

Article

# Mathematical Models and Nonlinear Optimization in Continuous Maximum Coverage Location Problem

Sergiy Yakovlev <sup>1,2,\*</sup>, Oleksii Kartashov <sup>1</sup> and Dmytro Podzheha <sup>1</sup>

<sup>1</sup> Mathematical Modelling and Artificial Intelligence Department, National Aerospace University “Kharkiv Aviation Institute”, 61072 Kharkiv, Ukraine; alexeykartashov@gmail.com (O.K.); dima.podzega@gmail.com (D.P.)

<sup>2</sup> Institute of Information Technology, Lodz University of Technology, 90-924 Lodz, Poland

\* Correspondence: s.yakovlev@khai.edu

**Abstract:** This paper considers the maximum coverage location problem (MCLP) in a continuous formulation. It is assumed that the coverage domain and the family of geometric objects of arbitrary shape are specified. It is necessary to find such a location of geometric objects to cover the greatest possible amount of the domain. A mathematical model of MCLP is proposed in the form of an unconstrained nonlinear optimization problem. Python computational geometry packages were used to calculate the area of partial coverage domain. Many experiments were carried out which made it possible to describe the statistical dependence of the area calculation time of coverage domain on the number of covering objects. To obtain a local solution, the BFGS method with first-order differences was used. An approach to the numerical estimation of the objective function gradient is proposed, which significantly reduces computational costs, which is confirmed experimentally. The proposed approach is shown to solve the maximum coverage problem of a rectangular area by a family of ellipses.

**Keywords:** geometric object; coverage; optimization; computational geometry software; Python



**Citation:** Yakovlev, S.; Kartashov, O.; Podzheha, D. Mathematical Models and Nonlinear Optimization in Continuous Maximum Coverage Location Problem. *Computation* **2022**, *10*, 119. <https://doi.org/10.3390/computation10070119>

Academic Editor: Ravi P. Agarwal

Received: 15 June 2022

Accepted: 6 July 2022

Published: 11 July 2022

**Publisher’s Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The maximum coverage problem is a classical problem in computer science, computational complexity theory, and operations research. In the original formulation, the problem is as follows.

There are a family of sets  $\Omega = \{S_1, \dots, S_n\}$  and integer number  $1 \leq k \leq n$ . Sets may have some elements in common. The objective is to find such a family  $\tilde{\Omega} \subseteq \Omega$  of  $k$  sets from  $\Omega$  so that the maximum number of elements is covered, i.e., the union  $\bigcup_{S_i \in \tilde{\Omega}} S_i$  of the selected sets has the maximum size.

The great interest of scientists in the problem of maximum coverage is associated with a wide range of its practical applications. The concept of coverage is associated with video surveillance cameras, emergency warning systems, mobile communications and the Internet, radio stations, environmental sensors, service systems, etc. This leads to the need to cover certain areas and is associated with an analysis of the location of objects of various nature. A major category of location analysis approaches involves coverage when an entity provides services based on spatial proximity. This may be based on travel time, line-of-sight, or hearing factors. These spatial footprints reflect services and may be regular or irregular, contiguous, or fragmented in area. One of the most common approaches to location modeling is the maximum coverage location problem (MCLP). Such a problem is the subject of study for both operations research theory and computational geometry. In this case, the goal is to place several objects in a given region in such a way as to maximize (minimize) some objective function associated with the considered type of objects.

In most of these problems, a demand point is considered to be covered by a facility if the distance or travel time between them is less than a given predetermined value. It is desirable that all points of demand are served by at least one enterprise. However, often full coverage of demand in the region is impossible due to restrictions on the possible number of objects to be placed. In this case, the task is to cover the maximum possible area of demand through the efficient use of limited resources.

Note that the problems associated with determining the location of objects are both discrete and continuous in nature. In the discrete case, there is a finite set of possible locations of potential objects, and it is required to choose their optimal location. Such tasks arise in the placement of physical services, when it is possible to predetermine a finite set of admissible locations. At the same time, there is a wide class of problems when it is allowed to place objects in the entire space or its continual subdomain. Such tasks are continuous (for example, in telecommunications networks, when placing sensors or radar stations, etc.). In this case, continuity conditions can be imposed not only on the sets of admissible locations, but also on the service area. This article is devoted to mathematical modeling and solving these coverage optimization problems.

## 2. Background

Reviews [1–5] are devoted to the issues of object location analysis based on the modeling of coverage problems. The main approaches to location analysis consider coverage in the sense that an object provides services within spatial proximity, i.e., each object has a spatial footprint that defines a service area. Article [6] provides extended overview of the maximal covering location problem, highlighting the use, application, solution, evolution, and generalization of this important location analytic approach.

The problem of placing a certain number of objects within an acceptable service distance to maximize demand in a given region was posed and formalized by Church and ReVelle [7]. This problem is called the maximum coverage location problem (MCLP). To solve this problem, two sets of discrete locations were considered, representing demand and possible locations. As a result, the MCLP is formulated as an integer linear programming problem that can be solved in various ways. Variations arise due to the use of different methods for determining the area of coverage of a potential object or assigning weights to sites on several factors and parameters.

The standard form of MCLP considers a finite set of potential locations for objects, and a set of discrete points represents demand. The paper [8] proposes a generalization that allows both facilities and demand points to be placed continuously on a plane. Such a problem is called planar MCLP. An exact planar MCLP solution, assuming that the need for coverage exists anywhere in the surrounding area and that objects can also be arbitrarily placed in this area, is impossible due to the NP-hard nature of the problem [9]. Several heuristic methods have been proposed in an attempt to obtain acceptable solutions. For example, paper [9] describes a greedy algorithm for MCLP that selects at each step the location containing the largest number of uncovered elements. This greedy algorithm is claimed to be the best approximation algorithm in polynomial time.

In the work [10], a planar MCLP with partial coverage and rectangular demand and service zones is studied. The problem is to position a given number of rectangular service zones on the two-dimensional plane to partially cover a set of existing (possibly overlapping) rectangular demand zones such that the total covered demand is maximized. Based on the properties of the model, the possibility of a significant reduction in the search space is theoretically substantiated and a modification of the branch and bound algorithm for solving the problem is presented. The authors of [10,11] use a computational geometry approach to study partial coverage with rectangular and circular service areas. However, according to [12], optimality is guaranteed only in some special cases.

The paper [13] presents a fuzzy maximally covering location problem in which travel time between any pair of nodes is considered to be a fuzzy variable. A hybrid algorithm

of fuzzy simulation and simulated annealing is used to solve MCLP. Some numerical examples are solved and analyzed to show the performance of the proposed algorithm.

In the article [14], a mixed integer non-linear programming model was formulated for distributing the position of a number of rectangular objects of unequal area in a continuum of a planar plant site with a predetermined fixed area. A continuous approach to the problem is taken. Constraints are developed to eliminate the possible overlap between the different facilities. A novel simulated annealing algorithm is developed to solve large instances of the problem. A unique heuristic algorithm is used for initial solution.

We especially note the work [15], in which the continuous setting of MCLP is analyzed. Continuity conditions are imposed on the location of objects, however, the discrete structure of the service domain is preserved, taking into account the graph of relations between enterprises. The problem of mixed integer non-linear programming is formulated. At the same time, taking into account the geometric properties, the authors propose its equivalent reformulation as an integer linear programming problem.

Other approaches use the division of demand into several smaller regions and the ratio between fully covered or uncovered areas [8,16–18]. A facility is assumed to cover the demand for a region if it covers the entire region. Partially covered areas are considered as uncovered areas. Such coverage is called binary. Consideration of the binary coverage makes it possible to obtain a finite set of potential service points determined by the set of polygon intersection points. As a result, an integer linear programming problem is formulated. However, although binary coverage makes planar MCLP manageable, this approach is prone to significant errors because all partially covered regions, which may have a significant coverage area, are left out and ignored in the final solution [12].

Note that obtaining even a partial maximum coverage is not trivial. Therefore, as an auxiliary problem, one often considers partial coverage by only one object. In the paper [19], a method for solving the problem of determining the location of a circular service area in a polygonal zone is described based on the calculation of the mean axis of the zone. Taking into account partial coverage, in the work [20] it is proposed to approximate the optimal position of a single circular footprint with bounded error in an unconnected area with holes bounded by line or circle segments. The authors describe an efficient algorithm for accurately calculating the overlap area of a circle and a polygon, and introduce the concept of a coverage map for more information. The paper [21] presents improved algorithms for matching two polygonal shapes to approximate their maximum overlap. The run time of algorithms is studied.

The article [22] describes an approach to solving the problem of locating a disk so that its overlap area with a piecewise circular domain is near-optimal when considering partial coverage. The concept of an overlap area map is introduced. Parallel algorithm on graphics processing units are discussed for calculating the overlap area map and deriving a set of near-optimal locations from the overlap area map. Particular attention is paid to the process of calculating the overlap area. The authors visualize the information obtained, highlighting the optimal location, and analyze the experimental results obtained by implementing the proposed algorithms.

