SIAM J. SCI. COMPUT.
Vol. 44, No. 2, pp. A798–A824

A SHAPE-NEWTON APPROACH TO THE PROBLEM OF COVERING WITH IDENTICAL BALLS*

ERNESTO G. BIRGIN†, ANTOINE LAURAIN‡, RAFAEL MASSAMBONE†, AND ARTHUR G. SANTANA†

Abstract. The problem of covering a region of the plane with a fixed number of minimum-radius identical balls is studied in the present work. An explicit construction of bi-Lipschitz mappings is provided to model small perturbations of the union of balls. This allows us to obtain analytical expressions for first- and second-order derivatives of the cost functional using nonsmooth shape optimization techniques under appropriate regularity assumptions. For regions defined as the union of disjoint convex polygons, algorithms based on Voronoi diagrams that do not rely on approximations are given to compute the derivatives. Extensive numerical experiments illustrate the capabilities and limitations of the introduced approach.

Key words. nonsmooth shape optimization, covering problem, optimization

AMS subject classifications. 49Q10, 49J52, 49Q12

DOI. 10.1137/21M1426067

1. Introduction.

The problem of covering a region of the plane with a fixed number of minimum-radius identical balls is studied in the present work by expanding the nonsmooth shape optimization approach introduced in [7]. The main challenge in this previous work was the the first-order shape sensitivity analysis with respect to perturbations of the balls' centers and radii. Therefore, investigating the second-order shape sensitivity is a natural, albeit challenging, extension of [7].

Shape optimization is the study of optimization problems where the variable is a geometric object; see [14, 19, 36]. One of the key concepts in this discipline is the notion of *shape derivative*, which measures the sensitivity of functions with respect to perturbations of the geometry. The theoretical study of second-order shape derivatives is a difficult topic in shape optimization. There exists an abundant literature on the shape Hessian in the smooth setting [11, 12, 14, 36]; while in the nonsmooth setting it is still an active research topic [26, 27]. Numerical methods based on second-order shape derivative are rarely used in shape optimization due to several obstacles. First, the second-order shape derivative is often difficult to compute and costly to implement numerically, especially when partial differential equations are involved. Second, the shape Hessian presents several theoretical issues, such as the *two norms-discrepancy* and lack of coercivity, which have been extensively studied in control problems; see [1, 11, 16] and the references therein. There exist only few attempts at defining numerical methods based on second-order information in shape optimization. In [15],

*Submitted to the journal's Methods and Algorithms for Scientific Computing section June 10, 2021; accepted for publication (in revised form) December 17, 2021; published electronically April 7, 2022.

https://doi.org/10.1137/21M1426067

Funding: This work was partially supported by FAPESP through grants 2013/07375-0, 2016/01860-1, 2018/24293-0, and 2019/25258-7, and by CNPq through grants 302682/2019-8, 304258/2018-0, and 408175/2018-4.

†Department of Computer Science, Institute of Mathematics and Statistics, University of São Paulo, Cidade Universitária, SP 05508-090, São Paulo, Brazil (egbirgin@ime.usp.br, rmassambone@ime.usp.br, ags@ime.usp.br).

‡Department of Applied Mathematics, Institute of Mathematics and Statistics, University of São Paulo, Cidade Universitária, SP 05508-090, São Paulo, Brazil (laurain@ime.usp.br).

a regularized shape-Newton method is introduced to solve an inverse problem for star-shaped geometries. Second-order preconditioning of the shape gradient has been used in [21] for image segmentation and in [3, 34] for aerodynamic optimization. Automatic shape differentiation has also been successfully employed to compute first- and second-order shape derivatives [18, 33]. We also observe that the numerical investigations using Newton-type algorithms [15, 21] are set in a relatively smooth setting. In [27], the shape Hessian was calculated for nonsmooth geometries and polygons in a form that was convenient for numerical experiments, but no numerical investigations were performed. To the best of our knowledge, the present paper is the first attempt at designing and analyzing a shape-Newton algorithm in a genuinely nonsmooth setting.

From a theoretical perspective, the main achievement of [7] was to build bi-Lipschitz transformations to model the geometry perturbations corresponding to covering with identical balls. In the present work, these transformations are key elements for the calculation of the second-order shape derivative, which, unlike the first-order shape derivative, differs from the expression that would be obtained in a smooth setting. Indeed, for the piecewise smooth shapes considered in the covering problem, various terms with a support at singular boundary points, typically circles intersection, appear in the shape Hessian.

Due to the generality of the regions to be covered considered in [7], in the presented numerical experiments, the function that measures the covering and its first-order derivatives were approximated with discretization strategies that may be very time consuming if high precision is required. In the present work, by restricting the region to be covered to be the union of disjoint convex polygons, algorithms based on Voronoi diagrams to compute the covering function and its first- and second-order derivatives *without* relying on approximations are given.

The problem of covering a two-dimensional region with identical balls has already been considered in the literature. Covering equilateral triangles and squares was considered in [31, 32], respectively, while covering the union and difference of polygons was considered in [37]. The covering of rectangles, triangles, squares and arbitrary regions was considered in [20, 29, 30, 40], respectively. However, the problem addressed in [40] actually consists of covering an arbitrary set of points, which is substantially different from the problem of covering an entire region. All of these papers approach the problem as an optimization problem. In [20, 29, 30] a simulated annealing approach with local search in which the centers of the balls are chosen as points on an adaptive mesh is considered. In [31, 32], a discrete rule is used to define the radius; while a Broyden–Fletcher–Goldfarb–Shanno (BFGS) method is used to solve subproblems in which the radius is fixed. A feasible direction method that requires solving a linear programming problem at each iteration was proposed in [37]. None of the mentioned works addresses the problem in a unified way as a continuous optimization problem, nor do they present first- or second-order derivatives of the functions that define the problem. In [5], the problem of covering an arbitrary region is modeled as a nonlinear semidefinite programming problem using convex algebraic geometry tools. The introduced model describes the covering problem without resorting to discretizations, but it depends on some polynomials of unknown degrees whose coefficients are difficult to compute, limiting the applicability of the method.

The rest of this paper is organized as follows. Section 2 presents a formal definition of the problem, the formula for the first-order derivative introduced in [7], and the formula for the second-order derivative being introduced in the present work. Section 3 presents the derivation of the second-order derivatives for nondegenerate cases. Algorithms based on Voronoi diagrams for the exact calculation of the cover-

ing function and its first- and second-order derivatives are introduced in section 4. Extensive numerical experiments are given in section 5. Final considerations are given in section 6.

Notation. $\|\cdot\|$ denotes the Euclidean norm. Given $x, y \in \mathbb{R}^n$, $x \cdot y = x^\top y \in \mathbb{R}$, while $x \otimes y = xy^\top \in \mathbb{R}^{n\times n}$. The divergence of a vector field $\mathbb{R}^2 \ni (x, y) \mapsto V(x, y) = (V_1(x, y), V_2(x, y)) \in \mathbb{R}^2$ is defined by $\operatorname{div} V := \frac{\partial V_1}{\partial x} + \frac{\partial V_2}{\partial y}$, and its Jacobian matrix is denoted DV. Given an open set $S \in \mathbb{R}^n$, $\overline{S}$ denotes its closure, $\partial S = \overline{S} \setminus S$ its boundary, and $\mathrm{Vol}(S)$ its volume. Let $B(x_i, r)$ denote an open ball with center $x_i \in \mathbb{R}^2$ and radius r. For a sufficiently smooth set $S \subset \mathbb{R}^2$, $\nu_S(z)$ denotes the unitary-norm outwards normal vector to S at z and $\tau_S(z)$ the unitary-norm tangent vector to ∂S at z (pointing counterclockwise). In the particular case $S = B(x_i, r)$, we use the simpler notation $\nu_i := \nu_{B(x_i,r)}$ and $\tau_i := \tau_{B(x_i,r)}$, and we have $\nu_i(z) = (\cos\theta_z, \sin\theta_z)^\top$ and $\tau_i(z) = (-\sin\theta_z, \cos\theta_z)^\top$, where θ_z is the angular coordinate of $z - x_i$. For intersection points $z \in \partial S \cap B(x_i, r)$, we also use the notation $\nu_{-i}(z) := \nu_S(z)$.

2. The shape optimization problem. Let $A \subset \mathbb{R}^2$ and $\Omega(\boldsymbol{x}, r) = \cup_{i=1}^m B(x_i, r)$ with $\boldsymbol{x} := \{x_i\}_{i=1}^m$. We consider the problem of covering A using a fixed number m of identical balls $B(x_i, r)$ with minimum radius r, i.e., we are looking for $(\boldsymbol{x}, r) \in \mathbb{R}^{2m+1}$ such that $A \subset \Omega(\boldsymbol{x}, r)$ with minimum r. The problem can be formulated as

$$\underset{(\boldsymbol{x},r)\in\mathbb{R}^{2m+1}}{\text{Minimize}} \; r \;\text{ subject to }\; G(\boldsymbol{x}, r) = 0, \tag{2.1}$$

where

$$G(\boldsymbol{x}, r) := \mathrm{Vol}(A) - \mathrm{Vol}(A \cap \Omega(\boldsymbol{x}, r)). \tag{2.2}$$

Note that $G(\boldsymbol{x}, r) = 0$ if and only if $A \subset \Omega(\boldsymbol{x}, r)$ up to a set of zero measure, i.e., when $\Omega(\boldsymbol{x}, r)$ covers A.

The derivatives of G can be computed using techniques of shape calculus [14, 19, 27, 28, 36]. In particular, it was shown in [7] that, under suitable assumptions,

$$\nabla G(\boldsymbol{x}, r) = -\left(\int_{\mathcal{A}_1} \nu_1(z)\, dz, \ldots, \int_{\mathcal{A}_m} \nu_m(z)\, dz, \int_{\partial\Omega(\boldsymbol{x},r)\cap A} dz\right)^\top, \tag{2.3}$$

where

$$\mathcal{A}_i = \partial B(x_i, r) \cap \partial\Omega(\boldsymbol{x}, r) \cap A \quad \text{ for } i = 1, \ldots, m. \tag{2.4}$$

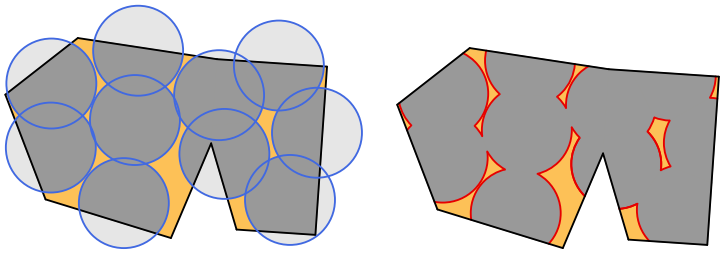
In the present work, we show that

$$\nabla^2 G(\boldsymbol{x}, r) = \begin{pmatrix} \nabla^2_{\boldsymbol{x}} G(\boldsymbol{x}, r) & \nabla^2_{\boldsymbol{x},r} G(\boldsymbol{x}, r) \\ \nabla^2_{\boldsymbol{x},r} G(\boldsymbol{x}, r)^\top & \nabla^2_r G(\boldsymbol{x}, r) \end{pmatrix}, \tag{2.5}$$

where $\nabla^2_{\boldsymbol{x}} G(\boldsymbol{x}, r) \in \mathbb{R}^{2m\times 2m}$, $\nabla^2_{\boldsymbol{x},r} G(\boldsymbol{x}, r) \in \mathbb{R}^{2m}$, and $\nabla^2_r G(\boldsymbol{x}, r) = \partial^2_r G(\boldsymbol{x}, r) \in \mathbb{R}$ are described below. Their description is based on the fact that each set $\mathcal{A}_i$ can be represented by a finite number $m_i \geq 0$ of arcs of the circle $\partial B(x_i, r)$, under suitable regularity assumptions on $\boldsymbol{x}$ and A (see Assumptions 1 and 2 in section 3). Note that, since $(\cup_{i=1}^m \partial B(x_i, r)) \cap \partial\Omega(\boldsymbol{x}, r) = \partial\Omega(\boldsymbol{x}, r)$, by (2.4),

$$\bigcup_{i=1}^m \mathcal{A}_i = \partial\Omega(\boldsymbol{x}, r) \cap A; \tag{2.6}$$

see Figure 1. (This means that, under a regularity assumption, namely, Assumption 1 stated in section 3, the union of all $\mathcal{A}_i$ represents a partition of $\partial\Omega(\boldsymbol{x}, r) \cap A$.) Each arc in $\mathcal{A}_i$ can be represented by a pair of points (v, w), named starting and ending points, in counterclockwise direction, i.e., such that the angular coordinates θ_v and θ_w of $v - x_i$ and $w - x_i$, respectively, satisfy $\theta_v \in [0, 2\pi)$ and $\theta_w \in (\theta_v, \theta_v + 2\pi]$; see Figure 2. If $\mathcal{A}_i$ is not a full circle, we denote by $\mathbb{A}_i$ the set of pairs (v, w) that represent the arcs in $\mathcal{A}_i$; otherwise, we define $\mathbb{A}_i = \emptyset$. In addition, if $\mathcal{A}_i$ is a full circle, then we set Circle($\mathbb{A}_i$) equal to true; otherwise, we set Circle($\mathbb{A}_i$) equal to false. We say a configuration $(\boldsymbol{x}, r)$ is nondegenerate if, for every $i = 1, \ldots, m$, every $(v, w) \in \mathbb{A}_i$, and every $z \in \{v, w\}$, there exists one and only one $\nu_{-i}(z)$ and $\nu_{-i}(z) \cdot \tau_i(z) \neq 0$. A characterization of nondegenerate configurations, which satisfy Assumptions 1 and 2, is given in the next section.



(a) (b)

FIG. 1. *Figure* (a) *represents a region A to be covered and an arbitrary configuration of balls $\Omega(\boldsymbol{x}, r)$. Figure* (b) *represents, in red, $\partial\Omega(\boldsymbol{x}, r) \cap A$. Each $\mathcal{A}_i$ corresponds to the red arcs that intersect $\partial B(x_i, r)$. Note that in this example, most sets $\mathcal{A}_i$ contain two or three maximal arcs; and there is only one set $\mathcal{A}_i$ with four maximal arcs. (Figure in color online.)*

Assuming $(\boldsymbol{x}, r)$ is nondegenerate, we have that $\nabla_r^2 G(\boldsymbol{x}, r)$ in (2.5) is given by

$$\nabla_r^2 G(\boldsymbol{x}, r) = -\frac{\mathrm{Per}(\partial\Omega(\boldsymbol{x}, r) \cap A)}{r} - \sum_{i=1}^{m} \sum_{(v,w)\in\mathbb{A}_i} \left[\!\!\left[\frac{|L(z)| - \nu_{-i}(z) \cdot \nu_i(z)}{\nu_{-i}(z) \cdot \tau_i(z)} \right]\!\!\right]_v^w, \tag{2.7}$$

where, for an arbitrary expression $\Phi(z)$, $[\![\Phi(z)]\!]_v^w := \Phi(w) - \Phi(v)$, $\mathrm{Per}(S)$ denotes the perimeter of the set S, for an extreme z of an arc represented by $(v, w) \in \mathbb{A}_i$, $L(z) = \{\ell \in \{1, \ldots, m\} \setminus \{i\} \mid z \in \partial B(x_\ell, r)\}$, and $|L(z)|$ denotes the cardinality of $L(z)$.

