On Solving the Weighted Orthogonal Art Gallery Problem under regular grid discretization by greedy methods and their hybridizations

Abstract

In this paper...

1. Introduction

For a given polygon P, Art Gallery Problem (AGP) asks for a set of points G of minimal cardinality, such that for each point $y \in P$ there is $x \in G$ such that $xy \subset P$. We say that the point y is covered by the point x, or y is visible from x. Set G is called guard set of P and the points from G as guards. In *The Orthogonal Art Gallery Problem* (OAGP) edges of the polygon are only horizontal and vertical, i.e. angles between adjacent edges are 90° or 270°. The original AGP was stated by Victor Klee in 1973. [8]. The problem can be motivated from installing the cameras inside a building (or gallery) such that the whole area of the building is covered. Orthogonality constraint naturally comes out from the orthogonality of the walls in buildings. Kahn et al. in [6] formulated and proofed that $\lfloor \frac{n}{4} \rfloor$ guards are sufficient to cover an orthogonal polygon with n vertices. In this study, we are interested in the variant of the OAGP which allows only that guards are positioned at the vertices of polygon P. This restricted problem is known to be NP- hard in [10, 7]. In real situations (like installing the cameras in a building) , it is justified to assume that the prices of cameras are not equal and may depends on several factors, like the quality of a camera (respecting the range of spectrum of view) or installation price at some specific parts of the building (like corners or tight places). In Weighted Orthogonal Art Gallery Problem (WOAGP), which we finally consider in this paper, the task is to place guards on some vertices of the orthogonal polygon which cover all points from P, such that the total sum of prices assigned to all chosen vertices is minimal.

It is well known that AGP can be reduced to the Minimum Set Cover Problem (MSCP) by a discretization of the set of all points of the polygon P. The appropriate discretization should be performed in such a way that if each point from the discretized set $D(P)$ is covered, then the whole polygon P is covered. After the discretization is made, for each vertex of the polygon P, a set of visible points from $D(P)$ is determined. In that way, the problem of determining the minimum number of guards covering the entire polygon is reduced to determining the minimum number of subsets of points, such that each point from $D(P)$ is included in at least one of the chosen subsets, which is MSCP. Analogously, WOAGP can be reduced to the Weighted Minimum Set Cover Problem (WMSCP).

Couto et al. presented an exact and efficient exact algorithm for the OAGP based on preprocessing and refinement phases of the the discretized instance. in [2]. In [3] an approximate solution of the minimum vertex guard problem, which can be computed in $O(n^4)$ time and the size of the solution is at most $O(\log n)$ times the optimal. After that, on these constructed sets Johnson's approximation algorithm [5] for the MSCP is applied. An anytime algorithm to compute successively better approximations of the optimum to Minimum Vertex Guard is proposed in [11]. A major idea of this approach is exploring dominance of visibility regions to first detect pieces that are more difficult to guard. The same problem is solved in [12] by applying successive approximations from [11]. Tozoni et a. [13, 14] presented an exact Integer linear programming (ILP)-based algorithm, which iteratively generates upper and lower bounds through the resolution of discretized space of the AGP. Although many variants AGP are present in literature, WOAGP has not been so intensively studied, which motivated us to consider this problem. Analysis of many greedy-like heuristic for the WMSCP was presented in [15]. More detailed overview of the extensive literature regarding SCP and AGP is out of the scope of this paper and for further reading we suggest review papers [1, 9, 4].

1.1. Problem definition

FiXme: TODO: Ovo mozda sad i ne treba?

1.2. Main contributions

The main contributions of this paper are:

- We developed a novel greedy approach which is based on balancing the trade off between the total sum of guards' costs and the total number of not yet covered points from the discretization.
- We developed a specially designed shaking procedure, which perturbs the current partial solution in such a way that some vertices are replaced with others, where the replacement is based on measuring distances between points in the polygon. FiXme: TODO: popraviti
- Both an existing and the novel greedy algorithms are hybridized with the proposed shaking procedure and with the ILP.
- We considered different types of weights for our benchmarks, based on an approximation of the costs in real situations.
- In a comprehensive computational experiment, FiXme: TODO: popraviti different variants of approximative algorithms we tested and analysed.

