The Dispersive Art Gallery Problem

Christian Rieck
Department of Computer Science, TU Braunschweig, Germany

Christian Scheffer
Faculty of Electrical Engineering and Computer Science, Bochum University of Applied Sciences, Germany

Abstract

We introduce a new variant of the art gallery problem that comes from safety issues. In this variant we are not interested in guard sets of smallest cardinality, but in guard sets with largest possible distances between these guards. To the best of our knowledge, this variant has not been considered before. We call it the DISPERSIVE ART GALLERY PROBLEM. In particular, in the dispersive art gallery problem we are given a polygon $\mathcal{P}$ and a real number ℓ, and want to decide whether $\mathcal{P}$ has a guard set such that every pair of guards in this set is at least a distance of ℓ apart.

In this paper, we study the vertex guard variant of this problem for the class of polyominoes. We consider rectangular visibility and distances as geodesics in the L_1-metric. Our results are as follows. We give a (simple) thin polyomino such that every guard set has minimum pairwise distances of at most 3. On the positive side, we describe an algorithm that computes guard sets for simple polyominoes that match this upper bound, i.e., the algorithm constructs worst-case optimal solutions. We also study the computational complexity of computing guard sets that maximize the smallest distance between all pairs of guards within the guard sets. We prove that deciding whether there exists a guard set realizing a minimum pairwise distance for all pairs of guards of at least 5 in a given polyomino is NP-complete.

We were also able to find an optimal dynamic programming approach that computes a guard set that maximizes the minimum pairwise distance between guards in tree-shaped polyominoes, i.e., computes optimal solutions; due to space constraints, details can be found in the full version of our paper [42]. Because the shapes constructed in the NP-hardness reduction are thin as well (but have holes), this result completes the case for thin polyominoes.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Art gallery, dispersion, polyominoes, NP-completeness, r-visibility, vertex guards, L_1-metric, worst-case optimal

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2022.67

Related Version *Full Version*: https://arxiv.org/abs/2209.10291 [42]

Acknowledgements We thank Joseph S.B. Mitchell for bringing this problem to our attention.

1 Introduction

How many guards are necessary to guard an art gallery? This question was first posed by Victor Klee in 1973 and opened a flourishing field of research in computational geometry; see for example the book by O'Rourke [41], or the surveys by Shermer [44], and Urrutia [46]. This question states the classic ART GALLERY PROBLEM as follows: Given a (simple) polygon $\mathcal{P}$ and an integer k, decide whether there is a guard set of cardinality k such that every point $p \in \mathcal{P}$ is seen by at least one guard, where a point is seen by a guard if and only if the connecting line segment is inside the polygon.

Suppose the following situation: Your art gallery is the victim of a robbery, or there is a fire outbreak and heavy smoke development in one part of the building. Because guards in an optimal solution to instances of the classic art gallery problem can be really close together, many cameras can be affected at the same time, see Figure 1. From safety and

33rd International Symposium on Algorithms and Computation (ISAAC 2022).
Editors: Sang Won Bae and Heejin Park; Article No. 67; pp. 67:1–67:18
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

security issues this would be a catastrophic scenario. We want to address these issues, i.e., for a given shape, we are interested in a guard set that realizes preferably large distances between any two guards of the respective set, rather than focusing on the minimum number of guards needed. Problems of this kind are called DISPERSION PROBLEMS, and are typically stated as follows: Given a set of n objects in the plane and an integer k, decide if there is a subset of k such objects, such that the distances between any pair in this subset is at least as large as a given threshold. We assume that the shortest paths that realize the distances between guards are within the shape, i.e., they do not leave and enter the shape.

In this paper, we introduce the following problem that combines art gallery and dispersion problems and is described as follows.

Dispersive Art Gallery Problem. Given a polygon $\mathcal{P}$ and a real number ℓ, decide whether there exists a guard set $\mathcal{G}$ for $\mathcal{P}$ such that the pairwise geodesic distances between any two guards in $\mathcal{G}$ are at least ℓ.

Note that in this problem we are not interested in the size of a particular guard set, but only in the distances between guards realized by the guard set. To the best of our knowledge, this problem has not been considered before. Additionally, a first intuitive thought might be that solutions to the classic art gallery problem are also solutions to this variant, since small cardinality guard sets should somehow yield larger pairwise distances. However, this is nowhere near the truth, see for example Figure 1 where doubling the size of the guard set results in an arbitrary growth of the dispersion distance.

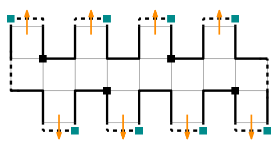


Figure 1 An adaption of the comb-like polyomino. The black vertices realize an optimal guard set for the classic AGP, while the dark cyan set is optimal for the dispersive AGP.

1.1 Our contributions

In this paper, we introduce the dispersive art gallery problem and investigate it for vertex guards in polyominoes, i.e., orthogonal polygons whose vertices have integer coordinates.

- We describe a (simple) thin polyomino where the minimum pairwise distance between any two guards in every feasible guard set is at most 3.
- We give a worst-case optimal algorithm for placing a set of guards at the vertices of a simple polyomino such that the pairwise distances between any two guards are at least 3.
- It is NP-complete to decide whether a pairwise distance of at least 5 can be guaranteed.
- We describe a dynamic programming approach that computes a guard set that maximizes the minimum pairwise distance between any two guards for tree-shaped polyominoes.

1.2 Previous work

The famous question from Klee was answered relatively quickly by Chvátal [17]. Not least because of the beautiful proof from Fisk [28] it is almost common knowledge that $\lfloor n/3 \rfloor$ guards are sufficient but sometimes necessary to monitor a simple polygonal region with n edges. Through their typical orthogonality, "traditional" galleries actually require less guards, i.e., for orthogonal polygons with n vertices already $\lfloor n/4 \rfloor$ guards are sufficient, but also sometimes necessary [30, 35, 40]. However, finding the optimal solution even in simple polygons is proven to be NP-hard by Lee and Lin [37], and by Schuchardt and Hecker [43] for simple orthogonal polygons. In the special case of r-visibility, computing the minimum guard set is polynomial in orthogonal polygons [11, 47]. More recently, Abrahamsen et al. [2, 3] first showed that irrational guards are sometimes needed in an optimal guard set (in general and orthogonal polygons), and subsequently that the art gallery problem is actually $\exists\mathbb{R}$-complete.

Restricting the class of galleries to polyominoes intuitively makes the problem a lot easier. However, as shown by Biedl et al. [8, 9] the problem remains NP-hard. On the positive side they showed that $\lfloor (m+1)/3 \rfloor$ point guards are always sufficient and sometimes necessary, where m is the number of squares of the polyomino. Additionally, they give an algorithm for computing optimal guard sets in the case of thin polyomino trees.

By now, there are many variations of the classic art gallery problem. At least in two of them the number of placed guards is irrelevant, as it is also the case in our problem setting. These are the Chromatic AGP [21, 22, 25, 33] where guards are associated by a color and no two guards of the same color class are allowed to have overlapping visibility regions, and the Conflict-free chromatic AGP [4, 5, 31] in which the overlapping constraint is relaxed in a way that at every point within the polygon a unique color must be visible. In both of these problems, only the number of used colors in a feasible guard set is of interest.

Other variations regard the region that has to be covered, e.g., the Terrain Guarding Problem [12, 36], or problems that arrive from restricting the visibility of the guards to cones of a certain angle, that can be summarized under the generic term of Floodlight Problems [1, 13, 18, 24, 32, 39, 45].

Dispersion problems are related to packing problems and involve arranging a set of objects "far away" from one another, or choosing a subset of objects that are "far apart". These naturally arrive as obnoxious facility location problems (see, e.g., the surveys by Cappanera [15], or Erkut and Neuman [23]), and as problems of distant representatives [27]. For more recent work in many different settings, e.g., in disks [20, 27], or on intervals [10, 38]; see also [6, 7, 14, 16, 26, 29, 34] for various other settings.

