Bicriteria Polygon Aggregation with Arbitrary Shapes

Lotte Blank
University of Bonn, Germany

David Eppstein
University of California, Irvine, CA, USA

Jan-Henrik Haunert
University of Bonn, Germany

Herman Haverkort
University of Bonn, Germany

Benedikt Kolbe
University of Bonn, Germany
Hausdorff Center for Mathematics, Bonn, Germany
Lamarr Institute, Bonn, Germany

Philip Mayer[1]
University of Bonn, Germany

Petra Mutzel
University of Bonn, Germany
Lamarr Institute, Bonn, Germany

Alexander Naumann
University of Bonn, Germany

Jonas Sauer
Karlsruhe Institute of Technology, Germany

Abstract

We study the problem of aggregating a set of polygons by covering them with disjoint representative regions, thereby inducing a clustering of the polygons. Equivalently, this can be seen as a fence enclosure problem, where the goal is to surround the polygons with a set of closed curves. Our objective is to minimize a weighted sum of the total area and the total perimeter of the regions, which naturally extends other fencing problems and has applications in geographical information systems. Previously, this objective was only studied in a restricted variant, in which the boundary curves of the regions must be selected from a fixed subdivision of the plane. It is natural to ask whether the problem is still tractable if this restriction is removed, allowing output regions to be bounded by arbitrary curves. We provide a positive answer in the form of an algorithm with runtime $\tilde{\mathcal{O}}(n^4)$, where n is the number of input vertices. To achieve this, we fully characterize the optimal solutions by showing that their boundaries are composed of input edges and circular arcs of constant radius. Additionally, we consider the parametric problem, where for every weighting factor we seek a solution that is optimal for that trade-off of area and perimeter. We show that $\mathcal{O}\left(n^2\right)$ combinatorial solutions suffice to describe all optimal solutions across all weighting factors, and provide both an exact algorithm and an approximation scheme. To make the algorithms scalable in practice, we develop engineering techniques that exploit structural properties of the solutions. Our experimental evaluation on real-world data shows linear runtime in practice, even for the parametric variant.

2012 ACM Subject Classification Theory of computation → Computational geometry; Information systems → Geographic information systems

Keywords and phrases polygon aggregation, fencing, minimum-perimeter clustering, map generalization, urban analytics

Digital Object Identifier 10.4230/LIPIcs.ESA.2026.11

Related Version *Full Version*: `https://arxiv.org/abs/2507.11212`

Supplementary Material
Software (Source Code): `https://github.com/GeometryCodes/ArbitraryBicriteriaShapes`
archived at `swh:1:dir:7026652d840d88c43eb4c6fa71e35bcda85d1720`

[1] corresponding author

Funding This research was partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant FOR-5361 – 459420781.
David Eppstein: Research supported in part by NSF grant CCF-2212129.
Benedikt Kolbe: This research has partly been funded by the Federal Ministry of Education and Research of Germany and the state of North-Rhine Westphalia as part of the Lamarr-Institute for Machine Learning and Artificial Intelligence.

Acknowledgements We thank Anne Driemel and Frederik Brüning for fruitful initial discussions. Additionally, we thank the anonymous reviewers for helping to improve this paper.

1 Introduction

We study the task of aggregating a set of polygons $\mathcal{B}$ in the plane by covering them with a set of disjoint representative regions $\mathcal{S}$, i.e., for every polygon $B \in \mathcal{B}$ there must be a region $S \in \mathcal{S}$ with $B \subseteq S$; see Figure 1. This problem formulation appears frequently in cartographic applications [31, 30, 18, 24]. In the context of computational geometry, it can be classified as a type of *fence enclosure*, or *fencing*, problem [1, 4, 12]. Here, given a set of points or polygons, we want to enclose them by a set of closed curves (fences) such that a certain objective is achieved. For example, Arkin et al. [4] consider the problem of surrounding resource-rich plots of land by one or multiple fences while minimizing construction costs.

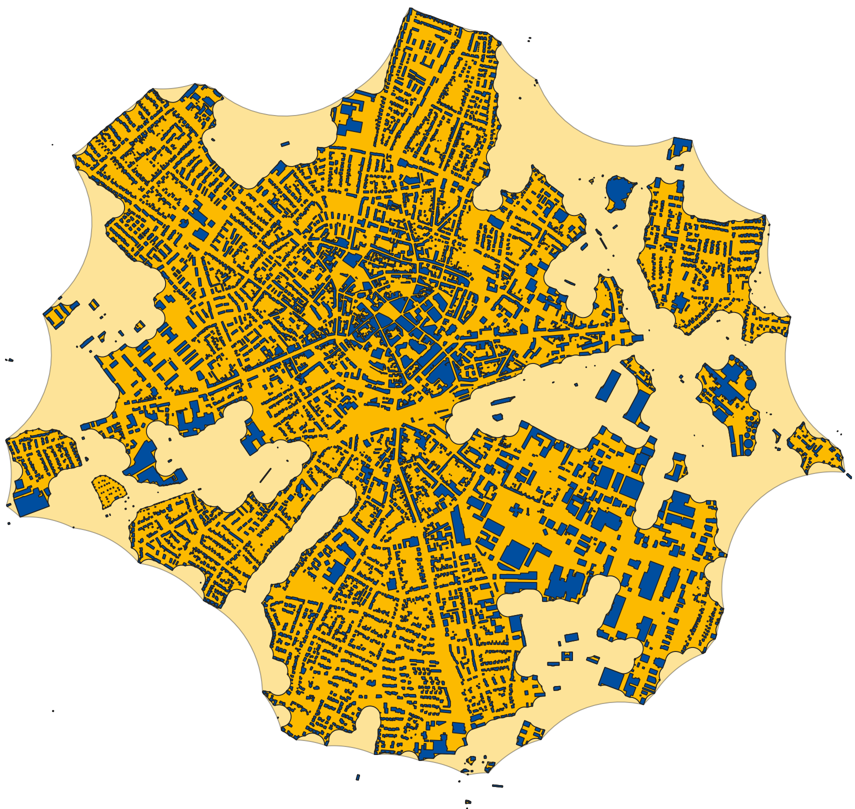


Figure 1 All figures show input polygons in blue and solutions in orange. Shown here is the unrestricted polygon aggregation problem with objective function g_α for the town of Euskirchen. Two optimal solutions are superimposed in increasingly lighter shades of orange, with $\alpha \in \{500, 3000\}$.

In our work, we consider a bi-criteria objective function, which was first proposed by Rottmann et al. [31] and is motivated by geographical information systems (GIS): a solution $\mathcal{S}$ is optimal for some parameter $\alpha \in [0, \infty]$ if it minimizes the linear combination $g_\alpha(\mathcal{S}) = A(\mathcal{S}) + \alpha P(\mathcal{S})$ of the overall area $A(\mathcal{S})$ and the overall perimeter $P(\mathcal{S})$. In the fencing analogy, the perimeter term corresponds to the cost of the fences and the area term to the cost of acquiring additional land. Both terms have been considered as separate objective functions for the fencing problem [4, 1], but not jointly. From a theoretical perspective, their combination into a bi-criteria cost function provides a natural generalization.

The work by Rottmann et al. [31] considers the objective function g_α in very restricted settings, leading to a constrained problem. Here, a subdivision $\mathcal{D}$ of the plane (e.g., a constrained Delaunay triangulation) is fixed and the boundary curves of the representative regions must be edges of $\mathcal{D}$. Rottmann et al. give a transformation of this problem to a graph-cut problem, allowing it to be solved in $\mathcal{O}(n^2/\log n)$ time. In addition to the problem where α is fixed, they also consider the parametric problem of finding a set $\mathfrak{S}$ that contains an optimal solution for every value of α. A powerful property of the objective function g_α is that the optimal solutions are *nested* with respect to α, i.e., an optimal solution for value α is contained in all optimal solutions for values $\alpha' > \alpha$ [31]. This implies that the optimal solutions for different values of α induce a hierarchical clustering of the input polygons. Using this, they show that $\mathfrak{S}$ has size $\mathcal{O}(n)$ and can be computed using a dichotomic scheme.

Fixing the subdivision $\mathcal{D}$ makes both problems much easier to solve, but this comes at the cost of constraining the solution space. This raises a natural question: is the problem still tractable if the restrictions imposed by the subdivision are lifted? We answer this question by investigating a more general problem setting: any set of regions with differentiable boundary curves that covers $\mathcal{B}$ is allowed, and we aim to find one that minimizes g_α.

GIS Applications. Rottmann et al. introduced the bicriteria polygon aggregation problem in a GIS context, where polygons represent map features such as buildings or bodies of water. One potential application is map generalization, where a given map must be transformed into a less detailed map to be displayed at a smaller scale. This requires aggregation and omission of details. Especially at the transition from large-scale to small-scale maps, buildings must be grouped into areas representing entire settlements [18, 31, 30]. The delineation of settlement areas is also an important task in urban analytics, e.g., to study urban growth and morphology, where the aggregations provide evidence of urban land use [5, 15, 20].

In such applications, the objective function g_α models a trade-off between two natural goals: The output regions should be compact and have low shape complexity, which is captured by minimizing the perimeter [26, 21]. On the other hand, they should remain as faithful as possible to the input polygons, which is captured by minimizing the area. The trade-off parameter α corresponds to the scale at which the map is viewed: larger scales (zoomed in) prioritize faithfulness to the original input, whereas smaller scales (zoomed out) prioritize simplicity. The nestedness property with respect to the parameter α guaranteed by our problem formulation provides visual stability across scale changes [18, 31, 30], which is a desirable property in map generalization. Furthermore, the spatial patterns encoded in the induced (hierarchical) clustering are useful for data analysis tasks in urban morphology [3], such as the classification and comparison of different settlement types [25, 33].

Our Contributions.

- We characterize solutions for the unrestricted polygon aggregation problem (denoted P_{F}^{α}) and show that their boundary curves consist of parts of the input boundaries and circular arcs. Unlike previous approaches, the shapes of the boundary curves are not specified in advance. Rather, they emerge naturally as a result of minimizing the objective.

- Using this characterization, we construct a subdivision $\mathcal{D}_C$ of size $\mathcal{O}(n^4)$ such that an optimal solution for P_{F}^{α} can be obtained as an optimal solution for the subdivision-restricted problem $P_{\mathcal{D}_C}^{\alpha}$ in $\tilde{\mathcal{O}}(n^4)$ time.
- We show that the optimal solutions for P_{F}^{α} are nested with respect to α. In addition, we show that the problem also exhibits a *subset-nestedness* property: if new polygons are added to the input, the optimal solution will only grow.
- We propose a preprocessing algorithm that leverages subset-nestedness by iteratively merging sets of nearby polygons that belong to the same output region. This drastically improves the runtime, especially for large α values.
- We prove that a solution set $\mathfrak{S}$ for the parametric unrestricted aggregation problem contains $\mathcal{O}(n^2)$ (combinatorial) solutions.
- We give a fully polynomial-time approximation scheme (FPTAS) for the parametric problem. We also give an exact polynomial-time algorithm under the assumption that we have access to an oracle that returns an intersection point of two functions.
- We present engineering ideas, based on α-nestedness, to improve the runtime in practice.
- We evaluate our algorithms in the map generalization task where real-world building footprints from towns and cities are aggregated. Here, we show that, despite the $\tilde{\mathcal{O}}(n^4)$ worst-case runtime, our approach scales linearly with the instance size and α in practice.