The study [23] presents an approach to the maximum coverage of a polygonal area by disk services of a given radius. Of particular interest is a parallel optimization method using simulated annealing based on a perturbation strategy and a probabilistic estimate of the covered area of a polygon. The system provides in a reasonable run time fairly good location of disks, starting with an arbitrary initial solution. The paper [24] considers the problem of covering compact polygonal set by identical circles of minimal radius. A mathematical model of the problem based on Voronoi polygons is constructed and its characteristics are investigated. A modification of the Zoutendijk feasible directions method is developed to search local minima. A special approach is suggested to choose starting points. Many computational examples are given.

Access to and solution of the MCLP is also possible in GIS software packages. For example, structuring and solving an MCLP application instance in *ArcGIS* and *TransCAD*,

among others, is easily accomplished. In *ArcGIS*, there are location-allocation functions in network analyst, and, in particular, “maximize coverage” would enable an MCLP to be set up as the problem type. Similarly, in *TransCAD*, one would use “facility location analysis” to accomplish this. There are in fact a number of reported application studies that have relied on this basic approach using *GIS*, such as [25,26]. It is important to note that *GIS* packages facilitating access to the MCLP typically solve problems using a heuristic [4].

The evolution of the application of MCLP has occurred concurrently in concert with the development, growth, and maturation of *GIS*. The MCLP access and solution is possible in various *GIS* software packages. The authors of [3,27] discuss that *GIS* is a combination of hardware, software, and procedures enabling data management and spatial query. Key components of *GIS* include data capture, modeling, management, manipulation, analysis, and display. It is important that in addition to generating and working with spatial information, *GIS* supports a number of analytical functions, many of which facilitate the MCLP specification and application. In the paper [4], the capabilities of a *GIS* for carrying out suitability analysis, as well as operations associated with containment, overlay, distance, buffering, and spatial interpolation, among others. Much spatial information now exists, expediting coverage location analysis as well.

### 3. Materials and Methods

#### 3.1. Mathematical Model of Continuous Maximum Coverage Location Problem

Let us formulate the continuous MCLP as a geometric problem. In the space  $R^d$ ,  $d = 2, 3$  there are a certain domain  $S_0$  and a family of geometric objects  $\{S_1, \dots, S_n\}$  of a given shapes and sizes. We will say that a family of objects  $\{S_1, \dots, S_n\}$  covers some subdomain  $\tilde{S} \subseteq S_0$  if each point  $\tilde{S}$  belongs to at least one of the objects of the family. The problem is to find such a location of objects  $S_1, \dots, S_n$  that they cover as much of the domain  $S_0$  as possible.  $S_0$  is called the coverage domain, and  $S_1, \dots, S_n$ —the covering objects.

To formalize the continuous MCLP we make the following notations:  $J_n = \{1, \dots, n\}$ ,  $J_n^0 = J_n \cup \{0\} = \{0, 1, \dots, n\}$ . Geometric object  $S_i \subset R^d$ ,  $i \in J_n^0$  means a geometric set of points  $P \in R^d$  that satisfy inequalities  $f_i(P, \mathbf{m}^i) \geq 0$ . The equation  $f_i(P, \mathbf{m}^i) = 0$  defines the boundary of the object  $S_i$  and determines its shape. In the general case, functions  $f_i(P, \mathbf{m}^i)$ ,  $i \in J_n^0$  contain constants  $\mathbf{m}^i = (m_1^i, \dots, m_{\alpha_i}^i)$ , which are called metric parameters of the shape of the object  $S_i$ . These parameters determine the linear sizes of the corresponding objects.

Consider a fixed Cartesian coordinate system in space  $R^d$ ,  $d = 2, 3$ , and with each object  $S_i \subset R^d$ ,  $i \in J_n^0$ , connects own coordinate system, the beginning of which is called the pole of this object. The relative position of these system will be characterized by parameters  $\mathbf{p}^i = (p_1^i, \dots, p_{\beta_i}^i) = (\mathbf{v}^i, \theta^i)$ ,  $i \in J_n^0$ , where  $\mathbf{v}^i$ —a coordinate vector of the pole of the object  $S_i$  in a fixed coordinate system, and  $\theta^i$ —a vector of angular parameters that determine the relative position of the axes of own and fixed coordinate systems. The coordinates of the vector  $\mathbf{p}^i = (p_1^i, \dots, p_{\beta_i}^i)$  indicates the position of the object  $S_i$  in  $R^d$ ,  $d = 2, 3$  and are called placement parameters. In this case, the position of the object  $S_i$  relative to the fixed coordinate system is given by the equation of the general position, which has the form

$$F_i(P, \mathbf{m}^i, \mathbf{v}^i, \theta^i) = f_i(A(p - \mathbf{v}^i), \mathbf{m}^i) = 0, \tag{1}$$

where  $A$  is the orthogonal operator expressed through angular parameters  $\theta^i$ .

In the papers [28,29], methods for constructing configurations of geometric objects are described, and classes of packing, layout, and covering configurations are identified. A configuration space of geometric objects is constructed, the generalized variables of which are placement parameters and metric parameters of objects.

Let us apply these results to build a model of the continuous MCLP. Let  $\Omega = \{S_1, \dots, S_n\}$  be a family of geometric objects with metric parameters of the shape  $\mathbf{m}^i = (m_1^i, \dots, m_{\alpha_i}^i)$  and placement parameters  $\mathbf{p}^i = (p_1^i, \dots, p_{\beta_i}^i) = (\mathbf{v}^i, \theta^i)$ ,  $i \in J_n^0$ . A parameters  $\mathbf{g}^i = (\mathbf{m}^i, \mathbf{p}^i) = (m_1^i, \dots, m_{\alpha_i}^i, p_1^i, \dots, p_{\beta_i}^i)$  is called a generalized variable of the

object  $S_i$ . The dimension of the vector  $\mathbf{g}^i$  is equal to  $\gamma_i = \alpha_i + \beta_i$ . The object  $S_i$  with the generalized variables  $\mathbf{g}^i$  denote by  $S_i(\mathbf{g}^i)$ ,  $i \in J_n^0$ .

Note that, depending on the specifics of the problem, some generalized variables can be fixed. To indicate this we will use a cap over the corresponding variable. In continuous MCLP, placement parameters of the coverage domain  $S_0$  are assumed to be fixed and usually equal to  $\hat{\mathbf{p}}^0 = (0, \dots, 0)$ . Metric parameters  $\mathbf{m}^0$  in the general case are variable, unless otherwise specified by the problem. Considering problems when the sizes of  $S_0$  are known, we will assume  $\mathbf{m}^0 = \hat{\mathbf{m}}^0$ , so  $\hat{\mathbf{g}}^0 = (\hat{\mathbf{m}}^0, \hat{\mathbf{p}}^0)$ .

Using set-theoretic operations, we build a complex object

$$\Omega = S_0 \cap \bigcup_{i=1}^n S_i. \tag{2}$$

We assume that Formula (2) determines the structure of a complex object  $\Omega$ , i.e., the rule of its formation.

Having determined the generalized variables  $\mathbf{g}^i$ ,  $i \in J_n^0$  of the constituent objects  $S_i$ , we form a parameterized object

$$\Omega(\mathbf{g}) = \Omega(\hat{\mathbf{g}}^0, \mathbf{g}^1, \dots, \mathbf{g}^n) = S_0(\hat{\mathbf{g}}^0) \cap \bigcup_{i \in J_n} S_i(\mathbf{g}^i). \tag{3}$$

In  $\mathbb{R}^d$ ,  $d = 2, 3$ , each fixed vector  $\hat{\mathbf{g}} = (\hat{\mathbf{g}}^0, \hat{\mathbf{g}}^1, \dots, \hat{\mathbf{g}}^n)$  of generalized variables corresponds to a complex geometric object of a given structure

$$\Omega(\hat{\mathbf{g}}) = S_0(\hat{\mathbf{g}}^0) \cap \bigcup_{i \in J_n} S_i(\hat{\mathbf{g}}^i). \tag{4}$$

Depending on the dimension of  $\mathbb{R}^d$ ,  $d = 2, 3$ , we calculate the measure (area or volume) of the formed object  $\Omega(\hat{\mathbf{g}})$ , which we denote by  $\mu(\Omega(\hat{\mathbf{g}}))$ . Thus, an arbitrary set of generalized variables  $\mathbf{g} = (\hat{\mathbf{g}}^0, \mathbf{g}^1, \dots, \mathbf{g}^n)$  can be associated with a measure of a complex object  $\Omega(\mathbf{g})$  of a given structure, i.e., define a function of generalized variables

$$\omega_\Omega(\mathbf{g}) = \mu(\Omega(\hat{\mathbf{g}})). \tag{5}$$

The described class of functions is called  $\omega$ -functions [30].