1.3. Discretization

FiXme: TODO: Milan treba da ovo objasni proces transofrmacije, zasto uzimamo ovakav tip diskretizacije, kako se onda problem svodi na SC, da obajsni zasto diskretni grid ne mora dati potpunu pokrivenost poligona – dvije slike. Proces transfomracije objasniti kratko (mozda i pseudokod ako je moguce da se prikaze)...

2. Exact methods

FiXme: TODO: Marko In this section we present the known exact ILP and constraint programming (CP) methods for solving WMSCP, which are used in the rest of the paper.

2.1. Integer linear programming method

Let us suppose we are given a polygon P and a discretization $D(P)$ of P. Our task is to cover all points from $D(P)$ by some vertices from P such that the sum of their weights is minimized. The problem is related to the known Minimal Weighted Set-Cover (MWSC) problem. Family $\mathcal{F} \subseteq D(P)$ of nonempty sets consists of sets $S_i \in \mathcal{F}$ which include point $p \in D(P)$ that is visible from guard $v_i \in V$, that is $pv_i \subset P$. Note that set S_i includes a point p_i which can also included by some other guard $v_j \in V$, $i \neq j$. In this was, the task is equivalent of finding a minimum cardinalty cover $\mathcal{C} \subseteq \{S_1, ..., S_n\}$ of the set of points $D(P)$, that is

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

The IP model for the Weighted Set Cover problem is well-known in the literature, and is here adapted for our task as follows:

$$\sum_{i=1}^{n} w_i x_i \longrightarrow \min \tag{1}$$

$$\text{s.t.} \tag{2}$$

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

$$x_j \in \{0, 1\}, j \in V, \tag{4}$$

where $a_{ij} = \begin{cases} 1, p_i \in V(j), \\ 0, \text{otherwise} \end{cases}$ and $x_i = \begin{cases} 1, \text{ if the point } p_i \in \mathcal{C}, \\ 0, \text{otherwise}, \end{cases}$
where $V(j)$ is the set of all points from $D(P)$ that are visible from j-th vertex of P. Set $Z = \{j \in V \mid x_i = 1\}$ represents a solution of the problem w.r.t. discretization $D(P)$ of polygon P. Constraint (3) enforces that any point $p_i \in D(P)$ will be visible from at least one guard from Z.

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

2.2. Constraint programming method

An equivalent CP model was implemented and tested by CP Optimizer. In this case, Constraint (3) is transformed into

$$\bigvee_{j \in V} (a_{ij} \wedge x_j) = 1, \tag{5}$$

whereas the other constraints and the objective function are the same like in the above MIP model. Note that CP appraoch works in a branch-and-bound manner imploying a constriant propagation and domain filtering **??**.

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** Extend(s^P) $\neq \emptyset$ **do**
5: Select component $e \in$ Extend(s^P) // w.r.t. some criterion
6: Extend s^P by e
7: **end while**

3. Algorithmic Approaches for WOAGP

3.1. Greedy approaches for solving WOAGP

Since the WOAGP can be seen as the WMSCP and since the best known heuristics to solve WSCP is an enhanced greedy heuristic FiXme: Marko, treba nam referenca da je greedy najbolja heuristika, our idea is to transform a WOAGP instance into a WSC problem instance and then apply greedy algorithms to solve WOAGP. These algorithms produce a solution of reasonable quality within a short interval of time. Efficiency of such greedy heuristics is related to a greedy criterion utilized to expand current (non-complete, i.e., 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 until the solution is complete (i.e., cover all points from $D(P)$). A general pseudocode of Greedy heuristics is given in Algorithm 1

3.1.1. An existing greedy method

FiXme: TODO: Milana

Concerning the literature for WMSC problem **??**, one of the most efficient greedy heuristic was based on the following greedy rule:

$$g(s^p, p_i) = \frac{w_{p_i}}{f(s^p \cup \{p_i\}) - f(s^p)}, \tag{6}$$

where $f(s^p) = |\bigcup_{s \in s^p} s|$. This heuristic is also a approximation algorithm which ensures $O(\log(n))$ approximation factor.

