arXiv:2111.13989v2 [cs.CG] 26 Jan 2022

Clustering Geometrically-Modeled Points in the Aggregated Uncertainty Model

Vahideh Keikha[*][†]
The Czech Academy of Sciences
Institute of Computer Science
Prague, Czech Republic
keikha@cs.cas.cz

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

Ali Mohades
Department of Mathematics and Computer Sci.
Amirkabir University of Technology
Tehran, Iran
mohades@aut.ac.ir

Mohammad Ghodsi
Department of Computer Engineering
Sharif University of Technology
Tehran, Iran
ghodsi@sharif.edu

Abstract. The k-center problem is to choose a subset of size k from a set of n points such that the maximum distance from each point to its nearest center is minimized. Let $Q = \{Q_1, \ldots, Q_n\}$ be a set of polygons or segments in the region-based uncertainty model, in which each Q_i is an uncertain point, where the exact locations of the points in Q_i are unknown. The geometric objects such as segments and polygons can be models of a point set. We define the uncertain version of the k-center problem as a generalization in which the objective is to find k points from Q to cover the remaining regions of Q with minimum or maximum radius of the cluster to cover at least one or all exact instances of each Q_i, respectively. We modify the region-based model to allow multiple points to be chosen from a region, and call the resulting model the *aggregated uncertainty model*.

All these problems contain the point version as a special case, so they are all NP-hard with a lower bound 1.822 for the approximation factor. We give approximation algorithms for uncertain k-center of a set of segments and polygons. We also have implemented some of our algorithms on a data-set to show our theoretical performance guarantees can be achieved in practice.

Keywords: k-center Uncertain data Approximation algorithms

[*]Address for correspondence: The Czech Academy of Sciences, Institute of Computer Science, Pod Vodárenskou věží 2, 182 07 Prague, Czech Republic.

[†]Also affiliated at: Department of Computer Science, University of Sistan and Baluchestan, Zahedan, Iran.

1.   Introduction

k-center is a classic problem in the fields of computational geometry, data mining, and approximation algorithms. Suppose a set of n points is given. The goal of metric k-center is to choose a subset of size k called C from a set of n points such that the maximum distance from each point to its nearest center is minimized.

Metric k-center is NP-hard and it has a tight 2-approximation algorithms in metric spaces [1]. According to Theorem 2.1 in [2], the lower bound on the approximation factor of k-center in the Euclidean plane is 1.822.

The data *uncertainty* usually comes from the error in the precision of representing numbers [3], the error in the measurement of input data such as GPS data [4] or other sources. Sometimes the probability of data is known based on previous data or measurement error [5, 6]. Uncertainty sometimes affects the correctness of geometric algorithms, for example, a convex hull might not be convex at all as a result of reduced precision. This has motivated a lot of uncertainty models over the years.

We introduce a new model for uncertainty, which we call the "aggregated uncertainty model". This allows the multiplicity of the realizations of an uncertain point to exceed one, which is a realistic assumption for many applications, including anonymized data, continuous data such as GPS data or sensor outputs, and repeated data which were summarized in the geometric version of the input as a single point. In this paper, when we refer to uncertainty, it is the aggregated region-based uncertainty model, which is the aggregated uncertainty model applied to a set of points in the region-based uncertainty model (we have defined it later in this section).

Through the paper, the covering means the distance between the covered point and its closest center is at most the radius of the covering, and the goal is to minimize the radius of covering. We introduce and study the following two problems: (see Section 3 for formal definitions):

1. **MinMax segments clustering**: covering a set of n segments with k segments as the centers, and the maximum distance from a point on a segment to its center is minimized. And,

2. **Domain-restricted k-center of polygons**: covering the points of a set of polygons by a set of k points inside those polygons, and the maximum distance from a point on a polygon to its center is minimized.

For each problem, we discuss two cases: the maximum version and the minimum version of the problem. In the maximum version, the goal is to cover the whole input, while in the minimum version the goal is to hit each input, i.e. to cover at least one point of each input.

Many variations of k-center have been studied before, in the following, we review the ones most related to our problems as explained below.

k-**center Variations**   In k-*line center* problem [7], the input is a set of lines and the goal is to find k lines (or cylinders in higher dimensions) as centers that minimize the distance from each point to the nearest line. In the same paper, a $(1 + \epsilon)$-approximation algorithm for this problem in $O(n \log n)$ time was given.