1.3 Preliminaries

We consider *polyominoes*, that are orthogonal polygons formed by joining unit squares edge to edge. These unit squares are called *cells*, and the edges of the cells are denoted as *sides*. The *boundary* $\partial\mathcal{P}$ of the polyomino $\mathcal{P}$ is the sequence of all cell sides each one lying between one cell from $\mathcal{P}$ and one cell not being part of $\mathcal{P}$. The *vertices* of a polyomino $\mathcal{P}$ are the vertices of the boundary of $\mathcal{P}$. A point $p \in \mathcal{P}$ *covers* or *sees* another point $q \in \mathcal{P}$ if there is an axis-aligned rectangle defined by p and q that is a subset of $\mathcal{P}$. In the literature this notion of visibility is called *r-visibility*. The area that is visible from a point p is its *visibility region* $\mathcal{V}(p)$. The *distance* $d(p, q)$ between two points $p, q \in \mathcal{P}$ is given by the L_1 geodesic shortest path connecting these two points, i.e., the distance is measured entirely within the interior of $\mathcal{P}$. A *guard set* $\mathcal{G}$ is a set of points of $\mathcal{P}$ such that every point of $\mathcal{P}$ is covered by at least on point of $\mathcal{G}$. We will restrict ourselves to *vertex guards*, i.e., guards that are placed on vertices of $\mathcal{P}$. The minimum over all pairwise distances between any two guards in a guard set $\mathcal{G}$ is called its *dispersion distance*. The *dual graph* of a polyomino $\mathcal{P}$ has a vertex for every cell of $\mathcal{P}$, and edges between vertices if their corresponding cells share a side. We say that a polyomino is *simple* if it has no holes, *thin* if it does not contain a 2×2 polyomino as a subpolyomino, and *tree-shaped* if its dual graph is a tree. We call a cell a *niche* if it is a degree 1 vertex in the dual graph of $\mathcal{P}$.

2 Worst-case optimality

In this section we prove that a dispersion distance of 3 is worst-case optimal for simple polyominoes. In particular, we describe the construction of thin polyominoes for which no guard set can have larger dispersion distance than 3, and describe an algorithm that constructs such guard sets for any simple polyomino.

▶ **Lemma 1.** *There are (simple) thin polyominoes such that every guard set has dispersion distance at most 3.*

Proof. Consider the dark magenta region in Figure 2. Note that this region has to be guarded by a guard g that is placed on one of the four vertices incident to this region. Let Π be one of the four niches that is closest to g. The guard g' that covers Π has distance at most 3 to g. ◀

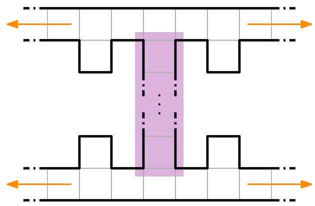


Figure 2 A simple, thin polyomino in that every guard set has dispersion distance at most 3.

Note that the polyomino given in Figure 2 can be used as a "building block", i.e., it can be extended (as indicated by orange arrows) and therefore be used to construct arbitrarily large polyominoes (as well as non-simple ones) in which the same upper bound holds.

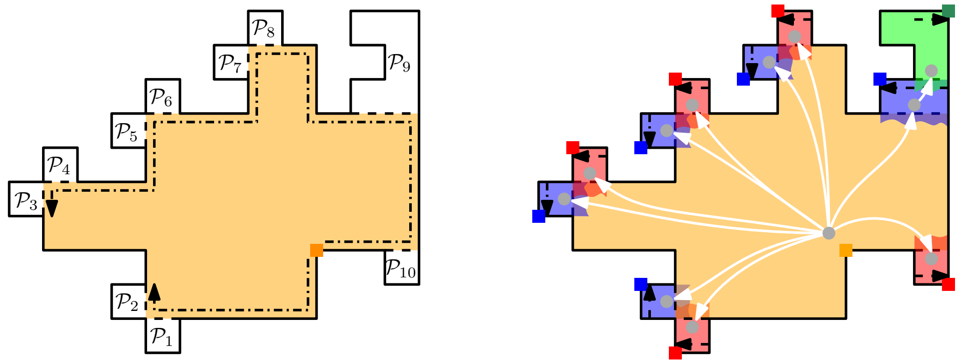
In the remainder of this section we show that every *simple* polyomino allows for a guard set with a dispersion distance of at least 3, implying worst-case optimality.

▶ **Theorem 2.** *For every simple polyomino there exists a guard set that has dispersion distance at least 3.*

In particular, we prove Theorem 2 constructively by giving an algorithm that constructs a guard set with dispersion distance of at least 3 in polynomial time. In a nutshell, the algorithm places guards greedily until the whole polyomino is guarded. The algorithm starts with a guard on an arbitrary vertex. Then the region that is visible from this guard is removed from the polyomino. This leads to a set of disjoint subpolyominoes that are guarded recursively, maintaining a distance of at least 3 between any two guards, see Figure 3.

2.1 Preliminaries for the algorithm

Let $\mathcal{P}'$ be a subpolyomino of $\mathcal{P}$, i.e., $\mathcal{P}' \subseteq \mathcal{P}$. The boundary $\partial\mathcal{P}'$ of $\mathcal{P}'$ is the union of all sides being part of exactly one cell from $\mathcal{P}'$. Note that the definition of $\partial\mathcal{P}'$ does not depend on $\mathcal{P}$. Assume that the guard g cannot see the entire polygon $\mathcal{P}$, i.e., $\mathcal{V}(g) \neq \mathcal{P}$. By removing $\mathcal{V}(g)$ from $\mathcal{P}$ we obtain $k \geq 1$ subpolyominoes $\mathcal{P}_1, \ldots, \mathcal{P}_k \subset \mathcal{P}$, being maximal subsets of unit squares such that each subset forms an orthogonal polygon. The *gate* G_i corresponding to $\mathcal{P}_i$ is $\partial\mathcal{V}(g) \cap \partial\mathcal{P}_i$. Without loss of generality we assume $G_1, \ldots, G_k$ to be ordered clockwise on $\partial\mathcal{P}$ starting from g. The *walls* of a gate G_i are the two sides from $\partial\mathcal{P} \setminus G_i$ being adjacent to G_i, see the red segments in Figure 5. Note that the first (second) wall of G_i can lie on $\partial\mathcal{P}_{i-1}$ ($\partial\mathcal{P}_{i+1}$) where from now on the indices $i+1$ and $i-1$ are considered modulo k.



(a) **(b)**

Figure 3 (a) A guard with its visibility region (orange), and the subpolyominoes $\mathcal{P}_1, \ldots, \mathcal{P}_{10}$. The corresponding gates G_1 and G_2 are clockwise, and $G_3, \ldots, G_{10}$ are counterclockwise. (b) The recursion tree T and a guarding computed by our algorithm.

2.2 Description of the algorithm

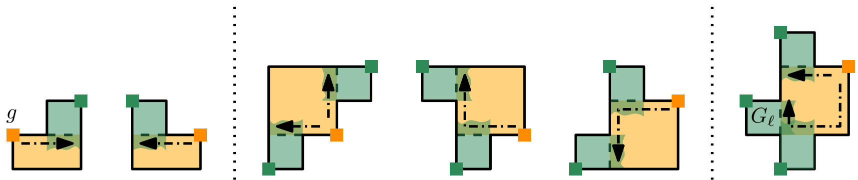
Based on the preliminaries above we provide the details of our algorithm. As initialization, we consider a guard g placed on an arbitrary vertex of the given polyomino $\mathcal{P}$.

A recursion step. Consider the subshapes $\mathcal{P}_1, \ldots, \mathcal{P}_k$ and the corresponding gates as defined above, see Figure 3(a). Let α and β be the number of sides from $\partial\mathcal{P}$ when walking clockwise along $\partial\mathcal{P}$ from g to G_1, and from G_k to g, respectively. Note that $\alpha, \beta \geq 1$. In the following we declare each gate to be *(oriented) clockwise* or *counterclockwise*. For this, we consider different cases regarding k, see Figure 4.