Matrix $\nabla_{\boldsymbol{x}}^2 G(\boldsymbol{x}, r)$ in (2.5) is given by the 2×2 diagonal blocks

$$\partial_{x_i x_i}^2 G(\boldsymbol{x}, r) = \frac{1}{r} \int_{\mathcal{A}_i} -\nu_i(z) \otimes \nu_i(z) + \tau_i(z) \otimes \tau_i(z)\, dz + \sum_{(v,w)\in\mathbb{A}_i} \left[\!\!\left[\frac{\nu_{-i}(z) \cdot \nu_i(z)}{\nu_{-i}(z) \cdot \tau_i(z)} \nu_i(z) \otimes \nu_i(z) \right]\!\!\right]_v^w \tag{2.8}$$

and the 2×2 off-diagonal blocks

$$\partial_{x_i x_\ell}^2 G(\boldsymbol{x}, r) = \sum_{v\in\mathcal{I}_{i\ell}} \frac{\nu_i(v) \otimes \nu_\ell(v)}{\nu_\ell(v) \cdot \tau_i(v)} - \sum_{w\in\mathcal{O}_{i\ell}} \frac{\nu_i(w) \otimes \nu_\ell(w)}{\nu_\ell(w) \cdot \tau_i(w)}, \tag{2.9}$$

where $\mathcal{I}_{i\ell} = \{v \in \partial B(x_\ell, r) \mid (v, \cdot) \in \mathbb{A}_i\}$ and $\mathcal{O}_{i\ell} = \{w \in \partial B(x_\ell, r) \mid (\cdot, w) \in \mathbb{A}_i\}$. (Note that $\mathcal{I}_{i\ell} = \mathcal{O}_{i\ell} = \emptyset$ for all $\ell \neq i$ if $\mathbb{A}_i = \emptyset$.) Finally, array $\nabla_{\boldsymbol{x},r}^2 G(\boldsymbol{x}, r)$ in (2.5) is given by the two-dimensional arrays

$$\partial_{x_i r}^2 G(\boldsymbol{x}, r) = -\frac{1}{r} \int_{\mathcal{A}_i} \nu_i(z)\, dz + \sum_{(v,w)\in\mathbb{A}_i} \left[\!\!\left[\frac{\nu_{-i}(z) \cdot \nu_i(z)}{\nu_{-i}(z) \cdot \tau_i(z)} \nu_i(z) - \sum_{\ell\in L(z)} \frac{\nu_i(z)}{\tau_i(z) \cdot \nu_\ell(z)} \right]\!\!\right]_v^w. \tag{2.10}$$

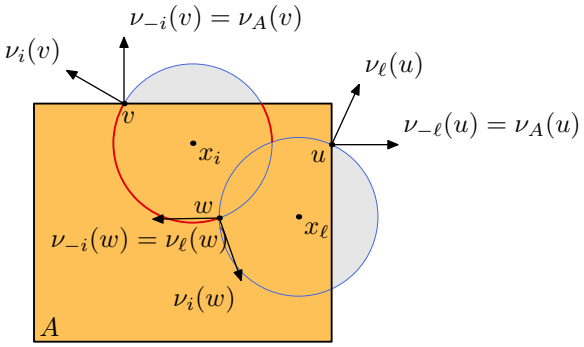


FIG. 2. *The set $\mathcal{A}_i = \partial B(x_i, r) \cap \Omega(\boldsymbol{x}, r) \cap A$ is composed of two arcs (in red). If $z \in \partial B(x_i, r) \cap \partial B(x_\ell, r)$ for some $\ell \neq i$, as for $z = w$, then $\nu_{-i}(z) = \nu_\ell(z)$, while if $z \in \partial B(x_i, r) \cap \partial A$, as for $z = u$, then $\nu_{-i}(z) = \nu_A(z)$. (Figure in color online.)*

3. Proof of second-order differentiability of G. In this section, we prove that the second-order derivatives of G, as defined in (2.2), are given by (2.5, 2.7, 2.8, 2.9, 2.10). In [7] we have built appropriate bi-Lipschitz mappings T_t in order to use integration by substitution for the differentiation of $G(\boldsymbol{x} + t\delta\boldsymbol{x}, r)$ and $G(\boldsymbol{x}, r + t\delta r)$. Some of the more technical aspects of these constructions were related to the fact that G is an area functional, which required defining T_t on $\Omega(\boldsymbol{x}, r) \cap A$ and on $\partial(\Omega(\boldsymbol{x}, r) \cap A)$. Since ∇G only involves boundary integrals that, in addition, can be decomposed into integrals on arcs of circle, this facilitates the construction of the mappings T_t required to compute $\nabla^2 G$, as T_t need only be defined on $\partial\Omega(\boldsymbol{x}, r) \cap A$.

We consider two types of transformations for the shape sensitivity analysis. First, in the case of fixed radius and center perturbations one needs a mapping T_t between the reference set $\partial\Omega(\boldsymbol{x}, r) \cap A$ and the perturbed set $\partial\Omega(\boldsymbol{x} + t\delta\boldsymbol{x}, r) \cap A$; see Theorem 2. Second, in the case of fixed centers and radius perturbation one needs a mapping T_t between $\partial\Omega(\boldsymbol{x}, r) \cap A$ and the perturbed set $\partial\Omega(\boldsymbol{x}, r + t\delta r) \cap A$; see Theorem 3. The shape sensitivity analysis of ∇G is then achieved through integration by substitution using T_t. The construction of these mappings T_t is similar to the constructions in [7]; however, the results are presented in a different way as we need specific properties of T_t to compute the derivatives of ∇G. One of the main differences with respect to [7] appears in Theorem 2, where one considers simultaneous perturbations of all the balls' centers, which allows us to simplify the calculations of the Hessian of G. On the one hand, T_t was used in [7] mainly to prove first-order shape differentiability and its unusual structure did not affect the expression of the first-order shape derivative, in the sense that a similar formula would have been obtained in a smooth setting. On the other hand, the expression of the second-order shape derivative of G at a nonsmooth reference domain Ω differs significantly from the expression that would be obtained for a smooth Ω, as it involves terms with a support at singular boundary points of Ω, and the particular structure of T_t now plays an important role in the calculation of those singular terms.

In [7], we have described detailed conditions to avoid degenerate situations and we also discussed various examples of such degeneracies and how they may affect the numerical algorithm. In the present paper we use the same conditions to prove second-order differentiability of G. To summarize, the main issues when studying the differentiability of G arise when two balls are tangent or exactly superposed, when the boundaries of more than two balls intersect at the same point, or when $\Omega(\boldsymbol{x}, r)$ and A

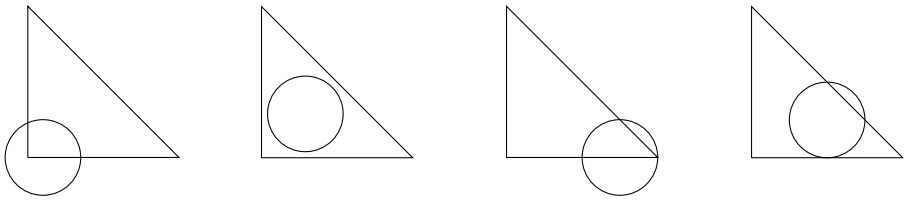
are not compatible in the sense of Definition 3.1. The role of Assumptions 1 and 2 is to avoid these singular cases, which allows us to prove second-order differentiability of G. We emphasize that these assumptions only exclude a null-measure set of balls' configurations in $\mathbb{R}^{2m+1}$, and in [6] we show via the study of several singular cases that the second-order differentiability of G fails when these assumptions are not satisfied.

ASSUMPTION 1. *The centers* $\{x_i\}_{i=1}^m$ *satisfy* $\|x_i - x_j\| \notin \{0, 2r\}$ *for* $1 \le i, j \le m$, $i \ne j$, *and* $\partial B(x_i, r) \cap \partial B(x_j, r) \cap \partial B(x_k, r) = \emptyset$ *for all* $1 \le i, j, k \le m$ *with* i, j, k *pairwise distinct.*

DEFINITION 3.1. *Let* ω_1, ω_2 *be open subsets of* $\mathbb{R}^2$. *We call* ω_1 *and* ω_2 *compatible if* $\omega_1 \cap \omega_2 \ne \emptyset$, ω_1 *and* ω_2 *are Lipschitz domains, and the following conditions hold:* (i) $\omega_1 \cap \omega_2$ *is a Lipschitz domain;* (ii) $\partial\omega_1 \cap \partial\omega_2$ *is finite;* (iii) $\partial\omega_1$ *and* $\partial\omega_2$ *are locally smooth in a neighborhood of* $\partial\omega_1 \cap \partial\omega_2$*; and* (iv) $\tau_1(x) \cdot \nu_2(x) \ne 0$ *for all* $x \in \partial\omega_1 \cap \partial\omega_2$, *where* $\tau_1(x)$ *is a tangent vector to* $\partial\omega_1$ *at* x *and* $\nu_2(x)$ *is a normal vector to* $\partial\omega_2$ *at* x.

ASSUMPTION 2. *Sets* $\Omega(\boldsymbol{x}, r)$ *and* A *are compatible.*

The notion of compatiblity from Definition 3.1 is illustrated in Figure 3: the ball ω_1 and the triangle ω_2 satisfy $\omega_1 \cap \omega_2 \ne \emptyset$ and are always compatible except when the ball is tangent to the boundary of the triangle or when the boundary of the ball meets a corner of the triangle. Thus, the set of points where they are not compatible has measure zero. Note that the examples depicted in Figure 3 are representative of the geometric configurations occurring in practice.



(a) compatible (b) compatible (c) not compatible (d) not compatible

FIG. 3. *Compatibility of the ball* ω_1 *and the triangle* ω_2 *in the sense of Definition* 3.1. *In* (c), *condition* (iii) *of Definition* 3.1 *fails as* $\partial\omega_2$ *is not locally smooth in a neighborhood of one of the three points in* $\partial\omega_1 \cap \partial\omega_2$. *In* (d), *condition* (iv) *of Definition* 3.1 *fails for one of the three points in* $\partial\omega_1 \cap \partial\omega_2$.

We observe that $\Omega(\boldsymbol{x}, r)$ is Lipschitz under Assumption 1, and if, in addition, the intersection of $\partial\Omega(\boldsymbol{x}, r)$ and ∂A is empty, then Assumption 2 is satisfied. Hence, in this particular case we can drop Assumption 2 in Theorems 2 and 3.

We also recall the following basic results, which are key ingredients for the calculation of the shape Hessian of G.

THEOREM 1 (tangential divergence theorem). *Let* $\Gamma \subset \mathbb{R}^2$ *be a* C^k *open curve,* $k \ge 2$, *with a parameterization* γ, *and denote* (v, w) *the starting and ending points of* Γ, *respectively, with respect to* γ. *Let* τ *be the unitary-norm tangent vector to* Γ, ν *the unitary-norm normal vector to* Γ, *and* $\mathcal{H}$ *the mean curvature of* Γ, *with respect to the parameterization* γ. *Let* $F \in W^{1,1}(\Gamma, \mathbb{R}^2) \cap C^0(\overline{\Gamma}, \mathbb{R}^2)$*; then we have*

$$\int_\Gamma \operatorname{div}_\Gamma(F)\, dz = \int_\Gamma \mathcal{H} F \cdot \nu\, dz + F(w) \cdot \tau(w) - F(v) \cdot \tau(v) = \int_\Gamma \mathcal{H} F \cdot \nu\, dz + [\![F(z) \cdot \tau(z)]\!]_v^w,$$

where $\operatorname{div}_\Gamma(F) := \operatorname{div}(F) - DF\nu \cdot \nu$ *is the tangential divergence of* F *on* Γ.

Proof. The result follows from [35, section 7.2] and [14, Chap. 9, section 5.5]. ∎

LEMMA 3.2 (change of variables for line integrals). *Let $\Gamma \subset \mathbb{R}^2$ be a C^k open curve, $k \geq 2$, and ν a unitary-norm normal vector to Γ. Let $F \in C^0(\overline{\Gamma}, \mathbb{R}^2)$ and $T_t : \overline{\Gamma} \to T_t(\overline{\Gamma})$ be a bi-Lipschitz mapping. Then*

$$\int_{T_t(\Gamma)} F(z)\, dz = \int_{\Gamma} F(T_t(z))\omega_t(z)\, dz,$$

where

$$\omega_t(z) := \|M(z,t)\nu(z)\| \tag{3.1}$$

and $M(z,t) := \det(DT_t(z))DT_t(z)^{-\top}$ is the cofactor matrix of $DT_t(z)$. Furthermore, we have

$$\partial_t \omega_t|_{t=0} = \operatorname{div}_\Gamma V \text{ with } V := \partial_t T_t|_{t=0} \text{ on } \Gamma. \tag{3.2}$$

Proof. See [19, Prop. 5.4.3] for the proof. ∎

3.1. Construction of a perturbation field for center perturbations. Theorem 2 below employs several ideas from [7, Thms. 3.2 and 3.6]. However, an important difference is that we consider simultaneous center perturbations for all balls instead of just one, which is more convenient for the calculation of $\nabla^2 G$. Theorem 2 provides an appropriate mapping T_t for the differentiation of $\partial_{x_i} G(\boldsymbol{x} + t\delta\boldsymbol{x}, r)$ that will be used in sections 3.4 and 3.5 and for the differentiation of $\partial_r G(\boldsymbol{x} + t\delta\boldsymbol{x}, r)$ in section 3.6.

THEOREM 2. *Suppose that Assumptions 1 and 2 hold. Then there exists $t_0 > 0$ such that for all $t \in [0, t_0]$ we have the following decomposition:*

$$\partial\Omega(\boldsymbol{x} + t\delta\boldsymbol{x}, r) \cap A = \bigcup_{k=1}^{\bar{k}} \mathcal{S}_k(t), \tag{3.3}$$

where $\mathcal{S}_k(t)$ are arcs parameterized by an angle aperture $[\theta_{k,v}(t), \theta_{k,w}(t)]$, $t \mapsto \theta_{k,v}(t)$, $t \mapsto \theta_{k,w}(t)$ are continuous functions on $[0, t_0]$, and $\bar{k}$ is independent of t.

Also, for all $t \in [0, t_0]$ there exists a bi-Lipschitz mapping $T_t : \partial\Omega(\boldsymbol{x}, r) \cap A \to \mathbb{R}^2$ satisfying $T_t(\partial\Omega(\boldsymbol{x}, r) \cap A) = \partial\Omega(\boldsymbol{x} + t\delta\boldsymbol{x}, r) \cap A$ and $T_t(\mathcal{S}_k(0)) = \mathcal{S}_k(t)$ for all $k = 1, \ldots, \bar{k}$. Furthermore,

$$V \cdot \nu_i = \delta x_i \cdot \nu_i \text{ on } \mathcal{S}_k(0) \subset \partial B(x_i, r), \tag{3.4}$$

$$V(z) = \delta x_i - \frac{\nu_A(z) \cdot \delta x_i}{\tau_i(z) \cdot \nu_A(z)} \tau_i(z) \text{ if } z \in \partial B(x_i, r) \cap \partial A, \tag{3.5}$$

$$V(z) = \delta x_i - \frac{\nu_\ell(z) \cdot (\delta x_i - \delta x_\ell)}{\tau_i(z) \cdot \nu_\ell(z)} \tau_i(z) \text{ if } z \in \partial B(x_i, r) \cap \partial B(x_\ell, r), i \neq \ell, \tag{3.6}$$

where $V := \partial_t T_t|_{t=0}$.