3.1.2. A Novel Greedy Heuristic

FiXme: TODO: Dragan In this subsection we present a novel greedy function, which is used in the greedy algorithm. Let us 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 denote by $incorrect_{total}$ the total number of incorrect points from discretization $D(P)$ for the current partial solution. Let w_{total} be the total sum of all weights among all vertices and $|D(P)|$ be total number of discretization set. The greedy function for the given partial solution s^{ps} and the guard v takes into account both the value of the objective function of the WOAGP applied to the partial solution if v is included in it and the total number of incorrect points, given by the formula:

$$g(s^{ps}; v) = \alpha \cdot \sum_{i \in s^{ps} \cup \{v\}} w_i + \beta \cdot incorrect_{total} \tag{7}$$

where α and β are positive real parameters.

Although many combination of parameters α and β may be used, in our investigation we used parameter α only to normalize the sum of prices, i.e. $\alpha = \frac{1}{w_{total}}$ and two values for β: $\beta = 1$ and $\beta = \frac{1}{|D(P)|}$. Thus, we consider two variants of the greedy function:

$$g_2(s^{ps}; v) = \frac{\sum_{i \in s^{ps} \cup \{v\}} w_i}{w_{total}} + incorrect_{total} \tag{8}$$

and

$$g_3(s^{ps}; v) = \frac{\sum_{i \in s^{ps} \cup \{v\}} w_i}{w_{total}} + \frac{incorrect_{total}}{|D(P)|} \tag{9}$$

From Equation (8) one can see that the greedy function g_2 mainly depends on the second term, since the value of the first one is not greater than 1. On other words, this function prefers such guards which inclusion in the partial solution will cover more points. On the other side, if the inclusion of some two guards covers the same number of points, then the function suggest the guard with lower cost. In the early stage of the greedy algorithm this approach enables that the guards which covers more points are preferable, while in a later stage, when less points remain uncovered, the function prefers the guards with lower cost.

In the Equation (9) the second term is also normalized and the trade off between two terms is more balanced. In this case, the algorithm does not ultimately prefer any of criteria for choosing next vertex. Instead, FiXme: TODO...

Ties occurred in the search are broken by using price-per-unit heuristic which i stated as follows. For each not yet considered guards p_i, we assign the region which is visible from p_i by $S(p_j)$. As the next candidate to extend s^P, we choose a guard p^* which covers set $V(j)$ by smallest price per unit, among the other candidate to extend s^P. More precisely, greedy criterion g is given as:

$$g(s^p, p_i) = \frac{w_{p_i}}{S_{p_i}}, \tag{10}$$

and p^* that minimizes $g(s^p, p^*)$ value is chosen as an extension of the current partial solution. In our experimental studies, we found out that this heuristic does not perform well on its own, but presents a reasonable tie-breaking mechanism and is able to boost quality of our greedy heuristics.

3.1.3. Partial calculation of objective function

In order to enable fast calculation of greedy functions described in previous subsections, we introduce several useful structures and auxiliary functions:

- structure `map<Point,list<Point>> Visibility` – for each point, we provide the list od guards which see that point;
- structure `map<Point,int> numberOfGuards` – for each point we keep the number of Guards in solution which see that point;
- structure `set<Point> CoveredPoints` – set of points covered by the partial solution;

- function `updateCoveredPointsAdd (vertex v)` – updates structures `CoveredPoints` and `numberOfGuards` by considering all new points which are covered by vertex `v`, when `v` is added to solution;

- fucntion `updateCoveredPointsRemove(vertex v)` – updates structures `CoveredPoints` and `numberOfGuards` by removing all points which are covered by `v`, when `v` is removed from solution.

3.1.4. A hybrid of the Greedy and shaking

FiXme: TODO: Ubaciti pricu o shaking $--->$ Dragan?