(1) If $k = 1$:
- **(1.1)** if $\alpha = 1$, we declare G_1 to be clockwise
- **(1.2)** otherwise, we declare G_1 to be counterclockwise.

(2) If $k = 2$:
- **(2.1)** if $\alpha = 1 = \beta$, we declare G_1 to be clockwise and G_2 to be counterclockwise,
- **(2.2)** if $\alpha = 1 < \beta$, we declare G_1 and G_2 to be clockwise,
- **(2.3)** if $\alpha > 1 = \beta$, we declare G_1 and G_2 to be counterclockwise.

(3) If $k \geq 3$, let G_ℓ be the first gate being not adjacent to its successor $G_{\ell+1}$, i.e., G_ℓ and $G_{\ell+1}$ are not sharing an endpoint. We declare $G_1, \ldots, G_\ell$ to be clockwise and $G_{\ell+1}, \ldots, G_k$ to be counterclockwise, see Figure 3(a).



(1.1) **(1.2)** **(2.1)** **(2.2)** **(2.3)** **(3)**

Figure 4 Case distinction for gate orientations (orange: guard g, green: guard $\overline{g}$ placed after g and whose position is influenced by orientation of corresponding gate).

For each $\mathcal{P}_i$ we make a recursive call for covering $\mathcal{P}_i$ separately. In particular, we make the following distinction: A gate is *parallel* (*orthogonal*) when its walls lie parallel (orthogonal) to each other.

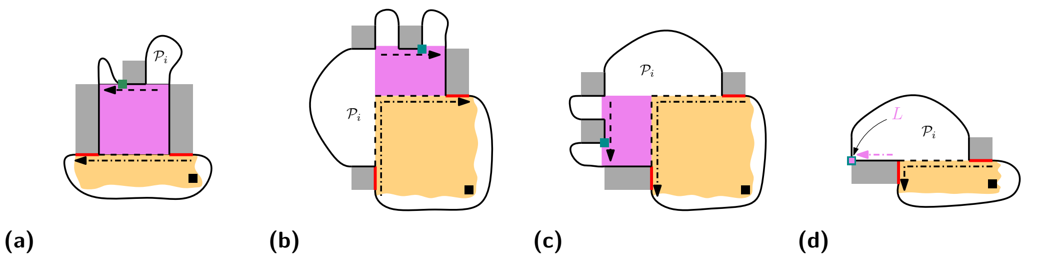


Figure 5 Placements of a guard $\overline{g}$ (green) depending on the gate's type and orientation (dashed line): (a) A parallel counterclockwise gate. (b) An orthogonal clockwise gate. (c) An orthogonal counterclockwise gate. (d) An orthogonal counterclockwise gate, and L degenerates to a single point.

A recursive call for a parallel gate. Without loss of generality, we assume that G_i is horizontal and $\mathcal{V}(g)$ lies below G_i, see Figure 5(a). Let $T \subseteq \mathcal{P}_i$ be the axis aligned rectangle with maximal height and bottom side G_i. If G_i is clockwise (counterclockwise), we choose the guard $\overline{g}$ as an arbitrary vertex on the boundary of T not lying on G_i and not lying on the left (right) side of T. Finally, we recurse on $\mathcal{P} := \mathcal{P}_i \cup \mathcal{V}(\overline{g})$ and $g := \overline{g}$.

A recursive call for an orthogonal gate. Without loss of generality, we assume that $\mathcal{P}_i$ lies above and to the left of G_i, see Figure 5(b)+(c). We distinguish two cases.

(1) G_i is counterclockwise: Let $\ell \subseteq G_i$ be the vertical segment of G_i. Note that ℓ denotes the left endpoint of G_i if it only consists of a horizontal segment, see Figure 5(d). Let $L \subseteq \mathcal{P}_i$ be maximal rectangle with right side ℓ, see Figure 5(c)+(d). We choose $\overline{g}$ as a vertex from the boundary of L not lying on G_i and not lying on the right side of L.

(2) G_i is clockwise: Consider by $t \subset G_i$ the horizontal segment of G_i. Note that t denotes the top endpoint of G_i if it consists only of a vertical segment. Let $T \subseteq \mathcal{P}_i$ be the maximal rectangle with bottom side t, see Figure 5(b). We choose $\overline{g}$ as a vertex from the boundary of T not lying on G_i.

Finally, we again recurse on $\mathcal{P} := \mathcal{P}_i \cup \mathcal{V}(\overline{g})$ and $g := \overline{g}$.

2.3 Analysis of the algorithm

We consider the recursion tree T of our algorithm. In particular, each guard placed in the corresponding recursion step is a node in T. An edge between a father node g and a child node $\overline{g}$ exists if g creates a subpolyomino $\mathcal{P}_i$ causing a recursive call on $\mathcal{P}_i \cup \mathcal{V}(\overline{g})$ and $\overline{g}$. We say that g_2 is an *indirect* child of g_1 if there is a sequence of nodes $g_1 = \overline{g}_1, \ldots, \overline{g}_\ell = g_2$ such that $\overline{g}_i$ is the father of $\overline{g}_{i+1}$ for $i = 1, \ldots, \ell - 1$.

For a clearer presentation, we say that g_1 is a *direct* child of g_2, if $\ell = 1$.

As $\overline{g}$ is chosen from the segment resulting from pushing a vertical or horizontal line of G_i until a vertex of $\mathcal{P}$ is hit for the first time, we obtain the following:

▶ **Observation 3.** *All cells from $\mathcal{P}_i$ sharing at least one point or a side with G_i are seen by $\overline{g}$, see the dark green areas in Figure 6.*

As our algorithm recurses on $\mathcal{P}_i \cup \mathcal{V}(\overline{g})$, we obtain the following as a direct consequence of Observation 3.

▶ **Corollary 4.** *For each recursive call on $\mathcal{P}_i \cup \mathcal{V}(\overline{g})$ and $\overline{g}$ the guard $\overline{g}$ is placed inside $\mathcal{P}_i$ within a distance of at least 1 to the corresponding gate G_i.*

The next lemma addresses neighbored gates and their orientations and is a key in the analysis of the algorithm:

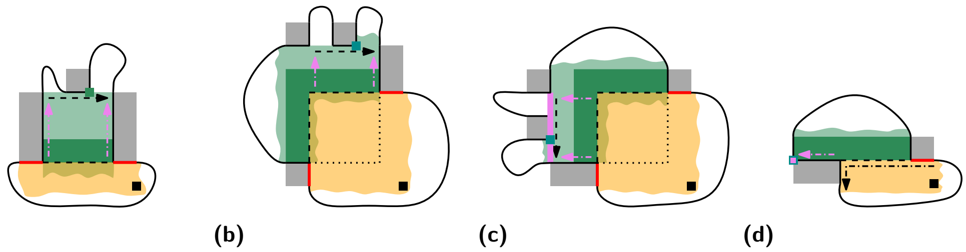


Figure 6 All cells from $\mathcal{P}_i$ that share at least a point or a side with G_i are seen by $\overline{g}$.

▶ **Lemma 5.** *Let G_1 and G_2 be two gates created in the same recursion step. If G_1 and G_2 share an endpoint, they have the same orientation.*

Proof. The proof follows the case distinction of the recursion step, and let k be the number of gates created in the considered recursion step. As $k \geq 2$, Case (1) is not relevant. If $k = 2$, the Cases (2.2) and (2.3) directly imply the same orientation of G_1 and G_2. In Case (2.1) both gates are connected via one side of the boundary of $\mathcal{P}$, i.e., $\gamma_1 = \gamma_2$. Thus, G_1 and G_2 cannot share an endpoint, see Figure 4 (2.1). Finally, let at least $k = 3$ gates be created during the considered recursion step. If G_1 and G_2 share an endpoint, the description of the case ensures that they are oriented in the same direction. ◀

We now consider the geometric form of gates and their positions relative to one another.