Proof. The decomposition (3.3) relies on Assumptions 1 and 2 and is obtained in a similar way as in [7, Thm. 3.2]. Therefore, in this proof we focus on the construction of the mapping T_t. We observe that each extremity of the arcs $\mathcal{S}_k(t)$ in (3.3) is either a point belonging to $\partial B(x_i + t\delta x_i, r) \cap \partial A$ or a point in $\partial B(x_i + t\delta x_i, r) \cap \partial B(x_\ell + t\delta x_\ell, r)$.

We first provide a general formula for the angle $\vartheta(t)$ in local polar coordinates with the pole $x_i + t\delta x_i$, describing a point in $\partial B(x_i + t\delta x_i, r) \cap \partial A$. Let ϕ be the

oriented distance function to A, defined as $\phi(x) := d(x, A) - d(x, A^c)$, where $d(x, A)$ is the distance from x to the set A, and $z \in \partial B(x_i, r) \cap \partial A$. Since $\Omega(\boldsymbol{x}, r)$ and A are compatible due to Assumption 2, it follows that ∂A is locally smooth around the points $\partial B(x_i, r) \cap \partial A$, hence there exists a neighborhood U_z of z such that the restriction of ϕ to U_z is smooth, $\phi(x) = 0$ and $\|\nabla\phi(x)\| = 1$ for all $x \in \partial A \cap U_z$.

Let (r, θ_z) be the polar coordinates of z with the pole x_i. Introduce the function

$$\sigma(t, \vartheta) = \phi\left(x_i + t\delta x_i + r\begin{pmatrix}\cos\vartheta\\ \sin\vartheta\end{pmatrix}\right)$$

and compute

$$\partial_\vartheta\sigma(0, \theta_z) = r\begin{pmatrix}-\sin\theta_z\\ \cos\theta_z\end{pmatrix} \cdot \nabla\phi\left(x_i + r\begin{pmatrix}\cos\theta_z\\ \sin\theta_z\end{pmatrix}\right) = r\tau_i(z) \cdot \nabla\phi(z).$$

As $\Omega(\boldsymbol{x}, r)$ and A are compatible, $B(x_i, r)$ is not tangent to ∂A, and using $\|\nabla\phi(z)\| = 1$ we obtain $\tau_i(z) \cdot \nabla\phi(z) \neq 0$. Thus, we can apply the implicit function theorem which yields the existence of a smooth function $[0, t_0] \ni t \mapsto \vartheta(t)$ with $\sigma(t, \vartheta(t)) = 0$ and $\vartheta(0) = \theta_z$. We also compute, using $\nabla\phi(z) = \|\nabla\phi(z)\|\nu_A(z)$ since ϕ is the oriented distance function to A, that

$$\vartheta'(0) = -\frac{\partial_t\sigma(0, \vartheta(0))}{\partial_\vartheta\sigma(0, \vartheta(0))} = -\frac{\nabla\phi(z) \cdot \delta x_i}{r\tau_i(z) \cdot \nabla\phi(z)} = -\frac{\nu_A(z) \cdot \delta x_i}{r\tau_i(z) \cdot \nu_A(z)}. \tag{3.7}$$

We now consider the second case of a point in $\partial B(x_i + t\delta x_i, r) \cap \partial B(x_\ell + t\delta x_\ell, r)$, $i \neq \ell$ and introduce the functions

$$\psi(t, \vartheta) = \|\zeta(t, \vartheta)\|^2 - r^2 \quad \text{with} \quad \zeta(t, \vartheta) = x_i + t\delta x_i - x_\ell - t\delta x_\ell + r\begin{pmatrix}\cos\vartheta\\ \sin\vartheta\end{pmatrix}.$$

Observe that $\vartheta \mapsto \zeta(t, \vartheta)$ is a parameterization of the circle $\partial B(x_i + t\delta x_i, r)$ in a coordinate system of center x_ℓ, which means that the solutions of the equation $\psi(t, \vartheta) = 0$ describe the intersections between $\partial B(x_i + t\delta x_i, r)$ and $\partial B(x_\ell + t\delta x_\ell, r)$. We compute $\partial_\vartheta\psi(0, \vartheta) = 2\zeta(0, \vartheta) \cdot \partial_\vartheta\zeta(0, \vartheta)$ with

$$\zeta(0, \vartheta) = x_i - x_\ell + r\begin{pmatrix}\cos\vartheta\\ \sin\vartheta\end{pmatrix} \quad \text{and} \quad \partial_\vartheta\zeta(0, \vartheta) = r\begin{pmatrix}-\sin\vartheta\\ \cos\vartheta\end{pmatrix}.$$

Now let $z \in \partial B(x_i, r) \cap \partial B(x_\ell, r)$, and let θ_z be the corresponding angle in a polar coordinate system with pole x_i. Since Assumption 1 is satisfied, it is easy to see that $\partial_\vartheta\psi(0, \theta_z) = 2\zeta(0, \theta_z) \cdot \partial_\vartheta\zeta(0, \theta_z) \neq 0$. Hence, the implicit function theorem can be applied to $(t, \vartheta) \mapsto \psi(t, \vartheta)$ in a neighborhood of $(0, \theta_z)$. This yields the existence, for t_0 sufficiently small, of a smooth function $t \mapsto \vartheta(t)$ in $[0, t_0]$ such that $\psi(t, \vartheta(t)) = 0$ in $[0, t_0]$ and $\vartheta(0) = \theta_z$. We also have the derivative

$$\vartheta'(t) = -\frac{\partial_t\psi(t, \vartheta(t))}{\partial_\vartheta\psi(t, \vartheta(t))} = -\frac{\zeta(t, \vartheta(t)) \cdot \partial_t\zeta(t, \vartheta(t))}{\zeta(t, \vartheta(t)) \cdot \partial_\vartheta\zeta(t, \vartheta(t))},$$

and in particular, using $\nu_i = (\cos\theta_z, \sin\theta_z)^\top$ and $\tau_i = (-\sin\theta_z, \cos\theta_z)^\top$,

$$\vartheta'(0) = -\frac{(x_i - x_\ell + r\nu_i) \cdot (\delta x_i - \delta x_\ell)}{(x_i - x_\ell + r\nu_i) \cdot (r\tau_i)} = -\frac{\nu_\ell \cdot (\delta x_i - \delta x_\ell)}{r\nu_\ell \cdot \tau_i}. \tag{3.8}$$

We are now ready to build the mapping T_t. Let $\mathcal{S}(t) \subset \partial B(x_i + t\delta x_i, r)$ be one of the arcs parameterized by the angle aperture $[\theta_v(t), \theta_w(t)]$ in the decomposition (3.3); we have dropped the index k for simplicity. Then, $\theta_v(t)$ and $\theta_w(t)$ are given by $\vartheta(t)$ with either $\theta_z = \theta_v(0)$ or $\theta_z = \theta_w(0)$, and $\vartheta(t)$ either corresponds to an intersection $\partial B(x_i + t\delta x_i, r) \cap \partial A$ or to an intersection $\partial B(x_i + t\delta x_i, r) \cap \partial B(x_\ell + t\delta x_\ell, r)$. Thus we define T_t on the arc $\mathcal{S}(0)$ as

$$T_t(x) := x_i + t\delta x_i + r \begin{pmatrix} \cos \xi(t,\theta) \\ \sin \xi(t,\theta) \end{pmatrix} \text{ with } x = x_i + r \begin{pmatrix} \cos \theta \\ \sin \theta \end{pmatrix} \in \mathcal{S}(0), \tag{3.9}$$

where

$$\xi(t,\theta) := \theta_v(t) + \alpha(t)(\theta - \theta_v(0)) \text{ for } (t,\theta) \in [0,t_0] \times [\theta_v(0), \theta_w(0)] \text{ and } \alpha(t) := \tfrac{\theta_w(t) - \theta_v(t)}{\theta_w(0) - \theta_v(0)}. \tag{3.10}$$

The bi-Lipschitz property of T_t on $\partial\Omega(\boldsymbol{x}, r) \cap A$ is obtained as in the proof of [7, Thm. 3.3].

Finally, differentiating in (3.9) with respect to t and using $\xi(0,\theta) = \theta$ we get (3.4). Then (3.10) yields $\xi(t, \theta_v(0)) = \theta_v(t)$, $\xi(t, \theta_w(0)) = \theta_w(t)$, $\partial_t \xi(0, \theta_v(0)) = \theta_v'(0)$, $\partial_t \xi(0, \theta_w(0)) = \theta_v'(0)$, consequently using (3.7) we obtain (3.5) and using (3.8) we obtain (3.6). ∎

Remark 1. A closer look at the construction of T_t in the proof of Theorem 2 shows that there are infinitely many ways of constructing T_t with the key property $T_t(\partial\Omega(\boldsymbol{x}, r) \cap A) = \partial\Omega(\boldsymbol{x} + t\delta\boldsymbol{x}, r) \cap A$ that is used to compute derivatives of G. However, the definition of the derivatives of G, and consequently the expressions of the gradient and Hessian of G, are independent of the choice of T_t.

3.2. Construction of a perturbation field for radius perturbations. Theorem 3 below relies on several ideas from [7, Thms. 3.3 and 3.8] and provides an appropriate mapping T_t for the differentiation of $\partial_r G(\boldsymbol{x}, r + t\delta r)$ that will be used in section 3.3. We omit the proof of Theorem 3 which is similar to the proof of Theorem 2 and refer to [6] for the detailed proof.

THEOREM 3. *Suppose that Assumptions* 1 *and* 2 *hold. Then there exists* $t_0 > 0$ *such that for all* $t \in [0, t_0]$ *we have the decomposition* $\partial\Omega(\boldsymbol{x}, r + t\delta r) \cap A = \bigcup_{k=1}^{\bar{k}} \mathcal{S}_k(t)$, *where* $\mathcal{S}_k(t)$ *are arcs parameterized by an angle aperture* $[\theta_{k,v}(t), \theta_{k,w}(t)]$, $t \mapsto \theta_{k,v}(t)$, $t \mapsto \theta_{k,w}(t)$ *are continuous functions on* $[0, t_0]$, *and* $\bar{k}$ *is independent of* t.

Also, for all $t \in [0, t_0]$ *there exists a bi-Lipschitz mapping* $T_t : \partial\Omega(\boldsymbol{x}, r) \cap A \to \mathbb{R}^2$ *satisfying* $T_t(\partial\Omega(\boldsymbol{x}, r) \cap A) = \partial\Omega(\boldsymbol{x}, r + t\delta r) \cap A$ *and* $T_t(\mathcal{S}_k(0)) = \mathcal{S}_k(t)$ *for all* $k = 1, \ldots, \bar{k}$. *In addition, we have*

$$V \cdot \nu_i = \delta r \text{ on } \mathcal{S}_k(0) \subset \partial B(x_i, r), \tag{3.11}$$

$$V(z) = \delta r \nu_i(z) - \delta r \frac{\nu_A(z) \cdot \nu_i(z)}{\tau_i(z) \cdot \nu_A(z)} \tau_i(z) \text{ if } z \in \partial B(x_i, r) \cap \partial A, \tag{3.12}$$

$$V(z) = \delta r \nu_i(z) + \delta r \frac{1 - \nu_\ell(z) \cdot \nu_i(z)}{\tau_i(z) \cdot \nu_\ell(z)} \tau_i(z) \text{ if } z \in \partial B(x_i, r) \cap \partial B(x_\ell, r), i \neq \ell, \tag{3.13}$$

where $V := \partial_t T_t|_{t=0}$.

3.3. Second-order derivative of G with respect to the radius. We have $\partial_r G(\boldsymbol{x}, r) = -\int_{\partial\Omega(\boldsymbol{x},r)\cap A} dz$; see (2.3) and [7, section 3.3] for the detailed calculation. As in [7], the calculation is achieved through integration by substitution using the mapping T_t given by Theorem 3, which requires that Assumptions 1 and

2 hold. According to Theorem 3, there exists a bi-Lipschitz mapping T_t satisfying $T_t(\partial\Omega(\boldsymbol{x},r)\cap A)=\partial\Omega(\boldsymbol{x},r+t\delta r)\cap A$, and this yields, using Lemma 3.2 on each arc of $\partial\Omega(\boldsymbol{x},r)\cap A$,

$$\partial_r G(\boldsymbol{x},r+t\delta r)=-\int_{\partial\Omega(\boldsymbol{x},r+t\delta r)\cap A} dz=-\int_{T_t(\partial\Omega(\boldsymbol{x},r)\cap A)} dz=-\int_{\partial\Omega(\boldsymbol{x},r)\cap A}\omega_t(z)\,dz.$$

Thus, using Lemma 3.2 and the decomposition (2.6), we compute

$$\frac{d}{dt}\partial_r G(\boldsymbol{x},r+t\delta r)\bigg|_{t=0}=-\int_{\partial\Omega(\boldsymbol{x},r)\cap A}\operatorname{div}_\Gamma V(z)\,dz=-\sum_{i=1}^m\int_{\mathcal{A}_i}\operatorname{div}_\Gamma V(z)\,dz.$$

Applying Theorem 1 for each arc in $\mathcal{A}_i$, we obtain

$$\frac{d}{dt}\partial_r G(\boldsymbol{x},r+t\delta r)\bigg|_{t=0}=-\sum_{i=1}^m\int_{\mathcal{A}_i}\mathcal{H}V\cdot\nu_i\,dz-\sum_{i=1}^m\sum_{(v,w)\in\mathbb{A}_i}[\![V(z)\cdot\tau_i(z)]\!]_v^w. \tag{3.14}$$

To get a more explicit formula we need to determine $V(v), V(w)$, and $V\cdot\nu_i$ on $\mathcal{A}_i$. For this we apply Theorem 3 to two different cases. On the one hand, if $v\in\partial B(x_i,r)\cap\partial B(x_\ell,r)$ for some $i\neq\ell$, then applying (3.13) we obtain

$$V(v)\cdot\tau_i(v)=\delta r\frac{1-\nu_\ell(v)\cdot\nu_i(v)}{\nu_\ell(v)\cdot\tau_i(v)}. \tag{3.15}$$

On the other hand, if $v\in\partial B(x_i,r)\cap\partial A$, then applying (3.12) we get

$$V(v)\cdot\tau_i(v)=-\delta r\frac{\nu_A(v)\cdot\nu_i(v)}{\tau_i(v)\cdot\nu_A(v)}. \tag{3.16}$$

Then, recalling that $L(z)=\{\ell\in\{1,\dots,m\}\setminus\{i\}\mid z\in\partial B(x_\ell,r)\}$ for $z\in\{v,w\}$, and that $\nu_{-i}(z):=\nu_\ell(z)$ if $z\in\partial B(x_i,r)\cap\partial B(x_\ell,r)$, $\ell\neq i$, and $\nu_{-i}(z):=\nu_A(z)$ if $z\in\partial B(x_i,r)\cap\partial A$, (3.15) and (3.16) can be merged into a single formula:

$$V(v)\cdot\tau_i(v)=\delta r\frac{|L(v)|-\nu_{-i}(v)\cdot\nu_i(v)}{\nu_{-i}(v)\cdot\tau_i(v)}. \tag{3.17}$$

We obtain a similar formula for $V(w)\cdot\tau_i(w)$. Gathering these results we get

$$\begin{aligned}\frac{d}{dt}\partial_r G(\boldsymbol{x},r+t\delta r)\bigg|_{t=0}&=-\delta r\frac{\operatorname{Per}(\partial\Omega(\boldsymbol{x},r)\cap A)}{r}\\&\quad-\delta r\sum_{i=1}^m\sum_{(v,w)\in\mathbb{A}_i}\left[\!\!\left[\frac{|L(z)|-\nu_{-i}(z)\cdot\nu_i(z)}{\nu_{-i}(z)\cdot\tau_i(z)}\right]\!\!\right]_v^w,\end{aligned}$$

where we have used $\mathcal{H}V\cdot\nu_i=\frac{\delta r}{r}$ on $\mathcal{A}_i\subset\partial B(x_i,r)$ due to (3.11) and $\mathcal{H}=1/r$. This yields (2.7).

