , we achieve a covering of 69% of the area (see figure 9a). Under the assumption that the cameras have also limited range of vision, for example 140° , we get a comparable coverage result by adding one camera to the solution (see figure 9c).



(a) 2 cameras of 360° :
69% covered.

(b) 2 cameras of 140° :
51% covered.

(c) 3 cameras of 140° :
71% covered.

Figure 9. Monitoring the polygon with cameras with limited depth of field.

We also found a Pareto frontier for the cameras with limited range of vision (140°) and depth of view (see figure 10). Using this frontier, one can decide at which point additional cameras or additional costs for this cameras make no sufficient improvement in the monitoring area.

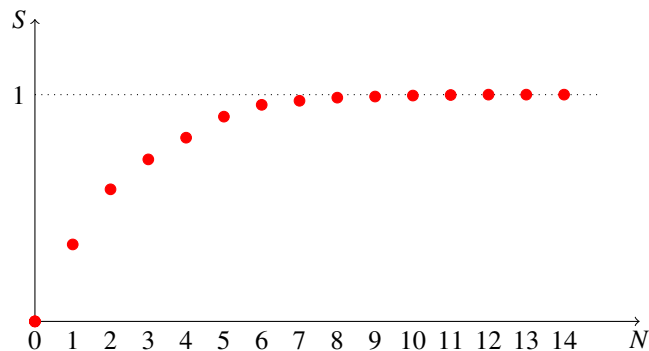


Figure 10. Pareto frontier for the problem with two conflicting objectives.

Conclusion

For the problem of finding optimal locations for surveillance cameras we adapted different approaches presented in the literature. Some of them are focused on getting an upper bound of the minimum number of cameras, which can dramatically differ from the optimal solution. The solution by means of set cover problem shows the best results even with help of heuristics. Decomposition of the polygon into grid cells can be obtained much easier and faster than the visibility cell decomposition, moreover it gives an opportunity to solve the problem under different assumptions on cameras capabilities. The presented bicriteria approach finds a set of Pareto optimal solutions for the problem with two conflicting objective functions: monitoring of the largest area and minimization of the number of surveillance cameras.

References

- [1] György Csizmadia and C  za T  th. Note on an art gallery problem. *Computational Geometry*, 10:47–55, 1998.
- [2] Matthias Ehrgott. *Multicriteria Optimization*. Springer Berlin Heidelberg, 2 edition, 2005.
- [3] Alan T. Murray et al. Coverage optimization to support security monitoring. *Computers, Environment and Urban Systems*, 31:133–147, 2007.
- [4] Antonio Sforza et al. Optimization models in a smart tool for the railway infrastructure protection. *Critical Information Infrastructures Security*, pages 191–196, 2013.
- [5] Prosenjit Bose et al. Efficient visibility queries in simple polygons. *Computational Geometry*, 23:313–335, 2002.
- [6] Thomas H. Cormen et al. *Introduction to Algorithms*, pages 1117–1122. MIT Press, 3 edition, 2009.
- [7] Steve Fisk. A short proof of chv  tal’s watchman theorem. *Journal of Combinatorial Theory*, 24:374, 1978.
- [8] A. A. Kooshesh and B. M. E. Moret. Three-coloring the vertices of a triangulated simple polygon. *Pattern Recognition*, 25:443, 1992.
- [9] Bernhard Korte and Jens Vygen. *Combinatorial Optimization*, volume 21. Springer Berlin Heidelberg, 5 edition, 2012.
- [10] Simeon Ntafos and Markos Tsoukalas. Optimum placement of guards. *Information Sciences*, 76:141–150, 1994.
- [11] Joseph O’Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, 1987.
- [12] Csaba D. T  th. Art gallery problem with guards whose range of vision is 180° . *Computational Geometry*, 17:121–134, 2000.