▶ **Lemma 6.** *If two gates G_1, G_2 created in the same recursion step share an endpoint, both G_1 and G_2 are parallel gates lying orthogonal to one another.*

Proof. The proof is by contradiction. Assume that at least G_1 is orthogonal. Two adjacent segments from different gates cannot be collinear, because otherwise there is a side from the boundary of $\mathcal{P}$ lying between cells from the polygon, see Figure 7b. As G_1 is orthogonal and adjacent to G_2, there is a sequence of consecutive segments s_1, s_2, s_3 from these gates, where s_1, s_2 and s_2, s_3 are adjacent to one another, see Figure 7c. As the guard g lies to the right of s_1, s_2, and s_3, it sees at least one cell outside of $\mathcal{V}(g)$ being a contradiction. ◀

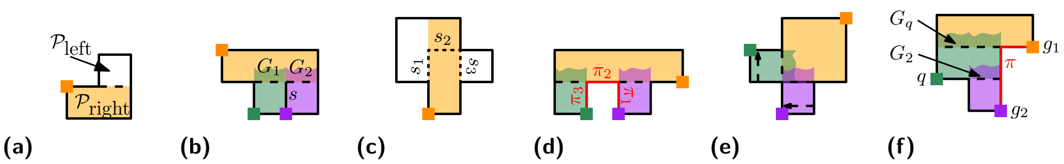


Figure 7 (a) A segment of a gate separates $\mathcal{P}$ into a left and a right polyomino. (b) Two parallel gates lying collinear are not possible. (c) Two gates laying adjacent where at least one is an orthogonal gate is not possible. (d) A shortest path connecting two guards not being (indirect) children of each other where the corresponding gates are not adjacent. (e) Two guards being not (indirect) children of each other where the corresponding gates are adjacent. (f) Sequence of children.

We now analyze the dispersion distance of the guard set constructed by our approach based on Corollary 4 and Lemmas 5 and 6.

▶ **Lemma 7.** *The constructed guard set has a dispersion distance of at least 3.*

Proof. In order to prove the lemma we consider an arbitrary pair of placed guards g_1, g_2 and distinguish three cases: (1) Neither g_1 is an indirect child of g_2 nor vice versa. (2) g_1 is an indirect but not a direct child of g_2 or vice versa. (3) g_1 is a direct child of g_2 or vice versa.

In the following, we consider all three cases separately. The intuitions for the three cases are the following: (1) If g_1 and g_2 have the common father g let G_1, G_2 be the corresponding gates. If G_1, G_2 are not adjacent these gates are within a distance of at least 1. Hence, applying Corollary 4 twice leads to a distance of at least 3, see Figure 7(d). If G_1, G_2 are adjacent, Lemma 5 implies a distance of at least 3, see Figure 7(e). If g_1 or g_2 ist not a direct child of g, similar arguments apply. (2) Applying Observation 3 yields two gates G_1, G_2 between g_1 and g_2 where each path between G_1 and G_2 has a length of 1, see Figure 7(f). Finally, applying Corollary 4 and the observation that g_2 does not lie on a gate caused by g_2 yields a distance of at least 3. (3) Intuitively speaking Figure 6 implies that the same arguments as used in (2) apply to (3).

Neither g_1 is an indirect child of g_2, nor vice versa. First consider the case in which g_1 and g_2 are direct children of the same father g. Let G_1 and G_2 be the gates created by g corresponding to g_1 and g_2. Let π be a shortest path connecting g_1 and g_2. Note that $\pi = (\pi_1, \pi_2, \pi_3)$ contains three subpaths, where π_1 connects g_1 and G_1, π_2 connects G_1 and G_2, and π_3 connects G_2 and g_2. Corollary 4 implies that π_1 and π_3 have a length of at least 1. If G_1 and G_2 do not share an endpoint, we obtain that π_2 has a length of at least 1, implying that π has a length of at least 3, see Figure 7(d).

If G_1 and G_2 share an endpoint, Lemma 5 implies that G_1 and G_2 have the same orientation. Without loss of generality, assume that G_1, G_2 are oriented clockwise. Furthermore, Lemma 6 implies that G_1, G_2 are parallel gates, whose segments lie orthogonal to one another. Without loss of generality, assume that G_1, G_2 are ordered clockwise and that G_1 (G_2) is horizontal (vertical), see Figure 7(e). As G_1 and G_2 are oriented clockwise, applying Corollary 4 simultaneously to g_1 and g_2 implies that the x-coordinate of g_1 is at least one larger than the x-coordinate of g_2 and the y-coordinate of g_1 is at least two smaller than the y-coordinate of g_2. Thus, g_1 and g_2 have a distance of at least 3.

g_1 is an indirect but not a direct child of g_2, or vice versa. Without loss of generality, assume that g_1 is an indirect child of g_2. Thus, there is at least one further guard q being placed between g_1 and g_2, i.e., such that q is a direct or indirect child of g_1 and g_2 is a direct or indirect child of q, see Figure 7(f). This implies that the shortest path π connecting g_1 and g_2 has to cross the visibility region $\mathcal{V}(q)$ of q. Observation 3 implies that this subpath of π has a length of at least 1. Let G_q and G_2 be the gates between g_1, q and q, g_2. Corollary 4 implies that the length of the subpath of π connecting G_2 with g_2 is at least 1. Furthermore, g_1 cannot lie on G_q implying that the length of the subpath of π connecting g_1 and G_q also has a length of 1. Hence, π has a length of at least 3.

g_1 is a direct child of g_2, or vice versa. Without loss of generality, assume that each segment of the gate G_1 corresponding to g_1 has a length of 1, see Figure 8 for the different cases. In the case of a parallel gate, assume without loss of generality that g_2 lies adjacent to an endpoint G_1 resulting in a distance of 3, see Figure 8(a). In the case of an orthogonal gate, and that G_1 consist of two segments, assume without loss of generality that g_2 lies as close as possible to both segments of G_1 resulting in a distance of 4 between g_1 and g_2, see Figure 8(b)+(c). Finally, in the case of an orthogonal gate that consist of a single segment, assume without loss of generality that g_2 lies on the wall collinear with G_1 resulting in a distance of at least 3, see Figure 8(d).

This concludes the proof of Lemma 7. ◀

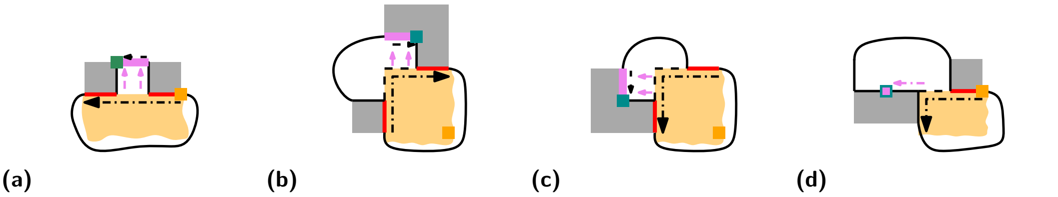


Figure 8 Assuming a position for g decreasing at most its distance to q: (a) A parallel gate. (b) A clockwise oriented orthogonal gate made up of two segments. (c) A counterclockwise oriented orthogonal gate made up of two segments. (d) An orthogonal gate made up of one segment.

As Lemma 1 provides an upper bound on the dispersion distance in simple polyominoes and Lemma 7 the matching lower bound, these lemmata together prove Theorem 2.

3 Computational complexity

In this section we study the computational complexity of computing guard sets that maximize the smallest distance between all pairs of guards within the guard set. In particular, we show the following.