3.4. Second-order derivative of G with respect to the centers. We have $\partial_{x_i}G(\boldsymbol{x},r)=-\int_{\mathcal{A}_i}\nu_i(z)\,dz$; see (2.3) and [7, section 3.4] for the detailed calculation. As in [7], the calculation is achieved through integration by substitution using the

mapping T_t from Theorem 2 with the specific perturbation $\delta\boldsymbol{x} = (0, \dots, 0, \delta x_i, 0, \dots, 0)$, which requires that Assumptions 1 and 2 hold. Using Lemma 3.2 yields

$$\begin{aligned}\partial_{x_i} G(\boldsymbol{x} + t\delta\boldsymbol{x}, r) &= -\int_{\partial B(x_i + t\delta x_i, r) \cap \partial\Omega(\boldsymbol{x}+t\delta\boldsymbol{x}, r) \cap A} \nu_t(z)\, dz = -\int_{T_t(\mathcal{A}_i)} \nu_t(z)\, dz \\ &= -\int_{\mathcal{A}_i} \nu_t(T_t(z))\omega_t(z)\, dz,\end{aligned}$$

where ν_t is the outward unit normal vector to $\partial B(x_i + t\delta x_i, r) \cap \partial\Omega(\boldsymbol{x} + t\delta\boldsymbol{x}, r) \cap A$ and ω_t is given by (3.1).

To obtain the derivative of $\partial_{x_i} G(\boldsymbol{x} + t\delta\boldsymbol{x}, r)$ with respect to t at $t = 0$ we need the so-called *material derivative* of the normal vector given by

$$\frac{d}{dt}\nu_t(T_t(z))|_{t=0} = -(D_\Gamma V)^\top \nu_i \text{ on } \mathcal{A}_i,$$

with $V := \partial_t T_t|_{t=0}$; see [39, Lemma 5.5, page 99], and $D_\Gamma V := DV - (DV)\nu_i \otimes \nu_i$ denotes the tangential Jacobian of V on $\mathcal{A}_i$. Then, using (3.2) we obtain

$$\frac{d}{dt}\partial_{x_i} G(\boldsymbol{x} + t\delta\boldsymbol{x}, r)\bigg|_{t=0} = -\int_{\mathcal{A}_i} -(D_\Gamma V)^\top \nu_i + \nu_i \operatorname{div}_\Gamma(V)\, dz.$$

This expression can be further transformed using the following properties of tensor calculus:

$$\operatorname{div}_\Gamma(\nu_i \otimes V) = \operatorname{div}_\Gamma(V)\nu_i + (D_\Gamma \nu_i)V \quad \text{and} \quad \nabla_\Gamma(V \cdot \nu_i) = D_\Gamma \nu_i^\top V + D_\Gamma V^\top \nu_i \text{ on } \mathcal{A}_i. \tag{3.18}$$

We show that $D_\Gamma \nu_i^\top V = D_\Gamma \nu_i V$ on $\mathcal{A}_i$. Indeed, let $W \in \mathbb{R}^2$ and denote V_τ and W_τ to be the tangential components of V and W on $\mathcal{A}_i$. Differentiating $\nu_i \cdot \nu_i = 1$ on $\mathcal{A}_i$, we get $(D_\Gamma \nu_i)^\top \nu_i = 0$ and then

$$(D_\Gamma \nu_i)^\top V \cdot W = (D_\Gamma \nu_i)^\top V_\tau \cdot W = (D_\Gamma \nu_i)^\top V_\tau \cdot W_\tau = (D_\Gamma \nu_i) V_\tau \cdot W_\tau,$$

using the fact that the second fundamental form $(V_\tau, W_\tau) \mapsto (D_\Gamma \nu_i) V_\tau \cdot W_\tau$ is symmetric. Further,

$$(D_\Gamma \nu_i)^\top V \cdot W = (D_\Gamma \nu_i)^\top W_\tau \cdot V_\tau = (D_\Gamma \nu_i)^\top W \cdot V = (D_\Gamma \nu_i) V \cdot W \text{ on } \mathcal{A}_i. \tag{3.19}$$

Now, using (3.18) and (3.19) we obtain

$$\frac{d}{dt}\partial_{x_i} G(\boldsymbol{x} + t\delta\boldsymbol{x}, r)\bigg|_{t=0} = -\int_{\mathcal{A}_i} \operatorname{div}_\Gamma(\nu_i \otimes V) - \nabla_\Gamma(V \cdot \nu_i)\, dz.$$

Applying Theorem 1 to the integral of $\operatorname{div}_\Gamma(\nu_i \otimes V)$ on each arc in $\mathcal{A}_i$ we get

$$\frac{d}{dt}\partial_{x_i} G(\boldsymbol{x} + t\delta\boldsymbol{x}, r)\bigg|_{t=0} = -\int_{\mathcal{A}_i} \mathcal{H}(\nu_i \otimes V) \cdot \nu_i - \nabla_\Gamma(V \cdot \nu_i)\, dz - \sum_{(v,w) \in \mathbb{A}_i} [\![(\nu_i(z) \otimes V(z)) \cdot \tau_i(z)]\!]_v^w,$$

and then, using $\mathcal{H} = 1/r$ on $\mathcal{A}_i$,

$$\frac{d}{dt}\partial_{x_i} G(\boldsymbol{x} + t\delta\boldsymbol{x}, r)\bigg|_{t=0} = -\frac{1}{r}\int_{\mathcal{A}_i} (V \cdot \nu_i)\nu_i - r\nabla_\Gamma(V \cdot \nu_i)\, dz - \sum_{(v,w) \in \mathbb{A}_i} [\![(V(z) \cdot \tau_i(z))\nu_i(z)]\!]_v^w. \tag{3.20}$$

Considering that $\delta x_\ell = 0$ for $\ell \neq i$ since we use the specific perturbation $\delta \boldsymbol{x} = (0, \dots, 0, \delta x_i, 0, \dots, 0)$, (3.5) and (3.6) actually provide the same formula in this particular case:

$$V(z) \cdot \tau_i(z) = \delta x_i \cdot \left(\tau_i - \frac{\nu_{-i}}{\tau_i \cdot \nu_{-i}} \right)(z) = - \left(\frac{\nu_{-i} \cdot \nu_i}{\nu_{-i} \cdot \tau_i} \delta x_i \cdot \nu_i \right)(z) \tag{3.21}$$

for $z \in \{v, w\}$ and $(v, w) \in \mathbb{A}_i$. We also have $\nabla_\Gamma(V \cdot \nu_i) = \nabla_\Gamma(\delta x_i \cdot \nu) = (D_\Gamma \nu)^\top \delta x_i$ and

$$D_\Gamma \nu_i = D_\Gamma \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix} = \begin{pmatrix} \nabla_\Gamma(\cos\theta)^\top \\ \nabla_\Gamma(\sin\theta)^\top \end{pmatrix} = \frac{1}{r} \begin{pmatrix} \partial_\theta(\cos\theta)\tau_i^\top \\ \partial_\theta(\sin\theta)\tau_i^\top \end{pmatrix} = \frac{1}{r} \tau_i \otimes \tau_i \text{ on } \mathcal{A}_i.$$

Applying (3.4) yields $V \cdot \nu_i = \delta x_i \cdot \nu_i$ on $\mathcal{A}_i$. Gathering these results we get

$$\frac{d}{dt} \partial_{x_i} G(\boldsymbol{x} + t\delta\boldsymbol{x}, r)\Big|_{t=0} = -\frac{1}{r} \int_{\mathcal{A}_i} (\delta x_i \cdot \nu_i)\nu_i - (\tau_i \otimes \tau_i)\delta x_i \, dz + \sum_{(v,w) \in \mathbb{A}_i} \left[\!\!\left[\frac{\nu_{-i} \cdot \nu_i}{\nu_{-i} \cdot \tau_i} \nu_i \otimes \nu_i \right]\!\!\right]_v^w \delta x_i,$$

which yields (2.8).

3.5. Second-order derivative with respect to x_i and x_ℓ of G. As in section 3.4 we use the mapping T_t from Theorem 2, which requires that Assumptions 1 and 2 hold, but now with the specific perturbation $\delta \boldsymbol{x} = (0, \dots, 0, \delta x_\ell, 0, \dots, 0)$. This yields a transformation T_t satisfying, in particular, $T_t(\mathcal{A}_i) = \partial B(x_i, r) \cap \partial\Omega(\boldsymbol{x} + t\delta\boldsymbol{x}, r) \cap A$. Then, using Lemma 3.2 we obtain

$$\begin{aligned} \partial_{x_i} G(\boldsymbol{x} + t\delta\boldsymbol{x}, r) &= - \int_{\partial B(x_i, r) \cap \partial\Omega(\boldsymbol{x} + t\delta\boldsymbol{x}, r) \cap A} \nu_t(z) \, dz = - \int_{T_t(\mathcal{A}_i)} \nu_t(z) \, dz \\ &= - \int_{\mathcal{A}_i} \nu_t(T_t(z)) \omega_t(z) \, dz, \end{aligned}$$

where ν_t is the outward unit normal vector to $\partial B(x_i, r) \cap \partial\Omega(\boldsymbol{x} + t\delta\boldsymbol{x}, r) \cap A$ and ω_t is given by (3.1). Applying (3.4) and considering that $\delta x_i = 0$ since we are using the specific perturbation $\delta \boldsymbol{x} = (0, \dots, 0, \delta x_\ell, 0, \dots, 0)$, we get $V \cdot \nu_i = 0$ on $\mathcal{A}_i$. Then, applying (3.6) with $\delta x_i = 0$ we get

$$V(z) \cdot \tau_i(z) = \frac{\delta x_\ell \cdot \nu_\ell(z)}{\tau_i(z) \cdot \nu_\ell(z)} \quad \text{if } z \in \partial B(x_\ell, r) \cap \partial B(x_i, r),\ i \neq \ell. \tag{3.22}$$

Next, the derivative of $\partial_{x_i} G(\boldsymbol{x} + t\delta\boldsymbol{x}, r)$ with respect to t at $t = 0$ is already calculated in (3.20), but the terms $(V \cdot \nu_i)\nu_i$ and $\nabla_\Gamma(V \cdot \nu_i)$ in (3.20) vanish due to $V \cdot \nu_i = 0$ on $\mathcal{A}_i$. We also observe that $V(z) = 0$ if $z \in \{v, w\}$ with $(v, w) \in \mathbb{A}_i$ and $z \notin \partial B(x_\ell, r)$. Finally, using (3.22), $\mathcal{I}_{i\ell} = \{v \in \partial B(x_\ell, r) \mid (v, \cdot) \in \mathbb{A}_i\}$ and $\mathcal{O}_{i\ell} = \{w \in \partial B(x_\ell, r) \mid (\cdot, w) \in \mathbb{A}_i\}$ we get

$$\begin{aligned} \frac{d}{dt} \partial_{x_i} G(\boldsymbol{x} + t\delta\boldsymbol{x}, r)\Big|_{t=0} &= \sum_{v \in \mathcal{I}_{i\ell}} V(v) \cdot \tau_i(v) \nu_i(v) - \sum_{w \in \mathcal{O}_{i\ell}} V(w) \cdot \tau_i(w) \nu_i(w) \\ &= \left[\sum_{v \in \mathcal{I}_{i\ell}} \frac{\nu_i(v) \otimes \nu_\ell(v)}{\nu_\ell(v) \cdot \tau_i(v)} - \sum_{w \in \mathcal{O}_{i\ell}} \frac{\nu_i(w) \otimes \nu_\ell(w)}{\nu_\ell(w) \cdot \tau_i(w)} \right] \delta x_\ell, \end{aligned}$$

which yields (2.9).

3.6. Second-order derivative with respect to x_i and r of G. In a similar way as in sections 3.4 and 3.5, we use the mapping T_t from Theorem 2 with the perturbation $\delta\boldsymbol{x} = (0, \dots, 0, \delta x_i, 0, \dots, 0)$. This yields, using Lemma 3.2,

$$\partial_r G(\boldsymbol{x} + t\delta\boldsymbol{x}, r) = -\int_{\partial\Omega(\boldsymbol{x}+t\delta\boldsymbol{x},r)\cap A} dz = -\int_{T_t(\partial\Omega(\boldsymbol{x},r)\cap A)} dz = -\int_{\partial\Omega(\boldsymbol{x},r)\cap A} \omega_t(z)\, dz.$$

Proceeding as in the calculation leading to (3.14), we get

$$\frac{d}{dt}\partial_r G(\boldsymbol{x} + t\delta\boldsymbol{x}, r)\bigg|_{t=0} = -\sum_{\ell=1}^{m}\int_{\mathcal{A}_\ell} \mathcal{H}V\cdot\nu_\ell\, dz - \sum_{\ell=1}^{m}\sum_{(v,w)\in\mathbb{A}_\ell} [\![V(z)\cdot\tau_\ell(z)]\!]_v^w. \tag{3.23}$$

Considering that $\delta x_\ell = 0$ for $\ell \neq i$, since $\delta\boldsymbol{x} = (0, \dots, 0, \delta x_i, 0, \dots, 0)$, (3.5) and (3.6) actually provide the same formula in this particular case:

$$V(z)\cdot\tau_i(z) = \delta x_i \cdot \left(\tau_i - \frac{\nu_{-i}}{\tau_i\cdot\nu_{-i}}\right)(z) = -\left(\frac{\nu_{-i}\cdot\nu_i}{\nu_{-i}\cdot\tau_i}\delta x_i\cdot\nu_i\right)(z), \tag{3.24}$$

for $z \in \{v, w\}$ and $(v, w) \in \mathbb{A}_i$, and also

$$\begin{aligned} V(z)\cdot\tau_\ell(z) &= \delta x_i\cdot\left(\tau_\ell - \frac{\nu_\ell}{\tau_i\cdot\nu_\ell}(\tau_i\cdot\tau_\ell)\right)(z) \\ &= \delta x_i\cdot\left(\frac{\mu}{\tau_i\cdot\nu_\ell}\right)(z) \text{ if } z\in\partial B(x_i,r)\cap\partial B(x_\ell,r) \text{ and } \ell\neq i, \end{aligned}$$

with $\mu := \tau_\ell(\tau_i\cdot\nu_\ell) - (\tau_i\cdot\tau_\ell)\nu_\ell$. This yields $\mu\cdot\tau_i = 0$ and

$$\mu\cdot\nu_i = (\tau_\ell\cdot\nu_i)(\tau_i\cdot\nu_\ell) - (\tau_i\cdot\tau_\ell)(\nu_\ell\cdot\nu_i) = -(\tau_i\cdot\nu_\ell)^2 - (\tau_i\cdot\tau_\ell)^2 = -1,$$

where we have used $\tau_\ell\cdot\nu_i = -\tau_i\cdot\nu_\ell$ and $\tau_i\cdot\tau_\ell = \nu_\ell\cdot\nu_i$. Thus $\mu = -\nu_i$ and we get

$$V(z)\cdot\tau_\ell(z) = -\frac{\delta x_i\cdot\nu_i(z)}{\tau_i(z)\cdot\nu_\ell(z)} \text{ if } z\in\partial B(x_i,r)\cap\partial B(x_\ell,r) \text{ and } \ell\neq i. \tag{3.25}$$

