Designing Art Galleries

Toon van Benthem
Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands
a.t.v.benthem@student.tue.nl

Kevin Buchin
Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands
k.a.buchin@tue.nl

Irina Kostitsyna
Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands
i.kostitsyna@tue.nl

Stijn Slot
Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands
s.j.slot@student.tue.nl

Abstract

We present a method for generating interesting levels based on several NP-hardness reductions for a puzzle game based on the Art Gallery problem.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Art Gallery problem, NP-hard, puzzle, level generation

Digital Object Identifier 10.4230/LIPIcs.SoCG.2020.80

Category Media Exposition

Supplementary Material game: `https://www.win.tue.nl/~kbuchin/proj/ruler/art/`, source code: `https://github.com/kbuchin/ruler/`

1 Introduction

The *Art Gallery problem* is a classic visibility problem in computational geometry: given a polygon (i.e., an art gallery), find the smallest number of points (guards) inside the polygon such that the guards together see the interior of the polygon. The Art Gallery problem was shown to be NP-hard by Lee and Lin [3] by a reduction from 3SAT.

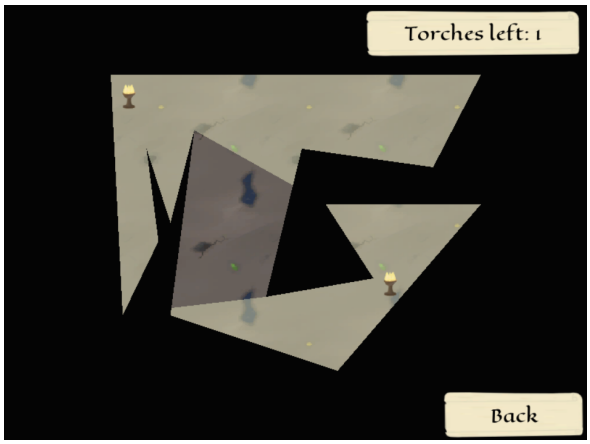

Figure 1 Screenshot from the Art Gallery Game: Two torches/guards have been placed so far, which is not sufficient to see the whole polygon.

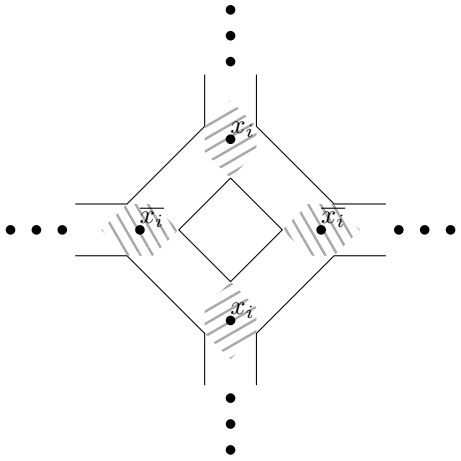

Figure 2 Variable gadget for 3SAT.

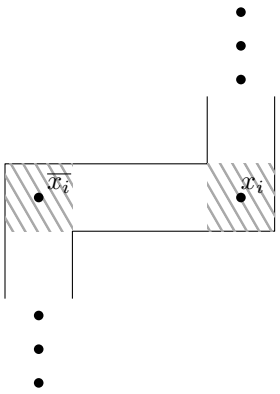

Figure 3 Wire gadget for 3SAT.

In a collection of games based on computational geometry [2], we recently published an Art Gallery puzzle[1], in which the player has to solve the Art Gallery problem, see Figure 1. The goal of this work is to develop methods for generating interesting levels for the Art Gallery puzzle based on NP-hardness reductions. We design levels, i.e., art galleries, based on three hardness reductions, from planar 3SAT, from planar vertex cover, and from geometric hitting set. We present the details of level generation using NP-hardness reductions in Section 2. We discuss how to measure the quality of the levels in Section 3. To evaluate the generated levels we have performed a user study, the results of which we discuss in Section 4.

