Exact Solutions for Area-Optimal Simple Polygonization Problems*

Natanael Ramos**, Pedro J. de Rezende***, and Cid C. de Souza†

Institute of Computing, UNICAMP, Campinas, Brazil
{nramos,rezende,cid}@ic.unicamp.br

Abstract. In problems of Area-Optimal Simple Polygonization, we are given a set S of points in the plane and the objective is to find a minimum- or maximum-area polygon with vertex-set S. These problems were recently featured in an international optimization contest and drew a lot of attention. Here, we propose a novel Integer Programming formulation for both problems and present promising preliminary results.

Keywords: Computational Geometry · Combinatorial Optimization · Integer Programming · Polygonization.

1 Introduction

Computational Geometry deals with the study of problems involving proximity, reachability, convexity and the construction of various geometric objects from a given set of points. In particular, one is often interested in building a single simple polygon from a set S of points in the plane that optimizes an objective function. These are referred to as *optimal polygonization* problems. Most commonly, this function requires us to maximize or minimize the area of the resulting polygon. This gives rise to Max-Area and Min-Area problems, respectively, which are the object of study in this paper.

Both problems were proved to be $\mathbb{NP}$-complete and there are exact, heuristic and approximation algorithms for them in the literature. Considerable attention was recently given to these problems during the 2019 CG Challenge[1], which prompted contestants to find good viable solutions for a wide spectrum of benchmark instances with up to 1,000,000 points.

As for exact methods, the best known results are by Fekete et al. [1], where instances of up to 16 points were solved to optimality for Min-Area, and of at most 19 points for Max-Area. Thus, there still is room for some serious investigative effort in this topic.

Here, we propose a new Integer Programming (IP) formulation for both Min-Area and Max-Area, which is described in Section 2.

* This work was supported in part by Grants from CAPES, Brazil – Finance Code 001; CNPq, Brazil: #140300/2020-1,#311140/2014-9, #304727/2014-8 and FAPESP, Brazil: #2014/12236-1, #2018/11100-0.

** [0002−3546−9787], *** [0000−0002−9529−4253], † [0000−0002−5945−0845].

[1] https://cgshop.ibr.cs.tu-bs.de/competition/cg-shop-2019/

2 Integer Programming Models

In this section, we introduce our IP formulation for both Max-Area and Min-Area, comprised of objective function (1) and constraints (2) to (6). Denote a segment between points $i, j \in S$ by ij and a triangle by ijk when its vertices are $i, j, k \in S$. The set of all segments between points of S is indicated by $E(S)$, and the set of all empty triangles in S is, henceforth, $\Delta(S)$. Also, $\mathcal{X}(E(S))$ is the set of pairs of segments that cross each other. The area of a given triangle ijk is written $\mathcal{A}(ijk)$. Given a subset of points $U \subset S$, $\delta(U)$ is the set $\{ij : i \in U \text{ and } j \in S \setminus U\}$.

We have two sets of binary variables $\{x_{ij}, ij \in E(S)\}$ and $\{t_{ijk}, ijk \in \Delta(S)\}$.

$$\max / \min \sum_{ijk \in \Delta(S)} \mathcal{A}(ijk) t_{ijk} \tag{1}$$

$$\sum_{ijk \in \Delta(S)} t_{ijk} = n - 2 \tag{2}$$

$$x_{ij} + x_{kl} \leq 1 \qquad \forall (ij, kl) \in \mathcal{X}(E(S)) \tag{3}$$

$$x_{ij} + x_{ik} + x_{jk} - t_{ijk} \leq 2 \qquad \forall ijk \in \Delta(S) \tag{4}$$

$$x_{ij} \leq \sum_{k \in S \setminus \{i,j\}} t_{ijk} \leq 2 x_{ij} \qquad \forall ijk \in \Delta(S) \tag{5}$$

$$\sum_{ij \in \delta(U)} x_{ij} \geq 2 \qquad \forall U \subset S \tag{6}$$

$$\sum_{jk \mid ijk \in \Delta(S)} t_{ijk} = \sum_{j \in S \setminus \{i\}} x_{ij} - 1 \qquad \forall i \in S \tag{7}$$

Constraint (2) sets the number of triangles in a solution. Crossing segments are forbidden by inequalities (3). Constraints (4) ensure that if three segments ij, ik and jk are part of a solution, then triangle ijk must also be in it. For each segment ij in a solution, at least one and at most two triangles must be supported by ij, which (5) encodes. Constraints (6) assure that a single connected component is obtained. Lastly, constraints (7) say that a point cannot be surrounded by triangles, so, from (6), a solution cannot have internal points. After solving our model, a solution to either Max-Area or Min-Area is the polygon formed by the sequence of segments that support a single triangle.

3 Results

We implemented the above model and solved it using the CPLEX solver v. 12.9, with a time limit of one hour. The CG Challenge instance set, with 6 instances for each $n \in \{10, 15, 20, 25\}$, was used as a benchmark. Considering Max-Area, our formulation was able to solve to optimality all instances with up to 15 points and two with 20. Furthermore, we observed that our model is more effective for Min-Area, solving to optimality all instances with up to 20 points.

References

[1] S. P. Fekete et al. "Area- and Boundary-Optimal Polygonalization of Planar Point Sets". In: *31st European Workshop on Computational Geometry*. Ljubljana, Slovenia, 2015, pp. 133–136. URL: http://eurocg15.fri.uni-lj.si/pub/eurocg15-book-of-abstracts.pdf.