For the formalization of the  $\omega$ -function, we introduce the characteristic function

$$\lambda_\Omega(P, \mathbf{g}) = \begin{cases} 1, & \text{if } P \in \Omega(\mathbf{g}); \\ 0, & \text{if } P \notin \Omega(\mathbf{g}). \end{cases} \tag{6}$$

If  $P \in \mathbb{R}^2$ , then

$$\omega_\Omega(\mathbf{g}) = \iint \lambda_\Omega(P, \mathbf{g}) dP \tag{7}$$

and for  $P \in \mathbb{R}^3$

$$\omega_\Omega(\mathbf{g}) = \iiint \lambda_\Omega(P, \mathbf{g}) dP. \tag{8}$$

Properties of  $\omega$ -functions were considered in the article [31].

Using  $\omega$ -functions allows us to state the maximum coverage problem as the following nonlinear unconstrained optimization problem

$$\omega_\Omega(\mathbf{g}) \rightarrow \max \tag{9}$$

where  $\omega$ -function is defined by the Formula (5), and the structure  $\Omega$  is given as (2).

This statement of the continuous MCLP follows from the fact that the function  $\omega_\Omega(\mathbf{g})$  determines the dependence of the measure (area) of the partial coverage of the domain on the generalized variables  $\mathbf{g}$  of the family of covering objects. It is the maximization of this measure that is the objective of the problem.

Thus, the solution of the continuous MCLP is inextricably linked, on the one hand, with methods for calculating the function  $\omega_{\Omega}(\mathbf{g})$ , and, on the other hand, with the choice of an effective method for solving the optimization problem (9).

Therefore, the general scheme for solving the problem includes the following stages:

- Formation of initial data on the coverage domain and the family of covering objects;
- Choice of generalized coverage configuration variables;
- Calculating a measure of the covered part of the coverage area;
- Choice of local optimization method for solving problem (9);
- Estimation of the gradient of the objective function, taking into account the geometric properties of the problem;
- Choice of global optimization method.

The first two stages were discussed earlier. The next stages are described in the next subsection.

### 3.2. Computer Geometry and Optimization Software

To formalize the function, one can use the analytical approach [31], which consists of constructing an equation for the boundary of a complex geometric object based on the known equations for the boundaries of composite objects. However, this approach is very time consuming and requires large computational costs, especially considering that the objects depend on parameters (generalized variables). The papers [32,33] propose ways to formalize the coverage conditions using *phi*-functions, but only for a narrow class of covering objects, such as rectangles, circles, and spheroids [24,34,35].

At the same time, considering the issue of working with geometric objects of a given shape, you can find a powerful list of libraries that allow you to work with such figures, in particular, *SymPy*, *Shapely*, *CGAL*, *SpaceFuncs*, and many others. Based on the analysis of existing libraries, taking into account the necessary functionality to calculate the  $\omega$ -function of spatial configurations (complex geometric objects), we chose the *Shapely* library [36]. *Shapely* is a *Python* package for theoretical analysis of points set and plane object manipulation using *Python ctypes* module functions from the well-known *GEOS* library. Note that *GEOS* as a port of the *Java Topology Suite (JTS)* is the geometry mechanism of the *PostGIS* spatial extension for the *PostgreSQL* database. On the one hand, the *Shapely* package is deeply rooted in the conventions of the world of geographic information systems (*GIS*). On the other hand, this package seeks to be equally useful for programmers working on non-traditional issues, including image processing and geometric design. It is with the help of the *Shapely* library that it is possible to construct complex shapes through set-theoretic operations on geometric objects (union, intersection, symmetric difference, product) from a set of basic shapes (circle, ellipse, polygon, etc.), as well as perform the same operations on arbitrary complex figures, built with the help of basic. Using the *unary\_union(.)* function in *shapely.ops* allows you to build unions at the same time, which is much more efficient than sequential accumulation using the *union(.)* operation. The set of affine transformation functions is contained in the *shapely.affinity* module, which transforms geometric shapes by directly assigning coefficients to the affine transformation matrix, or by means of a specially named transformation (rotation, scale, etc.). Note that geometric objects are created in the typical *Python* way, using the classes themselves as instance factories.

The most important feature of the *Shapely* library is the ability to calculate the area of any complex object using the *area* field.

The use of modern computer geometry software allows us to offer new effective methods for solving problems of maximum coverage. In particular, in *Python* has developed a package of applied mathematical procedures *SciPy* based on the *Numpy Python* extension. With *SciPy*, the *Python* interactive session becomes the same full-fledged data processing and prototyping environment for complex systems, such as any versions of *MATLAB*, *IDL*, *Octave*, *R-Lab*, and *SciLab* later than 2015.

On the other hand, the properties of the MCLP make it possible to use gradient methods of local optimization using first order differences. To do this, consider the *k*-th

component of the vector  $\hat{\mathbf{g}}^i = (\hat{g}_1^i, \dots, \hat{g}_k^i, \dots, \hat{g}_{\gamma_i}^i)$  corresponding to the object  $S_i$  and give it an increment  $\delta$ . Denote

$$\tilde{\mathbf{g}} = (\hat{g}^0, \dots, \hat{g}^{i-1}, \hat{g}_1^i, \dots, \hat{g}_k^i + \delta, \dots, \hat{g}_{\gamma_i}^i, \hat{g}^{i+1}, \dots, \hat{g}^n). \tag{10}$$

Then, to estimate the increment of a function  $\omega_\Omega(\hat{\mathbf{g}})$  at a point  $\hat{\mathbf{g}} = (\hat{g}^0, \dots, \hat{g}^{i-1}, \hat{g}_1^i, \dots, \hat{g}_k^i, \dots, \hat{g}_{\gamma_i}^i, \hat{g}^{i+1}, \dots, \hat{g}^n)$ , we obtain

$$\Delta\omega_\Omega(\hat{\mathbf{g}}) = \mu(\Omega(\tilde{\mathbf{g}})) - \mu(\Omega(\hat{\mathbf{g}})), \quad i \in J_n, \tag{11}$$

where the vectors  $\hat{\mathbf{g}}$  and  $\tilde{\mathbf{g}}$  are differ in only one coordinate  $\hat{g}_k^i$  changed to  $\delta$ .

This approach to determining the increment of a function  $\omega_\Omega(\hat{\mathbf{g}})$  is universal, but it requires the calculation of the values of function at points  $\hat{\mathbf{g}}$  and  $\tilde{\mathbf{g}}$ .

Using the properties of  $\omega$ -functions, one can propose an approach that has less computational complexity.

Let us form a symmetric matrix of areas of pairwise intersections of objects

$$M = [\mu_{ij}]_{(n+1) \times (n+1)}, \tag{12}$$

where

$$\mu_{ij} = \mu(S_i(\hat{\mathbf{g}}^i) \cap S_j(\hat{\mathbf{g}}^j)), \quad i, j \in J_n^0. \tag{13}$$

As a rule, such a matrix  $M$  is highly sparse. Let us use this property.

For the  $k$ -th component of the vector  $\hat{\mathbf{g}}^i = (\hat{g}_1^i, \dots, \hat{g}_k^i, \dots, \hat{g}_{\gamma_i}^i)$ , corresponding to the object  $S_i$ , we will increment  $\delta$  and form the object  $S_i^k(\tilde{\mathbf{g}})$ , where  $\tilde{\mathbf{g}}$  has the form (10). Then, the increment of the function  $\omega_\Omega(\hat{\mathbf{g}})$  at the point  $\hat{\mathbf{g}} = (\hat{g}^0, \dots, \hat{g}^n)$  will be equal to

$$\Delta_k^i \omega_\Omega(\hat{\mathbf{g}}) = \sum_{j \in J_n^0} \left( \mu(S_i^k(\tilde{\mathbf{g}}) \cap S_j(\hat{\mathbf{g}}^j)) - \mu_{ij} \right) \tag{14}$$

where the summation is over all  $j \in J_n^0$ , such that  $\mu_{ij} \neq 0$ .

The experiments described in the next section confirm that it takes an order of magnitude less time to evaluate the increment of a function using Formula (14) than when using Formula (11).

#### 4. Results

To choose and justify the method for solving the optimization problem (9), we studied the dependence of computational resources on the dimension of the problem, i.e., number of covering objects. We conducted a series of experiments during which we evaluated:

- Time for calculating the area of a complex object, the structure of which is determined by Formula (4);
- Run time of local optimization for an arbitrary starting point;
- Run time of forming the area matrix of pairwise intersections of objects.