▶ **Theorem 8.** *Deciding whether there exists a guard set with a dispersion distance of 5 for a given polyomino is* NP*-complete.*

3.1 Outline of the NP-hardness reduction

For showing NP-hardness we utilize the problem Planar Monotone 3Sat that is shown to be NP-complete by de Berg and Khosravi [19]. This problem asks to decide the satisfiability of a Boolean 3-CNF formula for which the literals in each clause are either all negated or all unnegated, and the corresponding variable-clause incidence graph is planar.

To this end, we will construct polyominoes that will represent variables and clauses. Because a variable may contribute to multiple clauses, we model a specific shape that duplicates the given assignment. Furthermore, we describe simple shapes that are used to connect different subshapes, while maintaining the given assignment from the variables. Figure 9 gives a high-level overview of the construction and the main gadgets.

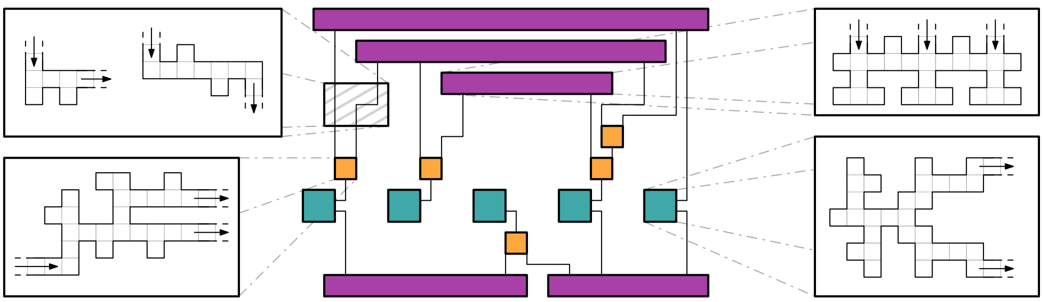


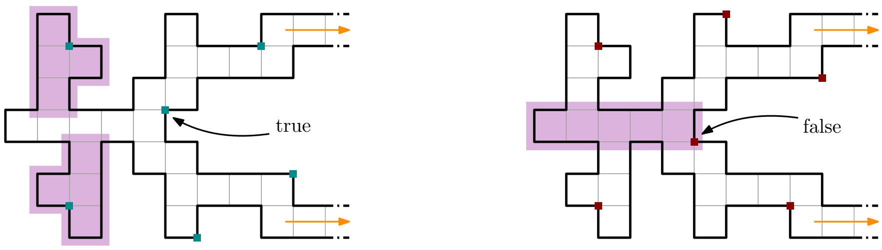
Figure 9 Symbolic overview of the NP-hardness reduction. The depicted instance is due to the Planar Monotone 3Sat formula $\varphi = (x_1 \vee x_2 \vee x_4) \wedge (x_2 \vee x_4) \wedge (x_1 \vee x_4 \vee x_5) \wedge (\overline{x_1} \vee \overline{x_3}) \wedge (\overline{x_3} \vee \overline{x_4} \vee \overline{x_5})$. Variables are in dark cyan, clauses in magenta, duplicator gadgets in orange, and connectors as lines.

The idea of the reduction is as follows: As shown in Figure 9, all gadgets have *open ends* (depicted by arrows) where they are connected to one another. These openings are distinguished as input and output depending on how the arrows are oriented. Starting from the variable gadgets there is a sequence of outputs and inputs that ends in the clauses gadgets. Through intensive use of niches, we force the possible guard sets within gadgets to essentially two sets. In particular, the first set covers the output of a gadget from within the gadget, and therefore also already the input of the next gadget in the sequence. The second set have to cover the input from within the gadget (because it is not already covered from before). We use these constraints and observations to propagate variable assignments through the construction, i.e., to obtain a *guarding direction*.

3.2 Setting up the gadgets

We now give the description of the involved gadgets. In addition we prove several lemma that will then put together to yield a proof of Theorem 8.

Variable gadget The gadget that is depicted in Figure 10 allows exactly two guard sets with a dispersion distance of 5, so it models true and false assignments. In order to do so, we force guards to unique positions within specific subregions, which results in the fact that all other feasible guard positions are restricted to two disjoint sets.



(a) **(b)**

Figure 10 The figure shows the variable gadget, and regions that are used in Lemmas 9 and 10. (a) shows the guard set for a true assignment, while (b) shows the respective set for a false assignment.

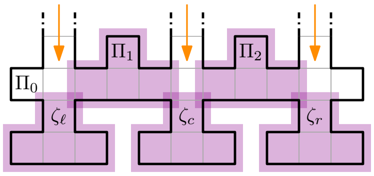
▶ **Lemma 9.** *Within the variable gadget, no guard set has dispersion distance larger than 5.*

Proof. Consider the dark magenta regions ("T-shapes") in Figure 10a. No guard set with at least two guards realizes a dispersion distance larger than 5 within such a region. The only vertex that could partly cover such a T-shape from the outside, is itself a vertex from another T-shape. Therefore, both these regions have to be covered from uniquely from within it. Thus, the largest possible distance between these guards is 5, as shown by cyan squares. ◀

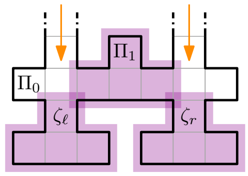
▶ **Lemma 10.** *Within the variable gadget there are exactly two guard sets realizing a dispersion distance of 5.*

Proof. As the guard placement within the T-shapes is unique (see Lemma 9), the only variability lies within the magenta region shown in Figure 10b. However, by maintaining a distance of 5 to the necessary guards placed within the T-shapes, there are exactly two vertices that remain for covering this region. By choosing one, all the other positions follow uniquely. Hence, there are exactly two guard sets with a dispersion distance of 5. ◀

Clause gadget. The clause gadget is depicted in Figure 11. The overall idea of this gadget is that it does not allow for a guard set with a dispersion distance of at least 5 if guards have to be placed only on vertices of this subshape. Hence, some specific cells have to be already covered from outside the shape, what will be related to satisfying the clause.



(a)



(b)

Figure 11 (a) The depicted shape represents a clause gadget containing three literals, while (b) shows a clause with two literals. The cells labeled with ζ can be covered from outside the gadget.

Note that a clause gadget "contains" basically two types of T-shapes, as shown in Figure 11. We call them *prospects* if they can partly be covered from outside, and *checkers* otherwise.

▶ **Lemma 11.** *There is no guard set with a dispersion distance of at least 5 for the shape representing the clause gadget when placing guards only within this shape.*

Proof. We will only argue this in detail for the case that the clause contains three literals; similar arguments hold for the remaining case. Consider one of the colored T-shapes in Figure 11a. Only a single guard can be placed within such a region if a dispersion distance of at least 5 is required. Consider three consecutive T-shapes, such that two of them are prospects. The shortest path connecting the six potential guard locations has a length of 9. Hence, no guard set with a dispersion distance of 5 exists. ◀

▶ **Lemma 12.** *If at least one cell of the clause gadget is covered from outside the gadget, a feasible guard set with a dispersion distance of 5 exists.*

Proof. Again, we will only argue the more complicated case, i.e., a clause containing three literals. For this, consider the marked region in Figure 11a and distinguish the following.

First assume that the central connector is covered from outside the gadget. Thus, in particular cell ζ_c is already covered, and we can place a guard in a bottom corner of this prospect. This results in two disjoint pairs of uncovered T-shapes, such that each of these pairs share a vertex of the shape. Without loss of generality, consider the left pair. Two guards can be placed at the bottom left vertex of ζ_ℓ and bottom right of Π_1 with a distance of 5. Because the guard that covers Π_1 already covers the bottom part of the other checker, we can place a guard in its niche, i.e., at the top right vertex of Π_2. Placing a guard at the bottom right vertex of ζ_r completes the guard set.