In (3.23), we observe that $V(z) = 0$ whenever $z \in \{v, w\}$ and $z \notin \partial B(x_i, r)$; this can be seen from (3.5)–(3.6) and the fact that we use $\delta\boldsymbol{x} = (0, \dots, 0, \delta x_i, 0, \dots, 0)$. Hence, recalling that $L(z) = \{\ell \in \{1, \dots, m\} \setminus \{i\} \mid z \in \partial B(x_\ell, r)\}$,

$$\begin{aligned} \sum_{\ell=1}^{m}\sum_{(v,w)\in\mathbb{A}_\ell} [\![V(z)\cdot\tau_\ell(z)]\!]_v^w &= \sum_{(v,w)\in\mathbb{A}_i} [\![V(z)\cdot\tau_i(z)]\!]_v^w + \sum_{\substack{\ell=1\\ \ell\neq i}}^{m}\sum_{(v,w)\in\mathbb{A}_\ell} [\![V(z)\cdot\tau_\ell(z)]\!]_v^w \\ &= \sum_{(v,w)\in\mathbb{A}_i} [\![V(z)\cdot\tau_i(z)]\!]_v^w - \sum_{(v,w)\in\mathbb{A}_i}\left[\!\!\left[\sum_{\ell\in L(z)} V(z)\cdot\tau_\ell(z)\right]\!\!\right]_v^w. \end{aligned}$$

Note that the negative sign in front of the last sum is due to the fact that if an ending point of an arc in $\mathbb{A}_\ell$ belongs to some arc in $\mathbb{A}_i$, then it is a starting point for this arc in $\mathbb{A}_i$, and vice versa. Using (3.4) we have $V\cdot\nu_\ell \equiv 0$ on $\mathcal{A}_\ell$ for all $\ell \neq i$. Since $\mathcal{H} = 1/r$, we may write (3.23) as

$$\frac{d}{dt}\partial_r G(\boldsymbol{x} + t\delta\boldsymbol{x}, r)\bigg|_{t=0} = -\frac{1}{r}\int_{\mathcal{A}_i} V\cdot\nu_i\, dz - \sum_{(v,w)\in\mathbb{A}_i}\left[\!\!\left[V(z)\cdot\tau_i(z) - \sum_{\ell\in L(z)} V(z)\cdot\tau_\ell(z)\right]\!\!\right]_v^w.$$

Using (3.4) we get $V \cdot \nu_i = \delta x_i \cdot \nu_i$ on $\mathcal{A}_i$. Finally, using (3.24)–(3.25) we get

$$\frac{d}{dt}\partial_r G(\boldsymbol{x}+t\delta\boldsymbol{x},r)\bigg|_{t=0} = -\frac{1}{r}\int_{\mathcal{A}_i} \delta x_i \cdot \nu_i \, dz + \sum_{(v,w)\in\mathbb{A}_i} \left[\!\!\left[\frac{\nu_{-i}(z)\cdot\nu_i(z)}{\nu_{-i}(z)\cdot\tau_i(z)}\delta x_i\cdot\nu_i(z) - \sum_{\ell\in L(z)} \frac{\delta x_i\cdot\nu_i(z)}{\tau_i(z)\cdot\nu_\ell(z)} \right]\!\!\right]_v^w,$$

which yields (2.10).

4. Exact calculation of G and its derivatives. In this section, we consider that $A = \cup_{j=1}^p A_j$ and $\{A_j\}_{j=1}^p$ are nonoverlapping convex polygons. (If not available, such decomposition can be computed in $\mathcal{O}(e_A + \bar{e}_A)$, where e_A is the number of vertices of A and $\bar{e}_A$ is its number of notches; see, for example, [25] and the references therein.) The key ingredient for the exact computation of G, ∇G, and $\nabla^2 G$ as stated in section 2 is to consider partitions

$$A_j \cap \Omega(\boldsymbol{x},r) = \bigcup_{i\in\mathcal{K}_{A_j}} S_{ij}, \quad j=1,\dots,p, \tag{4.1}$$

where $\mathcal{K}_{A_j} \subseteq \{i \in \{1,\dots,m\} \mid B(x_i,r)\cap A_j \neq \emptyset\}$ for $j=1,\dots,p$ and each S_{ij} is such that ∂S_{ij} is a simple and convex curve given by the union of segments and arcs of the circle $\partial B(x_i,r)$. It is worth noticing that, since $A_1, A_2, \dots, A_p$ are disjoint, then $\{S_{ij}\}_{(i,j)\in\mathcal{K}}$ with $\mathcal{K} = \{(i,j) \mid j \in \{1,\dots,p\} \text{ and } i \in \mathcal{K}_{A_j}\}$ is a partition of $A\cap\Omega(\boldsymbol{x},r)$, i.e.,

$$A \cap \Omega(\boldsymbol{x},r) = \bigcup_{(i,j)\in\mathcal{K}} S_{ij}; \tag{4.2}$$

see Figure 4. Note that 8 out of the 10 balls intersect with either A_1 or A_2 in Figure 4. Let us number the balls intersecting only A_1 from 1 to 5 and the balls intersecting only A_2 from 8 to 10. Thus, balls 1 to 5 contribute to (4.1) with $S_{11}, S_{21}, \dots, S_{51}$, i.e., they contribute to the partition of A_1 only; while balls 8, 9, and 10 contribute with S_{82}, S_{92}, and $S_{10,2}$, i.e., they contribute to the partition of A_2 only. Balls 6 and 7 intersect both A_1 and A_2 and contribute with S_{61} and S_{71} to the partition of A_1 and with S_{62} and S_{72} to the partition of A_2. Therefore, we have $\mathcal{K}_{A_1} = \{1,2,3,4,5,6,7\}$, $\mathcal{K}_{A_2} = \{6,7,8,9,10\}$, and $\mathcal{K} = \{(1,1),(2,1),(3,1),(4,1),(5,1),(6,1),(6,2),(7,1),(7,2),(8,2),(9,2),(10,2)\}$. In addition, for further use, we define $\mathcal{K}_{B_1} = \mathcal{K}_{B_2} = \mathcal{K}_{B_3} = \mathcal{K}_{B_4} = \mathcal{K}_{B_5} = \{1\}$, $\mathcal{K}_{B_6} = \mathcal{K}_{B_7} = \{1,2\}$, $\mathcal{K}_{B_8} = \mathcal{K}_{B_9} = \mathcal{K}_{B_{10}} = \{2\}$.

The computation of the partitions in (4.1) is based on Voronoi diagrams. For a given $(\boldsymbol{x},r)$, we first compute the Voronoi diagram with cells $\{V_i\}_{i=1}^m$ associated with the balls centers $x_1,\dots,x_m$. Each cell V_i is a (bounded or unbounded) polyhedron given by the points $y \in \mathbb{R}^2$ such that $\|y-x_i\| = \min_{\{\ell=1,\dots,m\}}\{\|y-x_\ell\|\}$. Then, for each $j=1,\dots,p$ and each $i=1,\dots,m$, we compute the convex polygons $W_{ij} = A_j \cap V_i$ and, in the sequence, $S_{ij} = W_{ij} \cap B(x_i,r)$. (Note that, by construction, $W_{ij}\cap B(x_i,r) = W_{ij}\cap\Omega(\boldsymbol{x},r)$; and so (4.1) and, in consequence, (4.2) hold.) In the construction process, we obtain the sets $\mathcal{K}_{A_j} = \{i \in \{1,\dots,m\} \mid S_{ij} \neq \emptyset\}$, $\mathcal{K}_{B_i} = \{j \in \{1,\dots,p\} \mid S_{ij} \neq \emptyset\}$, and $\mathcal{K}$ such that $(i,j) \in \mathcal{K}$ if and only if $S_{ij} \neq \emptyset$. Let $\mathcal{V}(S_{ij})$ be the set of vertices of S_{ij}, let $\mathcal{A}(S_{ij}) = \partial S_{ij} \cap \partial B(x_i,r)$ be the union of the arcs in ∂S_{ij}, and let $\mathcal{E}(S_{ij}) = \partial S_{ij} \setminus \mathcal{A}(S_{ij})$ be the union of the edges in ∂S_{ij}. Moreover, we associate with $\mathcal{E}(S_{ij})$ and $\mathcal{A}(S_{ij})$ the corresponding sets of maximal

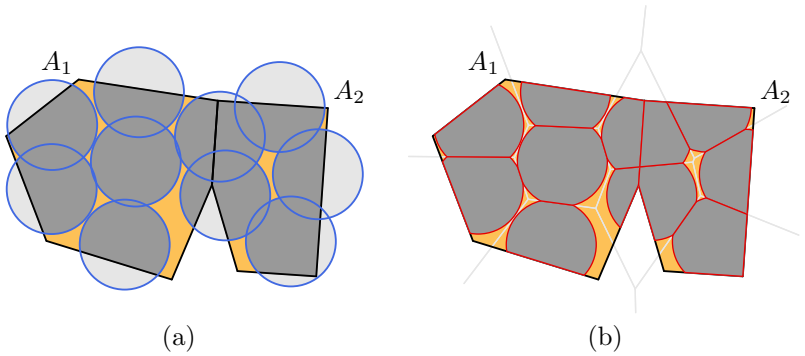


FIG. 4. *Figure* (a) *represents a region* A *to be covered given by* $A = \cup_{j=1}^{p} A_j$ *with* $p = 2$ *and an arbitrary configuration of balls* $\Omega(\boldsymbol{x}, r) = \cup_{i=1}^{m} B(x_i, r)$ *with* $m = 10$. *Figure* (b) *represents the partitions of* $A_1 \cap \Omega(\boldsymbol{x}, r)$ *and* $A_2 \cap \Omega(\boldsymbol{x}, r)$ *defined in* (4.1) *that, together, as expressed in* (4.2), *represent a partition of* $A \cap \Omega(\boldsymbol{x}, r)$. *In* (b), *the Voronoi diagram that allows the partitions to be computed is depicted.*

arcs $\mathbb{A}(S_{ij})$ and edges $\mathbb{E}(S_{ij})$. Strictly speaking, these are sets of pairs of points representing arcs and edges, respectively. Each edge is represented by a pair $[v, w]$ of vertices in counterclockwise order and each arc is represented by a pair (v, w) of vertices, in counterclockwise order, which unequivocally determines two angles. For each vertex $z \in \mathcal{V}(S_{ij})$, we save whether or not $z \in \partial A$. If $z \in \partial A$, then we save, whenever it exists, the unitary (Euclidean) norm outward normal vector to ∂A at z, named $\nu_A(z)$. Additionally, for each vertex $z \in \mathcal{V}(S_{ij})$, we save the set of indices $L(z) \subseteq \{1, \ldots, m\} \setminus \{i\}$ such that $z \in \partial B(x_\ell, r)$ for all $\ell \in L(z)$.

Each set $\mathcal{A}_i = \partial B(x_i, r) \cap \partial\Omega(\boldsymbol{x}, r) \cap A$ for $i = 1, \ldots, m$, defined in (2.4), corresponds to the union of the arcs in ∂S_{ij} for all $j \in \mathcal{K}_{B_i}$, i.e., it holds that

$$\mathcal{A}_i = \bigcup_{j \in \mathcal{K}_{B_i}} \mathcal{A}(S_{ij}) \tag{4.3}$$

for $i = 1, \ldots, m$. It is worth noticing (4.3) does not mean that every arc in $\mathbb{A}_i$ belongs to $\mathbb{A}(S_{ij})$ for some $j \in \mathcal{K}_{B_i}$ nor that $|\mathbb{A}_i| = \sum_{j \in \mathcal{K}_{B_i}} |\mathbb{A}(S_{ij})|$. Indeed, if z is an extremity of an arc in $\mathbb{A}(S_{ij})$, then either $z \in \partial B(x_\ell, r)$ for some $\ell \neq i$ or $z \in \partial A_j$. In the case $z \in \partial A_j$, it may happen that $z \notin \partial A$, and consequently z is not an extremity of an arc in $\mathbb{A}_i$. To construct $\mathbb{A}_i$, consecutive arcs (arcs with an extreme in common) in $\cup_{j \in \mathcal{K}_{B_i}} \mathbb{A}(S_{ij})$ must be merged into a single arc. So, what holds is that each arc in $\mathbb{A}_i$ belongs to $\mathbb{A}(S_{ij})$ for some $j \in \mathcal{K}_{B_i}$ or is the union of two or more consecutive arcs in $\cup_{j \in \mathcal{K}_{B_i}} \mathbb{A}(S_{ij})$. Thus $|\mathbb{A}_i| \leq \sum_{j \in \mathcal{K}_{B_i}} |\mathbb{A}(S_{ij})|$. The particular case $\mathcal{A}_i = \partial B(x_i, r)$ is considered separately; in this case, we set $\mathbb{A}_i = \emptyset$ and Circle$(\mathbb{A}_i)$ equal to true.

In a similar way, we also define

$$\mathcal{E}_i = \bigcup_{j \in \mathcal{K}_{B_i}} \mathcal{E}(S_{ij}), \tag{4.4}$$

and the associated set $\mathbb{E}_i$ of pairs $[v, w]$ representing edges, for $i = 1, \ldots, m$. These sets of edges play a role in the computation of G only. Thus, while the same principle of merging consecutive edges could be applied, it has no practical relevance because, one way or the other, the same result is obtained.

A second ingredient for the exact computation of G and its derivatives are the parameterizations

$$t \mapsto \begin{pmatrix} x_{\mathcal{E}}(t) \\ y_{\mathcal{E}}(t) \end{pmatrix} = v + t(w - v), \quad t \in [0, 1], \tag{4.5}$$

of each edge represented by $[v, w] \in \cup_{i=1}^{m} \mathbb{E}_i$, and the parameterizations

$$\theta \mapsto \begin{pmatrix} x_{\mathcal{A}}(\theta) \\ y_{\mathcal{A}}(\theta) \end{pmatrix} = x_i + r \begin{pmatrix} \cos\theta \\ \sin\theta \end{pmatrix}, \quad \theta \in [\theta_v, \theta_w], \tag{4.6}$$

of each arc represented by $(v, w) \in \mathbb{A}_i$ for $i = 1, \ldots, m$, where θ_v and θ_w are the angular coordinates of $v - x_i$ and $w - x_i$, respectively.