Another related problem is stabbing a set of line segments with two congruent squares of minimum size[8]. The authors introduced a $\sqrt{2}$-approximation algorithm with running time $O(n)$. They have

shown that their algorithm can also be applied to find two congruent disks of minimum size to cover the line segments. Note that this is different from the MinMax segments clustering since our objective is to find k segments from the input as centers, such that this selection minimizes the radius of the covering.

A constrained version of our second problem is covering the area of a convex polygon P of n vertices by k centers on its boundary so that each circle has the smallest possible radius [9]. The authors of [9] provide a 1.8841-approximation $O(nk)$ time algorithm, that first computes an approximate smallest bounding box R for P, and solves the problem for R. Then, they translate the centers on the boundary of R, to the points on the boundary of P. Later, a $(1 + \frac{7}{k} + O(\epsilon))$-factor approximation algorithm for any $\epsilon > 0$ and $k \geq 7$ was given, which runs in $O(n^2(\log r_{opt}) + \log \frac{1}{\epsilon})$ time [10].

Uncertainty Models Several formulations have been proposed for modeling data uncertainty over the years, including epsilon geometry [3], the probabilistic model [5, 6], the region-based model [4], and the domain-restricted models [11].

In the region-based model (which is equivalent to the locational model when each uncertain point has a uniform distribution over its region), two types of regions where each point can occur are usually discussed: the continuous and the discrete region. The problem asks for the minimum and maximum solutions, defined as the solutions that work in the luckiest (best possible distribution based on the objective function) and unluckiest (worst possible distribution based on the objective function) of input points.

Several problems are studied in the region-based uncertainty model, where the objective was minimizing or maximizing the size of the area/perimeter of the convex hull [12], the area of the bounding box, the length of width and diameter, and the area of the smallest enclosing disk [13], or the area of the inscribing convex k-gon [14], etc. The smallest area/perimeter convex hull that contains at least one point of each imprecise point also called *polygon transversal*. For a given set of line segments, the smallest area/perimeter convex polygon that intersects all the segments can be computed in $O(n \log n)$ time [15, 16, 17].

Uncertain k-center In the discrete model of uncertainty in which each uncertain point is modeled by a discrete set of points with an assigned probability of occurrence, the k-center problem was studied in [5, 18, 19], where, in all of them, the objective is to minimize the maximum expected distance from the uncertain points. Also, the authors of [5, 19] only consider the problem in 1D space. The 1-center of a set of uncertain points also studied for rectilinear distances [20], where the uncertainty of each point is modeled as a set of discrete points with assigned probability, but the objective is still minimizing the maximum expected rectilinear distance to the uncertain points.

In the following, we briefly compare the introduced uncertainty model with the existing models. In the region-based model, the objective is to compute the maximum and minimum possible feasible solutions, with the existing error bound estimations. In the domain-restricted model, the objective is to just find a class of feasible solutions. But, in the aggregated uncertainty model, the objective is to compute a compact subset of the input which gives a guaranteed approximation for any exact instance of the data, assuming the error bound estimation is given.

Aggregated Uncertainty Model A common event in modeling uncertain data is when multiple points are mapped to a single uncertain point. However, most existing uncertainty models allow only a single point to be chosen from each uncertainty region. We remove this assumption from region-based uncertainty models in our model, which is described in Definition 1.1. Also, in the current study, by an uncertain point we mean a point that is specified by a region in which the point may lie.

Definition 1.1. (Aggregated Uncertainty Model (AGU))
For a set of uncertain points represented with regions $Q_1, \ldots, Q_n$ in $\mathbb{R}^2$ as inputs to a problem $\mathcal{P}$ in the AGU model, a feasible solution for $\mathcal{P}$ is a solution which is feasible for all subsets of points in $\cup_{i=1}^n Q_i$, such that at least one point from each set Q_i is chosen. Indeed each uncertain point in the AGU model is defined by a continuous region, and corresponds to at least one point in the real world.

