

Greedy Heuristics for Solving the Weighted Orthogonal Art Gallery Problem

—

1 Introduction

The *Orthogonal Art Gallery Problem* (OAGP) is one of many variants of Art Gallery Problem (AGP). AGP asks for a set of points G of minimal cardinality on some polygon P such that for each point $y \in P$ there is $x \in G$ such that $xy \subset P$. Set P is called guard set of P , and the points from G as guards. The orthogonal AGP consider on arbitrary polygon but whose angles are 90^{deg} and 270^{deg} . This problem is stated by Victor Klee in 1973. [2]. The problem is motivated from installing the cameras in a building (or gallery) such that the whole surface of the building is covered. Orthogonality comes out from the walls which are orthogonal to some other wall in a building. A variant of the OAGP for which we are interested in this studies allows only that guards are positioned at the edges of polygon P . This problem is known to be NP-hard [1]. Let us denote set V as a set of all vertices of polygon P .

Since then, many approaches have been proposed to solve the problem.

FiXme: TODO: work on literature approaches

2 Preliminaries

Solvers which solves IP model for OAGP problem are one of the most efficient techniques to approach this problem. It is known that the problem is related to the known Minimal-Set-Cover (MSC) problem.

Let us suppose we are given a discretization $D(P)$ of the polygon P (with a family of rectangles). Then we relate the OAGP with the known MSC problem. Family $\mathcal{F} \subseteq D(P)$ of nonempty sets is given as: $S_i \in \mathcal{F}$ iff it includes any point from $D(P)$ which is visible from guard $i \in V$. Note that set S_i includes a point p_i which can also included by some other guard $j \in V$, $i \neq j$. So, the task of OAPG becomes finding a minimal cardinality cover $\mathcal{C} \subseteq \{S_1, \dots, S_n\}$ of set $D(P)$, that is

$$\bigcup_{c \in \mathcal{C}} c = D(P).$$

The Integer Programming model for the Set Cover problem is known, and is as follows:

$$\sum_{i=1}^n x_i \longrightarrow \min \quad (1)$$

$$\text{s.t.} \quad (2)$$

$$\sum_{j \in V} a_{ij} x_j \geq 1 \quad (\forall p_i \in D(P)) \quad (3)$$

$$x_j \in \{0, 1\}, j \in V, \quad (4)$$

where $a_{ij} = \begin{cases} 1, & p_i \in V(j), \\ 0, & \text{otherwise} \end{cases}$ and $x_i = \begin{cases} 1, & \text{if the point } i \in \mathcal{C}, \\ 0, & \text{otherwise,} \end{cases}$

presents a MIP model to solve OAGP where $V(j)$ is the set of all points from $D(P)$ that are visible from j -th vertex of P . Set $Z = \{j \in V \mid x_j = 1\}$ represents a solution of the problem. Constraint (3) enforces that any point $p_i \in D(P)$ will be visible from at least one guard from Z .

If we want to add weights into the OAGP problem such that all guard have no equal weight (of 1), we obtained the weighted variant of OAGP, that is the WOAGP problem. This can be augmented that the prices of cameras do not need to be equal and these prices might be different due to the quality (of the range) of the cameras that are installed at some specific corners. Let us suppose that for any guard p_i we assign a price w_i , $i = 1, \dots, n$. Then, the above model is may be extended to its weighted version where the objective function has to be replaced by

$$\sum_i x_i w_i,$$

and all other constraints remain the same.

In order to solve this model, we propose to use a general purpose solver CPLEX.

3 Algorithmic Approaches for WOAGP

Since the WOAGP can be seen as the Weighted Set Cover problem (WSCP) and since the best known heuristics to solve WSCP is an enhanced greedy heuristic, our idea is to apply greedy algorithms to solve WOAGP. These algorithms produce a solution of reasonable quality within a short interval of time. Efficient of greedy heuristics is related to a greedy criterion utilized to expand current (non-complete, that is partial) solution to complete one. Among all candidates (solution components for expansion, that is not-yet-considered guards, we choose one with the smallest greedy value etc. until the solution is complete (i.e., cover all regions of the polygon). Pseudocode of Greedy heuristic is given in Algorithm 1

Algorithm 1 Greedy Heuristic

```
1: Input: an instance of a problem
2: Output: A (feasible) non-expandable solution (or reporting that no feasible solution)
3:  $s^P \leftarrow ()$  // partial solution set to empty solution
4: while  $\text{Extend}(s^P) \neq \emptyset$  do
5:   Select component  $e \in \text{Extend}(s^P)$  // w.r.t. some criterion
6:   Extend  $s^P$  by  $e$ 
7: end while
```

3.1 Greedy Heuristic based on Price-per-Unit

Our greedy criterion to expand current partial solution s^P is based on the WOAGP characteristics, that is considering not-yet-covered regions of polygon P , that is, its discretisation $D(P)$. For each not yet considered guards p_i , we assign the region he is able to see by S_{p_i} . As the next candidate to extend s^P , we choose not-yet-considered guard p^* which is able to cover his part S_{p_i} by smallest price per unit, among the other not considered guards. More precisely, greedy criterion g is given as:

$$g(s^P, p_i) = \frac{w_{p_i}}{S_{p_i}}, \quad (5)$$

and p^* that minimizes $g(s^P, p^*)$ value is chosen as an extension of the current partial solution. Afterwards the region S_{p^*} is further then dropped off from P , i.e., $P = P \setminus S_{p^*}$, and update $s^P = s^P \cup \{p^*\}$. These steps are repeated until s^P is complete, which means $P = \emptyset$ (polygon P is covered by the guards from s^P).