2 Reductions

Next we present three NP-hardness reductions for the Art Gallery problem. The constructions described can be used to generate new levels, starting from an instance of the respective original problem, for the Art Gallery game.

Planar 3SAT. The first method is based on the reduction from the planar 3SAT [4]. Given an instance of the planar 3SAT, we construct an instance of the Art Gallery problem for polygons with holes. The 3SAT formula is satisfiable if and only if the resulting polygon can be guarded by K or fewer guards, for some K. For each variable and clause of the 3SAT instance we construct a *variable-* and *clause-gadget* respectively, and connect them with *wire gadgets*. To connect the same variable to multiple clauses we use *split gadgets*.

Variable gadget (Fig. 2) is a cycle with an even number of corners. The essential property of the gadget is that it (ignoring outgoing corridors) can be optimally covered in two possible ways. By increasing the size of the gadget (to a $2k$-gon for $k \geq 2$) we can create more instances of x_i and $\overline{x_i}$ for a 3SAT variable x_i. This can also be achieved using a split gadget.

Wire gadget consists of a zig-zag corridor (Fig. 3), which can be optimally covered by placing the guards in every other corner. The wire thus has two settings, which depend on the initial corner in which the first guard is placed. Define the wire to be *active* for a given direction when guard are placed in every other corners starting from the second turn.

[1] `http://www.win.tue.nl/~kbuchin/proj/ruler/`

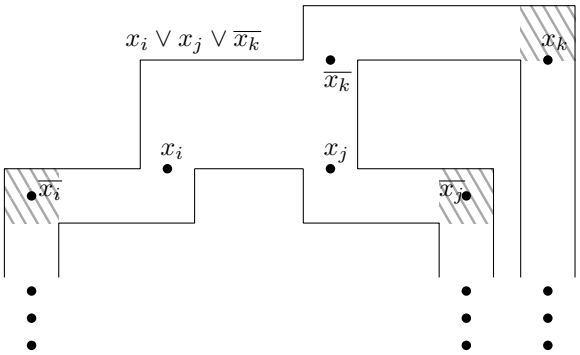

Figure 4 Clause gadget for 3SAT.

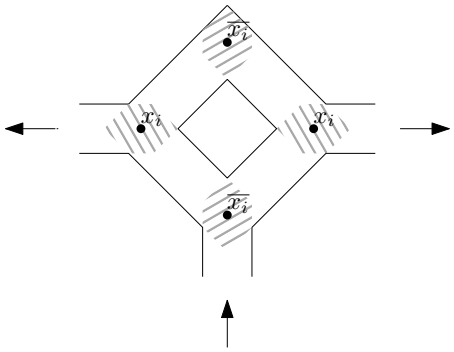

Figure 5 Split gadget for 3SAT.

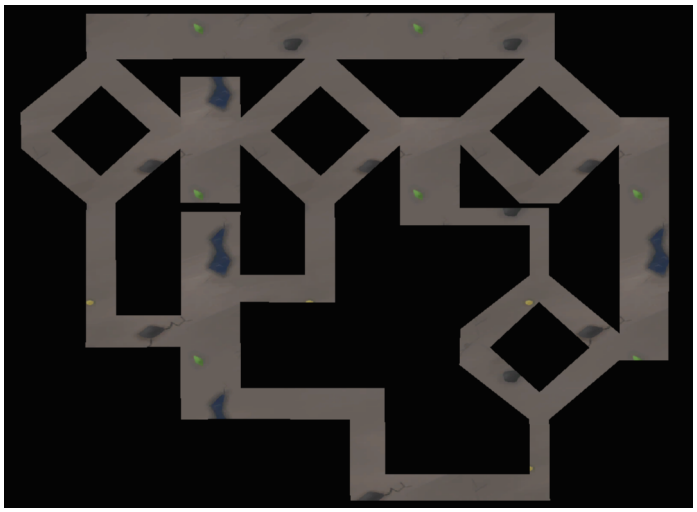

Figure 6 Art Gallery puzzle generated by reduction from planar 3SAT formula $(a \vee b \vee c) \wedge (\neg a \vee \neg b) \wedge (\neg b \vee \neg c \vee d) \wedge (\neg c \vee \neg d) \wedge (a \vee b \vee d)$; 13 guards are necessary for coverage ($K = 13$).