Computational experiments were carried out using a computer with the following configuration: Intel Core i7-5557U processor, CPU Speed 3.1 GHz, 2 cores, 4 threads; RAM 16 GB DDR3 1866 MHz; Graphics processor Intel Iris Graphics 6100 with 1.5 GB of video memory; SSD 512 GB; and operating system Mac OS X11.0 Big Sur.

The calculations were made using *Python* packages, such as *Shapely* and *SciPy*.

We chose a rectangle as the coverage domain  $S_0$ , and ellipses as the covering objects  $S_i, i \in J_n$ . The metric parameters  $\hat{\mathbf{m}}^0 = (a_0, b_0)$  of  $S_0$  are lengths of their sides, and the placement parameters are  $\hat{\mathbf{p}}^0 = (0, 0, 0)$ . The metric parameters  $\mathbf{m}^i = (a_i, b_i), i \in J_n$  of ellipses are their semi-axis, and the placement parameters  $\mathbf{p}^i = (x_i, y_i, \theta_i), i \in J_n$  are

the coordinates  $(x^i, y^i)$  of the symmetry centers and the angle of rotation  $\theta^i$ . The general equation of the ellipse in accordance with (1) has the form

$$1 - \frac{((x - x_i) \cos \theta_i + (y - y_i) \sin \theta_i)^2}{a_i^2} + \frac{(-(x - x_i) \sin \theta_i + (y - y_i) \cos \theta_i)^2}{b_i^2} = 0. \quad (15)$$

So,  $\mathbf{g}^0 = (\hat{\mathbf{m}}^0, \hat{\mathbf{p}}^0) = (a_0, b_0, 0, 0, 0)$ ,  $\mathbf{g}^i = (\hat{\mathbf{m}}^i, \mathbf{p}^i) = (a_i, b_i, x_i, y_i, \theta_i)$ ,  $i \in J_n$ ,  $\mathbf{g} = (\mathbf{g}^0, \mathbf{g}^1, \dots, \mathbf{g}^n) = (a_0, b_0, 0, 0, 0, a_1, b_1, x_1, y_1, \theta_1, \dots, a_n, b_n, x_n, y_n, \theta_n)$ .

For each n from 10 to 500 with step 10, we will randomly form the initial data of the problem, generating semi-axes  $(a_i, b_i)$ ,  $i \in J_n$  of ellipses evenly distributed on the interval  $(0, 100)$ . Then, we calculate the total area of the constructed ellipses

$$Q = \pi \sum_{i=1}^n a_i b_i, \quad (16)$$

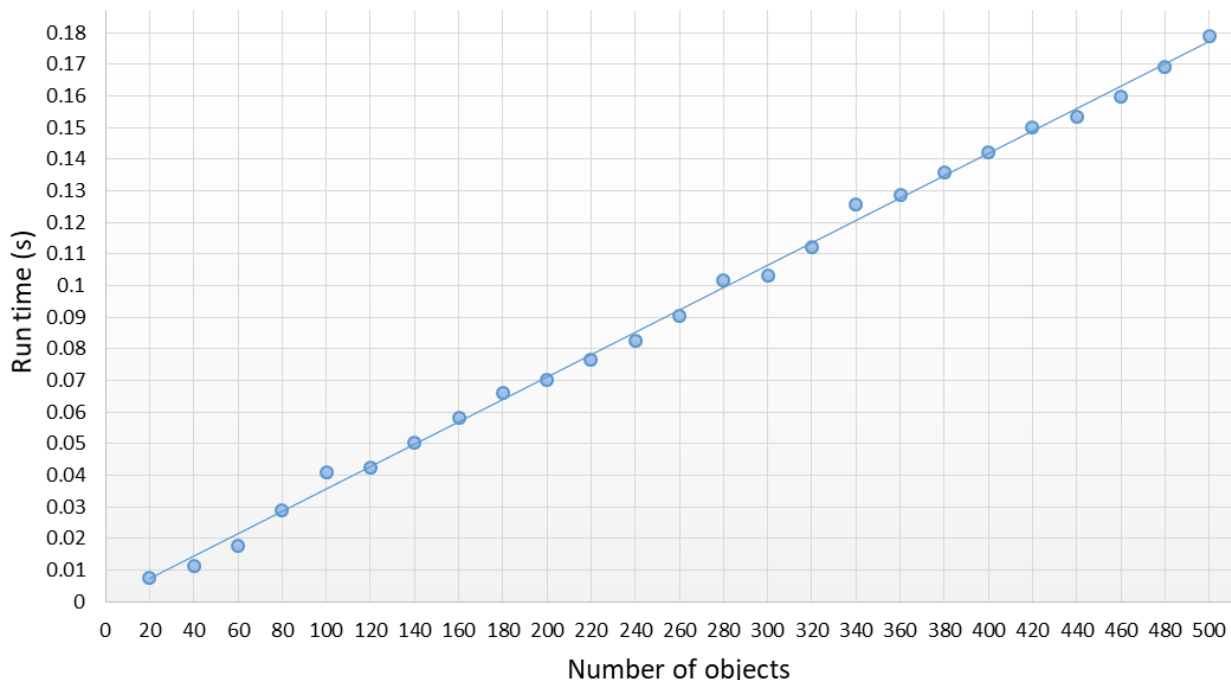
and determine the metric parameters of the coverage domain  $S_0$ —rectangle with sides  $a_0 = \sqrt{2Q}$ ,  $b_0 = \frac{\sqrt{2Q}}{2}$ .

Thus, we form 50 tests that differ from each other in the number n of ellipses, their semi-axis  $(a_i, b_i)$ ,  $i \in J_n$  and the sizes  $(a_0, b_0)$  of the coverage domain.

The next step is to generate ellipse placement parameters, which we also choose for each test evenly distributed on the intervals  $(-\frac{a_0}{2}, \frac{a_0}{2})$ ,  $(-\frac{b_0}{2}, \frac{b_0}{2})$ ,  $(0, \pi)$ , respectively.

Based on the generated initial data, we calculate the area  $\mu(\Omega(\mathbf{g}))$  of a complex object  $\Omega(\mathbf{g})$ , the structure of which is determined by Formula (3). The *Shapely* library was used to build  $\Omega(\mathbf{g})$  and calculate its area.

Figure 1 shows the dependence of the average time of calculating the area  $\mu(\Omega(\mathbf{g}))$  on the number of ellipses  $S_i$ ,  $i \in J_n$  covering the rectangle  $S_0$ . Averaging was performed on 20 independent tests.



**Figure 1.** Dependence of the time of calculation of a complex object area on the number of objects.

The next experiment is to study the time of obtaining a local solution to problem (9). For local optimization, the *SciPy* package was chosen, which implements the *BFGS* method [37] using first order differences. The gradient was estimated by Formula (14). For each of the previously formed test tasks, the placement parameters  $\mathbf{p}^i = (x_i, y_i, \theta_i)$ ,  $i \in J_n$



randomly generated at the previous stage during the formation of a complex object were chosen as a starting point.

Figure 2 shows the dependence of the average time to obtain a local solution on the number of ellipses. Averaging was carried out over 10 independent tests. Formula (14) was used instead of (11) to estimate first-order differences when calculating the gradient of function  $\omega_{\Omega}(\mathbf{g})$ .

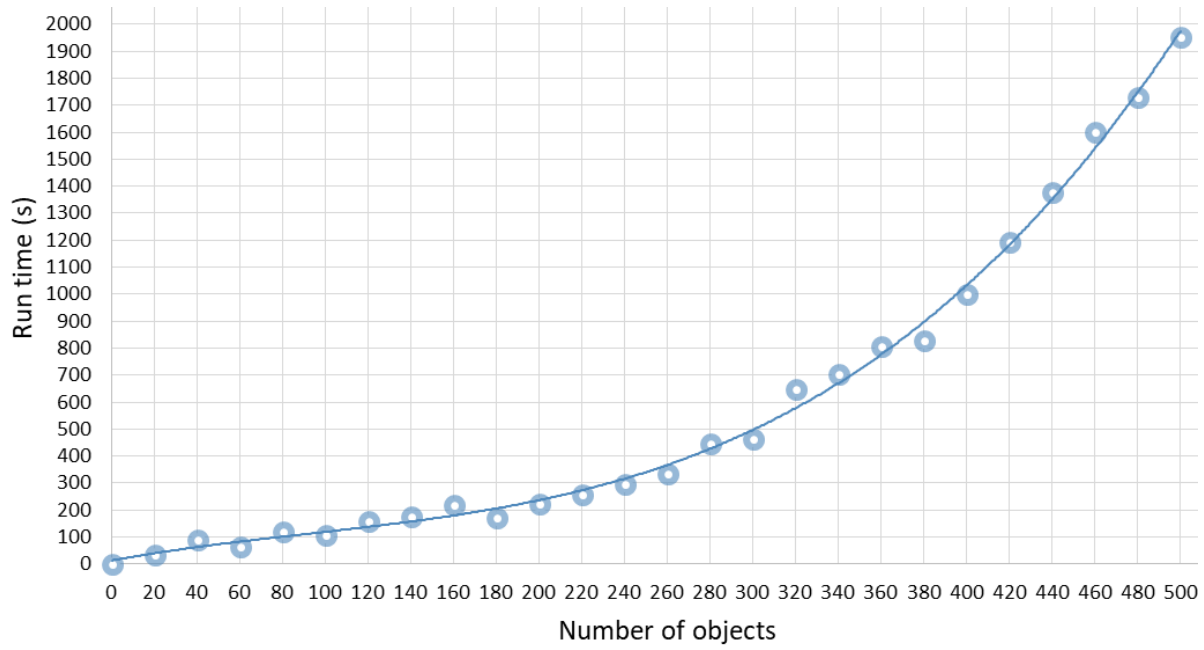


Figure 2. Dependence of local optimization run time on the number of objects.