2 Related Work

Fencing. Several fencing problems for point sets in the plane were first introduced by Arkin, Khuller, and Mitchell [4], as well as Capoyleas, Rote, and Woeginger [12]. Capoyleas et al. mainly study the k-clustering variant, where the goal is to enclose the input points with k fences, whereas Arkin et al. allow either exactly one or an arbitrary number of fences and consider several possible cost measures, including the fence length and the enclosed area.

Abrahamsen et al. [1] give polynomial-time algorithms for two fencing variants: (1) Find a set of enclosing curves of minimal length where each curve additionally incurs a fixed opening cost. (2) Find a set of k enclosing curves of minimal length. Another fencing variant is *geometric multicut* [2], where input polygons are divided into k different color classes and the objective is to find a minimum-perimeter fencing that separates polygons with different colors from each other. For $k = 2$, this can be reduced to a min-cut problem and solved in $\tilde{\mathcal{O}}(n^4)$ time. For $k \geq 3$, it is NP-hard, but an approximation algorithm is given.

Polygon Aggregation. Polygon aggregation can be viewed as a combination of two sub-problems: partitioning the polygons into clusters and computing a representative shape for each cluster. Several popular approaches specify the type of representative shape in advance, which reduces it to a clustering problem. For example, minimum-perimeter fencing [1] corresponds to choosing convex hulls as the representative shapes. A drawback of this approach is that it tends to include large empty areas in the solution. The same problem occurs when using the χ-hull, a generalization of the convex hull for aggregation purposes [16]. Other generalizations of the convex hull, such as the concave hull introduced by Moreira and Santo [28], or α-shapes [17], mitigate this problem. However, both of these approaches can lead to narrow bridges in the output (see Appendix A). Recently, Funke and Storandt [18] proposed an approach that first clusters the polygons based on a distance threshold β and then computes an α-shape for each cluster. Clusters with fewer points than some threshold value μ are discarded altogether. Their approach runs in $\mathcal{O}(n \log n)$ time. Although this allows for the aggregation of country-sized instances within minutes, a major drawback is that the three parameters (α, β and μ) must be specified by the user, and the narrow bridge problem is only alleviated if they are chosen appropriately for the given instance.

Two approaches deviate from the paradigm of choosing the type of representative shape in advance: the method of Damen, van Kreveld and Spaan [14], which applies morphological closure and opening operators to the input polygons, and the previously mentioned approach by Rottmann et al. [31]. The parametric problem proposed by Rottmann et al. has recently been reduced to the source-sink-monotone parametric min-cut problem [6], which enables the use of faster algorithms in practice. Using this reduction, instances representing large cities can be solved optimally within seconds.

Map Generalization. Besides polygon aggregation, map generalization involves a range of well-studied operators, including elimination, displacement, exaggeration, detail elimination, squaring, and typification (see [27, 14] for further reading). Some approaches focus primarily on the simplification (and local aggregation) of polygons in large-scale maps, where individual buildings are still distinguishable [22, 11, 10]. However, maps of scales smaller than $1 : 50\,000$ usually do not show individual buildings but rather representatives of entire settlements [35].

3 Problem Definition

We introduce some basic notation. A curve is a continuous map $f\colon [0,1] \to \mathbb{R}^2$. We will also identify a curve with its image. The length of a piecewise differentiable curve f is denoted by $\ell(f)$. The area of a (Lebesgue-measurable) closed region S is denoted by $A(S)$. The perimeter $P(S)$ is the length of its boundary ∂S. If S is a polygon, its vertices are denoted by $V(S)$, its edges by $E(S)$, and the convex hull of S by $\operatorname{conv}(S)$. Additionally, we assume that outer boundary curves of regions are oriented counter-clockwise and inner boundary curves (which enclose holes) are oriented clockwise. Thus, along each boundary curve, the interior of S lies locally to the left. We extend the above definitions to sets $\mathcal{S} = \{S_1, \dots, S_k\}$ of (disjoint) closed regions or polygons. Most applications focus on polygonal inputs, but during the preprocessing algorithm discussed at the end of Section 4, we have to consider intermediate sub-instances that are bounded by circular arcs in addition to straight lines.

▶ **Definition 1** (Circular polygon)**.** *In a circular polygon, every edge, i.e., a boundary segment connecting two vertices, is a straight line or a circular arc that bends inwards and has central angle not larger than π. In an α-circular polygon, all circular arcs have radius α.*

▶ **Problem 1** (Unrestricted polygon aggregation problem P_{F}^{α})**.** *Let $\mathcal{B}$ be a set of interior-disjoint (circular) polygons and $\alpha \geq 0$. A (feasible) solution to P_F^{α} is a set $\mathcal{S}$ of disjoint closed regions with piecewise differentiable boundary curves that cover all input polygons, i.e., for every $B \in \mathcal{B}$ there exists an $S \in \mathcal{S}$ with $B \subseteq S$. A solution is optimal if it minimizes the objective*

$$g_\alpha(\mathcal{S}) = A(\mathcal{S}) + \alpha P(\mathcal{S})$$

among all feasible solutions. For $\alpha = \infty$, the objective function becomes $g_\infty(\mathcal{S}) = P(\mathcal{S})$.

This objective function is equivalent to the function $f_\lambda(\mathcal{S}) = \lambda A(\mathcal{S}) + (1-\lambda)P(\mathcal{S})$ from [31] with the reparameterization $\alpha = \frac{1-\lambda}{\lambda}$. Additionally, most of the following results extend to inputs with arbitrary piecewise differentiable boundaries; this extension is discussed in the full version of this paper.

▶ **Definition 2.** *A curve f from u to v is a **boundary piece** of a solution $\mathcal{S}$ if $f \subseteq \partial\mathcal{S}$ and $\mathcal{S}$ includes the area to the left of f. A boundary piece f is **constrained** if $f \subseteq \partial\mathcal{B}$ and **free** if $f \cap \partial\mathcal{B} \subseteq \{u, v\}$. The set of (inclusion-)maximal free pieces in $\mathcal{S}$ is denoted by $F_{\mathcal{B}}(\mathcal{S})$.*

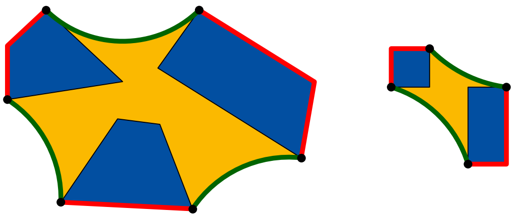


Figure 2 An example solution in which every solution region is a circular polygon. Free pieces are shown in dark green, and constrained pieces are shown in red. The vertices are the endpoints of the inclusion-maximal free pieces.

Note that every boundary curve can be described as an alternating sequence of constrained and maximal free boundary pieces; see Figure 2 for an example. The constrained pieces may consist of a single point.

4 Solving the Unrestricted Polygon Aggregation Problem

In this section, we investigate the unrestricted problem P_{F}^{α}. We begin by characterizing the shape of the free boundary pieces and show that only a polynomial number of curves need to be inspected to find an optimal solution. We then use these curves to build a subdivision $\mathcal{D}_C$ of $\operatorname{conv}(\mathcal{B}) \setminus \mathcal{B}$ that has polynomial size, thereby reducing P_{F}^{α} to the subdivision-restricted problem $P_{\mathcal{D}_C}^{\alpha}$. This allows us to apply the minimum cut-based algorithm proposed in [31] to solve the unrestricted problem in polynomial time. Finally, we note that the optimal solutions to P_{F}^{α} are nested with respect to α and with respect to introducing new input polygons. We sketch some proofs and omit others; the complete proofs can be found in the full version of this paper.

We start by characterizing the optimal solutions of the corner cases $\alpha = 0$ and $\alpha = \infty$. In the former, the input polygons $\mathcal{B}$ form an optimal solution because they minimize the area. In the latter, the perimeter must be minimized, so every region S in an optimal solution must be the convex hull $\operatorname{conv}(\mathcal{B}(S))$ of its contained (circular) polygons $\mathcal{B}(S)$. It is easy to see that the general case $\alpha \in (0, \infty)$ can be reduced to the case $\alpha = 1$ by scaling the input.

▶ **Observation 3.** *For a set $\mathcal{R}$ of regions and $c > 0$, let $\sigma_c(\mathcal{R})$ denote the same set of regions scaled uniformly by the factor c. A solution $\mathcal{S}$ for P_F^{α} with input polygons $\mathcal{B}$ and $\alpha \in (0,\infty)$ is optimal iff $\sigma_{1/\alpha}(\mathcal{S})$ is optimal for P_F^{1} with input polygons $\sigma_{1/\alpha}(\mathcal{B})$.*

Thus, we only consider the cost function g_1 for the proofs in this section. Observation 3 describes a crucial property of the problem formulation: the parameter α corresponds directly to the scale at which the input polygons are viewed. It follows that the set of solutions that are optimal for at least one value of α does not change when scaling the input.

We continue by investigating which curves locally optimize the area-perimeter trade-off.

▶ **Definition 4.** *Let f be a curve from u to v that does not intersect the line through u and v properly. Let R_f^{uv} denote the region enclosed by $\overline{uv}$ and f. The **efficiency** of f is defined as*

$$e_{uv}(f) = \begin{cases} -\ell(f) + \ell(\overline{uv}) + A(R_f^{uv}) & \text{if } f \text{ is to the left of } \overrightarrow{uv}, \\ -\ell(f) + \ell(\overline{uv}) - A(R_f^{uv}) & \text{if } f \text{ is to the right of } \overrightarrow{uv}. \end{cases}$$

If we consider a solution that uses $\overrightarrow{uv}$ as a free boundary piece, then $e_{uv}(f)$ is the improvement in the objective value if we replace $\overrightarrow{uv}$ with f. We now characterize the curves that have maximum efficiency and show that any solution that uses a curve of non-maximum efficiency can be improved locally.

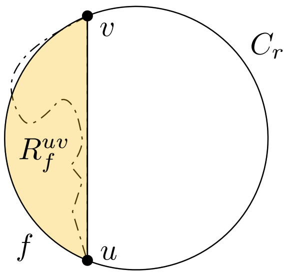


Figure 3 For fixed $\ell(f)$, the enclosed area and in turn the efficiency is maximized if f is circular.

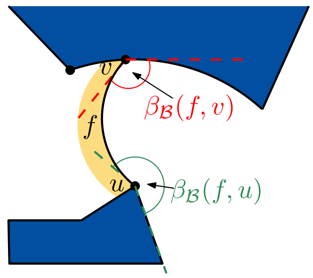


Figure 4 Two examples for the attachment angle $\beta_{\mathcal{B}}(f,x)$.