We are now ready to compute G and its derivatives. By (2.2),

$$G(\boldsymbol{x}, r) = \mathrm{Vol}(A) - \mathrm{Vol}(A \cap \Omega(\boldsymbol{x}, r)) = \mathrm{Vol}(A) - \sum_{(i,j)\in\mathcal{K}} \mathrm{Vol}(S_{ij}). \tag{4.7}$$

By Green's theorem,

$$\mathrm{Vol}(S_{ij}) = \int_{S_{ij}} dxdy = \int_{\partial S_{ij}} x\, dy = \sum_{[v,w]\in\mathbb{E}(S_{ij})} \int_0^1 x_{\mathcal{E}}(t)\, dy_{\mathcal{E}}(t) + \sum_{(v,w)\in\mathbb{A}(S_{ij})} \int_{\theta_v}^{\theta_w} x_{\mathcal{A}}(\theta)\, dy_{\mathcal{A}}(\theta) \tag{4.8}$$

for all $(i, j) \in \mathcal{K}$, while, by (4.5),

$$\int_0^1 x_{\mathcal{E}}(t)\, dy_{\mathcal{E}}(t) = \frac{(v_1 + w_1)(w_2 - v_2)}{2} \tag{4.9}$$

for all $[v, w] \in \mathbb{E}(S_{ij})$ and all $(i, j) \in \mathcal{K}$, and, by (4.6),

$$\begin{aligned} \int_{\theta_v}^{\theta_w} x_{\mathcal{A}}(\theta)\, dy_{\mathcal{A}}(\theta) \;&=\; (x_i)_1\, r\,(\sin\theta_w - \sin\theta_v) \\ &+\; \frac{r^2}{2}\,(\theta_w - \theta_v + \sin\theta_w\cos\theta_w - \sin\theta_v\cos\theta_v) \end{aligned} \tag{4.10}$$

for all $(v, w) \in \mathbb{A}(S_{ij})$ and all $(i, j) \in \mathcal{K}$. The computation of G as defined in (2.2) using (4.7)–(4.10) is summarized in Algorithm 4.1.

For computing $\nabla G(\boldsymbol{x}, r) = (\partial_{x_1} G(\boldsymbol{x}, r), \ldots, \partial_{x_m} G(\boldsymbol{x}, r), \partial_r G(\boldsymbol{x}, r))^\top$, by (2.3) and (2.4), we have that

$$\begin{aligned} \partial_{x_i} G(\boldsymbol{x}, r) \;&=\; -\int_{\mathcal{A}_i} \nu_i(z)\, dz = -\sum_{(v,w)\in\mathbb{A}_i} \int_{\theta_v}^{\theta_w} r\,(\cos\theta, \sin\theta)^\top\, d\theta \\ &=\; \sum_{(v,w)\in\mathbb{A}_i} r\,(\sin\theta_v - \sin\theta_w, \cos\theta_w - \cos\theta_v)^\top \end{aligned} \tag{4.11}$$

for $i = 1, \ldots, m$, and, by (2.3) and (2.6), we have that

$$\partial_r G(\boldsymbol{x}, r) = -\int_{\cup_{i=1}^m \mathcal{A}_i} dz = -\sum_{(v,w)\in\cup_{i=1}^m \mathbb{A}_i} \int_{\theta_v}^{\theta_w} r\, d\theta = -\sum_{(v,w)\in\cup_{i=1}^m \mathbb{A}_i} r\,(\theta_w - \theta_v). \tag{4.12}$$

The computation of ∇G as defined in (2.3) using (4.11) and (4.12) is summarized in Algorithm 4.2.

Algorithm 4.1 COMPUTES $G(\boldsymbol{x}, r)$.

Input: $\mathrm{Vol}(A)$, $(\boldsymbol{x}, r)$, and sets $\{\mathbb{E}_i, \mathbb{A}_i\}_{i=1}^m$.
Output: $G(\boldsymbol{x}, r)$.
$\gamma \leftarrow 0$
for $i = 1, \dots, m$ **do**
  if Circle($\mathbb{A}_i$) **then**
    $\gamma \leftarrow \gamma + \pi r^2$
  else
    foreach $[v, w] \in \mathbb{E}_i$ **do**
      $\gamma \leftarrow \gamma + \frac{1}{2}(v_1 + w_1)(w_2 - v_2)$
    foreach $(v, w) \in \mathbb{A}_i$ **do**
      $\gamma \leftarrow \gamma + (x_i)_1 r(\sin\theta_w - \sin\theta_v) + \frac{r^2}{2}(\theta_w - \theta_v + \sin\theta_w \cos\theta_w - \sin\theta_v \cos\theta_v)$
return $G = \mathrm{Vol}(A) - \gamma$

Algorithm 4.2 COMPUTES $\nabla G(\boldsymbol{x}, r)$.

Input: $(\boldsymbol{x}, r)$, and sets $\{\mathbb{A}_i\}_{i=1}^m$.
Output: $\nabla G(\boldsymbol{x}, r)$.
$g_r \leftarrow 0$ and $g_{x_i} \leftarrow (0, 0)^\top$ for $i = 1, \dots, m$.
for $i = 1, \dots, m$ **do**
  if Circle($\mathbb{A}_i$) **then**
    $g_r \leftarrow g_r - 2\pi r$
  else
    foreach $(v, w) \in \mathbb{A}_i$ **do**
      $g_r \leftarrow g_r - r(\theta_w - \theta_v)$
      $g_{x_i} \leftarrow g_{x_i} + r(\sin\theta_v - \sin\theta_w, \cos\theta_w - \cos\theta_v)^\top$
return $\nabla G(\boldsymbol{x}, r) = (g_r, g_{x_1}^\top, \dots, g_{x_m}^\top)^\top$

For computing $\nabla^2 G$, we use that, in (2.7),

$$-\frac{\mathrm{Per}(\partial\Omega(\boldsymbol{x}, r) \cap A)}{r} = \sum_{(v,w) \in \cup_{i=1}^m \mathbb{A}_i} (\theta_v - \theta_w), \tag{4.13}$$

in (2.8),

$$\begin{aligned}
&\frac{1}{r}\int_{\mathcal{A}_i} -\nu_i(z) \otimes \nu_i(z) + \tau_i(z) \otimes \tau_i(z)\, dz \\
&= \sum_{(v,w) \in \mathbb{A}_i} \int_{\theta_v}^{\theta_w} -\begin{pmatrix} (\cos\theta)^2 & \sin\theta\cos\theta \\ \sin\theta\cos\theta & (\sin\theta)^2 \end{pmatrix} + \begin{pmatrix} (\sin\theta)^2 & -\sin\theta\cos\theta \\ -\sin\theta\cos\theta & (\cos\theta)^2 \end{pmatrix} d\theta \\
&= \sum_{(v,w) \in \mathbb{A}_i} \begin{pmatrix} \sin(\theta_v - \theta_w)\cos(\theta_v + \theta_w) & (\cos\theta_w)^2 - (\cos\theta_v)^2 \\ (\cos\theta_w)^2 - (\cos\theta_v)^2 & \sin(\theta_w - \theta_v)\cos(\theta_v + \theta_w) \end{pmatrix},
\end{aligned} \tag{4.14}$$

and, in (2.10),

$$-\frac{1}{r}\int_{\mathcal{A}_i} \nu_i(z)\, dz = \sum_{(v,w) \in \mathbb{A}_i} (\sin\theta_v - \sin\theta_w, \cos\theta_w - \cos\theta_v)^\top. \tag{4.15}$$

Recall that in (2.7), (2.8), and (2.10), for $z \in \partial B(x_i, r)$, $\nu_{-i}(z)$ represents the unitary-norm outwards normal vector to the set intersecting $\partial B(x_i, r)$ at z. If this set is ∂A, then $\nu_{-i}(z) = \nu_A(z)$. If this set is $\partial B(x_\ell, r)$ for some $\ell \in L(z)$, then $\nu_{-i}(z) = \nu_\ell(z) = (\cos\vartheta_z, \sin\vartheta_z)^\top$, where ϑ_z is the angular coordinate of $z - x_\ell$. With these definitions and substituting (4.13), (4.14), and (4.15) in (2.7), (2.8), (2.9), and (2.10), we arrive at Algorithm 4.3.

Algorithm 4.3 COMPUTES $\nabla^2 G(\boldsymbol{x}, r)$.

Input: $(\boldsymbol{x}, r)$ and sets $\{\mathbb{A}_i\}_{i=1}^m$.
Output: The lower triangle of $H = \nabla^2 G(\boldsymbol{x}, r) \in \mathbb{R}^{2m+1,2m+1}$.

$H \leftarrow 0$.
for $i = 1, \dots, m$ **do**
  if Circle($\mathbb{A}_i$) **then**
    $h_{2m+1,2m+1} \leftarrow h_{2m+1,2m+1} - 2\pi$
  else
    foreach $(v, w) \in \mathbb{A}_i$ **do**
      let $a \odot b$ mean $a \leftarrow a + b$

$$\begin{pmatrix} h_{2i-1,2i-1} & \\ h_{2i,2i-1} & h_{2i,2i} \end{pmatrix} \odot \begin{pmatrix} \sin(\theta_v - \theta_w)\cos(\theta_v + \theta_w) & \\ (\cos\theta_w)^2 - (\cos\theta_v)^2 & \sin(\theta_w - \theta_v)\cos(\theta_v + \theta_w) \end{pmatrix}$$

$$(h_{2m+1,2i-1}, h_{2m+1,2i}) \odot (\sin\theta_v - \sin\theta_w, \cos\theta_w - \cos\theta_v)$$

$$h_{2m+1,2m+1} \odot \theta_v - \theta_w$$

      for $z \in \{v, w\}$ **do**
        if $z = v$ **then** let $a \odot b$ mean $a \leftarrow a - b$ **else** let $a \odot b$ mean $a \leftarrow a + b$
        if $z \in \partial A$ **then**

$$\alpha \leftarrow \left(-\nu_A(z) \cdot (\cos\theta_z, \sin\theta_z)^\top\right) / \left(\nu_A(z) \cdot (-\sin\theta_z, \cos\theta_z)^\top\right)$$

$$\begin{pmatrix} h_{2i-1,2i-1} & \\ h_{2i,2i-1} & h_{2i,2i} \end{pmatrix} \odot \alpha \begin{pmatrix} (\cos\theta_z)^2 & \\ \sin\theta_z\cos\theta_z & (\sin\theta_z)^2 \end{pmatrix}$$

$$(h_{2m+1,2i-1}, h_{2m+1,2i}) \odot \alpha(\cos\theta_z, \sin\theta_z)$$

$$h_{2m+1,2m+1} \odot \alpha$$

        foreach $\ell(z) \in L(z)$ **do**

$$\begin{pmatrix} h_{2i-1,2i-1} & \\ h_{2i,2i-1} & h_{2i,2i} \end{pmatrix} \odot \begin{pmatrix} \operatorname{cotan}(\vartheta_z - \theta_z)(\cos\theta_z)^2 & \\ \operatorname{cotan}(\vartheta_z - \theta_z)\sin\theta_z\cos\theta_z & \operatorname{cotan}(\vartheta_z - \theta_z)(\sin\theta_z)^2 \end{pmatrix}$$

$$h_{2m+1,2i-1} \odot \operatorname{cotan}(\vartheta_z - \theta_z)\cos\theta_z - \cos\theta_z / \sin(\vartheta_z - \theta_z)$$

$$h_{2m+1,2i} \odot \operatorname{cotan}(\vartheta_z - \theta_z)\sin\theta_z - \sin\theta_z / \sin(\vartheta_z - \theta_z)$$

$$h_{2m+1,2m+1} \odot (\cos(\vartheta_z - \theta_z) - 1) / \sin(\vartheta_z - \theta_z)$$

          if $\ell(z) > i$ **then**

$$\begin{pmatrix} h_{2\ell(z)-1,2i-1} & h_{2\ell(z)-1,2i} \\ h_{2\ell(z),2i-1} & h_{2\ell(z),2i} \end{pmatrix} \odot -(\sin(\vartheta_z - \theta_z))^{-1} \begin{pmatrix} \cos\theta_z\cos\vartheta_z & \sin\theta_z\cos\vartheta_z \\ \cos\theta_z\sin\vartheta_z & \sin\theta_z\sin\vartheta_z \end{pmatrix}$$

return H

Algorithms 4.1, 4.2, and 4.3 depend on the computation of sets $\mathbb{E}_i$ and $\mathbb{A}_i$ for $i = 1, \dots, m$. Computing these sets requires (a) computing the Voronoi diagram with cells $\{V_i\}_{i=1}^m$ associated with the balls' centers $x_1, \dots, x_m$ and (b) for each $i \in \{1, \dots, m\}$ and $j \in \{1, \dots, p\}$ computing $W_{ij} = V_i \cap A_j$ and $S_{ij} = W_{ij} \cap B(x_i, r)$. Computing the Voronoi diagram, using, for example, Fortune's algorithm [17], has known time complexity $\mathcal{O}(m \log m)$ [13, Lem. 7.9, p.158]. Since the intersection between a two-dimensional polyhedron defined by a half-planes and a convex polygon with b sides

can be computed in $\mathcal{O}(ab)$ [22], all W_{ij} can be computed in

$$\mathcal{O}\Big(\sum_{i=1}^{m}\sum_{j=1}^{p} e_{V_i} e_{A_j}\Big), \tag{4.16}$$

where e_{V_i} is the number of half-planes that define V_i, for $i = 1, \ldots, m$, and e_{A_j} is the number of sides of each A_j, for $j = 1, \ldots, p$. However, it is also known [13, Thm. 7.3, p. 150] that a Voronoi diagram generated by $m \geq 3$ points has at most $3m - 6$ edges; and since each edge is part of exactly two cells, we have that $\sum_{i=1}^{m} e_{V_i} = \mathcal{O}(m)$. Thus, (4.16) reduces to $\mathcal{O}(m \sum_{j=1}^{p} e_{A_j})$. By construction, it also holds that $\sum_{i=1}^{m} |\mathbb{E}_i|$ is $\mathcal{O}(m \sum_{j=1}^{p} e_{A_j})$. Finally, [6, Alg. A.4] is used to compute $S_{ij} = W_{ij} \cap B(x_i, r)$, and a simple inspection of this algorithm shows that the computational effort required to compute all S_{ij}, as well as $\sum_{i=1}^{m} |\mathbb{A}_i|$, are both $\mathcal{O}(m \sum_{j=1}^{p} e_{A_j})$. This implies that the worst-case time complexity of Algorithms 4.1, 4.2, and 4.3 is $\mathcal{O}(m \log m + m \sum_{j=1}^{p} e_{A_j})$.

5. Numerical experiments. In this section, we aim to illustrate the capabilities and limitations of the proposed approach. We implemented Algorithms 4.1, 4.2, and 4.3 in Fortran 90. Given the balls' centers $\{x_i\}_{i=1}^{m}$, the Voronoi diagram is computed with subroutine DTRIS2 from Geompack [24] (available at https://people.math.sc.edu/Burkardt/f_src/geompack2/geompack2.html). In fact, DTRIS2 provides a Delaunay triangulation from which the Voronoi diagram is extracted. The intersection W_{ij} of each Voronoi cell V_i (that is, a bounded or unbounded polyhedron) and each convex polygon A_j is computed with the Sutherland–Hodgman algorithm [38]. For each convex polygon W_{ij}, the intersection S_{ij} with the ball $B(x_i, r)$ is computed with an adaptation of a single iteration of the Sutherland–Hodgman algorithm; see [6, Alg. A.4].

Since Algorithms 4.1, 4.2, and 4.3 are based on Assumptions 1 and 2, the question of enforcing these conditions in the numerical experiments arises. From a theoretical point of view, these assumptions are convenient to obtain first- and second-order differentiability. Nevertheless, one can prove first- and second-order differentiability or Gateaux semidifferentiability, using asymptotic analysis, for various singular cases for which Assumptions 1 and 2 do not hold; see [7, section 3.5] and [6]. From a numerical point of view, there is no need to enforce these conditions: the initial configuration is random and, therefore, is never singular, and these singular cases constitute a set of null measure so that they rarely occur in practice. Of course, they occur in the limit if the solution of the problem under consideration is degenerate. However, since, in practice, iterative algorithms stop "before reaching the limit," degenerate solutions do not represent a practical issue. This was illustrated by a numerical experiment in [7, section 5] for the singular case where A is the union of two tangent unitary-diameter balls to be covered with $m = 2$ balls. Despite the solution being degenerate, it was found, with the desired preset precision, by visiting only points at which Assumptions 1 and 2 hold and, therefore, the derivatives are well defined.