An optimization problem in this model is solved by computing a minimum-cost solution with at least one point in each region and a maximum-cost solution with the same constraints. i.e, the optimal solution of the minimization version of a problem $\mathcal{P}$ in the AGU model is a minimum of the feasible solutions.

k-**Center in the aggregated uncertainty model.** Since we allow multiple points to be chosen from each uncertainty region, the uncertainty models of the input points and the chosen centers can be different. In the domain-restricted k-center of polygons problem that we discussed in this paper, we assume the minimum cost for the set of centers is intended. More specifically, we allow each center to be an arbitrary point in an input polygon (uncertainty region). Both the minimum and the maximum models for the output of an uncertainty problem, as defined for the region-based uncertainty models, have been discussed for the input points.

Motivation As an application, suppose for a huge set of N data points we are allowed to draw a set of $n \ll N$ regression lines. Then each line segment in the line segment uncertainty model coincides with one of these regression lines (which we have bounded the length of each line by the range of the corresponding input points), where we require to classify all the points based on the computed regression lines.

In the domain-restricted k-center of polygons, we assume each polygon represents a geometrical label, e.g. areas on a map. Conversely, we may have a set of labeled points, and for the points with equal labels, we have computed the corresponding region which here the region is modeled by a polygon. Then the objective is to make a classification for such polygons, which is persistent for any point which lies in the polygons in the future.

Contributions

We introduce a problem which we call multi-interval set cover, which appears as a sub-problem in many problems, e.g., the wireless network problems in which each interval can be seen as a moving sensor with bounded range [21, 22], or admissible choices in a game environment [23].

- In the MinMax segment clustering the objective is that of choosing k segments as centers to cover the remaining regions modeled by a set of segments. We study both the maximization and

the minimization versions that give some bounds on the radius of the clusters. We show that k-center of segments is NP-hard by an approximation-preserving reduction from the set cover problem, so it cannot be approximated by a factor better than $\Omega(\log n)$ in polynomial time. However, the approximation factor of the multi-interval set cover affects only the number of clusters, which results in a bicriteria approximation for the problem. A summary of the results is given in Table 1.

Table 1. The summary of the results. In our bi-criteria approximation for clustering, we assume βk centers are used and the radius is αr. n is the number of input objects. The algorithms that assume the aspect ratio is fixed are marked with ‡.

Problem	Region	Type	β	α	Time	Refs.
MinMax segment clustering ($k = 1$)	Segments	Max	1	1	$O(n^2)$	Alg. 3
MinMax segment clustering	Segments	Max	$O(\log n)$	$1 + \epsilon$	$O(\frac{n^3}{\epsilon^4})$	Alg. 5‡
MinMax segment clustering	Segments	Max	1	≥ 1.822	$O(poly(n))$	[2]
MinMax segment clustering ($k = 1$)	Segments	Min	1	1	$O(n^2)$	Alg. 6
MinMax segment clustering	Segments	Min	$O(\log n)$	$1 + \epsilon$	$O(\frac{n^3}{\epsilon^4})$	Alg. 8

- We prove a special case of the geometric set cover problem, which we call the multi-interval set cover problem is NP-hard, and prove it is NP-hard to approximate it by a factor better than $\Omega(\log n)$.

- In the domain-restricted k-center of polygons in the AGU model, we solve the k-center of a set of polygons which is that of choosing k points from the input as the centers to cover the area of the input polygons. We study both the maximization and the minimization versions that give some bounds on the radius of the clusters over all possible distributions of exact instances. We give constant factor approximation algorithms for this problem. The results are summarized in Table 2.

Table 2. The summary of the results. In our bi-criteria approximation for clustering, we assume βk centers are used and the radius is αr. n is the number of input objects. In polygon clustering problems, N is the total complexity of the input polygons. The algorithms that assume the aspect ratio is fixed are marked with ‡.

Problem	Region	Type	β	α	Time	Refs.
k-Center	Convex Polygons	Max	1	$2 + \epsilon$	$O(N + \frac{1}{\epsilon^2})$	Alg. 10‡
k-Center	Polygons	Max	1	$1 + \frac{\sqrt{3}}{2}$	$O(kN)$	Alg. 11
k-Center	Polygons	Max	1	≥ 1.822	$O(poly(n))$	[2]
k-Center	Polygons	Min	$O(1)$	1	$n^{O(1)} + O(\frac{1}{\epsilon^2})$	Alg. 12‡

- We have implemented our algorithms for solving (1) the maximum version of domain-restricted k-center of polygons of a data-set of size 4,491,143 of labeled points. (2) the existing algorithm to solve this problem. We have not observed any significant loss of accuracy of the optimal solutions by our algorithms in practice.