▶ **Lemma 5.** *Let u, v be two points. For fixed length l, the (unique) most efficient curve f from u to v with $\ell(f) = l$ that does not intersect the line through u and v properly is a circular arc that lies to the left of $\overrightarrow{uv}$.*

Proof. Let C_r be the circular arc with radius r from u to v in the half-plane to the left of $\overrightarrow{uv}$ such that $\ell(C_r) + \ell(f) = 2\pi r$ (see Figure 3). The curve f lies to the left of $\overrightarrow{uv}$ since the efficiency is non-positive otherwise. Hence, C_r together with f is a closed curve with perimeter $P = 2\pi r$. The isoperimetric inequality [32, 23] states that for every closed curve C with enclosed region R, it holds that $4\pi A(R) \leq \ell(C)^2$. Furthermore, equality holds if and only if C is a circle. It follows that C_r together with f has to be a circle to maximize the enclosed area and in turn maximize the efficiency. Consequently, f is a circular arc. ◀

In the remainder of this paper, we denote by C_r^{uv} the circular arc of radius r and central angle $\theta \leq \pi$ that connects the points u and v and lies to the left of $\overrightarrow{uv}$. The next lemma follows by combining Lemma 5 with an analytical argument and a local improvement step.

▶ **Lemma 6.** *Let $\mathcal{S}$ be a solution for P_F^1 and u, v two points on the boundary of $\mathcal{S}$. If u and v are connected by a free piece other than C_1^{uv}, then $\mathcal{S}$ is not optimal.*

We briefly describe how free boundary pieces attach to the input boundaries. At any point where a curve meets an input boundary, we compare their directions via their tangents; at corner points, we use the one-sided tangents of the adjacent boundary segments. This allows us to measure the counterclockwise angle between the free piece and the input boundary at the point of attachment. We denote this angle by $\beta_{\mathcal{B}}(f,x)$ and interpret it as the exterior angle at x of a solution region that uses f; see Figure 4. A formal definition is given in the full version of this paper. Using this, we characterize the free pieces that appear in an optimal solution.

▶ **Proposition 7.** *Let $\mathcal{B}$ be an instance of P_F^α for $\alpha \in (0, \infty)$. There exists a solution $\mathcal{S}$ that minimizes $g_\alpha(\mathcal{S})$, such that the boundary $\partial\mathcal{S}$ is a collection of piecewise differentiable curves and every free boundary piece f with endpoints u and v has the following properties:*
P1: *The distance $d = \ell(\overline{uv})$ of the endpoints is at most 2α.*
P2: *The piece f is circular with radius α.*
P3: *The central angle θ of f is at most π.*
P4: *If $x \in \{u, v\}$ lies on the input boundary $\partial\mathcal{B}$, then $\beta_{\mathcal{B}}(f,x) \geq \pi$.*
P5: *The piece f bends to the left (i.e., it is to the left of $\overrightarrow{uv}$).*

Proof sketch. The existence of an optimal solution follows from results in geometric measure theory and is derived in detail in the full version of this paper. Properties P1, P2, and P5 follow directly from Lemma 6. If f has a central angle $\theta > \pi$, we can replace it by the corresponding arc f' with central angle $2\pi - \theta$ that bends in the same direction. A comparison shows that the area gain is always outweighed by the perimeter loss, which implies P3. For P4, one can show (cf. full version) that if $\beta_{\mathcal{B}}(f, x) < \pi$, then a generalized version of Lemma 6 applies to the partially constrained piece formed by f concatenated with the next segment on ∂B. ◀

With this we can define a set $F_\alpha(\mathcal{B})$ of free pieces such that there exists an optimal solution that uses only maximal free pieces from $F_\alpha(\mathcal{B})$.

▶ **Definition 8.** *Let $\mathcal{B}$ be an instance of P_F^α. For $\alpha \in (0, \infty)$, let $F_\alpha(\mathcal{B})$ denote the set of circular arcs $f = C_\alpha^{uv}$ such that*

1. *the endpoints u and v lie on the input boundary $\partial\mathcal{B}$,*
2. *u and v are not in the interior of two parallel straight-line edges,*
3. *if $u \in V(\mathcal{B})$ and v lies in the interior of a circular boundary arc h (or vice versa), then u(or v) is not the center point of h,*
4. *the arc f fulfills properties P1–P5 of Proposition 7, and*
5. *the arc f does not intersect $\mathcal{B}$ properly.*

We define $F_0(\mathcal{B}) = \emptyset$ and $F_\infty(\mathcal{B})$ as the set of straight-line segments $\overline{uv}$ with $u, v \in V(\mathcal{B})$ that do not intersect $\mathcal{B}$ properly. Let $\mathcal{D}_C$ denote the subdivision of conv$(\mathcal{B}) \setminus \mathcal{B}$ induced by $F_\alpha(\mathcal{B})$ and $E(\mathcal{B})$, and let $|\mathcal{D}_C|$ denote the total number of vertices, edges, and cells in $\mathcal{D}_C$.

Properties (2) and (3) handle ambiguous cases in which an infinite number of arcs connecting two boundary objects with the same objective value may exist (the details can be found in the full version of this paper). We show that $\mathcal{D}_C$ has polynomial size and is sufficient to construct an optimal solution.

▶ **Lemma 9.** *For an instance $\mathcal{B}$ of problem P_F^α with $|V(\mathcal{B})| = n$, the subdivision $\mathcal{D}_C$ has size $|\mathcal{D}_C| \in \mathcal{O}(n^4)$ and every optimal solution for $P_{\mathcal{D}_C}^\alpha$ is an optimal solution for P_F^α.*

Proof sketch. Given a pair b_1, b_2 of boundary objects, we consider the possible center locations for an arc $f = C_\alpha^{uv} \in F_\alpha(\mathcal{B})$ with $u \in b_1$ and $v \in b_2$. Simple geometric arguments show that these center locations are intersections of line segments, circles, and circular arcs. After resolving ambiguous center locations (such as coinciding line segments and circles), each pair of boundary objects induces at most two arcs in $F_\alpha(\mathcal{B})$. Because the number of boundary objects is in $\mathcal{O}(n)$, it follows that $|F_\alpha(\mathcal{B})| \in \mathcal{O}(n^2)$. Each pair of arcs in $F_\alpha(\mathcal{B}) \cup E(\mathcal{B})$ intersects properly at most a constant number of times, so the number of vertices, edges and cells in $\mathcal{D}_C$ is in $\mathcal{O}(n^4)$. By construction, every boundary piece in $\mathcal{S}$ is a path in $\mathcal{D}_C$. Hence, $\mathcal{S}$ is a solution for $P_{\mathcal{D}_C}^\alpha$. Every optimal solution for $P_{\mathcal{D}_C}^\alpha$ is also a solution for P_{F}^α. To be optimal for $P_{\mathcal{D}_C}^\alpha$, it must have the same objective value as $\mathcal{S}$, so it is also optimal for P_{F}^α. ◀

Note that although there may be two arcs connecting the same pair b_1, b_2 of boundary objects, they are uniquely defined by their direction because one of the arcs starts at b_1 and the other one at b_2. Thus, for input edges $e, h \in E(\mathcal{B})$, we can write C_r^{eh} to denote the circular arc C_r^{uv} that satisfies P4 in Proposition 7, starts at u on e and ends at v on h.

By leveraging planar multi-source-multi-sink min-cut algorithms [8, 19] and a graph transformation adapted from [2], we show in Appendix B that for any subdivision $\mathcal{D}$, the problem $P_{\mathcal{D}}^\alpha$ where we only consider free pieces that are paths in $\mathcal{D}$ can be solved in quasilinear-time. Applying these results to $P_{\mathcal{D}_C}^\alpha$ yields a polynomial-time algorithm for P_{F}^α.

▶ **Theorem 10.** *For an instance $\mathcal{B}$ of problem P_F^α with $|V(\mathcal{B})| = n$, an optimal solution can be computed in $\tilde{\mathcal{O}}(|\mathcal{D}_C|) = \mathcal{O}\left(|\mathcal{D}_C| \frac{\log^3 |\mathcal{D}_C|}{\log^2 \log |\mathcal{D}_C|}\right) \subseteq \mathcal{O}\left(n^4 \frac{\log^3 n}{\log^2 \log n}\right)$ time.*

Finally, we observe that optimal solutions are nested with respect to changing α as well as adding polygons, properties we will crucially exploit to obtain a practical algorithm.

▶ **Proposition 11** (α-nestedness). *For an instance $\mathcal{B}$ of P_F^α and two values $0 \leq \alpha < \alpha'$, let $\mathcal{S}$ and $\mathcal{S}'$ be the respective optimal solutions. Then $\mathcal{S} \subseteq \mathcal{S}'$.*

▶ **Proposition 12** (subset-nestedness). *Let $\mathcal{B}$ and $\mathcal{B}'$ be two sets of input polygons, such that $\mathcal{B}' \subseteq \mathcal{B}$. For a given parameter α, let $\mathcal{S}_{\mathcal{B}'}$ be a solution for $\mathcal{B}'$ minimizing g_α. Then there exists a solution $\mathcal{S}_{\mathcal{B}}$ minimizing g_α for $\mathcal{B}$, such that $\mathcal{S}_{\mathcal{B}'} \subseteq \mathcal{S}_{\mathcal{B}}$.*

A Local Preprocessing Algorithm. The main contributor to the runtime of the algorithm is the size of the subdivision $\mathcal{D}_C$, which is quadratic in the size of the arc set $F_\alpha(\mathcal{B})$. While $|F_\alpha(\mathcal{B})|$ is manageable for small α, it grows significantly with increasing α, leading to subdivisions that become intractable for large α. This is because more pairs of potential endpoints satisfy the distance bound of 2α imposed by property P1. This effect is partially offset by α-nestedness: as α increases, regions in an optimal solution grow, rendering more arcs irrelevant since they intersect an optimal region. Although the exact optimal regions are unknown a priori, subset-nestedness allows us to approximate them from below.

Our preprocessing follows a simple idea: iteratively merge nearby polygons that belong to the same region in an optimal solution. Let $\texttt{UPA-Opt}_\alpha(\mathcal{B})$ denote the solution of our algorithm applied to $\mathcal{B}$. First, we replace each polygon $B \in \mathcal{B}$ by $\texttt{UPA-Opt}_\alpha(\{B\})$. Then we compute a Delaunay triangulation DT on the centroids of $\mathcal{B}$ and process edges $e \in \text{DT}$ in increasing order of length. For each edge connecting B_i and B_j, we compute $\texttt{UPA-Opt}_\alpha(\{B_i, B_j\})$. If this yields a single region, we replace B_i and B_j by the merged region $\texttt{UPA-Opt}_\alpha(\{B_i, B_j\})$. Due to subset-nestedness (Proposition 12), this merged representative is a subset of the globally optimal solution. Note that, during the preprocessing it may happen that some merged representatives of different polygon clusters overlap properly. We show in the full version of this paper that our algorithm can also handle this case without increasing the runtime.

In theory, the preprocessing may increase the runtime by a factor of $\mathcal{O}(n)$, but in practice the subproblems are solved quickly. Further details are given in the full version of this paper.