First of all, to put the practical performance of Algorithm 4.1 in perspective in relation to the practical performance of the method implemented in [7], consider a trivial configuration given by a square of side three with the bottom-left corner at the origin and two unitary-radius balls with centers $x_1 = (0, 3)^\top$ and $x_2 = (1.2, 1.7)^\top$. The covered area can be computed analytically and is given by $\mathrm{Vol}(A \cap \Omega(\boldsymbol{x}, r)) = 5\pi/4 - 2\arccos(d/2) + d\sqrt{1 - (d/2)^2} \approx 3.781718647855564$, where $d = \|x_1 - x_2\|$. Algorithm 4.1 computes this quantity up to the machine precision in 10^{-6} seconds

of CPU time. Algorithm 4.1 from [7], devised to cover more general nonpolygonal regions, approximates a covered area with precision $O(h)$ at cost $O(h^2)$ by partitioning a region D that contains A in small squares of side h, where $h > 0$ is a given parameter. In this specific trivial example, it takes 271.92 seconds of CPU time to compute the covered area with half of the machine precision using $h = 10^{-5}$. (With $h = 10^{-3}$ and $h = 10^{-4}$, four and six correct decimal digits are obtained, by consuming 0.024 and 2.4 seconds of CPU time, respectively. Moreover, the cost is proportional to the area of D, which is as small as possible since we considered $D = A$.) So, in this trivial example, we showed that Algorithm 4.1 allows us to compute the covered area with twice the number of correct digits with a computational cost that is eight orders of magnitude smaller (i.e., a hundred million times faster) than the cost of the approach proposed in [7], thus dramatically improves the computational efficiency.

Problem (2.1) is a nonlinear programming problem of the form

$$\text{Minimize } f(\boldsymbol{x}, r) := r \text{ subject to } G(\boldsymbol{x}, r) = 0 \text{ and } r \geq 0 \tag{5.1}$$

that can be tackled with an Augmented Lagrangian (AL) approach [9]. In the numerical experiments, we considered the AL method Algencan [2, 9, 10]. Algencan 4.0, implemented in Fortran 90 and available at http://www.ime.usp.br/~egbirgin/tango/, was considered. Algencan is an AL method with safeguards that, at each iteration, solves a bound-constrained subproblem. Since, in the present work, second-order derivatives are available, subproblems are solved with an active-set Newton's method; see [8] and [9, Chap .9] for details. When Algencan is applied to problem (5.1), on success, it finds $(\boldsymbol{x}^\star, r^\star, \lambda^\star)$ with $r^\star > 0$ satisfying

$$\|\nabla f(\boldsymbol{x}^\star, r^\star) + \lambda^\star \nabla G(\boldsymbol{x}^\star, r^\star)\|_\infty \leq \varepsilon_{\text{opt}} \text{ and } \|G(\boldsymbol{x}^\star, r^\star)\|_\infty \leq \varepsilon_{\text{feas}}, \tag{5.2}$$

where $\varepsilon_{\text{feas}} > 0$ and $\varepsilon_{\text{opt}} > 0$ are given feasibility and optimality tolerances, respectively; i.e., it finds a point that approximately satisfies KKT conditions for problem (5.1). Following [7], in order to enhance the probability of finding an approximation to a global minimizer, a simple multistart strategy with random initial guesses is employed; see [7, section 5] for details. We considered $\varepsilon_{\text{feas}} = \varepsilon_{\text{opt}} = 10^{-8}$.

In the numerical experiments, we considered (i) a nonconvex polygon with holes already considered in [37], (ii) a sketch of a map of the Americas available from [9, section 13.2] and already considered in [7], and (iii) iteration three of the Cesàro fractal; see Figures 6a–8a. (Two additional problems can also be found in [6].) In Figures 6b–8b, the way in which the problems were partitioned into convex polygons is made explicit. A description of each problem (namely, the vertices of each convex polygon that compose each problem) can be found in [6, App. B].

Fortran source code of Algorithms 4.1, 4.2, and 4.3, the source code of the considered problems, as well as the source code necessary to reproduce all numerical experiments, is available at http://www.ime.usp.br/~egbirgin/. All tests were conducted on a computer with an AMD Opteron 6376 processor and 256GB 1866 MHz DDR3 of RAM memory, running Debian GNU/Linux (version 9.13–stretch). Code was compiled by the GFortran compiler of GCC (version 6.3.0) with the -O3 optimization directive enabled.

In the experiments, we covered the three considered regions with $m \in \{10, 20, \ldots, 100\}$ balls. For each problem and each considered value of m, the multistart strategy makes 10,000 attempts, i.e. 10,000 different random initial guesses are considered. Table 1 and Figures 6–8 show the results. In Table 1, r^* represents the smallest obtained radius, $G(\boldsymbol{x}^*, r^*)$ corresponds to the value of G at the obtained solution,

and "trial" is the ordinal of the initial guess that yields the smallest radius. In addition, some performance metrics are also displayed in the remaining columns of the table. Columns "outit" and "innit" correspond to the so-called outer and inner iterations of the AL method, respectively, "Alg.1", "Alg.2", and "Alg.3" correspond to the number of calls to Algorithms 4.1, 4.2, and 4.3, respectively, i.e., to the number of evaluations of G, ∇G, and $\nabla^2 G$ that were required in the optimization process, and "CPU time" corresponds to the elapsed CPU time in seconds. All of these performance metrics correspond to the trial that lead to the smallest radius for a given problem and a given number of balls m. Thus, the whole process took approximately 10,000 times this effort. Clearly, the overall cost can be reduced by reducing the number of trials. Figure 5 illustrates, for the "nonconvex with holes problem" with $m \in \{10, 20, \ldots, 100\}$, the best obtained radius as a function of the number of trials. The picture shows that, for all values of m, good quality local minimizers are found with less than 100 trials and that in the remaining 99% additional trials only marginal improvements are obtained.

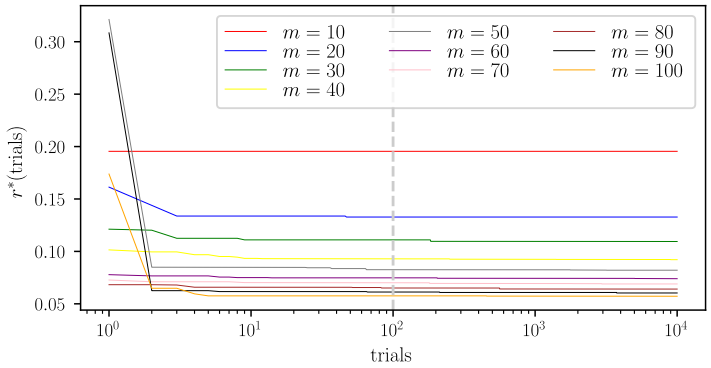


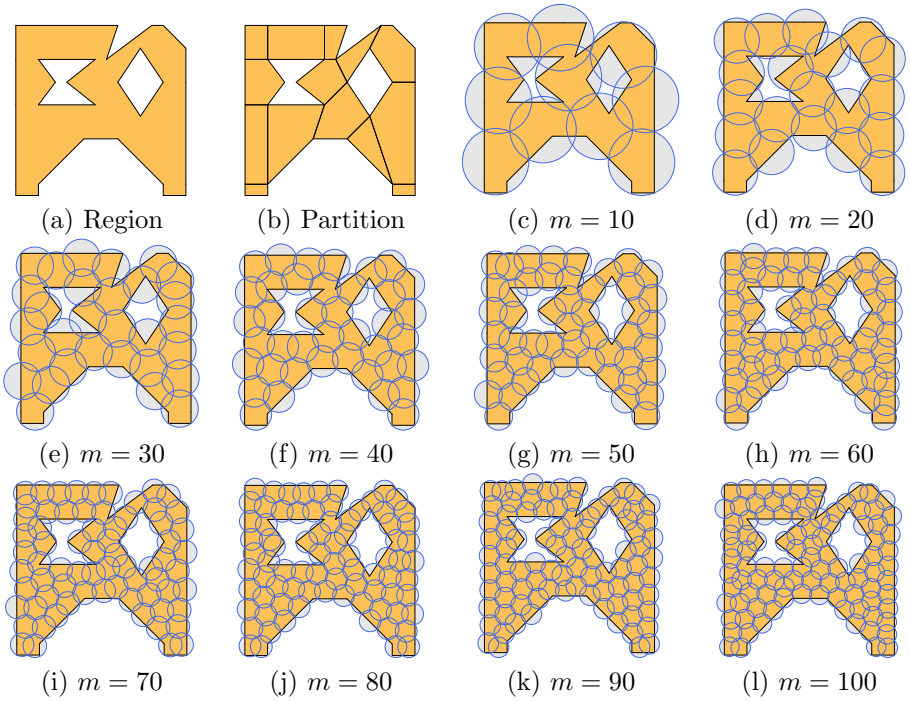
FIG. 5. *Best radius r^* for $m \in \{10, 20, \ldots, 100\}$ as a function of the number of trials in the multistart globalization strategy.*

As a whole, numerical experiments show that, by using second-order information, the AL method is able to find high-precision local solutions efficiently. It is worth noticing that, as shown in column $G(\boldsymbol{x}^*, r^*)$ of Table 1, using $\varepsilon_{\text{feas}} = 10^{-8}$ means that the area of the region A to be covered and the covered region $A \cap \Omega(\boldsymbol{x}^*, r^*)$ coincide in eight significant digits. Since, in the considered problems, no region has an area larger than 100 (see [6, App. B]), this means that reported solutions cover more than 99.999999% of the region. This precision is in contrast with the relatively low-quality solutions obtained with the approximate procedure considered in [7]. A scaled version of the nonconvex region with holes considered in the present work was also considered in [37], where radii $r^* = 16.6176655/150 \approx 0.110784446$ and $r^* = 14.07100757/150 \approx 0.09380671713$ for the cases with $m = 30$ and $m = 40$ were reported. A direct comparison is not possible because the balls' centers and the covering's precision of these solutions was not reported in [37]. Anyway, smaller radii were found for these two cases in the present work, namely, $r^* = 0.1094496310$ and $r^* = 0.09211041653$, respectively. The region that represents a sketch of the map of the Americas was also considered in [7]. Solutions presented in [7] are not comparable to the ones presented here. The latter are much more precise and can be found with much less effort.

TABLE 1

Details of the obtained solutions and performance metrics of the application of Algencan to the three considered covering problems.

	m	r^*	$G(\boldsymbol{x}^*, r^*)$	Trial	outit	innit	Alg. 1	Alg. 2	Alg. 3	CPU time
Nonconvex with holes	10	1.954663097e−01	5.2e−09	7078	23	154	538	388	384	0.33
	20	1.327772115e−01	4.2e−09	4580	21	123	426	345	333	0.56
	30	1.094496310e−01	9.9e−09	7155	22	187	1154	413	407	1.48
	40	9.211041653e−02	9.3e−09	8981	21	209	847	432	419	1.85
	50	8.205969668e−02	9.0e−09	3176	21	218	937	450	428	2.57
	60	7.397252994e−02	8.4e−09	7718	22	245	1750	484	465	4.54
	70	6.895468329e−02	9.0e−09	2942	20	209	1228	421	409	4.35
	80	6.406536859e−02	7.5e−09	8908	21	209	1366	419	419	5.69
	90	6.034584051e−02	7.7e−09	3741	23	263	2595	500	493	9.71
	100	5.722651130e−02	6.9e−09	2619	20	225	1390	448	425	5.45
Sketch of the Americas map	10	1.102268022e−01	6.2e−09	7191	22	226	1198	434	446	0.91
	20	7.056619375e−02	4.2e−09	558	21	256	1541	455	466	2.14
	30	5.672894538e−02	3.7e−09	3341	20	241	1451	428	441	3.36
	40	4.847968184e−02	5.3e−09	7518	21	274	1227	506	484	4.18
	50	4.307962390e−02	4.6e−09	9471	22	190	915	405	410	3.82
	60	3.866922338e−02	9.0e−09	6539	22	328	2124	544	548	9.17
	70	3.547953624e−02	9.3e−09	2774	20	290	1864	508	490	10.81
	80	3.303521347e−02	3.7e−09	9176	23	281	1098	529	511	8.94
	90	3.108185943e−02	9.4e−09	1815	20	296	967	528	496	11.20
	100	2.918558241e−02	7.3e−09	2427	21	302	1271	525	512	10.55
Cesàro fractal	10	2.127686460e−01	5.5e−09	7054	22	180	1348	377	400	0.63
	20	1.332687821e−01	3.8e−09	4870	23	278	1152	421	508	1.22
	30	1.052216365e−01	4.1e−09	7850	23	245	1219	486	475	2.18
	40	9.342803510e−02	9.4e−09	2646	21	193	871	424	403	2.43
	50	8.331418075e−02	9.4e−09	1317	23	197	1322	441	427	3.68
	60	7.841515385e−02	8.8e−12	9229	31	433	3704	674	743	10.83
	70	7.046047099e−02	8.4e−09	5859	22	289	1698	544	509	6.96
	80	6.611079200e−02	8.6e−09	7697	21	329	1694	581	539	9.38
	90	6.195627851e−02	7.8e−09	5722	23	318	3185	568	548	15.12
	100	5.846596190e−02	8.6e−09	3205	21	296	1729	548	506	8.32



(a) Region (b) Partition (c) $m = 10$ (d) $m = 20$

(e) $m = 30$ (f) $m = 40$ (g) $m = 50$ (h) $m = 60$

(i) $m = 70$ (j) $m = 80$ (k) $m = 90$ (l) $m = 100$

FIG. 6. (a) *The nonconvex polygon with holes considered in* [37], *partitioned into* $p = 14$ *convex polygons as depicted in* (b). *Pictures from* (c) *to* (l) *display the solutions found with* $m \in \{10, 20, \ldots, 100\}$.