3.2 An Alternative Greedy Heuristic

FiXme: TODO: work in progress... Greedy criterion:

- prefer those guards with a smaller cost w_i ;
- If there are more guards with the smallest cost, prefer one with the largest S_{p_i} .

FiXme: TODO: work in progress... Another approach DRAGAN:

- introduce a penalty function
- introduce a term “incorrect point”. For a point from $D(P)$ we say that it is incorrect if it is not covered by any guard from a current partial solution s^{ps} .
- let $incorrect_{total}$ be the total number of incorrect points from discretization $D(P)$
- let w_{total} be the total sum of all weights among all vertexes
- involve the penalty function in the objective function

$$obj(s^{ps}; v) = \sum_{i \in s^{ps} \cup \{v\}} w_i + incorrect_{total}$$

or

$$obj(s^{ps}; v) = \frac{\sum_{i \in s^{ps} \cup \{v\}} w_i}{w_{total}} + \frac{incorrect_{total}}{|D(P)|}$$

FiXme: moguće i skaliranje ove dvije vrijednosti?

- start Greedy with empty solution
- in each iteration add such a guard to s^{ps} for which the obj value is minimal
- end when $incorrect_{total}$ becomes 0

3.3 An improvement of Greedy Solution

Local search and Large Neighborhood search **FiXme: TODO...** Local search (LS) technique is a basis of the LS-based metaheuristics. Let us define the structure of neighborhood. We define them over the set of guards. The first configuration of neighbourhood of some partial solution is operation of swapping; that means from a given partial solution s^P , we choose one guard from s^P and then pick another guard from $V \setminus s^P$ w.r.t. value $obj(;)$.

Another criterion for the neighbourhood's configuration (k -th) would be taking arbitrary k guards $\{p_1, \dots, p_d\}$ from s^P , then drop them from s^P . Afterwards, for each p_i which are dropped off, we take the guard which is nearest by distance from p_i and add them into s^P .

4 Computational Results

We used the instance of a specific OAGP and assigned the weights to each vertex of polygons. We have generated three kind of benchmarks:

- *random benchmarks*. For each vertex i of polygon P we take a random value $w_i \in \{X_1, \dots, X_q\}$ **FiXme: da li je domen ovdje OK?** as its weight (X_i are some random values for prices), $q \in \mathbb{N}$.
- *topological-type benchmarks*. For each vertex i of polygon P let us denote by l_i and l_{i+1} the lengths of edges that comes out of vertex i . Then, $w_i := \frac{l_i + l_{i+1}}{2}$. This can be augmented by the fact that if the arithmetic length of both edges that comes out of vertex i is longer, it is expected that vertex is a guard can see a larger pieces of polygon P . This implies that the range of camera i has to be larger, which again means that it has to be of a higher price.
- ...

5 Conclusions and Future Work

References

1. S. L. Devadoss and J. O'Rourke. *Discrete and computational geometry*. Princeton University Press, 2011.
2. J. O'Rourke. *Art gallery theorems and algorithms*, volume 57. Oxford University Press Oxford, 1987.

Algorithm 2 Local Search

```
1: Input: a (partial) solution  $s_{greedy} = \{v_1, \dots, v_k\}$ ,  $iter_{allow} > 0$ ., probability  $p$ 
2: Output: and improved solution  $s$ 
3: while  $iter < iter_{allow}$  do
4:    $D(P)_s$  the number of points covered by  $s_{greedy}$ 
5:   if  $|D(P)_s| == |D(P)|$  then // a complete solution
6:     return  $s_{greedy}$ 
7:   end if
8:   choose random vertex  $v_{i'}$  from  $s_{greedy}$ 
9:    $v_{i''}, v_{i'''}$   $\leftarrow$  two neighbor vertices of vertex  $v_{i'}$ 
10:   $s_{greedy}^1 \leftarrow s_{greedy} \cup \{v_{i''}\} \setminus \{v_{i'}\}$ 
11:   $s_{greedy}^2 \leftarrow s_{greedy} \cup \{v_{i'''}\} \setminus \{v_{i'}\}$ 
12:  if  $|D(P)_{s_{greedy}^1}| > |D(P)_{s_{greedy}}|$  then
13:     $s_{greedy} \leftarrow s_{greedy}^1$ 
14:  end if
15:  if  $|D(P)_{s_{greedy}^1}| == |D(P)_{s_{greedy}}|$  then
16:     $s_{greedy} \leftarrow s_{greedy}^1$  with probability  $p$ 
17:  end if
18:  if  $|D(P)_{s_{greedy}^2}| > |D(P)_{s_{greedy}}|$  then
19:     $s_{greedy} \leftarrow s_{greedy}^2$ 
20:  end if
21:  if  $|D(P)_{s_{greedy}^2}| == |D(P)_{s_{greedy}}|$  then
22:     $s_{greedy} \leftarrow s_{greedy}^2$  with probability  $p$ 
23:  end if
24:   $iter \leftarrow iter + 1$ 
25: end while
26: return  $s_{greedy}$ 
```