5 Solving the Parametric Problem

We turn to the parametric variant of the problem P_F^α, in which the objective is to find an optimal solution for every $\alpha \in [0, \infty]$. For the subdivision-restricted variant, α-nestedness implies that there are $\mathcal{O}(n)$ optimal solutions [31], but this implication does not hold for P_F^α. Because our definition of solutions is geometric, even an infinitesimal change in α alters the radii of the circular arcs and, in turn, changes the optimal solution. Hence, the number of optimal geometric solutions is infinite. To remedy this, we derive a solution description that is geometry-independent and is strictly given combinatorially with respect to the polygon edges and vertices. We show that an optimal parametric solution contains $\mathcal{O}(n^2)$ combinatorial solutions, and we present a bisection scheme to compute such a solution. We only give intuitive definitions and omit technical details and proofs; these are given in the full version of this paper.

The Parametric Problem. We already established that for fixed α, every free arc $f \in F_\alpha(\mathcal{B})$ is defined by the ordered pair x, y of boundary objects that are connected by it. Hence, for two objects (vertices or edges) x, y and two values α, α', the arcs C_α^{xy} and $C_{\alpha'}^{xy}$ are different

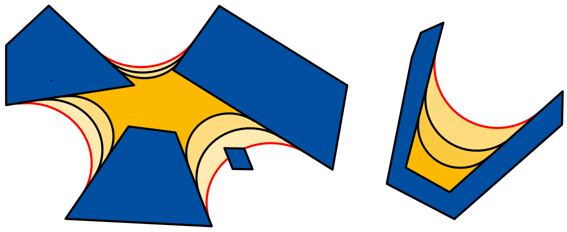


Figure 5 Three realizations of the same combinatorial solution for different α. The arcs move with α but connect the same edge pairs. The red realization is invalid because it intersects a polygon.

geometric realizations of the same combinatorial arc C^{xy}. We define a *combinatorial boundary* as a cyclic sequence of combinatorial arcs and (combinatorial) pieces of the input boundary. Its realization for α is the closed piecewise curve formed by the respective realizations of the combinatorial arcs and the pieces of the input boundary connecting them. A *combinatorial region* R is described by a set of combinatorial boundaries, one of which is designated as the outer boundary and the rest as inner boundaries. We call R *valid* for α if its realization $R[\alpha]$ forms a region. A *combinatorial solution* $\mathcal{K}$ is a set of combinatorial regions; see Figure 5. It is *valid* for α if its realization $\mathcal{K}[\alpha]$ is a valid geometric solution for P_{F}^{α}. Notably, α-nestedness also holds for realizations of the same combinatorial region.

▶ **Corollary 13.** *For a combinatorial region R and two values $\alpha_1 < \alpha_2$ such that R is valid, we have $R[\alpha_1] \subseteq R[\alpha_2]$.*

With this, we can show the following, which is crucial for solving the parametric problem.

▶ **Proposition 14.** *For a combinatorial solution $\mathcal{K}$, the set $\mathcal{I}(\mathcal{K}) \subseteq (0,\infty)$ of values for which $\mathcal{K}$ is valid is a single open interval.*

Using combinatorial solutions, we define the parametric problem.

▶ **Problem 2** (The Parametric Problem). *Let $\mathcal{B}$ be a set of interior-disjoint polygons in the plane. A solution to the parametric unrestricted polygon aggregation problem P_F is a set $\mathfrak{K}$ of combinatorial solutions such that for every $\alpha \in [0,\infty]$, there exists a combinatorial solution $\mathcal{K} \in \mathfrak{K}$ such that $\mathcal{K}[\alpha]$ is an optimal solution for P_F^{α}.*

By exploiting α-nestedness (see Proposition 11), we show that the set $\mathfrak{K}$ has polynomial size.

▶ **Proposition 15.** *Given a set $\mathcal{B}$ of interior-disjoint polygons in the plane with $V(\mathcal{B}) = n$, every solution $\mathfrak{K}$ for P_F has size $\mathcal{O}(n^2)$.*

Proof sketch. Iterate over the solutions in $\mathfrak{K}$ in increasing order of the α values for which they are optimal. We can show that each new solution either introduces a free arc for the first time or merges two regions. Taken together, both events happen at most $\mathcal{O}(n^2)$ times. ◀

To solve the parametric problem, we consider the objective values of geometric and combinatorial solutions as functions parameterized in α. For a geometric solution $\mathcal{S}$, the parametric objective function given by $g(\mathcal{S})(\alpha) := g_\alpha(\mathcal{S})$ is linear and its slope is the perimeter $P(\mathcal{S})$. For a combinatorial solution $\mathcal{K}$, we define

$$g(\mathcal{K})(\alpha) := \begin{cases} g_\alpha(\mathcal{K}[\alpha]) & \text{if } \alpha \in \mathcal{I}(\mathcal{K}), \\ \infty & \text{otherwise.} \end{cases}$$

It is easy to see that for all $\alpha \in \mathcal{I}(\mathcal{K})$, the slope of $\mathcal{K}$ is monotonically decreasing. In particular, $g(\mathcal{K}[\alpha])$ is the tangent of $g(\mathcal{K})$ at α. The objective of P_F can be stated as finding the lower envelope of the functions $g(\mathcal{K})$ for all possible combinatorial solutions $\mathcal{K}$.

For the subdivision-restricted problem variant, the number of geometric solutions is finite and it can be solved using a *dichotomic scheme* [31]. Given an interval $[\alpha_{\min}, \alpha_{\max}]$ and optimal solutions $\mathcal{S}_L$ at $\alpha_{\min}$ and $\mathcal{S}_U$ at $\alpha_{\max}$, it explores the interval by recursive bisection. At the crossing point α_N of $g(\mathcal{S}_L)$ and $g(\mathcal{S}_U)$, it computes an optimal solution $\mathcal{S}_N$. If $\mathcal{S}_N$ is better than $\mathcal{S}_L$ and $\mathcal{S}_U$ for α_N, the algorithm adds $\mathcal{S}_N$ to the solution set and recurses on the intervals $[\alpha_{\min}, \alpha_N]$ and $[\alpha_N, \alpha_{\max}]$. Otherwise, it terminates with the solution set $\{\mathcal{S}_L, \mathcal{S}_U\}$.

The correctness of the termination condition requires that a function cannot appear on the lower envelope more than once. In our scenario, this is not immediately clear because the objective functions of the combinatorial solutions are not linear and they jump to ∞ outside of the validity interval. However, we show in the full version of this paper that this is indeed the case by using another nesting property that relates realizations of combinatorial solutions.

▶ **Proposition 16.** *For every combinatorial solution $\mathcal{K}$, the set of values $\alpha \in [0, \infty]$ such that $\mathcal{K}[\alpha]$ is optimal is a single interval.*

Under the assumption that we have an oracle which, given solutions $\mathcal{K}_1$, $\mathcal{K}_2$, returns the exact intersection point of $g(\mathcal{K}_1)$ and $g(\mathcal{K}_2)$, we present a polynomial-time algorithm for P_F in the full version of this paper. In practice, we are satisfied with approximating the intersection point, as numerical precision is limited anyway.

Algorithm 1 Approximative dichotomic scheme for P_F.

$\mathcal{K}_L \leftarrow$ Combinatorial solution optimal for $\lambda = 0$, Report $\mathcal{K}_L$
$\mathcal{K}_U \leftarrow$ Combinatorial solution optimal for $\lambda = 1$, `Recurse`$(\mathcal{K}_L, 0, \mathcal{K}_U, 1)$, Report $\mathcal{K}_U$
Procedure `Recurse`$(\mathcal{K}_L, \lambda_{min}, \mathcal{K}_U, \lambda_{max})$
    if $\mathcal{K}_L = \mathcal{K}_U$ **or** $f(\mathcal{K}_L[\lambda_{\min}]) = f(\mathcal{K}_U[\lambda_{\max}])$ **then return**
    $\lambda_N \leftarrow (\lambda_{\max} - \lambda_{\min})/2$
    $\mathcal{K}_N \leftarrow$ Combinatorial solution optimal for λ_N
    `// Note` $f(\mathcal{K})(\lambda) = \infty$ `if not valid`
    $\mathrm{dom}_L \leftarrow f(\mathcal{K}_N)(\lambda_N) < \min(f(\mathcal{K}_L)(\lambda_N), f(\mathcal{K}_L[\lambda_{\min}])(\lambda_N)/(1+\varepsilon))$
    $\mathrm{dom}_U \leftarrow f(\mathcal{K}_N)(\lambda_N) < \min(f(\mathcal{K}_U)(\lambda_N), f(\mathcal{K}_U[\lambda_{\max}])(\lambda_N)/(1+\varepsilon))$
    if dom_L **then** `Recurse`$(\mathcal{K}_L, \lambda_{\min}, \mathcal{K}_N, \lambda_N)$
    if dom_L **and** dom_U **then** Report $\mathcal{K}_N$
    if dom_U **then** `Recurse`$(\mathcal{K}_N, \lambda_N, \mathcal{K}_U, \lambda_{\max})$

Approximation Scheme. In this section, we define the notion of an approximate solution to the parametric problem, and give an approximation scheme for computing such a solution.

▶ **Definition 17.** *Let $\varepsilon > 0$. A $(1+\varepsilon)$-approximate solution is a set $\mathfrak{K}$ of combinatorial solutions such that for every $\alpha \in [0, \infty]$, there exists $\mathcal{K} \in \mathfrak{K}$ and a value α' such that $g_\alpha(\mathcal{K}[\alpha']) \leq (1+\varepsilon) \cdot OPT(\alpha)$, where $OPT(\alpha)$ is the value of an optimal solution to P_F^α.*

To avoid dealing with the unbounded α-domain, we switch to the equivalent objective function $f_\lambda(\mathcal{S}) = (1-\lambda) \cdot A(\mathcal{S}) + \lambda \cdot P(\mathcal{S})$ with $\lambda \in [0, 1]$. In contrast to the standard dichotomic scheme, our approach (given in Algorithm 1) does not compute an intersection point but instead performs a binary bisection with $\lambda_N = (\lambda_{\max} + \lambda_{\min})/2$. Consequently, $\mathcal{K}_L$ and $\mathcal{K}_U$ generally do not have the same objective value at λ_N, since it is not the crossing point. Therefore, the algorithm performs two independent comparisons and may recurse on only one side. Since binary bisection does not guarantee reaching the exact crossing point, we introduce an approximation test: After computing the optimal solution $\mathcal{K}_N$ at λ_N, we compare it to the evaluated geometric solutions $\mathcal{K}_L[\lambda_{\min}]$ and $\mathcal{K}_U[\lambda_{\max}]$ at λ_N. If one of them is within a factor of $(1+\varepsilon)$ of $\mathcal{K}_N$, we do not recurse on that side.

We can show that the proposed algorithm is a fully polynomial-time approximation scheme (FPTAS). In addition to n and $1/\varepsilon$, the runtime also depends on the smoothness of the function $f(\mathcal{K})$, which in turn depends on geometric properties of the input. To bound this, let $\delta_{\text{seg}}(\mathcal{B})$ be the minimum distance between any vertex v and any polygon edge not incident to v. Intuitively, this is a lower bound on the thickness of each polygon and the minimum distance between any pair of polygons. The *segment spread* $\Phi := \frac{\operatorname{diam}(V(\mathcal{B}))}{\delta_{\text{seg}}(\mathcal{B})}$ represents the ratio between the maximum and minimum distance/thickness in the input.