To substantiate this approach, we studied the dependence of the formation time of the matrix of pairwise intersections (12) on its dimension. Figure 3 illustrates such dependence.

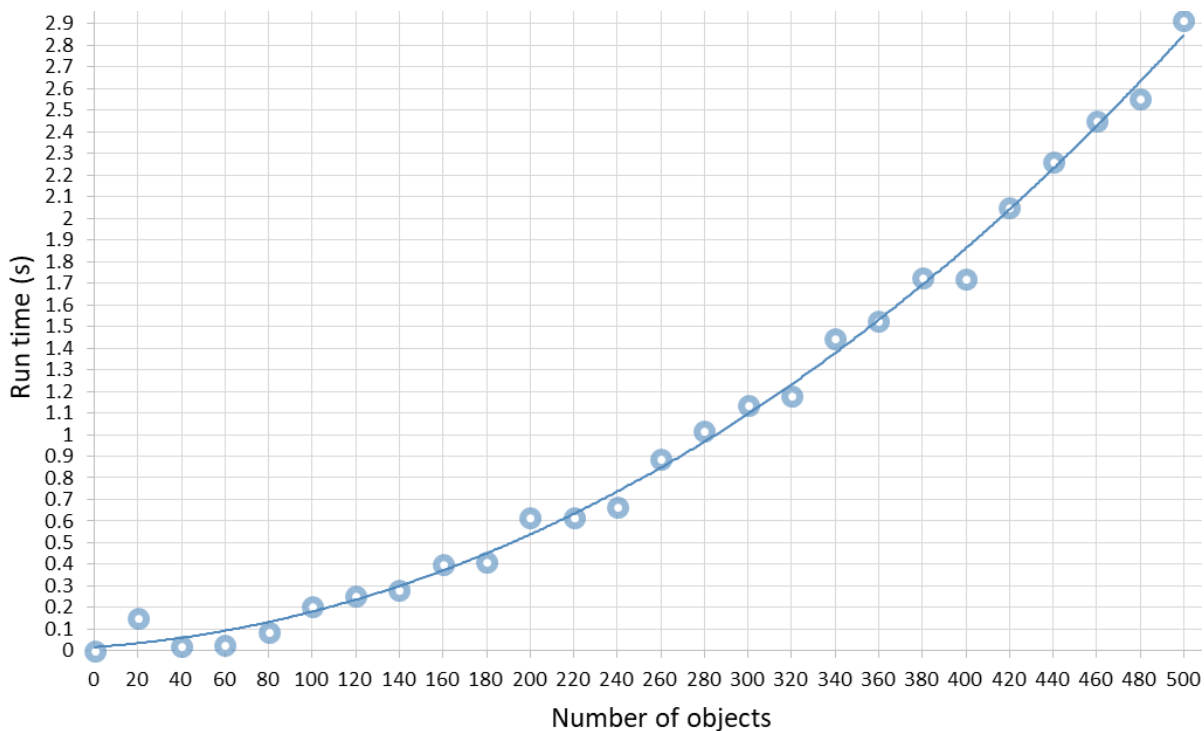


Figure 3. Dependence of the formation time of the matrix M on its dimension.

The data shown in Figure 3 allows us to estimate the average time for calculating the area of intersection of two objects. To do this, you need to sum all the elements of the matrix and divide by their number. Analyzing the obtained results, it can be argued that the estimation of the gradient using the matrix  $M$  requires much less time than by calculating the function  $\mu(\Omega(\mathbf{g}))$ . Indeed, with  $n=100$ , the time for calculating the increment of a function by Formula (11) is  $0.04 \times 2 = 0.08$  s, and the average time for the formation of the upper triangular matrix  $M$  is 0.2 s. Therefore, to calculate the area of intersection of each two objects,  $0.2 \times 2 / (100 \times 101) = 0.000039604$  s is required. Since to estimate  $\Delta_k^i \omega_\Omega(\hat{\mathbf{g}})$  using Formula (14) it is necessary to sum the areas of pairwise intersections of one object with the other 100, this will require 0.0039604 s. You can see the same when  $n = 500$ . The function  $\omega_\Omega(\hat{\mathbf{g}})$  calculation time is 0.18 s and the matrix  $M$  formation time is 2.7 s. The function increment  $\Delta_k^i \omega_\Omega(\hat{\mathbf{g}})$  according to Formula (14) will be calculated approximately in  $2.7 \times 2 / 501 = 0.01078$  s.

Let us apply the described approach to solving the following test problems of maximum coverage. Let the covering objects be ellipses with semi-axis  $(a_i, b_i)$ ,  $i \in J_n$ ,  $n = 100$  listed in Table A1 (Appendix A).

The problems of unconditional local optimization (9) for  $n = 30, 50, 75, 100$  was solved. Squares with sides 30, 40, 70, and 70 for  $n = 30, 50, 70$ , and 100, respectively, were chosen as the coverage domain. *BFGS* method using first-order differences was used. The gradients was estimated by Formula (14). The starting point for *BFGS* method was chosen by random generation of ellipse placement parameters by analogy with the experiments described earlier.

Figures 4–7 show the initial location of the ellipses in the coverage domain and the placement obtained as a result of local optimization for  $n = 30, 50, 75$ , and 100. The run times to solve the problems are 22 s, 51 s, 79 s, and 103 s, respectively.

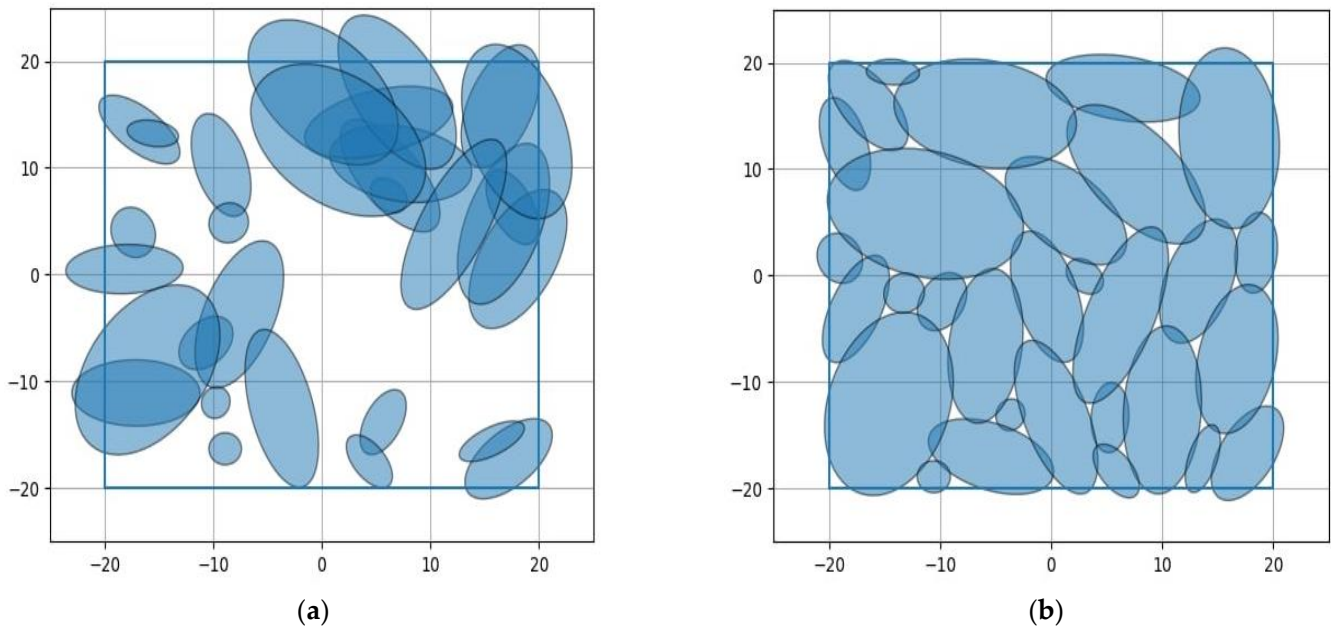


Figure 4. The continuous MCLP of 30 ellipses: (a) starting point and (b) local optimum.