On the other hand, without loss of generality, let the left connector be already covered, i.e., the cell ζ_ℓ is covered. A guard can be placed in the leftmost niche Π_0 to cover the bottom part of both checker. Therefore, we only have to cover Π_1 and Π_2 by placing guards at their respective top vertices. It remains to cover the prospects. Because guards are placed at the top vertices of the checker's niches, we can place guards in appropriate distances to obtain a guard set with dispersion distance 5. ◀

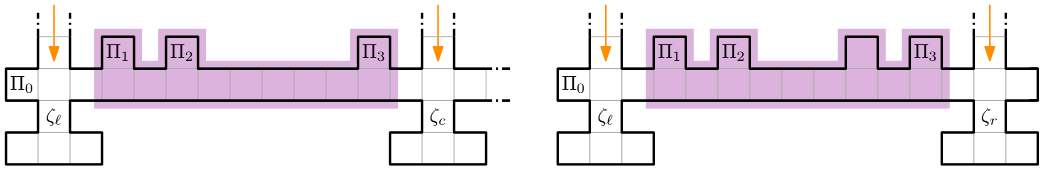
Due to the respective embedding of the overall shape it may be necessary to enlarge the clause gadget, see Figure 12.

▶ **Lemma 13.** *A clause gadget can be enlarged in a way that all functionalities are maintained.*

Proof. If the clause contains three literals, we replace the T-shape checker by the colored region in Figure 12a. Note that this region is mirrored vertically along the center connector, and that the region between Π_2 and Π_3 can be enlarged arbitrarily.

A crucial observation is that the niches Π_1 and Π_3 coincide in the short clause gadget, and therefore apply the same restrictions as before. The additional niche Π_2 guarantees that we cannot place a guard at the bottom right vertex of Π_1, assuming that the clause is not satisfied through an assignment.

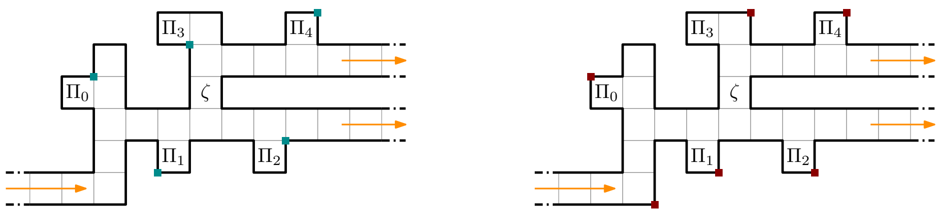
If the clause only contains two literals, the T-shape checker will be replaced by the colored region in Figure 12b. The correctness follows analogously. ◀



(a) (b)

Figure 12 (a) shows the left part of an enlarged clause gadget containing three literals, while (b) shows the respective enlarged shape for clauses containing two literals.

Duplicator gadget. Because a variable may contribute to more than one clause, we need to duplicate the respective assignment. For this purpose, we construct the duplicator gadget that is depicted in Figure 13. It works as follows: if the incoming connector is covered from outside the gadget, both outgoing connectors can be covered from within the gadget. Similarly, if the incoming connector has to be covered from within the gadget, the outgoing connectors must be covered from outside the gadget.



(a) (b)

Figure 13 The figure shows the duplicator gadget. (a) shows a set of guards duplicating a true assignment, while (b) shows the respective guard set for a false assignment.

▶ **Lemma 14.** *The duplicator gadget is correct, i.e., any output is equal to the input.*

Proof. First consider the situation given in Figure 13a. Because the incoming connector is covered from the outside, we want to cover the outgoing connectors from the inside. We will argue that the configuration in the marked region is unique and fulfills the requirements. Because of niche Π_3, covering Π_1 by a guard placed at vertices of ζ is not possible, and because of Π_0 and Π_2 the guard covering Π_1 is uniquely defined. Because this position is fixed, all other positions follow.

Now consider the situation in Figure 13b. The incoming connector has to be covered from the inside. Because of Π_0, the position of the guard covering the incoming connector is uniquely defined. Therefore, there are two positions left to cover the niche Π_1; however, because of Π_3, we cannot choose the vertex of ζ. It follows that the positions for guarding Π_1 and Π_2 are uniquely defined. Because ζ cannot be covered from the outside, the guard covering this square is also uniquely defined and also covers Π_3 simultaneously. This again leaves only a single position to cover Π_4. Overall, this leaves some squares of the outgoing connectors uncovered, so that they have to be covered from outside the gadget. ◀

Because a necessary condition to the problem is that the guard set have to cover the polyomino completely, we have to ensure that visibility regions that are induced by guards from clause gadgets do not interfere the assignment given due to guards from within the variable gadgets. So, if a clause $C_i = (x_j, x_k, x_\ell)$ is satisfied by say x_j, we have to make sure that other clauses containing x_k or x_ℓ but not x_j are not become automatically satisfied by *backward guarding* from guards in C_i. Preventing this is also the job of the duplicator gadget.

▶ **Lemma 15.** *Backward guarding of an output of the duplicator gadget cannot result in covering the other output from within the gadget.*

Proof. Consider without loss of generality that the duplicator gadget propagates false from the respective variable. A critical situation would occur if the assignment could be flipped within a duplicator gadget due to the coverage coming from a clause gadget, i.e., propagating true to the other output.

As argued above, due to the positions of the niches, the positions for guards are highly restricted. The coverage from the outside does not cover any of these niches. Therefore, it does not change the possible set of guard positions. ◀

Connector gadget. Now that we have the main components, we need to connect them. For this, we introduce two different connector gadgets, see Figure 14.

(a) **(b)**

Figure 14 (a) shows an L-connector, and (b) shows a Z-connector. The dark cyan and red colored guard sets propagate whether a variable is set to true or false, respectively.

▶ **Lemma 16.** *All connector gadgets fulfill the property that either the input, or the output can be guarded from within the gadgets by a guard set with a dispersion distance of 5.*

Proof. As these gadgets connect the previous ones, we distinguish between the cases that their input is already covered or not. Remark that if the input is already covered, we want to cover the output within the connector, and vice versa.

We prove this by providing specific sets of guards, regarding the different settings. For the case that the input is already covered, consider the dark cyan placed guards. The distance between guards is at least 5, and everything is covered. For the case that the input has to be covered, consider the red guarding positions. The placement of the niches force the position of the guard that covers the input. All other positions follow uniquely. It is easy to see that no more guards can be placed, so the output remains uncovered. ◀

3.3 Completing the NP-hardness proof

We described several gadgets that will now be used to construct an instance $\mathcal{P}_\varphi$ of the Dispersive Art Gallery Problem from any Boolean formula φ that is an instance of Planar Monotone 3SAT; this yields a proof for Theorem 8 that we restate here.

▶ **Theorem 8.** *Deciding whether there exists a guard set with a dispersion distance of 5 for a given polyomino is* NP*-complete.*

Proof. Note that the problem is obviously in NP. It is easy to verify whether a potential set of vertices is in fact a guard set for the given polyomino. Furthermore we can compute the dispersion distance of a guard set in polynomial time.

To show that the problem is NP-hard, we reduce from PLANAR MONOTONE 3SAT. For a given formula φ we create an instance $\mathcal{P}_\varphi$ of DISPERSIVE AGP as follows; again, see Figure 9 for the high-level idea of the construction.

Consider the rectilinear embedding of the graph given by φ. For every variable, we place a variable gadget horizontally in a row. Each clause is represented by a clause gadget. Due to the rectilinear embedding, we can place them vertically behind one another, and expand them appropriately if necessary as shown above. Without loss of generality, we place the clauses containing only unnegated literals above the variables, and below otherwise. If a literal x_i occurs in m_i many clauses, we construct $m_i - 1$ duplicator gadgets between vertically between clauses and variables. We properly place a set of connector gadgets to connect variables to duplicator gadgets, as well as the outputs of duplicator gadgets to respective inputs, and duplicator gadgets to the respective clauses. Note that variables are connected to clauses if they contribute only to a single clause.

If φ is satisfiable, then there is a guard set with dispersion distance 5 for $\mathcal{P}_\varphi$.