▶ **Theorem 18.** *Algorithm 1 computes a $(1+\varepsilon)$-approximate solution to P_F in time*

$$\mathcal{O}\left(n^6 \frac{\log^3 n}{\log^2 \log n}(\log n + \log \Phi + \log \tfrac{1}{\varepsilon})\right).$$

Exploiting α-Nestedness. Let $\mathcal{S}_L := \mathcal{K}_L[\alpha_{\min}]$, $\mathcal{S}_N := \mathcal{K}_N[\alpha_N]$ and $\mathcal{S}_U := \mathcal{K}_U[\alpha_{\max}]$. Due to the α-nestedness of optimal solutions, we have $\mathcal{S}_L \subseteq \mathcal{S}_N \subseteq \mathcal{S}_U$. For the subdivision-restricted case, this can be exploited by contracting the problem instance that is solved for α_N [6]. This optimization does not carry over to P^α_{F}, but we can still exploit α-nestedness:

1. Since $\mathcal{S}_L \subseteq \mathcal{S}_N$ and `UPA-Opt` can handle circular polygons as inputs, we can run the optimization at α_N with $\mathcal{S}_L$ as the input instance instead of $\mathcal{B}$. Particularly for larger α values, this can significantly reduce the complexity of the input.
2. Let $\mathcal{B}(S)$ denote the polygons contained in a region S. It follows from $\mathcal{S}_N \subseteq \mathcal{S}_U$ that for every region $S \in \mathcal{S}_N$, there exists a region $S' \in \mathcal{S}_U$ with $\mathcal{B}(S) \subseteq \mathcal{B}(S')$. Therefore, $\mathcal{S}_N$ can be computed by solving the subproblem with input $\mathcal{B}(S')$ for each region $S' \in \mathcal{S}_U$ independently. This allows us to discard arcs between different regions of $\mathcal{S}_U$.
3. Now, consider an arbitrary but fixed region $S_L \in \mathcal{S}_L$, and let $S_U \in \mathcal{S}_U$ be the region such that $S_L \subseteq S_U$. If S_L and S_U contain the same polygons, then we can find the corresponding region $S_N \in \mathcal{S}_N$ by solving a simplified sub-instance for every free arc $f = C^{xy}$ in S_U (see Figure 6). Let $s = \langle x = v_0, \ldots, v_k = y \rangle$ be the sequence of combinatorial vertices between u and v on ∂S_L. The boundary segment between x and y in S_N must lie fully within the region enclosed by f and s, and the only vertices within this region are those in s. Thus, it suffices to consider arcs between vertices in s for this sub-instance.

6 Experiments

We implemented our algorithms in C++ and compiled the code using GCC 13.3.0 with the `-O3` optimization flag. For geometric computations, we used the CGAL library [34]. Minimum cuts were computed with the Boykov-Kolmogorov max-flow algorithm [9], using the implementation from the Boost library [13]. All experiments were run sequentially on a single core of an AMD Threadripper 3970X machine with 128 GB of DDR4-RAM. As input instances, we used building footprints of German towns and cities extracted from OpenStreetMap [29] (cf. Table 3) with input sizes ranging from 1 800 to 400 000 input vertices. All distances are measured in decimeters (10 centimeters), which means that the arcs in an optimal solution for P^α_{F} have a radius of α decimeters.

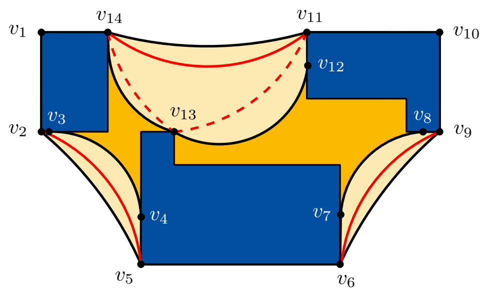


Figure 6 Input polygons are shown in blue, $\mathcal{S}_L$ in dark yellow and $\mathcal{S}_U$ in light yellow. By exploiting α-nestedness, we can isolate the vertex sequences $\langle v_2, v_3, v_4, v_5\rangle$, $\langle v_6, v_7, v_8, v_9\rangle$, and $\langle v_{11}, v_{12}, \ldots, v_{14}\rangle$ of $\mathcal{S}_L$ as sub-instances, where arcs in $\mathcal{S}_N$ may only connect vertices from the respective sequence.

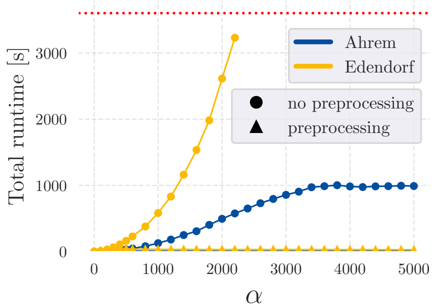


Figure 7 Runtime comparison with and without preprocessing. Runs that exceeded the time limit of 1 hour are omitted.

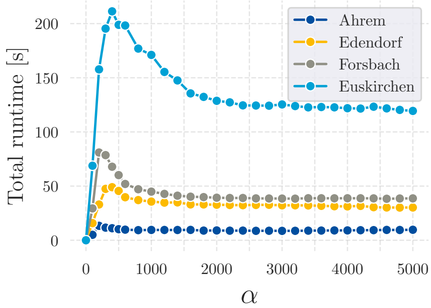


Figure 8 Runtime by α value, with preprocessing enabled.

6.1 Fixed Parameter Problem

Figure 7 shows the impact of the local preprocessing on the instances Ahrem ($|V(\mathcal{B})| = 3\,683$) and Edendorf ($|V(\mathcal{B})| = 10\,536$). Without preprocessing, the runtime grows quickly with α. For Ahrem, it plateaus around $\alpha = 4000$. This is because the set $F_\alpha(\mathcal{B})$ of free arcs stops growing, since larger endpoint distances make circular arcs likely to intersect other input polygons. However, already for Edendorf, the one-hour time limit is reached before this plateau. With preprocessing, the runtime remains in the range of seconds for all α values.

Figure 8 shows only the runtime with preprocessing enabled, now including two additional medium-sized instances. Furthermore, Figure 9 plots the runtime on all instances for selected α values. The runtime scales linearly with the instance size and, for a fixed instance, varies by less than a factor of 4 depending on α. For all instances, the runtime peaks at some $\alpha \in [200, 500]$ and then declines, reaching a plateau around $\alpha = 3000$; see Figure 10 for example aggregations. Next, we highlight some noteworthy observations; for detailed measurements and an in-depth discussion, see Appendix C.1. Overall, the preprocessing drastically reduces the number of generated arcs, especially for large α, where the reduction reaches orders of magnitude. On all instances, the size of the subdivision for an `UPA-Opt` call during preprocessing lies between 4 and 7, so most calls are nearly trivial to solve. Finally, we investigate why the runtime peaks for medium α values. For $\alpha \geq 3000$, most merges succeed, and the preprocessing yields a solution that is close to or identical to the final optimal solution. For $\alpha \leq 200$, only few arcs are generated due to the distance threshold;

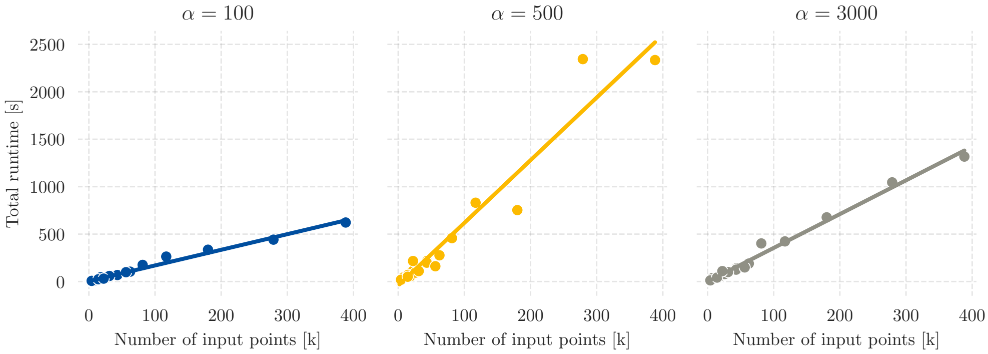


Figure 9 Runtime with preprocessing on all instances for selected α-values.

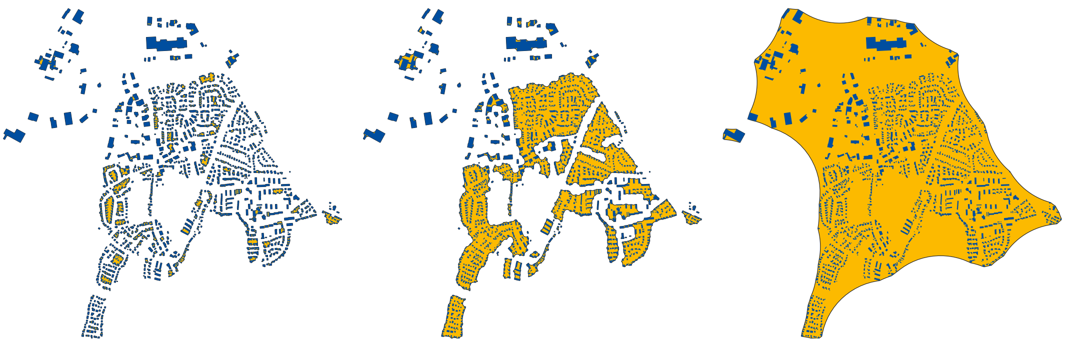


Figure 10 Instance Edendorf. Left: For $\alpha = 100$, few polygons are aggregated. Middle: For $\alpha = 260$, there are multiple clusters. Right: For $\alpha = 3000$, we have one large cluster and one outlier.

however, the effectiveness of the preprocessing also decreases, as only few merges succeed. In the range $\alpha \in [200, 500]$, solutions are the most volatile: some polygons already merge into large representatives, but still many merges can fail, resulting in larger subdivisions during the preprocessing, and in particular for the final `UPA-Opt` call after the preprocessing.

6.2 Parametric Problem

Figure 11 shows the runtime and number of solutions of the engineered approximation scheme on all instances with fewer than 100 000 vertices for $\varepsilon \in \{10^{-2}, 10^{-4}, 10^{-6}\}$. Despite the theoretical $\mathcal{O}(n^2)$ upper bound on the number of solutions, the algorithm produces significantly fewer than n solutions, and the solution size scales linearly for fixed ε. Independent of ε, the runtime also scales linearly with input size, albeit with large constants. Figure 12 show that the number of `Recurse` calls per reported solution is at most 4 on all instances, indicating that the approximation of the crossing point converges quickly in practice and that the runtime is essentially proportional to the number of solutions.

Table 1 reports the performance of the approximation scheme with partially disabled α-nestedness exploits on the smaller Edendorf and Andernach datasets with $\varepsilon = 10^{-6}$. Exploiting α-nestedness from above and below individually reduces the runtime by a factor of at most 2, while combining both and adding the decomposition-based optimization yields speedups of 5.9 for Edendorf and 7.1 on Andernach.