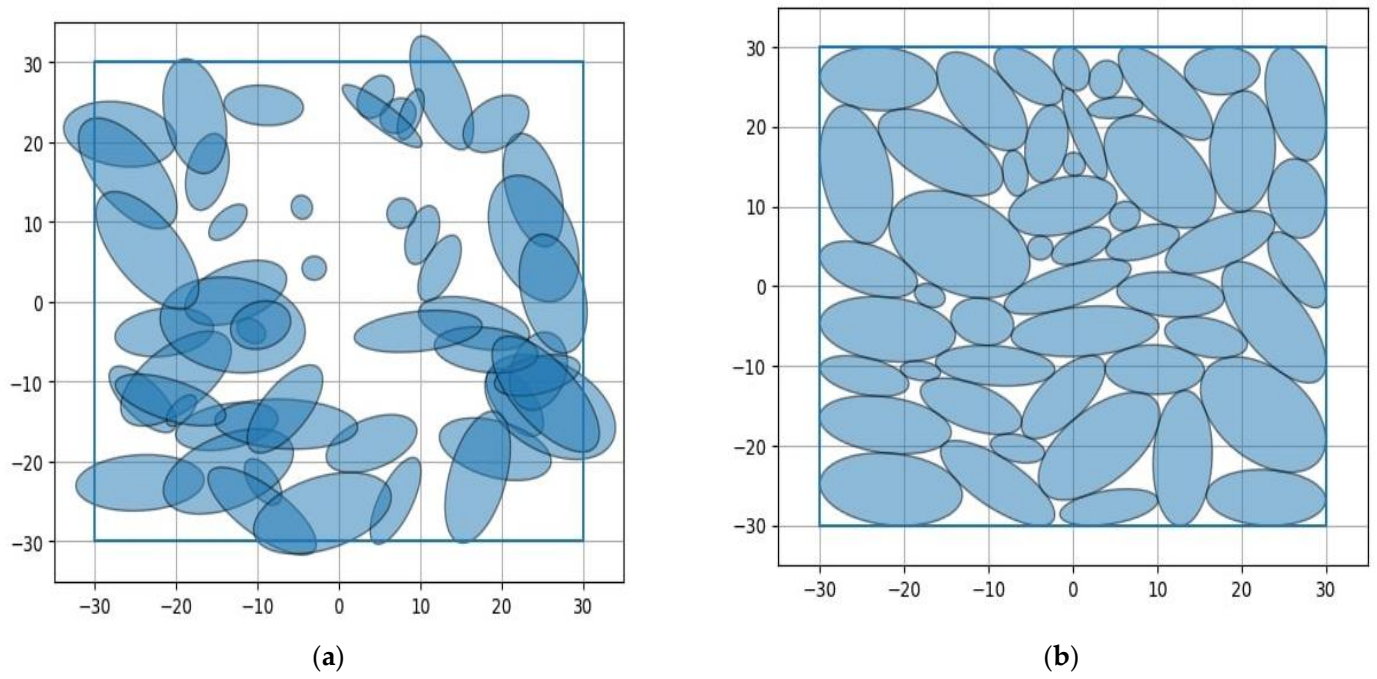


Figure 5. The continuous MCLP of 50 ellipses: (a) starting point and (b) local optimum.

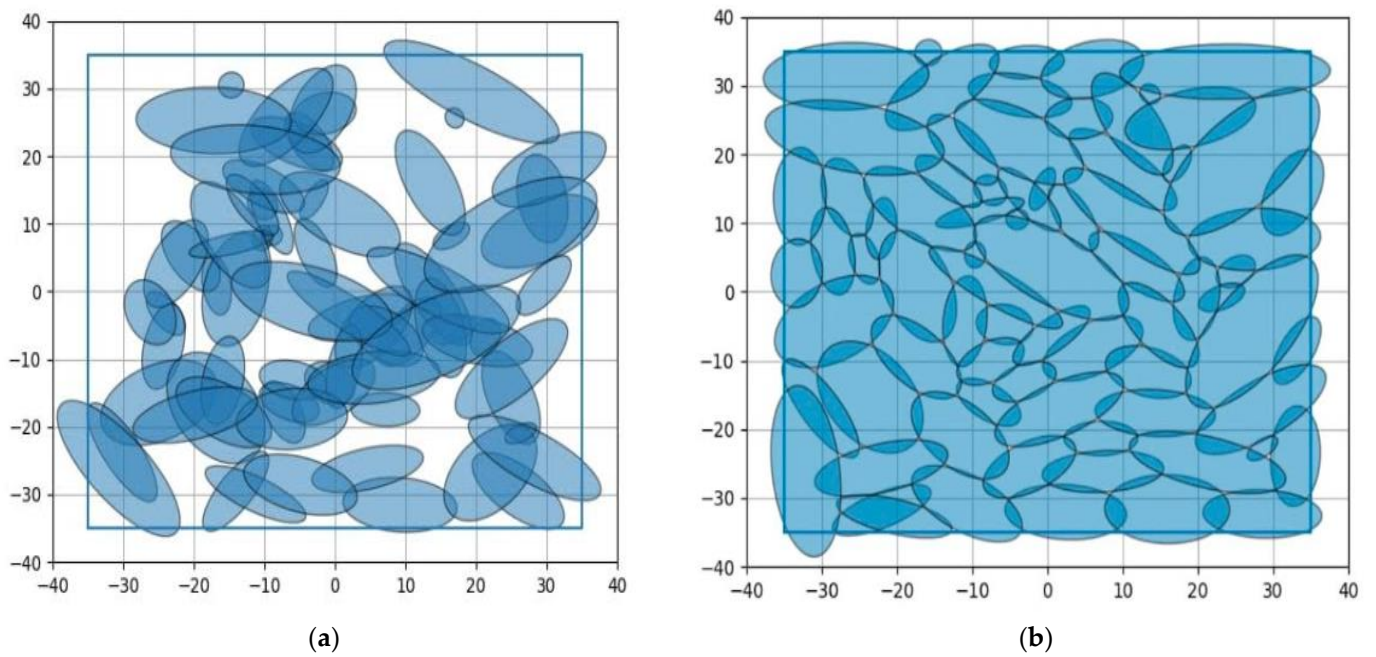
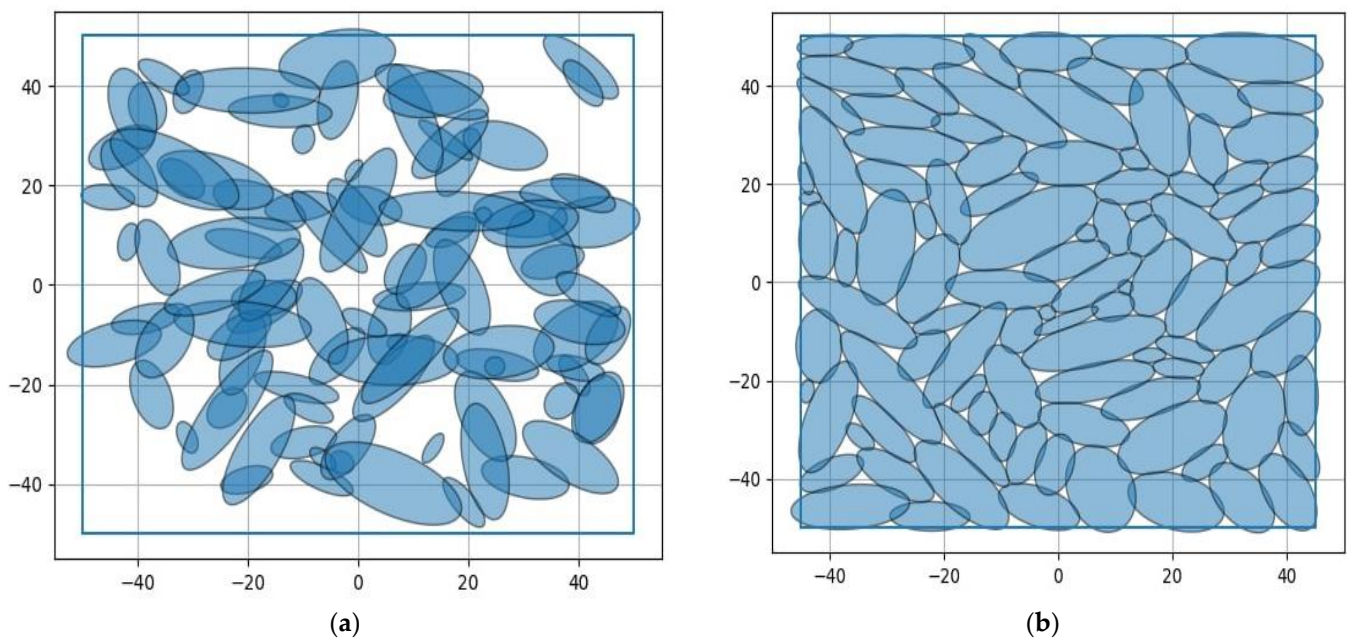


Figure 6. The continuous MCLP of 75 ellipses: (a) starting point and (b) local optimum.