Consider a satisfying assignment of φ. A guard set with a dispersion distance of 5 for $\mathcal{P}_\varphi$ can be constructed as follows: From the given assignment of the variable x_i the respective set of guards within the variable gadget is chosen. For every connector and duplicator gadget, there is a set of guards that maintains the assignment. Because we propagate the satisfying assignment through the gadgets, at least one literal satisfies each clause. Hence, we can choose guards within each clause gadget that has dispersion distance of 5, because in each of these gadgets at least one of the cells are covered from the outside.

If there is a guard set with dispersion distance 5 for $\mathcal{P}_\varphi$, then φ is satisfiable.

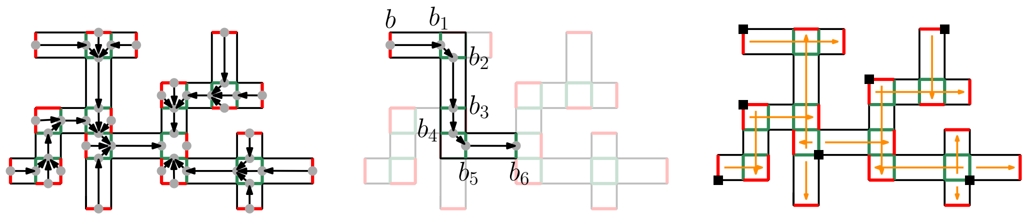
Consider a guard set for $\mathcal{P}_\varphi$ that has a dispersion distance of 5. As argued above, at least one cell of each clause gadget are covered from outside of the respective gadget, because otherwise there is no such desired guard set. Furthermore, there is no guard set for the variable gadget that has a dispersion distance larger than 5, and there are only two sets that realize this pairwise minimum distance. For every path from variables to clauses, the duplicator and connector gadgets provide specific locations for guards that maintain a dispersion distance of 5. Hence, the guards within the variable gadget of $\mathcal{P}_\varphi$ realize a satisfying assignment for φ.

This concludes the proof. ◀

4 Trees

While computing guard sets with maximum dispersion distance is NP-hard in general, we present a linear-time algorithm to compute optimal solutions in tree-shaped polyominoes. Recall that a polyomino $\mathcal{P}$ is tree-shaped if the dual graph of $\mathcal{P}$ is a tree. In particular, these polyominoes do not contain a 2×2 subpolyomino.

▶ **Theorem 17.** *Given a tree-shaped polyomino $\mathcal{P}$ with n vertices, there is an $O(n)$ dynamic programming approach for computing guard sets of maximum dispersion distance.*



(a) **(b)** **(c)**

Figure 15 (a) The directed tree (gray dots and black directed edges) and borders (red and green segments) of a tree-shaped polyomino. (b) The unique path from an outer border b to the inner border b_6 of the "root cell". (c) An optimal guard set (black squares) generated by our approach, with "screening" directions (orange arrows).

The main structure used in our algorithm are *borders*, which are defined as follows: Let $R \subseteq \mathcal{P}$ be the set of all maximal rectangles, see Figure 15. Note that R covers $\mathcal{P}$. We refer to the part of the boundary of a rectangle to be an *inner border* if it is not part of the boundary of the polyomino, see the green segments in Figure 15a. The two boundary parts of each rectangle having length 1 are called *outer borders*, see the red segments in Figure 15a. Note that every outer border is on the boundary of $\mathcal{P}$, and every border is either an inner or an outer border.

For every border we consider *states* describing (1) which of its endpoints are chosen as guards, (2) weather there is already a guard placed "behind" this border, and (3) the ratio of the shortest distances of its endpoints to a placed guard.

Applying a straightforward dynamic programming approach on a directed tree structure induced by the above-mentioned borders yields an algorithm for computing guard sets of maximum dispersion distance for the class of tree-shaped polyominoes. The observation that (3) does not cause $\Omega(n)$ but only a constant many states, results in the fact that the runtime of this algorithm is linear in the number of vertices of the input polyomino.

For the detailed description of the algorithm we refer to the full version [42].

5 Conclusion and future work

In this paper we introduced the dispersive AGP and investigated it for vertex guards in polyominoes. We described an algorithm that constructs worst-case optimal solutions of dispersion distance 3, and showed that it is NP-complete to decide whether a dispersion distance of 5 can be achieved. We were also able to find a linear-time dynamic programming approach to compute guard sets of maximum dispersion distance for tree-shaped polyominoes, see the full version [42].

Several open questions remain. Is it possible to close the gap to the worst-case, i.e., is deciding whether a dispersion distance of 4 can be achieved NP-hard as well? Is it possible to compute worst-case solutions in the case of non-simple polyominoes? It seems very promising that our method can be extended. Other open questions concern approximation algorithms. Is there a constant-factor approximation for this problem?

What can be said about the ratio between the cardinality of guard sets in optimal solutions for the dispersive and the classic art gallery problem? As shown in Figure 1 this ratio is at least 2, while the ratio between the dispersion distances increases arbitrarily.

What can be said about the dispersive art gallery problem in terrains, or general polygons?

References

1 James Abello, Vladimir Estivill-Castro, Thomas C. Shermer, and Jorge Urrutia. Illumination of orthogonal polygons with orthogonal floodlights. *International Journal of Computational Geometry and Applications*, 8(1):25–38, 1998. doi:10.1142/S0218195998000035.

2 Mikkel Abrahamsen, Anna Adamaszek, and Tillmann Miltzow. Irrational guards are sometimes needed. In *Symposium on Computational Geometry (SoCG)*, pages 3:1–3:15, 2017. doi:10.4230/LIPIcs.SoCG.2017.3.

3 Mikkel Abrahamsen, Anna Adamaszek, and Tillmann Miltzow. The art gallery problem is $\exists\mathbb{R}$-complete. *Journal of the ACM*, 69(1):4:1–4:70, 2022. doi:10.1145/3486220.

4 Andreas Bärtschi, Subir Kumar Ghosh, Matús Mihalák, Thomas Tschager, and Peter Widmayer. Improved bounds for the conflict-free chromatic art gallery problem. In *Symposium on Computational Geometry (SoCG)*, pages 144–153, 2014. doi:10.1145/2582112.2582117.

5 Andreas Bärtschi and Subhash Suri. Conflict-free chromatic art gallery coverage. *Algorithmica*, 68(1):265–283, 2014. doi:10.1007/s00453-012-9732-5.

6 Christoph Baur and Sándor P. Fekete. Approximation of geometric dispersion problems. *Algorithmica*, 30(3):451–470, 2001. doi:10.1007/s00453-001-0022-x.

7 Marc Benkert, Joachim Gudmundsson, Christian Knauer, René van Oostrum, and Alexander Wolff. A polynomial-time approximation algorithm for a geometric dispersion problem. *International Journal of Computational Geometry and Applications*, 19(3):267–288, 2009. doi:10.1142/S0218195909002952.

8 Therese C. Biedl, Mohammad T. Irfan, Justin Iwerks, Joondong Kim, and Joseph S. B. Mitchell. Guarding polyominoes. In *Symposium on Computational Geometry (SoCG)*, pages 387–396, 2011. doi:10.1145/1998196.1998261.

9 Therese C. Biedl, Mohammad T. Irfan, Justin Iwerks, Joondong Kim, and Joseph S. B. Mitchell. The art gallery theorem for polyominoes. *Discrete Computational Geometry*, 48(3):711–720, 2012. doi:10.1007/s00454-012-9429-1.

10 Therese C. Biedl, Anna Lubiw, Anurag Murty Naredla, Peter Dominik Ralbovsky, and Graeme Stroud. Dispersion for intervals: A geometric approach. In *Symposium on Simplicity in Algorithms (SOSA)*, pages 37–44, 2021. doi:10.1137/1.9781611976496.4.