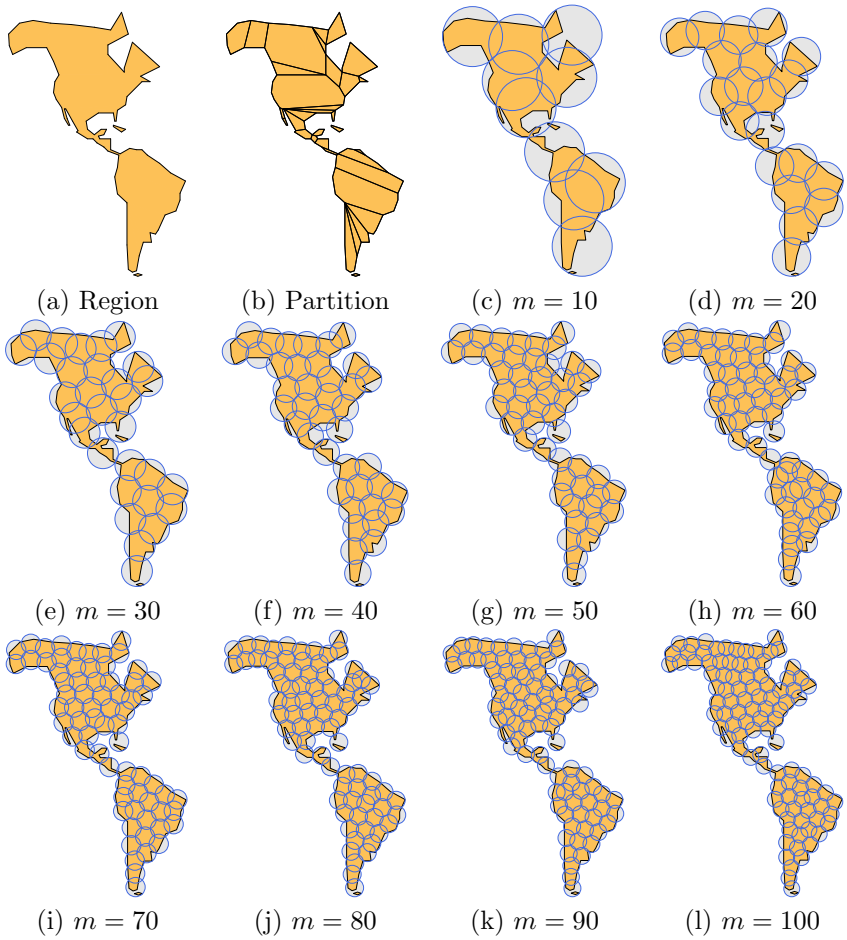


FIG. 7. (a) *Sketch of the Americas available from* [9, section 13.2] *and already considered in* [7]*, partitioned into* $p = 34$ *convex polygons as depicted in* (b)*. Pictures from* (c) *to* (l) *display the solutions found with* $m \in \{10, 20, \ldots, 100\}$.

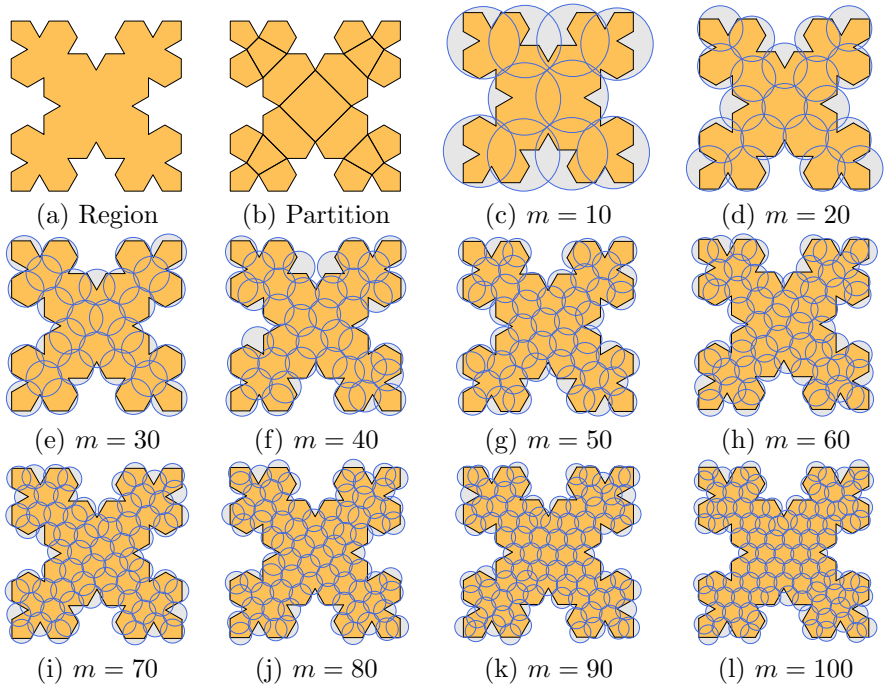


FIG. 8. (a) *Cesàro fractal, partitioned into* $p = 21$ *convex polygons as depicted in* (b). *Pictures from* (c) *to* (l) *display the solutions found with* $m \in \{10, 20, \ldots, 100\}$.

6. Final considerations. From the shape optimization perspective, the present work completes [7] with a second-order shape sensitivity analysis for nonsmooth domains defined as a union of balls intersected with the domain to be covered. The analysis of several singular cases in [6] seems to indicate that the assumptions used to derive $\nabla^2 G$ cannot be weakened. From the practical point of view, the exact calculation of G and its first- and second-order derivatives represents the possibility, absent in [7], of solving very efficiently and with high accuracy, problems in which the area to be covered is given by a (union of) nonconvex polygons.

We now discuss potential extensions of our approach. Redefining $\Omega(\boldsymbol{x}, \boldsymbol{r}) := \cup_{i=1}^{m} B(x_i, r_i)$ and $G(\boldsymbol{x}, \boldsymbol{r}) := \mathrm{Vol}(A \setminus \Omega(\boldsymbol{x}, \boldsymbol{r}))$, where $\boldsymbol{r} := \{r_i\}_{i=1}^{m}$, expressions and algorithms to approximate $G(\boldsymbol{x}, \boldsymbol{r})$, $\nabla G(\boldsymbol{x}, \boldsymbol{r})$, and $\nabla^2 G(\boldsymbol{x}, \boldsymbol{r})$ can be obtained with straightforward modifications to the introduced approach. From the practical point of view, underlying partitions that lead to exact calculations might be implemented using power diagrams [4, 23]. We observe that formulae (2.5), (2.7), (2.8), (2.9), and (2.10) are valid for general sets A satisfying Assumptions 1 and 2, but the exact numerical computation of G, ∇G, and $\nabla^2 G$ requires A to be a union of nonoverlapping convex polygons. The exact calculation of ∇G and $\nabla^2 G$ can actually be performed for any set A such that the intersections of ∂A with circles can be computed analytically. However, the possibilities of computing G exactly are more restricted as this requires the computation of integrals on subsets of ∂A. In some specific cases, this calculation could be done exactly, for instance when A is a union of balls. Nevertheless, in more general cases the integrals on subsets of ∂A could be efficiently approximated with high accuracy; or, alternatively, the original A may be approximated by a polygon.

The case where $\Omega(\boldsymbol{x}, r)$ is a union of objects with arbitrary (sufficiently smooth)

shapes is challenging and would require a generalization of the techniques developed in [7] and in the present paper. A key idea of our construction of the mappings T_t, which is still valid for objects with arbitrary shapes, is that the value of T_t at the intersection points of the objects' boundaries (or the intersections with ∂A) is fully determined by the motion of these singular points, whereas the value of T_t at the regular points of $\partial\Omega(\boldsymbol{x}, r)$ is underdetermined. When the objects are balls, this underdetermination is conveniently resolved using polar coordinates to extend T_t to the regular parts of $\partial\Omega(\boldsymbol{x}, r)$. However, in the case of arbitrary shaped-objects, for which rotations and free-form deformations become relevant, a more general construction of T_t is required. A generalization to three dimensions of the nonsmooth shape optimization techniques developed in [7] and in the present paper is conceivable but would also require a more general approach to build T_t. Another interesting direction for future investigations would be the application of these techniques for optimization problems involving partial differential equations. The calculation of the shape derivatives would depend on the specific partial differential equation, but the construction of the transformations T_t would remain the same.

REFERENCES

[1] L. Afraites, M. Dambrine, and D. Kateb, *On second order shape optimization methods for electrical impedance tomography*, SIAM J. Control Optim., 47 (2008), pp. 1556–1590, https://doi.org/10.1137/070687438.

[2] R. Andreani, E. G. Birgin, J. M. Martínez, and M. L. Schuverdt, *On Augmented Lagrangian methods with general lower-level constraints*, SIAM J. Optim., 18 (2008), pp. 1286–1309, https://doi.org/10.1137/060654797.

[3] E. Arian and V. N. Vatsa, *A preconditioning method for shape optimization governed by the Euler equations*, Int. J. Comput. Fluid Dyn., 12 (1999), pp. 17–27, https://doi.org/10.1080/10618569908940813.

[4] F. Aurenhammer, *Power diagrams: Properties, algorithms and applications*, SIAM J. Comput., 16 (1987), pp. 78–96, https://doi.org/10.1137/0216006.

[5] E. G. Birgin, W. Gómez, G. Haeser, L. M. Mito, and D. S. Viana, *An Augmented Lagrangian algorithm for nonlinear semidefinite programming applied to the covering problem*, Comput. Appl. Math., 39 (2020), 10.

[6] E. G. Birgin, A. Laurain, R. Massambone, and A. G. Santana, *A Shape-Newton Approach to the Problem of Covering with Identical Balls*, preprint, https://arxiv.org/abs/2106.03641, 2021.

[7] E. G. Birgin, A. Laurain, R. Massambone, and A. G. Santana, *A shape optimization approach to the problem of covering a two-dimensional region with minimum-radius identical balls*, SIAM J. Sci. Comput., 43 (2021), pp. A2047–A2078, https://doi.org/10.1137/20M135950X.

[8] E. G. Birgin and J. M. Martínez, *Large-scale active-set box-constrained optimization method with spectral projected gradients*, Comput. Optim. Appl., 23 (2002), pp. 101–125.

[9] E. G. Birgin and J. M. Martínez, *Practical Augmented Lagrangian Methods for Constrained Optimization*, Fundam. Algorithms 10, SIAM, Philadelphia, 2014, https://doi.org/10.1137/1.9781611973365.

[10] E. G. Birgin and J. M. Martínez, *Complexity and performance of an Augmented Lagrangian algorithm*, Optim. Methods Softw., 35 (2020), pp. 885–920, https://doi.org/10.1080/10556788.2020.1746962.

[11] M. Dambrine and M. Pierre, *About stability of equilibrium shapes*, M2AN Math. Model. Numer. Anal., 34 (2000), pp. 811–834, https://doi.org/10.1051/m2an:2000105.

[12] M. Dambrine, J. Sokolowski, and A. Zochowski, *On stability analysis in shape optimization: Critical shapes for Neumann poblem*, Control Cybernet., 32 (2003), pp. 503–528.

[13] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry*, Algorithms and Applications, 3rd ed., Springer-Verlag, Berlin, Heidelberg, 2008, https://doi.org/10.1007/978-3-540-77974-2.

[14] M. C. Delfour and J.-P. Zolésio, *Shapes and Geometries*, 2nd ed., Adv. Des. Control 22, SIAM) Philadelphia, 2011, https://doi.org/10.1137/1.9780898719826.
[15] K. Eppler and H. Harbrecht, *A regularized Newton method in electrical impedance tomography using shape Hessian information*, Control Cybernet., 34 (2005), pp. 203–225.
[16] K. Eppler, H. Harbrecht, and R. Schneider, *On convergence in elliptic shape optimization*, SIAM J. Control Optim., 46 (2007), pp. 61–83, https://doi.org/10.1137/05062679X.
[17] S. Fortune, *A sweep line algorithm for Voronoi diagrams*, Algorithmica, 2 (1987), pp. 153–174, https://doi.org/10.1007/BF01840357.
[18] D. A. Ham, L. Mitchell, A. Paganini, and F. Wechsung, *Automated shape differentiation in the unified form language*, Struct. Multidiscip. Optim., 60 (2019), pp. 1813–1820, https://doi.org/10.1007/s00158-019-02281-z.
[19] A. Henrot and M. Pierre, *Shape Variation and Optimization*, A Geometrical Analysis, EMS Tracts Math. 28, European Mathematical Society (EMS), Zürich, 2018, https://doi.org/10.4171/178, English version of the French *Variation et optimisation de formes* with additions and updates.
[20] A. Heppes and J. B. M. Melissen, *Covering a rectangle with equal circles*, Period. Math. Hungar., 34 (1997), pp. 65–81.
[21] M. Hintermüller and W. Ring, *A second order shape optimization approach for image segmentation*, SIAM J. Appl. Math., 64 (2004), pp. 442–467, https://doi.org/10.1137/s0036139902403901.
[22] E. Horowitz and M. Papa, *Polygon Clipping: Analysis and Experiences*, in Theoretical Studies in Computer Science, J. D. Ullman, ed., Academic Press, Boston, 1992, pp. 315–339, https://doi.org/10.1016/B978-0-12-708240-0.50016-2.
[23] H. Imai, M. Iri, and K. Murota, *Voronoi diagram in the Laguerre geometry and its applications*, SIAM J. Comput., 14 (1985), pp. 93–105, https://doi.org/10.1137/0214006.
[24] B. Joe, *GEOMPACK - A software package for the generation of meshes using geometric algorithms*, Adv. Eng. Softw., 13 (1991), pp. 325–331, https://doi.org/10.1016/0961-3552(91)90036-4.
[25] J. M. Keil, *Decomposing a polygon into simpler components*, SIAM J. Comput., 14 (1985), pp. 799–817, https://doi.org/10.1137/0214056.
[26] A. Laurain, *Structure of shape derivatives in nonsmooth domains and applications*, Adv. Math. Sci. Appl., 15 (2005), pp. 199–226.
[27] A. Laurain, *Distributed and boundary expressions of first and second order shape derivatives in nonsmooth domains*, J. Math. Pures Appl. (9), 134 (2020), pp. 328–368, https://doi.org/10.1016/j.matpur.2019.09.002.
[28] A. Laurain and K. Sturm, *Distributed shape derivative via averaged adjoint method and applications*, ESAIM Math. Model. Numer. Anal., 50 (2016), pp. 1241–1267, https://doi.org/10.1051/m2an/2015075.
[29] J. B. M. Melissen, *Loosest circle coverings of an equilateral triangle*, Math. Mag., 70 (1997), pp. 118–124.
[30] J. B. M. Melissen and P. C. Schuur, *Improved coverings of a square with six and eight equal circles*, Electron. J. Combin., 3 (1996), 32.
[31] K. J. Nurmela, *Conjecturally optimal coverings of an equilateral triangle with up to* 36 *equal circles*, Exp. Math., 9 (2000), pp. 241–250.
[32] K. J. Nurmela and P. R. J. Östergård, *Covering a Square with Up to* 30 *Equal Circles*, Technical Report HUT-TCS-A62, Helsinki University of Technology, Helsinki, Finland, 2000.
[33] S. Schmidt, *Weak and strong form shape Hessians and their automatic generation*, SIAM J. Sci. Comput., 40 (2018), pp. C210–C233, https://doi.org/10.1137/16m1099972.
[34] S. Schmidt and V. Schulz, *Impulse response approximations of discrete shape Hessians with application in CFD*, SIAM J. Control Optim., 48 (2009), pp. 2562–2580, https://doi.org/10.1137/080719844.
[35] L. Simon, *Lectures on Geometric Measure Theory*, Proc. Centre Math. Anal. Austral. Nat. Univ. 3, Australian National University, Centre for Mathematical Analysis, Canberra, 1983.
[36] J. Sokołowski and J.-P. Zolésio, *Introduction to Shape Optimization*, Springer Ser. Comput. Math. 16, Springer-Verlag, Berlin, Heidelberg, 1992.
[37] Y. G. Stoyan and V. M. Patsuk, *Covering a compact polygonal set by identical circles*, Comput. Optim. Appl., 46 (2010), pp. 75–92.
[38] I. E. Sutherland and G. W. Hodgman, *Reentrant polygon clipping*, Commun. ACM, 17

(1974), pp. 32–42, https://doi.org/10.1145/360767.360802.
[39] S. W. Walker, *The Shapes of Things: A Practical Guide to Differential Geometry and the Shape Derivative*, Adv. Des. Control 28, SIAM, Philadelphia, 2015, https://doi.org/10.1137/1.9781611973969.
[40] A. E. Xavier and A. A. Fernandes de Oliveira, *Optimal covering of plane domains by circles via hyperbolic smoothing*, J. Global Optim., 31 (2005), pp. 493–504.