**Figure 7.** The continuous MCLP of 100 ellipses: (a) starting point and (b) local optimum.

Let us pay special attention to the test problems when  $n = 50$  and  $n = 75$ . At  $n = 50$ , due to the large size of the coverage domain, such an placement of ellipses was obtained in which they do not intersect and are located inside the area. This is a global solution to problem (9), which corresponds to the so-called packing configuration [28,29]. For  $n = 75$ , each point of the coverage domain belongs to at least one of the covering objects. This is also a global solution to problem (9), and the resulting location defines the so-called covering configuration [28,29].

## 5. Discussion

The maximum coverage location problem is related to the determination of the placement of objects in order to cover as much as possible of the domain served by them. Such problems are formulated in discrete, continuous, and mixed statements, taking into account their geometric features.

The choice of one or another model depends on the restrictions on the location of objects and the properties of the service area. The classical discrete setting assumes that both object locations and service areas are isolated points. The weakening of the discreteness conditions leads to continuous and mixed formulations.

This article is perhaps one of the first to raise questions of formalization of completely continuous MCLP. In this case, both the locations and the service area are continual and suggest the possibility of their continuous transformation.

Of course, by discretizing the coverage area, for example, by grid methods, one can reduce continuous problems to discrete formulations. However, in this case, we will rather talk about the method of solution, and not about an adequate mathematical statement of the problem.

The geometric features of the continuous MCLP are related to both the shape of the covering objects and the service area. In turn, the problems go beyond the class of classical coverage problems, in which each point of the domain must belong to at least one of the covering objects. Therefore, continuous MCLP require the construction of special mathematical models and optimization methods that take into account their specifics.

The continuous MCLP model as a non-linear unconstrained optimization problem opens up prospects for using both modern effective methods for solving them and existing software in the form of powerful packages for various computer platforms.

The conducted studies allowed us to approach the analysis of the mathematical model of the problem from a unified standpoint. On the one hand, knowing the equations of general position of geometric objects  $S_i(\mathbf{g}^i), i \in J_n^0$ , used in the formation of a complex object  $\Omega(\mathbf{g})$ , one can write down the equation of its boundary. Therefore, formally we will find the analytical form of the function  $\omega_\Omega(\mathbf{g}) = \mu(\Omega(\mathbf{g}))$  depending on the generalized variables  $\mathbf{g}$ . As a result, we will obtain an expression in the form of an integral depending on a vector parameter, for the calculation of which we will have to use numerical methods. This approach is very time consuming and is associated with the consideration of various options for the acceptable location of objects.

The approach proposed in the paper is devoid of these shortcomings. All the difficulties associated with the analytical specification of the function  $\omega_\Omega(\mathbf{g})$  are overcome by using computer geometry packages that work great with geometric objects of complex spatial shape. As a result, it takes milliseconds to calculate the function value  $\omega_\Omega(\hat{\mathbf{g}})$  for the fixed  $\hat{\mathbf{g}}$ , which is confirmed by the experiments described in the Section 4. This justifies the possibility of using metaheuristic methods of global optimization [38,39]. The efficiency of methods increases significantly if it is possible to use local optimization methods. The simplest is the multistart scheme for generation  $\Omega(\hat{\mathbf{g}})$  with subsequent local optimization of the function  $\omega_\Omega(\mathbf{g})$ , choosing  $\hat{\mathbf{g}}$  as the starting point. With limited time, the data illustrated in Figure 2 allows an estimate of the number of iterations in the multistart method. If temporary resources allow, hybrid methods can be proposed, where local optimization is applied after improvements have been received.

The geometrical features of the continuous MCLP model have made it possible to significantly increase the efficiency of local optimization methods using first order differences. This is confirmed by the data shown in Figure 3.

In general, the proposed approach opens up prospects for the development of new methods for solving both packing and covering problems. Indeed, in the course of the experiments, we obtained such locations of objects that corresponded to the packing configuration (Figure 5b) and covering configuration (Figure 6b). Thus, the introduction of additional conditions on the value of the function will allow us to propose new mathematical models for classical packing and covering problems and methods for their solution. In particular, the constraint

$$\omega_\Omega(\mathbf{g}) = \sum_{i=1}^n \mu(S_i), \tag{17}$$

is the condition for both non-intersection of objects  $S_i(\mathbf{g}^i), i \in J_n$ , and their inclusion inside the container  $S_0$ .

The constraint

$$\omega_\Omega(\mathbf{g}) = \mu(S_0), \tag{18}$$

is a condition for covering the domain  $S_0$  by a family of geometric objects  $S_i(\mathbf{g}^i), i \in J_n^0$ .

Note that the choice of the BFGS method for local optimization was based on high user ratings. Of course, further studies of continuous MCLP require a comparative analysis of modern optimization methods, in particular [40]. Potential decision improvements can be obtained by using a novel optimization scheme, such as deep learned recurrent type-3 fuzzy system [41].

Another promising direction is the study of optimization problems of packing and coverage in domains with variable metric parameters. The results described in this article are directly related to this problem as well.

**Author Contributions:** Conceptualization, S.Y.; methodology, S.Y. and O.K.; software, O.K. and D.P.; validation, O.K. and D.P.; formal analysis, S.Y. and O.K.; investigation, O.K. and D.P.; resources, S.Y., O.K. and D.P.; writing—original draft preparation, S.Y.; writing—review and editing, S.Y. and O.K.; visualization, D.P.; supervision, S.Y.; project administration, S.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** The study was funded by the Ministry of Education and Science of Ukraine in the framework of the research project on the topic “Technologies, tools for mathematical modeling, optimization and systematic analysis of coverage problems in space monitoring systems”.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Generated data and test tasks are used.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A

**Table A1.** Metrical parameters of ellipses.

i	a <sub>i</sub>	b <sub>i</sub>	i	a <sub>i</sub>	b <sub>i</sub>	i	a <sub>i</sub>	b <sub>i</sub>	i	a <sub>i</sub>	b <sub>i</sub>
1	1.5	1.3	26	1.5	1.5	51	4.5	2.5	76	4.5	2.0
2	1.9	1.4	27	1.9	1.8	52	4.9	2.8	77	4.9	2.2
3	2.4	2.0	28	2.4	1.2	53	5.4	2.2	78	5.4	3.2
4	2.9	2.0	29	2.9	1.5	54	5.9	3.5	79	5.9	3.5
5	3.3	1.7	30	3.3	1.3	55	6.3	2.3	80	6.3	3.3
6	3.8	1.9	31	3.8	2.9	56	6.8	3.5	81	6.8	2.8
7	4.5	2.0	32	4.5	3.0	57	7.5	3.3	82	7.5	4.0
8	4.9	2.5	33	4.9	2.5	58	7.9	4.5	83	7.9	3.4
9	5.0	2.4	34	5.0	3.4	59	8.0	3.4	84	8.0	4.4
10	5.4	2.3	35	5.4	2.3	60	8.4	2.7	85	8.4	4.1
11	5.9	3.1	36	5.9	2.1	61	8.9	4.1	86	8.9	3.3
12	6.1	3.0	37	6.1	1.5	62	9.1	3.5	87	9.1	4.6
13	6.4	2.8	38	6.4	2.8	63	9.4	3.8	88	9.4	3.2
14	6.6	3.5	39	6.6	2.9	64	9.6	4.5	89	9.6	3.5
15	7.0	3.0	40	7.0	4.0	65	10.1	4.0	90	10.1	5.2
16	7.1	3.5	41	7.1	2.5	66	10.4	2.5	91	10.4	4.5
17	7.3	3.3	42	7.3	3.7	67	10.7	3.7	92	10.7	5.7
18	7.6	2.9	43	7.6	3.9	68	11.0	4.9	93	11.0	3.9
19	7.9	3.5	44	7.9	2.5	69	11.4	3.5	94	11.4	4.5
20	8.1	4.0	45	8.1	3.0	70	11.8	5.0	95	11.8	4.0
21	8.3	5.0	46	8.3	4.0	71	12.1	5.0	96	12.1	4.0
22	8.5	4.5	47	8.5	3.5	72	12.5	4.5	97	12.5	3.5
23	8.8	5.5	48	8.8	4.5	72	13.1	5.5	98	13.1	4.5
24	8.8	3.1	49	8.8	4.1	74	13.6	6.1	99	13.6	4.1
25	9.0	5.9	50	9.0	3.9	75	14.0	4.0	100	14.0	3.9