11 Therese C. Biedl and Saeed Mehrabi. On r-guarding thin orthogonal polygons. In *Symposium on Algorithms and Computation (ISAAC)*, pages 17:1–17:13, 2016. doi:10.4230/LIPIcs.ISAAC.2016.17.

12 Édouard Bonnet and Panos Giannopoulos. Orthogonal terrain guarding is NP-complete. *Journal of Computational Geometry*, 10(2):21–44, 2019. doi:10.20382/jocg.v10i2a3.

13 Prosenjit Bose, Leonidas J. Guibas, Anna Lubiw, Mark H. Overmars, Diane L. Souvaine, and Jorge Urrutia. The floodlight problem. *International Journal of Computational Geometry and Applications*, 7(1/2):153–163, 1997. doi:10.1142/S0218195997000090.

14 Sergio Cabello. Approximation algorithms for spreading points. *Journal of Algorithms*, 62(2):49–73, 2007. doi:10.1016/j.jalgor.2004.06.009.

15 Paola Cappanera. A survey on obnoxious facility location problems. Technical report, Università di Pisa, 1999.

16 Barun Chandra and Magnús M. Halldórsson. Approximation algorithms for dispersion problems. *Journal of Algorithms*, 38(2):438–465, 2001. doi:10.1006/jagm.2000.1145.

17 Vasek Chvátal. A combinatorial theorem in plane geometry. *Journal of Combinatorial Theory*, 18(1):39–41, 1975. doi:10.1016/0095-8956(75)90061-1.

18 Jurek Czyzowicz, Eduardo Rivera-Campo, and Jorge Urrutia. Optimal floodlight illumination of stages. In *Canadian Conference on Computational Geometry (CCCG)*, pages 393–398, 1993.

19 Mark de Berg and Amirali Khosravi. Optimal binary space partitions for segments in the plane. *International Journal on Computational Geometry and Applications*, 22(3):187–206, 2012. doi:10.1142/S0218195912500045.

20 Adrian Dumitrescu and Minghui Jiang. Dispersion in disks. *Theory of Computing Systems*, 51(2):125–142, 2012. doi:10.1007/s00224-011-9331-x.

21 Lawrence H. Erickson and Steven M. LaValle. A chromatic art gallery problem. Technical report, University of Illinois, 2010.

22 Lawrence H. Erickson and Steven M. LaValle. An art gallery approach to ensuring that landmarks are distinguishable. In *Robotics: Science and Systems VII*, 2011. doi:10.15607/RSS.2011.VII.011.

23 Erhan Erkut and Susan Neuman. Analytical models for locating undesirable facilities. *European Journal of Operational Research*, 40(3):275–291, 1989. doi:10.1016/0377-2217(89)90420-7.

24 Vladimir Estivill-Castro, Joseph O'Rourke, Jorge Urrutia, and Dianna Xu. Illumination of polygons with vertex lights. *Information Processing Letters*, 56(1):9–13, 1995. doi:10.1016/0020-0190(95)00129-Z.

25 Sándor P. Fekete, Stephan Friedrichs, Michael Hemmer, Joseph S. B. Mitchell, and Christiane Schmidt. On the chromatic art gallery problem. In *Canadian Conference on Computational Geometry (CCCG)*, 2014. URL: http://www.cccg.ca/proceedings/2014/papers/paper11.pdf.

26 Sándor P. Fekete and Henk Meijer. Maximum dispersion and geometric maximum weight cliques. *Algorithmica*, 38(3):501–511, 2004. doi:10.1007/s00453-003-1074-x.

27 Jirí Fiala, Jan Kratochvíl, and Andrzej Proskurowski. Systems of distant representatives. *Discrete Applied Mathematics*, 145(2):306–316, 2005. doi:10.1016/j.dam.2004.02.018.

28 Steve Fisk. A short proof of Chvátal's watchman theorem. *Journal of Combinatorial Theory*, 24(3):374, 1978. doi:10.1016/0095-8956(78)90059-X.

29 Michael Formann and Frank Wagner. A packing problem with applications to lettering of maps. In *Symposium on Computational Geometry (SoCG)*, pages 281–288, 1991. doi:10.1145/109648.109680.

30 Frank Hoffmann. On the rectilinear art gallery problem. In *International Colloquium on Automata, Languages and Programming (ICALP)*, pages 717–728, 1990. doi:10.1007/BFb0032069.

31 Frank Hoffmann, Klaus Kriegel, Subhash Suri, Kevin Verbeek, and Max Willert. Tight bounds for conflict-free chromatic guarding of orthogonal art galleries. *Computational Geometry*, 73:24–34, 2018. doi:10.1016/j.comgeo.2018.01.003.

32 Hiro Ito, Hideyuki Uehara, and Mitsuo Yokoyama. NP-completeness of stage illumination problems. In *Japanese Conference on Discrete and Computational Geometry (JCDCG)*, pages 158–165, 1998. doi:10.1007/978-3-540-46515-7_12.

33 Chuzo Iwamoto and Tatsuaki Ibusuki. Computational complexity of the chromatic art gallery problem for orthogonal polygons. In *Conference and Workshops on Algorithms and Computation (WALCOM)*, pages 146–157, 2020. doi:10.1007/978-3-030-39881-1_13.

34 Minghui Jiang, Sergey Bereg, Zhongping Qin, and Binhai Zhu. New bounds on map labeling with circular labels. In *Symposium on Algorithms and Computation (ISAAC)*, pages 606–617, 2004. doi:10.1007/978-3-540-30551-4_53.

35 Jeff Kahn, Maria Klawe, and Daniel Kleitman. Traditional galleries require fewer watchmen. *SIAM Journal on Algebraic Discrete Methods*, 4(2):194–206, 1983. doi:10.1137/0604020.

36 James King and Erik Krohn. Terrain guarding is NP-hard. *SIAM Journal on Computing*, 40(5):1316–1339, 2011. doi:10.1137/100791506.

37 D. T. Lee and Arthur K. Lin. Computational complexity of art gallery problems. *IEEE Transactions on Information Theory*, 32(2):276–282, 1986. doi:10.1109/TIT.1986.1057165.

38 Shimin Li and Haitao Wang. Dispersing points on intervals. *Discrete Applied Mathematics*, 239:106–118, 2018. doi:10.1016/j.dam.2017.12.028.

39 Bengt J. Nilsson, David Orden, Leonidas Palios, Carlos Seara, and Pawel Zylinski. Illuminating the x-axis by α-floodlights. In *Symposium on Algorithms and Computation (ISAAC)*, pages 11:1–11:12, 2021. doi:10.4230/LIPIcs.ISAAC.2021.11.

40 Joseph O'Rourke. An alternate proof of the rectilinear art gallery theorem. *Journal of Geometry*, 21(1):118–130, 1983. doi:10.1007/BF01918136.

41 Joseph O'Rourke. *Art gallery theorems and algorithms*. Oxford New York, NY, USA, 1987.

42 Christian Rieck and Christian Scheffer. The dispersive art gallery problem, 2022. arXiv:2209.10291.

43 Dietmar Schuchardt and Hans-Dietrich Hecker. Two NP-hard art-gallery problems for ortho-polygons. *Mathematical Logic Quarterly*, 41:261–267, 1995. doi:10.1002/malq.19950410212.

44 Thomas C. Shermer. Recent results in art galleries (geometry). *Proceedings of the IEEE*, 80(9):1384–1399, 1992.

45 William L. Steiger and Ileana Streinu. Illumination by floodlights. *Computational Geometry*, 10(1):57–70, 1998. doi:10.1016/S0925-7721(97)00027-8.

46 Jorge Urrutia. Art gallery and illumination problems. In *Handbook of Computational Geometry*, pages 973–1027, 2000. doi:10.1016/b978-044482537-7/50023-1.

47 Chris Worman and J. Mark Keil. Polygon decomposition and the orthogonal art gallery problem. *International Journal on Computational Geometry and Applications*, 17(2):105–138, 2007. doi:10.1142/S0218195907002264.