Clause gadget. In the clause gadget (Fig. 4) three corridors meet in a large room. It is covered if one of the wires is active, i.e., one of the wires has a guard in the corner closest to the room. The room can be of any size, as long as it cannot be covered by three inactive wires.

Split gadget. Though a split gadget (Fig. 5) is not technically necessary, for generating interesting puzzles we believe it might be better than a large variable gadget. For the split gadget we can reuse part of the variable gadget.

Puzzle level design. Figure 6 shows an example of an Art Gallery level generated from a small instance of planar 3SAT, with three variables and three clauses. Starting from a 3SAT instance, variables and clauses are replaced by gadgets and connected using corridors. The shape (e.g. variable-clause graph and corridors) for a level is designed by hand.

Planar Vertex Cover. Planar VC is another classic NP-complete problem which can be reduced to Art Gallery. For each vertex and edge, we build a *vertex-* and *edge-gadget*.

Vertex and edge gadgets. Figure 7 shows a vertex gadget. We represent the edges with positive width corridors which meet together in a crossing (highlighted in orange) corresponding to a vertex. The main property is that placing a guard at the crossing will cover all adjacent corridors. Additionally, if all corridors are covered by guards the crossing will also be covered.

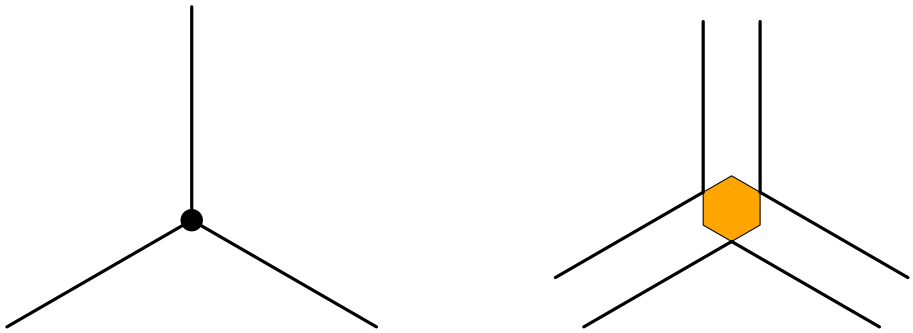

Figure 7 Vertex gadget. Left: original vertex. Right: simple vertex gadget.

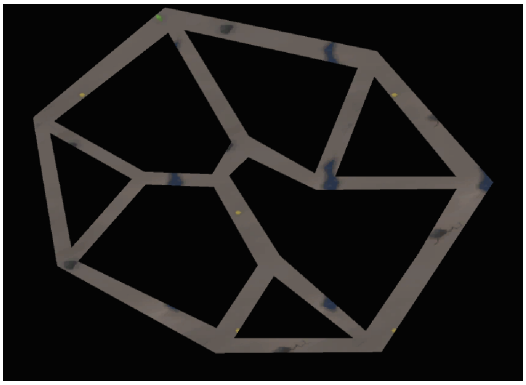

Figure 8 From Vertex Cover ($K = 7$).

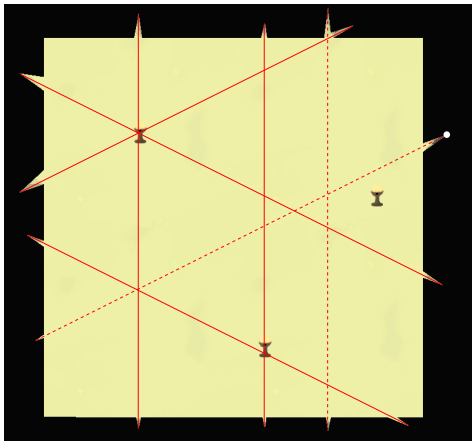

Figure 9 From Geometric Hitting Set ($K = 3$).