## References

- Owen, S.H.; Daskin, M.S. Strategic Facility Location: A Review. *Eur. J. Oper. Res.* **1998**, *111*, 423–447. [[CrossRef](#)]
- Hamacher, H.W.; Drezner, Z. (Eds.) *Facility Location: Applications and Theory*; Springer: Berlin, Germany, 2002.
- Church, R.L.; Murray, A.T. *Business Site Selection, Location Analysis, and GIS*; Wiley: New York, NY, USA, 2009.
- Murray, A.T. Advances in location modeling: GIS linkages and contributions. *J. Geogr. Syst.* **2010**, *12*, 335–354. [[CrossRef](#)]
- Eiselt, H.A.; Marianov, V. (Eds.) *Foundations of Location Analysis*; Springer: New York, NY, USA, 2011.
- Murray, A.T. Maximal Coverage Location Problem: Impacts, Significance, and Evolution. *Int. Reg. Sci. Rev.* **2016**, *39*, 5–27. [[CrossRef](#)]

7. Church, R.L.; ReVelle, C. The maximal covering location problem. *Pap. Reg. Sci. Assoc.* **1974**, *32*, 101–118. [[CrossRef](#)]
8. Church, R.L. The planar maximal covering location problem. *J. Reg. Sci.* **1984**, *24*, 185–201. [[CrossRef](#)]
9. Hochbaum, D.S. Approximating Covering and Packing Problems: Set Cover, Vertex Cover, Independent Set, and Related Problems. In *Approximation Algorithms for NP-Hard Problems*; Hochbaum, D.S., Ed.; PWS Publishing Co.: Boston, MA, USA, 1996; pp. 94–143.
10. Bansal, M.; Kianfar, K. Planar Maximum Coverage Location Problem with Partial Coverage and Rectangular Demand and Service Zones. *INFORMS J. Comput.* **2017**, *29*, 152–169. [[CrossRef](#)]
11. Murray, A.T.; Matisziw, T.; Wei, H.; Tong, D. A Geocomputational Heuristic for Coverage Maximization in Service Facility Siting. *Trans. GIS* **2008**, *12*, 757–773. [[CrossRef](#)]
12. Wei, R.; Murray, A.T. Continuous space maximal coverage: Insights, advances and challenges. *Comput. Oper. Res.* **2015**, *62*, 325–336. [[CrossRef](#)]
13. Davari, S.; Zarandi, M.H.F.; Hemmati, A. Maximal covering location problem (MCLP) with fuzzy travel times. *Expert Syst. Appl.* **2011**, *38*, 14535–14541. [[CrossRef](#)]
14. Allahyari, M.Z.; Azab, A. Mathematical modeling and multi-start search simulated annealing for unequal-area facility layout problem. *Expert Syst. Appl.* **2018**, *91*, 46–62. [[CrossRef](#)]
15. Blanco, V.; Gázquez, R. Continuous maximal covering location problems with interconnected facilities. *Comput. Oper. Res.* **2021**, *132*, 105310. [[CrossRef](#)]
16. Murray, A.T.; Tong, D. Coverage optimization in continuous space facility siting. *Int. J. Geogr. Inf. Sci.* **2007**, *21*, 757–776. [[CrossRef](#)]
17. Murray, A.T.; O’Kelly, M.E.; Church, R.L. Regional service coverage modeling. *Comput. Oper. Res.* **2008**, *35*, 339–355. [[CrossRef](#)]
18. Tong, D.; Murray, A.T. Maximising coverage of spatial demand for service. *Pap. Reg. Sci.* **2009**, *88*, 85–97. [[CrossRef](#)]
19. Matisziw, T.C.; Murray, A.T. Siting a facility in continuous space to maximize coverage of a region. *Socio-Econ. Plan. Sci.* **2009**, *43*, 131–139. [[CrossRef](#)]
20. Coll, N.; Fort, M.; Sellarés, J.A. On the overlap area of a disk and a piecewise circular domain. *Comput. Oper. Res.* **2019**, *104*, 59–73. [[CrossRef](#)]
21. Mount, D.M.; Silverman, R.; Wu, A.Y. On the Area of Overlap of Translated Polygons. *Comput. Vis. Image Underst.* **1996**, *64*, 53–61. [[CrossRef](#)]
22. Coll, N.; Fort, M.; Saus, M. Coverage area maximization with parallel simulated annealing. *Expert Syst. Appl.* **2022**, *202*. [[CrossRef](#)]
23. Cheng, S.W.; Lam, C.K. Shape matching under rigid motion. *Comput. Geom. Theory Appl.* **2013**, *46*, 591–603. [[CrossRef](#)]
24. Stoyan, Y.G.; Patsuk, V. Covering a compact polygonal set by identical circles. *Comput. Optim. Appl.* **2010**, *46*, 75–92. [[CrossRef](#)]
25. A Gerrard, R.; Church, R.L.; Stoms, D.M.; Davis, F.W. Selecting conservation reserves using species-covering models: Adapting the ARC/INFO GIS. *Trans. GIS* **1997**, *2*, 45–60. [[CrossRef](#)]
26. García-Palomares, J.C.; Gutiérrez, J.; Latorre, M. Optimizing the location of stations in bike-sharing programs: A GIS approach. *Appl. Geogr.* **2012**, *35*, 235–246. [[CrossRef](#)]
27. Longley, P.A.; Goodchild, M.F.; Maguire, D.J.; Rhind, D.W. *Geographic Information System and Science*, 4th ed.; John Wiley: Chichester, UK, 2015.
28. Stoyan, Y.G.; Yakovlev, S.V. Configuration Space of Geometric Objects. *Cybern. Syst. Anal.* **2018**, *54*, 716–726. [[CrossRef](#)]
29. Yakovlev, S.V. On Some Classes of Spatial Configurations of Geometric Objects and their Formalization. *J. Autom. Inf. Sci.* **2018**, *50*, 38–50. [[CrossRef](#)]
30. Yakovlev, S.V. Formalizing Spatial Configuration Optimization Problems with the Use of a Special Function Class. *Cybern. Syst. Anal.* **2019**, *55*, 581–589. [[CrossRef](#)]
31. Rvachev, V.L. *Theory of the R-Function and Some of Its Applications*; Naukova Dumka: Kyiv, Ukraine, 1982; 552p.
32. Stoyan, Y.; Scheithauer, G.; Gil, N.; Romanova, T.  $\Phi$ -functions for complex 2D-objects. *Q. J. Oper. Res.* **2004**, *2*, 69–84.
33. Bennell, J.; Scheithauer, G.; Stoyan, Y.; Romanova, T. Tools of mathematical modeling of arbitrary object packing problems. *Ann. Oper. Res.* **2010**, *179*, 343–368. [[CrossRef](#)]
34. Stoyan, Y.G.; Romanova, T.; Scheithauer, G.; Krivulya, A. Covering a polygonal region by rectangles. *Comput. Optim. Appl.* **2011**, *48*, 675–695. [[CrossRef](#)]
35. Pankratov, A.; Romanova, T.; Litvinchev, I.; Marmolejo-Saucedo, J.A. An Optimized Covering Spheroids by Spheres. *Appl. Sci.* **2020**, *10*, 1846. [[CrossRef](#)]
36. Gillies, S. The Shapely User Manual. Available online: <https://shapely.readthedocs.io/en/stable/manual.html> (accessed on 29 April 2022).
37. Fletcher, R. *Practical Methods of Optimization*, 2nd ed.; John Wiley Sons: New York, NY, USA, 2013; ISBN 978-0-471-91547-8.
38. Glover, F.; Sorensen, K. Metaheuristics. *Scholarpedia* **2015**, *10*, 6532. [[CrossRef](#)]
39. Salhi, S.; Thompson, J. An Overview of Heuristics and Metaheuristics. In *The Palgrave Handbook of Operations Research*; Salhi, S., Boylan, J., Eds.; Palgrave Macmillan: Cham, Switzerland, 2022. [[CrossRef](#)]
40. Feng, W.; Salgado, A.J.; Wang, C.; Wise, S.M. Preconditioned steepest descent methods for some nonlinear elliptic equations involving p-Laplacian terms. *J. Comput. Phys.* **2017**, *334*, 45–67. [[CrossRef](#)]
41. Cao, Y.; Raise, A.; Mohammadzadeh, A.; Rathinasamy, S.; Band, S.S.; Mosavi, A. Deep learned recurrent type-3 fuzzy system: Application for renewable energy modeling/prediction. *Energy Rep.* **2021**, *7*, 8115–8127. [[CrossRef](#)]