2. Preliminaries

Bicriteria Approximation and Pseudo-Approximation   In bicriteria approximation [24], two objective functions are approximated simultaneously by the solution of the algorithm. The approximation factor is usually reported as a pair (α, β), where α is the approximation factor of the first objective function and β is the approximation factor of the second objective function.

Smallest Enclosing Disk (SED)   Given is a set P of points in the plane. The aim is to place a point c that minimizes the maximum distance to the points of P. Then c determines the center of the *smallest enclosing disk* of P. Megido [25] showed that the smallest enclosing disk can be computed in linear time.

Uncertain Data   In epsilon-geometry [3] the authors introduced a framework that provides algorithms to determine the perturbation ϵ based on the rounding errors in which the output remains correct. If only the topology of the structure of the output needs to be kept, the value ϵ is called the *tolerance* of the structure [26].

In most probabilistic methods, a probability distribution is assigned to each uncertain point. These models are further divided into the *locational* models and the *existential* models. In the locational model, each uncertain point always exists but its exact location is unknown. Then, the location of each uncertain point will be determined by a specific probability distribution, but in the existential model, each uncertain point has a certain exact location but its existence is unknown. We refer the reader to [27, 28] and the references therein.

The domain-restricted uncertainty models [11, 29, 30] focus on the solution set of the problem and therefore does not require an objective function, unlike the region-based models [12] whose results highly depend on the objective function. For example, the convex hull of the domain-restricted uncertainty model is defined by an interior region and an exterior region, where the convex hull includes the interior region and is contained in the exterior region [11].

In the discrete model, each uncertain point is usually modeled by a discrete set of colored points, and the objective is selecting one point from each color such that a specific measure of points is minimized or maximized. There are several recent studies on this topic, see, e.g., [31].

Smallest Color-Spanning Disk   Given a set of points with different colors, the goal is to find the smallest disk that contains at least one point from each color [32]. There is a $O(\tau n \log n)$ time algorithm, for τ colors by computing the upper envelope of Voronoi surfaces [33, 34].

Colorful k-Center   The colorful k-center is a generalization of k-center where each point has a color and there is a minimum requirement for each color that must be covered in the final solution. A $O(1)$-approximation $n^{O(c)}$ time algorithm for this problem with c colors was given in [35].

ϵ-Net   For any subset $N \subset A$ and $0 \leq \epsilon \leq 1$, and a shape r from range space R, if the inequality $|r \cap A| \geq \epsilon|A|$ implies that r contains at least one point of N, then N is called an ϵ-net [36].

When the range space is a set of disks, the number of independent samples for building an ϵ-net is $O(\frac{1}{\epsilon} \log \frac{1}{\epsilon\delta})$, where $1 - \delta$ is the probability of the resulting set to be an ϵ-net and $\epsilon > 0$ is a given constant (this is a special case of the theorem in [36]).

For computing the set cover of a set of objects with a fixed VC dimension d, there exists an $O(d\log(nd))$-approximation algorithm by using an ϵ-net, where n is the complexity of the optimal solution. However, for a set of polygons of n vertices, the VC dimension at least equals $2n + 1$, which implies that the approximation factor of the set cover equals $O((2n + 1)\log((2n + 1)n))$ in the worst-case.

Geometric Set Cover and Geometric Hitting Set   In the geometric set cover problem, a range space $\Sigma = (X, R)$ is given as the input, where X is a set of points in $\mathbb{R}^d$ and R is a family of subsets of X. The goal is to select a minimum-size subset $C \subseteq R$, such that any point in X is inside a shape in C.

The geometric hitting set problem, finds the smallest subset $H \subseteq X$, such that any shape in R contains at least one point of H.

By reduction from facility location, the problem was proved to be NP-hard for disks in 2D, and $o(\log n)$-approximation algorithms for fat triangles, pseudo-disks, and other fat objects exist [37]. A randomized constant-factor approximation based on the concept of ϵ-nets exists for the geometric hitting set and set cover of disks with $O(n \text{ polylog}(n))$ time in $\mathbb{R}^2$ [38, 39]. The approximation factor of hitting set was later improved to $13.4/\epsilon$ [40]. Several PTAS algorithms for hitting set [41] and set cover [42] of pseudo-disks also exist.