One subtlety for the vertex gadget is that corridors must not be collinear for the reduction to work. Potential issues may also arise from the interplay of the thickness of the corridors and the general layout of the construction. However, by carefully designing the layout of the graph and by adjusting the thickness of the corridors, these issues can be avoided.

Puzzle level design. In the final construction we use a planar embedding of the graph to create the Art Gallery level. Figure 8 shows an example of a generated level.

Geometric Hitting Set. A third NP-hardness reduction, used for level generation, is from hitting the lines of a line arrangement with a minimum number of intersection points, the so called "spike box" construction [1]. For a given arrangement of lines in the plane, construct a large box containing all the intersections of the lines. For each line attach two spikes sticking out of the box. Guarding the spike box then is equivalent to hitting all the lines in the line arrangement. A spike box construction is shown in Figure 9, where solid lines are covered by the two guards, and dashed lines are not covered.

3 Defining Interestingness

To be able to compare the three described methods, and to evaluate the quality of generated puzzles, we measure how interesting the resulting puzzles are to a human. It may occur that a certain puzzle is trivial to an algorithmic solver but a human may have a very hard time in finding the right step. We believe the quality of a puzzle to be a combination of the following factors: *niceness* of the polygon shape, *difficulty* of the puzzle, *fun* in solving the puzzle.

The intuition behind this is the following. A puzzle that is too easy is often not interesting, as the solution will be too obvious. A puzzle that is not aesthetically pleasing, for example containing too fine features, can be frustrating. A puzzle that is not fun, e.g., the logic behind which is obvious, is often not interesting, since players may get bored.

4 User Study

In a user study, participants ($n = 17$) were asked to solve puzzles and answer a set of questions. Specifically the flow for each art gallery was: *(i)* observe an Art Gallery puzzle level for five seconds, *(ii)* answer a pre-solve survey, *(iii)* attempt to solve the Art Gallery puzzle level, and *(iv)* answer a post-solve survey.

The conclusions we draw from the user study is that the puzzles generated based on the reduction from the planar 3SAT were the most difficult ones (indicated by high solve time and placement attempts), yet also the most enjoyable. The single most enjoyed puzzle is a 3SAT puzzle (a puzzle similar to Fig. 6). Vertex cover puzzles were the most logical to solve, yet also the easiest set of puzzles. The spike box puzzles were considered the least nice. Their too fine features resulted in significantly lower average scores for shape aesthetics, solving fun, logical approach, and shape uniqueness. The single least enjoyed puzzle is a spike box puzzle (one similar to Fig. 9), the main problem being the lack of clarity regarding what part is still not illuminated. As a consequence, we added a *feedback point* which highlights some point in the polygon (white dot in Fig. 9) that is not yet covered.

References

1 Yoav Amit, Joseph S.B. Mitchell, and Eli Packer. Locating guards for visibility coverage of polygons. *International Journal of Computational Geometry & Applications*, 20(05):601–630, 2010.

2 Sander Beekhuis, Kevin Buchin, Thom Castermans, Thom Hurks, and Willem Sonke. Ruler of the plane – games of geometry (multimedia contribution). In *Proc. 33rd International Symposium on Computational Geometry (SoCG)*, pages 63:1–63:5, 2017.

3 Der-Tsai Lee and Arthur K. Lin. Computational complexity of art gallery problems. *IEEE Transactions on Information Theory*, 32(2):276–282, 1986.

4 David Lichtenstein. Planar formulae and their uses. *SIAM Journal on Computing*, 11(2):329–343, 1982.