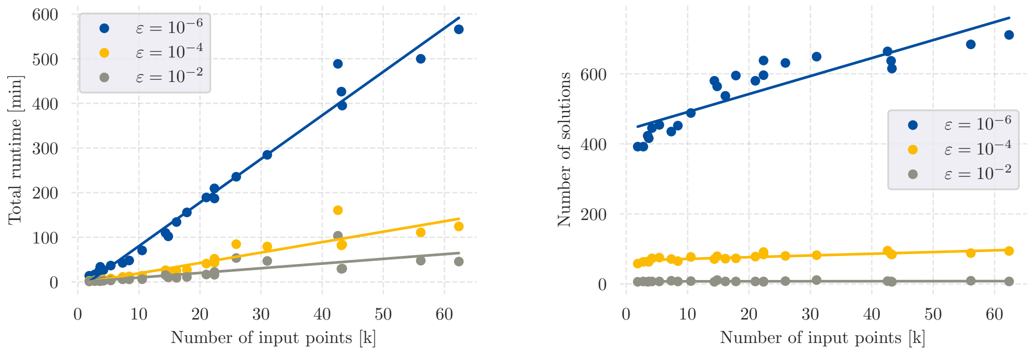


Figure 11 Runtime and # solutions of the approximation scheme on medium-sized instances.

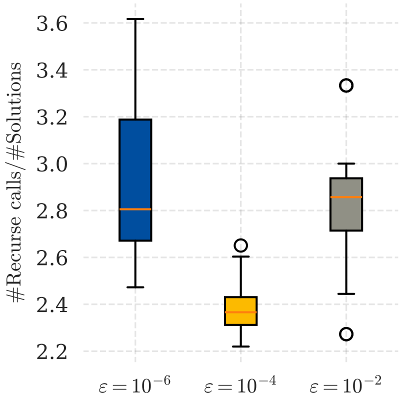


Figure 12 Number of recursive calls per reported solution across all instances, grouped by ε values.

Table 1 Performance of the approximation scheme for $\varepsilon = 10^{-6}$. Subset and Superset indicate whether we exploit α-nestedness from below and above, respectively.

Dataset	Subset	Superset	Runtime [min]
Edendorf	○	○	398.48
	○	•	323.02
	•	○	217.12
	•	•	67.14
Andernach	○	○	1246.47
	○	•	1005.51
	•	○	890.64
	•	•	176.49

7 Conclusion

We showed that the parametric polygon aggregation problem can still be solved in polynomial time if the subdivision-based restriction [31] is lifted. With careful engineering that exploits structural properties, our approach also achieves linear runtime on real-world data. While the constant factors are higher, it can still process city-sized instances (see Figure 13) within a timeframe of a few hours. Investing the extra computation time can be beneficial for at least two reasons. Firstly, situations arise in practice where the subdivision-restricted approach fails to find a sufficiently representative solution because the subdivision is “in the way”, especially at small scales (cf. Appendix C.2). More importantly, removing the restriction guarantees subset-nestedness, which is a structural advantage that ensures consistency across solutions, not only across parameter values but also under the addition or filtering of input polygons. This makes the approach particularly well-suited for applications in map generalization and urban analytics where consistency is crucial, for example, when filtering by temporal attributes (e.g., construction dates) or by building types. Here, it ensures that resulting clusterings remain consistent with the entire input, in the sense that more specific subsets refine, rather than contradict, the overall clustering. Beyond this, our preprocessing scheme may be of independent interest for other fencing problems that exhibit subset-nestedness.

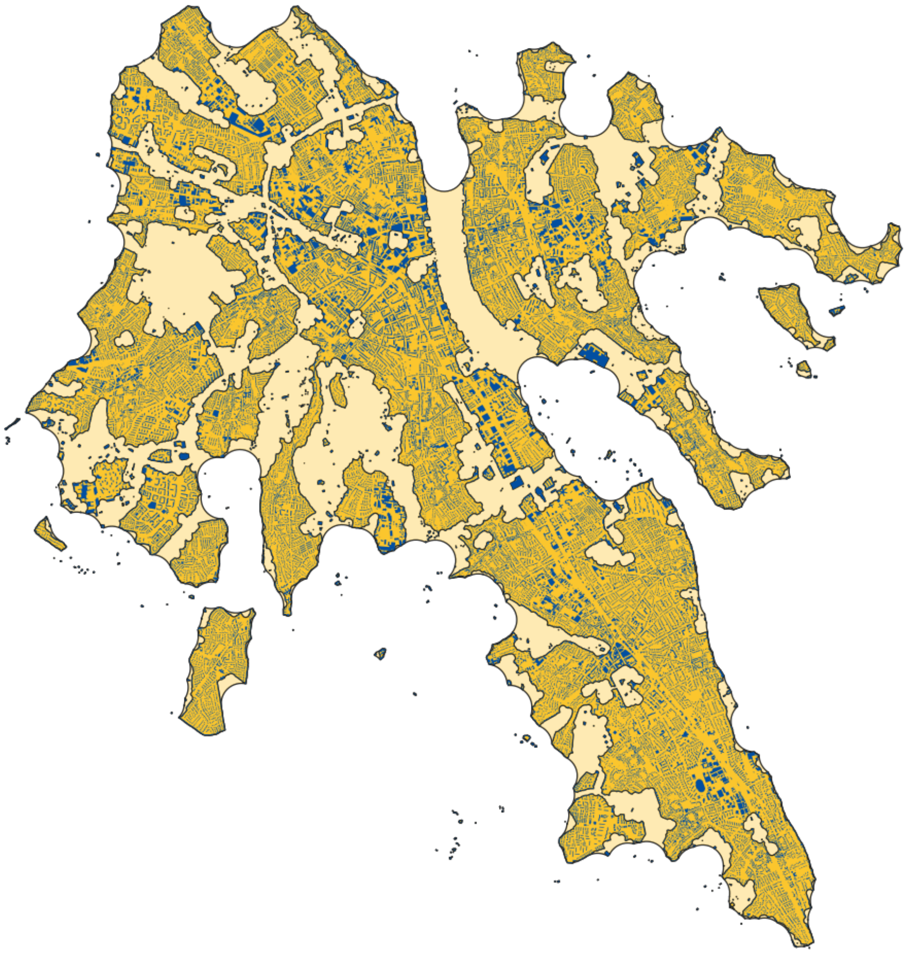


Figure 13 Aggregations of the city of Bonn, Germany, for $\alpha = 500$ and $\alpha = 3000$.

References

1 Mikkel Abrahamsen, Anna Adamaszek, Karl Bringmann, Vincent Cohen-Addad, Mehran Mehr, Eva Rotenberg, Alan Roytman, and Mikkel Thorup. Fast fencing. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018*, pages 564–573. Association for Computing Machinery, 2018. doi:10.1145/3188745.3188878.

2 Mikkel Abrahamsen, Panos Giannopoulos, Maarten Löffler, and Günter Rote. Geometric multicut: Shortest fences for separating groups of objects in the plane. *Discrete & Computational Geometry*, 64(3):575–607, 2020. doi:10.1007/S00454-020-00232-W.

3 Karl Heinrich Anders and Monika Sester. Parameter-free cluster detection in spatial databases and its application to typification. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 33:75–82, 2000. URL: https://www.isprs.org/proceedings/xxxiii/congress/part4/75_XXXIII-part4.pdf.

4 Esther M. Arkin, Samir Khuller, and Joseph S. B. Mitchell. Geometric knapsack problems. In *Algorithms and Data Structures, 2nd Workshop WADS '91*, volume 519 of *Lecture Notes in Computer Science*, pages 165–176. Springer, 1991. doi:10.1007/BFB0028259.

5 Daniel Arribas-Bel, M.-À. Garcia-López, and Elisabet Viladecans-Marsal. Building(s and) cities: Delineating urban areas with a machine learning algorithm. *Journal of Urban Economics*, 125:103217, 2021. Delineation of Urban Areas. doi:10.1016/j.jue.2019.103217.

6 Arne Beines, Michael Kaibel, Philip Mayer, Petra Mutzel, and Jonas Sauer. A simpler approach for monotone parametric minimum cut: Finding the breakpoints in order. In *Proceedings of the 27th Workshop on Algorithm Engineering and Experiments (ALENEX'25)*, pages 29–41. Society for Industrial and Applied Mathematics (SIAM), 2025. doi:10.1137/1.9781611978339.3.

7 Annika Bonerath, Benjamin Niedermann, and Jan-Henrik Haunert. Retrieving α-shapes and schematic polygonal approximations for sets of points within queried temporal ranges. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, SIGSPATIAL 2019*, pages 249–258. Association for Computing Machinery, 2019. doi:10.1145/3347146.3359087.

8 Glencora Borradaile, Philip N. Klein, Shay Mozes, Yahav Nussbaum, and Christian Wulff-Nilsen. Multiple-source multiple-sink maximum flow in directed planar graphs in near-linear time. *SIAM Journal on Computing*, 46(4):1280–1303, 2017. doi:10.1137/15M1042929.

9 Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, 2004. doi:10.1109/TPAMI.2004.60.

10 Kevin Buchin, Wouter Meulemans, André Van Renssen, and Bettina Speckmann. Area-preserving simplification and schematization of polygonal subdivisions. *ACM Transactions on Spatial Algorithms and Systems*, 2(1), 2016. doi:10.1145/2818373.

11 Kevin Buchin, Wouter Meulemans, and Bettina Speckmann. A new method for subdivision simplification with applications to urban-area generalization. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '11, pages 261–270. Association for Computing Machinery, 2011. doi:10.1145/2093973.2094009.

12 Vasilis Capoyleas, Günter Rote, and Gerhard J. Woeginger. Geometric clusterings. *J. Algorithms*, 12(2):341–356, 1991. doi:10.1016/0196-6774(91)90007-L.

13 Boost Contributors. *Boost C++ Libraries*. Boost.org, 2024. Version 1.74. URL: https://www.boost.org/.

14 Jonathan Damen, Marc van Kreveld, and Bert Spaan. High quality building generalization by extending the morphological operators. In *Proceedings of the 11th ICA Workshop on Generalisation and Multiple Representation*, pages 1–12, 2008. URL: https://kartographie.geo.tu-dresden.de/downloads/ica-gen/workshop2008/04_Damen_et_al.pdf.

15 Marie-Pierre de Bellefon, Pierre-Philippe Combes, Gilles Duranton, Laurent Gobillon, and Clément Gorin. Delineating urban areas using building density. *Journal of Urban Economics*, 125:103226, 2021. doi:10.1016/j.jue.2019.103226.

16 Matt Duckham, Lars Kulik, Mike Worboys, and Antony Galton. Efficient generation of simple polygons for characterizing the shape of a set of points in the plane. *Pattern Recognition*, 41(10):3224–3236, 2008. doi:10.1016/j.patcog.2008.03.023.

17 Herbert Edelsbrunner and Ernst P. Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13(1):43–72, 1994. doi:10.1145/174462.156635.