Ply   A set of 1-dimensional intervals $S_i, i = 1, \ldots, n$ are given, and $U = \cup_{i=1}^{n} S_i$ is the union of all input intervals (the universe). The number of intervals $S_i, i = 1, \ldots, n$ that contain a single point of the universe interval is called the ply of that universe [43].

3.   Problem definition

The k-center problem on points can be generalized to multiple versions on segments and polygons. In one of our generalizations, all the points of an object which is chosen as a center can cover other points. We study this model on a set of segments and call it the *MinMax Segment Clustering*. In the other version, only k points chosen from input objects can act as centers and cover other points. We study this model on a set of polygons and call it *Domain-Restricted k-center of polygons*.

3.1.   MinMax segments clustering

Maximum Cost   We are given a set $S = \{s_1, \ldots, s_n\}$ of n segments and we want to choose a subset C of size k of S as the set of centers, such that the distance from any point on any segment in S to its nearest center in C is minimized. In other words, the goal is to cover the points of all segments, using points from k segments. An example of 1-center of a set of segments is shown in Figure 1.

Formally, the distance from a segment s to the centers from C is defined as follows:

$$d(s, C) = \max_{p \in s} \min_{c \in C} d(p, c).$$

The objective function of the k-center is therefore:

$$\min_{\substack{C \subset S, \\ |C|=k}} \max_{\substack{p \in s, \\ s_i \in S}} \min_{c \in C} d(p, c).$$

Note that the distance defined over a set of segments (from other segments) is not a metric space.

Figure 1.   The maximum 1-center of a set of segments with $C = \{s_6\}$. Note that the distance from any point on any segment to s_6 is minimized. The point c is the center of 1-center.

Figure 2.   The minimum 1-center of a set of segments with $C = \{s_6\}$. Note that the distance from at least one point of each segment to s_6 is minimized. The point c determines the center of the 1-center.

Minimum Cost   For a given set of segments, the goal of the *minimum k-center of segments* is to choose k of these segments such that the distance from at least one point of each segment to its nearest center is at most r, and r is minimized. The objective function of the minimum k-center of a set S of segments is as follows:

$$\min_{\substack{C \subset S, \\ |C| \leq k}} \max_{s \in S} \min_{p \in s} \min_{c \in C} d(p, c).$$

See Figure 2 for an example of 1-center of a set of segments.

3.2.   Domain-restricted k-center of polygons

Maximum Cost   Let $Q = \{Q_1, \ldots, Q_n\}$ be a set of polygons. The maximum k-center of polygons solves the k-center problem on a set Q of polygonal shapes by finding a set C of k points inside the input polygons as centers such that the maximum distance from each point inside these polygons to its nearest center is minimized. An example of 2-center of polygons is shown in Figure 3.

Formally, for a set Q of polygons, the radius of clustering is defined as:

$$\max_{Q_i \in Q} \max_{p \in Q_i} \min_{c \in C} d(c, p),$$

and the objective function of the k-center of Q is:

$$\min_{\substack{C \subset (\cup_i Q_i), \\ |C|=k}} \max_{\substack{p \in Q_i, \\ Q_i \in Q}} \min_{c \in C} d(c, p).$$

In the domain-restricted k-center of polygons, the set of centers must lie inside the polygons.

Figure 3. The maximum 2-center of three polygons Q_1, Q_2 and Q_3 with $C = \{p_2, p_3\}$. The maximum distance from each point inside Q_1, Q_2 and Q_3 to its nearest center is minimized.

Figure 4. The minimum 1-center of three polygons with $C = \{p_3\}$. The maximum distances from at least one point inside each of Q_1, Q_2 and Q_3 to the center p_3 are minimized.

Minimum Cost   If we require at least one point of each input polygon to be covered by k centers chosen from points inside the input polygons, we call the problem *minimum k-center of polygons*. An example of 1-center of polygons is shown in Figure 4.

Formally, the objective function of the minimum k-center of a set of polygons Q is:

$$\min_{\substack{C \subset (\cup_i Q_i), \\ |C|=k}} \max_{Q_i \in Q} \min_{p \in Q_i} \min_{c \in C} d(c, p),$$

3.3.   Multi-interval set cover

We define a restricted version of the geometric set cover problem over a set of intervals, and call this special case *multi-interval set cover*. Formally, the problem is defined as follows:

Definition 3.1. (Multi-Interval Set Cover)
A set of n sets of intervals $Q_i, i = 1, \ldots, n$ are given. The goal of multi-interval set cover is to find the minimum size subset of sets $Q_i, i = 1, \ldots, n$, such that $\cup_{i=1}^{n} \cup_{s \in Q_i} s$ is covered.

