Catching a Polygonal Fish with a Minimum Net

Sepideh Aghamolaei
Sharif University of Technology, Tehran, Iran
aghamolaei@ce.sharif.edu

Abstract. Given a polygon \( P \) in the plane that can be translated, rotated and enlarged arbitrarily inside a unit square, the goal is to find a set of lines such that at least one of them always hits \( P \) and the number of lines is minimized. We prove the solution is always a regular grid or a set of equidistant parallel lines, whose distance depends on \( P \).

Keywords: Stabbing Problem, Nets.

1 Introduction

This was an open problem of CCCG 2020 conference, posed by Joseph O’Rourke. We show the cases discussed in the conference are the only possible cases and they give the optimal solution for any convex shape.

Such a net must stab any subset of copies of a shape, and the minimum number of lines for stabbing all copies is a lower bound on the minimum number of lines in the minimum net.

2 Catching the Fish!

Algorithm. For the case with axis-parallel lines, if the smallest enclosing square has side length \( a \) and the smallest bounding rectangle (arbitrarily oriented) has sides \( b \) and \( c \), \( b \leq c \), then if \( a < c \), both vertical and horizontal grid lines are required with distance \( a \) from each other, otherwise \( a = b = c \), and only one set of parallel lines is enough with distance \( a \) from each other.

Example. In the example of the slides from CCCG 2020, the \( 1 \times 3 \) rectangle has the smallest bounding square of side length \( 2\sqrt{2} \approx 2.8 \). So, this is the first case in our algorithm, and a regular grid of side length \( 2\sqrt{2} \) is the solution. The other example was a square of side length 1, for which the second case in our algorithm is used which gives a set of parallel vertical (or a set of horizontal lines) with distance 1 from each other.

Lemma 1. Rotate \( P \) to make the maximum distance of \( P \) parallel to one of the axes. Consider all translations of \( P \) in the direction of one of the axes by the width of \( P \) in the direction of that axis, and then in the perpendicular direction until they are disjoint with respect to the perpendicular axis as well. The number of these shapes is a lower bound on the number of lines (including the boundary lines).

Proof. Any of the shapes in the statement of the lemma is a valid movement of \( P \), so, the lines must hit them. Since these shapes are all disjoint, each of them requires a line, if we want to use
only axis-parallel lines. If the optimal solution is in another direction, we rotate the shape until its longest edge is parallel to one of the axes, and scale down $P$ such that the same number of copies fit in the direction of that axis and build a new bounding square with side $\rho$. This gives a scaling of the original problem by $\rho$, and some corners. So, the solution is at least as much as the original problem, if we had scaled the original instance by $\rho^{-1}$. So, this bound is still a lower bound on the optimal solution. The minimum number of lines is given when the largest width (the diameter of $P$) is used, since the side-length of the bounding square is fixed.

Since the algorithm gives a solution with at most twice as many lines as in Lemma 1, so far we know the algorithm is a 2-approximation. Now, we show it is an optimal solution.

Now, we give a better lower bound on the number of lines.

**Lemma 2.** Consider the shape $R$ with maximum width among the smallest bounding rectangle and the smallest bounding square of $P$. Build a grid with cells of shape $R$, with a copy of $P$ inside each of them. The lines of the grid (1D or 2D based on the algorithm) are the smallest subset of lines that stab all these shapes.

**Proof.** In the case where only one set of parallel lines are used, based on Lemma 1 that is the optimal solution. The other case is based on the minimum enclosing square, so, if only one direction is chosen, the shapes can be translated in the perpendicular direction. Among the pairs of orthogonal lines with equal distances between each set of parallel lines, the ones that are parallel to the sides of the bounding box have the minimum size, since other directions inside the bounding box have higher directional width. The minimum side-length happens when $R$ has sides parallel to the bounding box, which also gives the maximum number of lines required to stab all the translated shapes. □

The grid lines of the algorithm cover all shapes, so that is an upper bound on the number of lines required to solve the problem. Based on Lemma 2 the same number of lines is also the lower bound. So, it is the optimal solution:

**Theorem 1.** Our algorithm finds the net of minimum size that stabs any transformation (translation, rotation, enlargement) of a convex polygon $P$. 