18 Stefan Funke and Sabine Storandt. Smooth building footprint aggregation with alpha shapes. In *Web and Wireless Geographical Information Systems: 21st International Symposium, W2GIS 2024, Proceedings*, number 14673 in Lecture Notes in Computer Science, pages 117–129. Springer, 2024. doi:10.1007/978-3-031-60796-7_9.

19 Pawel Gawrychowski and Adam Karczmarz. Improved bounds for shortest paths in dense distance graphs. In *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018*, volume 107 of *LIPIcs*, pages 61:1–61:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.ICALP.2018.61.

20 Oliver Harig, Robert Hecht, Dirk Burghardt, and Gotthard Meinel. Automatic delineation of urban growth boundaries based on topographic data using Germany as a case study. *ISPRS International Journal of Geo-Information*, 10(5), 2021. doi:10.3390/ijgi10050353.

21 Lars Harrie, Hanna Stigmar, and Milan Djordjevic. Analytical estimation of map readability. *ISPRS International Journal of Geo-Information*, 4(2):418–446, 2015. doi:10.3390/ijgi4020418.
22 Jan-Henrik Haunert and Alexander Wolff. Optimal and topologically safe simplification of building footprints. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '10, pages 192–201. Association for Computing Machinery, 2010. doi:10.1145/1869790.1869819.
23 Adolf Hurwitz. Sur le probleme des isopérimètres. *Comptes rendus de l'Académie des Sciences*, 132:401–403, 1901.
24 Safe Software Inc. *FME AreaAmalgamator Transformer*, 2025. Accessed: 2025-03-24. URL: https://docs.safe.com/fme/html/FME-Form-Documentation/FME-Transformers/Transformers/areaamalgamator.htm.
25 Warren C. Jochem, Douglas R. Leasure, Oliver Pannell, Heather R. Chamberlain, Patricia Jones, and Andrew J. Tatem. Classifying settlement types from multi-scale spatial patterns of building footprints. *Environment and Planning B: Urban Analytics and City Science*, 48(5):1161–1179, 2021. doi:10.1177/2399808320921208.
26 Alan M. MacEachren. Map complexity: Comparison and measurement. *The American Cartographer*, 9(1):31–46, 1982. doi:10.1559/152304082783948286.
27 Robert B. McMaster and K. Stuart Shea. *Generalization in Digital Cartography*. Association of American Geographers, 1992.
28 Adriano J. C. Moreira and Maribel Yasmina Santos. Concave hull: A k-nearest neighbours approach for the computation of the region occupied by a set of points. In *GRAPP 2007, Proceedings of the Second International Conference on Computer Graphics Theory and Applications*, pages 61–68. INSTICC – Institute for Systems and Technologies of Information, Control and Communication, 2007. doi:10.5220/0002080800610068.
29 OpenStreetMap Contributors. OpenStreetMap. https://www.openstreetmap.org, 2004–*present*.
30 Dongliang Peng and Guillaume Touya. Continuously generalizing buildings to built-up areas by aggregating and growing. In *Proceedings of the 3rd ACM SIGSPATIAL Workshop on Smart Cities and Urban Analytics*, UrbanGIS'17. Association for Computing Machinery, 2017. doi:10.1145/3152178.3152188.
31 Peter Rottmann, Anne Driemel, Herman Haverkort, Heiko Röglin, and Jan-Henrik Haunert. Bicriteria shapes: Hierarchical grouping and aggregation of polygons with an efficient graph-cut approach. *ACM Transactions on Spatial Algorithms and Systems*, 11(1), 2025. doi:10.1145/3705001.
32 Jacob Steiner. Einfache Beweise der isoperimetrischen Hauptsätze. *Journal für die reine und angewandte Mathematik*, 18:281–296, 1838. URL: http://eudml.org/doc/147053.
33 H. Taubenböck, H. Debray, C. Qiu, M. Schmitt, Y. Wang, and X.X. Zhu. Seven city types representing morphologic configurations of cities across the globe. *Cities*, 105:102814, 2020. doi:10.1016/j.cities.2020.102814.
34 The CGAL Project. *CGAL User and Reference Manual*, 5.6 edition, 2023. URL: https://doc.cgal.org/5.6/Manual/packages.html.
35 Guillaume Touya and Marion Dumont. Progressive Block Graying and Landmarks Enhancing as Intermediate Representations between Buildings and Urban Areas. In *Proceedings of 20th ICA Workshop on Generalisation and Multiple Representation*, July 2017. URL: https://hal.science/hal-02097486.

A A Remark on α-Shapes

Unlike our approach, α-shapes tend to connect nearby point sets with long, narrow bridges, which are undesirable in the context of map generalization [7, 31]. Consider the example in Figure 14. Here, a narrow bridge appears as α increases, causing both the area and the perimeter of the solution to increase. Thus, in terms of our objective function g_α, the solution becomes worse in both relevant criteria.

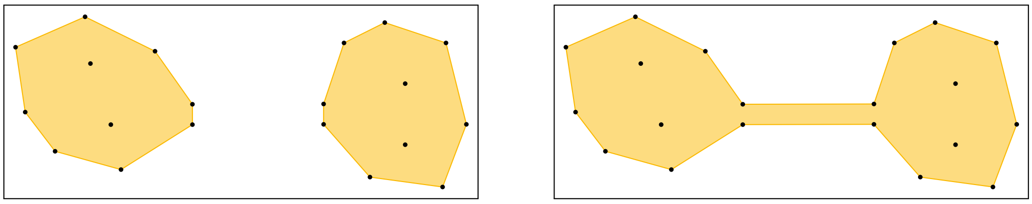
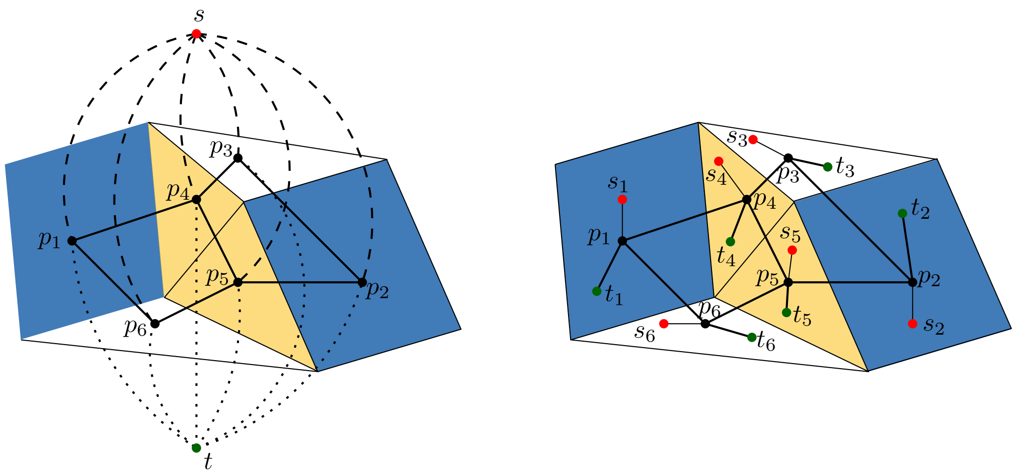


Figure 14 Given the depicted point set, the α-shape for some value of α is shown on the left. As α increases, a bridge forms (right), increasing the perimeter.

B The Minimum-Cut Approach for Polygon Aggregation



(a) Single-source-single-sink graph G_{st}. **(b)** Planar multi-sink-multi-source graph G_{M}.

Figure 15 Illustrations of the two transformations of $P_{\mathcal{D}}^{\alpha}$ into minimum cut problems on graphs given in this section.

Rottmann et al. [31] give a transformation of the subdivision-restricted polygon aggregation problem $P_{\mathcal{D}}^{\alpha}$ into a minimum (s,t)-cut problem on an almost-planar graph. In this section, we give the details of this transformation. Additionally, we give an alternative transformation, analogous to [2], that yields a planar graph with multiple source and sink vertices. On this graph, faster algorithms for planar multi-source-multi-sink minimum cut can be used.

Original Transformation. Let $\mathcal{D}$ be a planar subdivision of $\operatorname{conv}(\mathcal{B}) \setminus \mathcal{B}$. Then $\mathcal{A} = \mathcal{D} \cup \mathcal{B}$ is a planar subdivision of $\operatorname{conv}(\mathcal{B})$. Let G denote the geometric dual graph of $\mathcal{A}$. Here, two vertices $p_i, p_j \in V(G)$ are connected by an edge if the corresponding faces $a_i, a_j \in \mathcal{A}$ share at least one boundary edge. In the augmented graph G_{st}, we introduce an additional source vertex s and sink vertex t, as well as edges $\{s, p_i\}$ and $\{p_i, t\}$ for every vertex $p_i \in V(G)$. The graph construction is depicted in Figure 15a. Let $b(a_i)$ denote the length of the boundary shared by a face a_i and the outer face and let $b(a_i, a_j)$ denote the length of the boundary shared by a_i and a_j. We define edge weights $w\colon E(G_{st}) \to \mathbb{R}_{\geq 0}$ as follows:

- For $e = \{p_i, p_j\}$ with $p_i, p_j \in V(G)$, we set $w(e) = \alpha \cdot b(a_i, a_j)$.
- For $e = \{p_i, t\}$ with $p_i \in V(G)$, we set $w(e) = A(p_i) + \alpha \cdot b(a_i)$
- For $e = \{s, p_i\}$ with $p_i \in V(G)$, we set $w(e) = \begin{cases} \infty & \text{if } a_i \in \mathcal{B}, \\ 0 & \text{otherwise.} \end{cases}$

Note that Rottmann et al. [31] use the objective function $f_\lambda(\mathcal{S}) = \lambda A(\mathcal{S}) + (1-\lambda)P(\mathcal{S})$, whereas we use the equivalent function $g_\alpha(\mathcal{S}) = A(\mathcal{S}) + \alpha P(\mathcal{S})$. Accordingly, the edge weights in our construction differ slightly. A solution to the subdivision-restricted polygon aggregation problem $P^\alpha_\mathcal{D}$ is a selection of faces $\mathcal{S} \subseteq \mathcal{A}$ with $\mathcal{B} \subseteq \mathcal{S}$. Let $V(\mathcal{S})$ denote the corresponding set of vertices in G. Rottmann et al. show that $g_\alpha(\mathcal{S})$ is exactly the value of the (s,t)-cut with source component $(\{s\} \cup V(\mathcal{S})$ and sink component $V(G) \setminus V(\mathcal{S}) \cup \{t\}$. Using the fastest known algorithm for minimum (s,t)-cut in general graphs, they obtain the following result.