3.1.5. A Hybrid of the Greedy + CPLEX

CPLEX will soon or later degrades its performance w.r.t. instance size. On the other hand, the later stage of Greedy increase the chance to worsen the final greedy solution. So it makes sense to combine partial solutions generated by Greedy over a few interactions and then to use CPLEX up to completion of the partial solution. Our approach consists of the following steps:

1. Run a Greedy method up to K iterations (parameter) to obtain a partial solution C (therefore, $|C| = K$);
2. Take solution C and make it complete via. CPLEX model.
   - CPLEX solves corresponding sub-model which is formed by adding constraints $x_{p_i} = 1$, for all $p_i \in C$ into the existing WOAGP model;
   - we obtain a complete solution C';
3. return $f(C')$.

4. Computational Results

We used the instance of the specific OAGP up to a 200 vertices, and assigned the weights to each vertex of polygons. We included two kind of weights into these benchmarks:

- *topologically-based 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 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.

- *point-based benchmarks:* For each vertex j, we consider the number of points in $D(P)$ that are visible from the given vertex ($|V(j)|$). Based on this number, we assign prices to the vertices on the following way:

$$w_j = n \cdot \frac{|V(j)|}{|D(P)|}, j = 1, ..., n. \tag{11}$$

4.1. Results on exact methods

ovo za cp i 5min, što je Marko pominjao

5. Conclusions and Future Work

[1] A. Caprara, P. Toth, and M. Fischetti. Algorithms for the set covering problem. *Annals of Operations Research*, 98(1-4):353–371, 2000.

[2] M. C. Couto, C. C. De Souza, and P. J. De Rezende. An exact and efficient algorithm for the orthogonal art gallery problem. In *XX Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI 2007)*, pages 87–94. IEEE, 2007.

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

[4] S. K. Ghosh. Approximation algorithms for art gallery problems in polygons and terrains. In *International Workshop on Algorithms and Computation*, pages 21–34. Springer, 2010.

[5] D. S. Johnson. Approximation algorithms for combinatorial problems. *Journal of computer and system sciences*, 9(3):256–278, 1974.

[6] J. Kahn, M. Klawe, and D. Kleitman. Traditional galleries require fewer watchmen. *SIAM Journal on Algebraic Discrete Methods*, 4(2):194–206, 1983.

[7] M. J. Katz and G. S. Roisman. On guarding the vertices of rectilinear domains. *Computational Geometry*, 39(3):219–228, 2008.

[8] J. O'rourke. *Art gallery theorems and algorithms*, volume 57. Oxford University Press Oxford, 1987.

[9] Z.-G. Ren, Z.-R. Feng, L.-J. Ke, and Z.-J. Zhang. New ideas for applying ant colony optimization to the set covering problem. *Computers & Industrial Engineering*, 58(4):774–784, 2010.

[10] D. Schuchardt and H.-D. Hecker. Two np-hard art-gallery problems for ortho-polygons. *Mathematical Logic Quarterly*, 41(2):261–267, 1995.

[11] A. P. Tomás, A. L. Bajuelos, and F. Marques. Approximation algorithms to minimum vertex cover problems on polygons and terrains. In *International Conference on Computational Science*, pages 869–878. Springer, 2003.

[12] A. P. Tomás, A. L. Bajuelos, and F. Marques. On visibility problems in the plane–solving minimum vertex guard problems by successive approximations. In *ISAIM*, 2006.

[13] D. C. Tozoni, P. J. de Rezende, and C. C. de Souza. A practical iterative algorithm for the art gallery problem using integer linear programming. *Optimization Online*, pages 1–21, 2013.

[14] D. C. Tozoni, P. J. D. Rezende, and C. C. D. Souza. Algorithm 966: a practical iterative algorithm for the art gallery problem using integer linear programming. *ACM Transactions on Mathematical Software (TOMS)*, 43(2):1–27, 2016.

[15] F. J. Vasko, Y. Lu, and K. Zyma. What is the best greedy-like heuristic for the weighted set covering problem? *Operations Research Letters*, 44(3):366–369, 2016.