▶ **Theorem 19** (Rottmann et al. [31]). *For any minimum (s,t)-cut in G_{st}, the set of cells in $\mathcal{A}$ corresponding to the source component is an optimal solution to $P^\alpha_\mathcal{D}$. Hence, $P^\alpha_\mathcal{D}$ can be solved in $O\left(\frac{|\mathcal{A}|^2}{\log|\mathcal{A}|}\right)$ time.*

Planar Transformation. The addition of s and t does not preserve the planarity of G, preventing us from using faster min-cut algorithms for planar graphs. Therefore, we construct an alternate augmented graph G_M in which s and t are replaced with sets $S = \{s_i \mid p_i \in V(G)\}$ and $T = \{t_i \mid p_i \in V(G)\}$, i.e., every vertex $p_i \in V(G)$ is associated with a separate source s_i and sink t_i. The corresponding edges $\{s_i, p_i\}$ and $\{p_i, t_i\}$ are given the same weights as $\{s, p_i\}$ and $\{p_i, t\}$ in G_{st}, respectively. We call a cut in G_M an (S,T)-cut if all vertices in S lie in the source component and all vertices in T in the sink component. By construction, every (s,t)-cut in G_{st} has a corresponding (S,T)-cut in G_M and vice versa. Furthermore, a planar embedding of G can be extended to a planar embedding of G_M: for each vertex p_i, we place the vertices s_i and t_i inside the face a_i (see Figure 15b). Thus, we can apply algorithms for planar multi-source-multi-sink minimum cut. The fastest known algorithm for this problem is due to Borradaile et al. [8] and runs in time $\mathcal{O}\left(|\mathcal{A}|\frac{\log^3|\mathcal{A}|}{\log^2\log|\mathcal{A}|}\right)$ when combined with the data structure by Gawrychowski and Karczmarz [19] for shortest paths in dense distance graphs.

▶ **Theorem 20.** *For any minimum (S,T)-cut in G_M, the set of cells in $\mathcal{A}$ corresponding to the source component is an optimal solution to $P^\alpha_\mathcal{D}$. Hence, $P^\alpha_\mathcal{D}$ can be solved in $\mathcal{O}\left(|\mathcal{A}|\frac{\log^3|\mathcal{A}|}{\log^2\log|\mathcal{A}|}\right)$ time.*

C Data and Additional Experimental Results

In Table 2, we provide an overview of the datasets used in our experimental evaluation. We report characteristics of the datasets that have an influence on the performance of the exact algorithm for the unrestricted problem.

C.1 Additional Experiments for the Fixed Parameter Problem

We consider the two instance Andernach and Euskirchen in more detail. The runtime peak for Andernach is at $\alpha = 260$ and $\alpha = 500$ for Euskirchen. Using the detailed measurements for the two medium-size instances given in Table 3, we now discuss the runtime fluctuation for different values α. At $\alpha = 100$, few arcs exist and the preprocessing performs few successful merges, so the preprocessed instance and the solution contain many polygons. At $\alpha = 3000$, over 90% of merges succeed, and the number of regions after preprocessing is close to the final solution. Additionally, for sub-instances, on average, each `UPA-Opt` call involves only 4–5 cells, making most of them trivial to solve and the final global subdivision is also close to trivial. In the range $\alpha \in [200, 500]$, solutions are the most volatile: some polygons already merge into large representatives, but still many merges can fail, resulting in larger subdivisions during the preprocessing and in particular for the final `UPA-Opt` call after the preprocessing. Hence, different factors can lead to the peak in runtime: For Andernach, it is due to many generated free arcs and failed merges; for Euskirchen, it is mainly due to the size of the final arrangement, though free arcs and average subdivision size also contribute.

Overall, we observe that the preprocessing drastically reduces the number of generated arcs that need to be tested for intersection with $\mathcal{B}$ (for $\alpha = 3000$ this reduction is by more then one order of magnitude). In addition, for larger α, the number of regions after preprocessing is already close to that in the final solution and the preprocessing solution is already very close to optimal.

C.2 Comparison to the Subdivision-Restricted Variant

Figure 16 displays a situation in which the unrestricted and subdivision-restricted approaches produce substantially different solutions for the same value of α. The chosen subdivision is a constrained Delaunay triangulation (as suggested in [31]). As can be seen, the outer limits of settlement areas tend to be sparsely developed, so a decision has to be made whether to include outlier buildings in the main component or not. The subdivision has to incorporate these buildings somehow, which leads to long edges that span across large empty areas. These edges prescribe certain directions in the solution boundary and preclude others. In this case, the boundary arc in the optimal (unconstrained) solution is perpendicular to many edges of the Delaunay triangulation. Hence, the subdivision-based only has the choice between staying closer to the buildings near the boundary (which increases the perimeter) or including the outlier buildings (which vastly increases the area). For aggregations that place a high emphasis on the perimeter, the better choice is to include the outlier buildings, which leads to a drastically different aggregation.

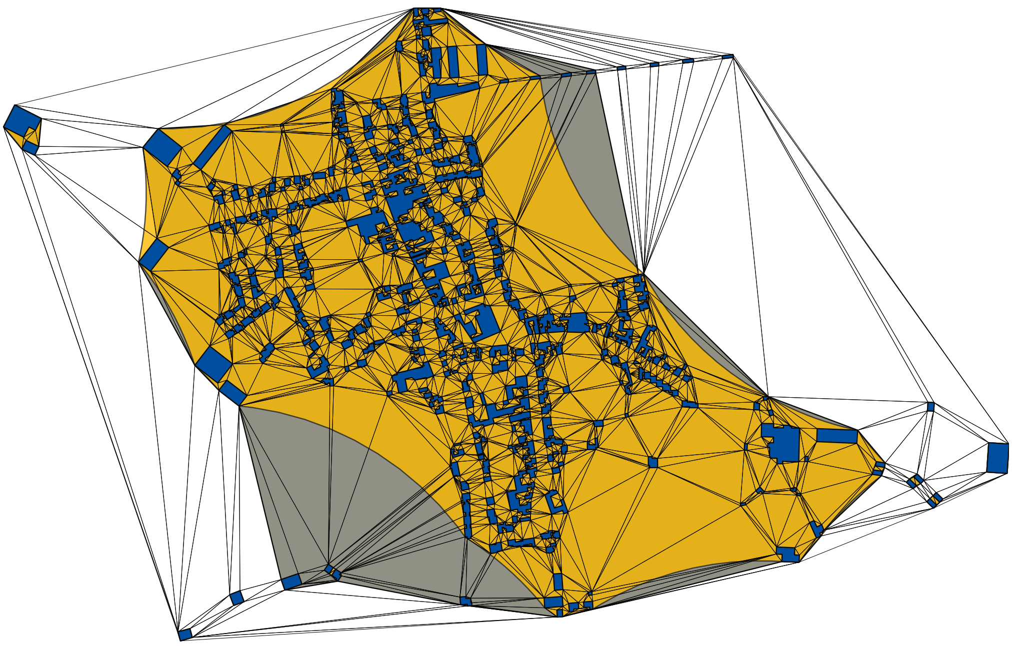


Figure 16 An example in which the triangulation-based aggregation approach by Rottmann et al. [31] produces a significantly different solution than the optimal one. The dataset is Ahrem with $\alpha = 5000$ for both solutions.

Table 2 An overview of the different datasets used for our experiments. "Diameter" refers to the diameter of the vertex set. "Density" is the proportion of the convex hull of the vertex set that is covered by input polygons. Column Φ is the segment spread, i.e., the diameter divided by the minimum distance from any polygon edge to a non-adjacent vertex.

Instance	Vertices	Polygons	Diameter	Density	Φ	$\log_2(\Phi)$
bockelskamp	1 883	360	1 324	10.7 %	36 819	15.17
bokeloh	2 793	579	1 612	11.7 %	23 900	14.54
erlenbach	3 505	650	1 789	12.0 %	17 892	14.13
ahrem	3 683	334	1 754	7.3 %	60 604	15.89
goddula	4 202	731	2 214	7.7 %	10 025	13.29
friesheim	5 402	862	2 090	9.6 %	99 379	16.60
gerolstein	7 344	1 312	3 291	8.0 %	33 149	15.02
belm	8 420	1 722	3 589	11.4 %	35 896	15.13
edendorf	10 536	1 464	2 605	9.0 %	26 794	14.71
gruppe8	14 365	2 437	10 554	0.8 %	373 983	18.51
gruppe2	14 814	2 192	4 941	3.6 %	134 884	17.04
forsbach	16 156	1 801	2 439	12.4 %	42 529	15.38
gruppe3	17 864	2 866	3 560	6.9 %	668 344	19.35
andernach	21 036	3 450	5 476	11.5 %	72 523	16.15
jena	22 365	3 970	6 643	9.9 %	288 002	18.14
bad_neuenahr	22 371	3 989	6 533	9.8 %	1 270 417	20.28
gruppe4	25 933	4 305	9 140	1.9 %	2 618 358	21.32
gruppe6	31 011	4 730	7 936	2.7 %	2 741 171	21.39
weimar	42 572	6 426	5 797	10.7 %	638 680	19.28
euskirchen	43 146	4 865	4 248	14.8 %	68 756	16.07
bergedorf	43 283	4 865	6 384	11.1 %	96 323	16.56
gruppe10	56 132	6 752	9 704	3.1 %	1 485 154	20.50
celle	62 364	8 411	6 155	10.0 %	974 941	19.89
ludwigshafen	81 262	11 341	11 094	13.9 %	1 955 340	20.90
mainz	116 944	17 551	9 821	13.9 %	16 997 667	24.02
koblenz	180 073	20 679	11 404	9.6 %	1 483 791	20.50
aachen	279 046	28 460	15 785	9.2 %	2 766 948	21.40
bonn	388 237	38 258	16 814	8.0 %	19 453 005	24.21

Table 3 Performance for $\alpha \in \{10, 3000\}$ and the α value corresponding to the maximum runtime. We measure the number of useful arcs before ($|\hat{F}_\alpha(\mathcal{B})|$) and after ($|F_\alpha(\mathcal{B})|$) testing for intersection with polygons, both with and without preprocessing. In the preprocessing case, the values are accumulated over all sub-problems as well as the final one. Merges (Tested) is the number of calls to `UPA-Opt` during the preprocessing and Succ is the percentage of successful merges. The average number of cells in the subdivision across all `UPA-Opt` calls in the preprocessing is listed under Cells (PP) and the number of cells in the final `UPA-Opt` call is given in Cells (Final). Finally, we list the number of regions after the preprocessing (PP) and in the solution (Sol).

			$\lvert\hat{F}_\alpha(\mathcal{B})\rvert$		$\lvert F_\alpha(\mathcal{B})\rvert$		Merges		Cells		Regions	
Dataset	$\lvert V(\mathcal{B})\rvert$	α	no PP	PP	no PP	PP	Tested	Succ	PP	Final	PP	Sol
Andernach	21 036	100	27 220	27 892	14 185	14 813	2 591	49 %	4.2	8 616	2 180	1 879
Andernach	21 036	260	468 642	121 414	82 389	29 845	15 509	64 %	5.0	5 272	218	147
Andernach	21 036	3 000	4 651 862	217 147	69 163	12 057	3 778	90 %	4.5	2705	50	11
Euskirchen	43 146	100	68 201	65 885	28 328	27 700	3 948	56 %	4.3	12 367	2 651	1 941
Euskirchen	43 146	500	1 025 116	233 526	64 510	24 773	5 748	80 %	4.7	47 780	259	123
Euskirchen	43 146	3 000	8 018 632	386 706	92 171	22 794	5 403	89 %	4.5	368	66	20