Two differences between multi-interval set cover problem and the well-known geometric set cover of intervals are first in the elements of the universe, which in the geometric set cover of intervals are points, and in the multi-interval set cover problem are intervals; and second, in the number of intervals in each set: each set in the multi-interval set cover is a set of intervals, while each set in the geometric set cover of intervals is a single interval (and therefore a continuous set of points).

We introduce this problem and prove that it is NP-hard in Definition 4.3. Also, we give approximation algorithms for this problem, using a reduction from set cover in Section 4.1.2.

4.   MinMax segments clustering

Here, we discuss the k-center of a set of segments, where all the points on a set of segments must be within distance at most r to their nearest point on a subset of size k of the segments, such that r is minimized.

The Hardness of MinMax Segment Clustering   Observe that the distance between the line segments is not symmetric as we have illustrated an example in Figure 5. Recall that we define the distance between a segment s_i to s_j as the distance between the furthest point of s_i to the closest point of s_j. Also, the triangle inequality no longer holds for this problem; consider a line segment s_i and two points (degenerate segments) s_j and s_k on a line which is intersecting s_i, so that each of s_j and s_k lies on one side of s_i, and $d(s_i, s_j)$ and $d(s_i, s_k)$ is a very small positive constant, as we have illustrated in Figure 6. Observe that $d(s_j, s_k) > d(s_i, s_j) + d(s_i, s_k)$. Therefore, the proof of metric k-center (including Euclidean k-center) no longer applies to this case, and we need to prove the approximation factor.

Figure 5.   The distance $d(s_i, s_j)$ (shown in green) does not equal to $d(s_j, s_i)$ (shown in purple).

Figure 6.   The distance $d(s_j, s_k)$ (shown in purple) is larger than $d(s_i, s_j) + d(s_i, s_k)$; and the triangle inequality does not hold.

4.1.   The maximum k-center

4.1.1.   The hardness of maximum k-center

We define a version of the set cover problem for intervals and prove its hardness. Then, we use it to prove the hardness of k-center of segments beyond the lower bound 1.822 from the k-center of a set of points [2], which holds for polygons because a point is a degenerate polygon.

Figure 7.   The Minkowski sum of a segment s_i with a disk of radius r. Only the blue segments are covered by selecting s_i as a center.

In Algorithm 1, $\oplus$ denotes the Minkowski sum.

Lemma 4.1. The time complexity of Algorithm 1 is $O(n^2 + T(n))$, where $T(n)$ is the time complexity of the multi-interval set cover.

Proof:
Computing the Minkowski sum of a segment with a disk of radius r takes $O(1)$ time (see Figure 7 for an illustration), and finding the intersections between the resulting shape with $n - 1$ segments takes

---

Algorithm 1 Reduction from k-Center of Segments to Multi-Interval Set Cover

---

Input: A set of segments S, an integer k, a constant r
Output: k segments as centers
1: **for** $s_i \in S$ **do**
2:     **for** $s_j \in S$ **do**
3:         $Q_j \leftarrow Q_j \cup \{$ the part of s_j inside $s_i \oplus$ disk of radius $r\}$.
4: $C=$ the solution of multi-interval set cover with $\{Q_i\}_{i=1}^{n}$ as sets and S as the universal set.
5: return C

$O(n)$ time. The nested for loops are repeated $O(n^2)$ times, each with $O(1)$ time. The last step of the algorithm runs an instance of the multi-interval set cover. So, the overall time complexity of the algorithm is $O(n^2 + T(n))$.   □

Theorem 4.2. Algorithm 1 computes a solution with αk centers, if C is an α-approximation of multi-interval set cover.

Proof:
The set Q_i is the set of segments covered by radius r of a segment s_i. So, the minimum number of the sets $Q_i, i = 1, \ldots, n$ that covers all other segments is the set of centers s_i with the same indices. So, the number of centers is the number of sets in C. So, an α-approximation for multi-interval set cover, is a solution with αk centers and radius r.   □

Based on Algorithm 1, in the clustering problem for segments, the ply of the intervals is equal to the number of segments that can cover the same point on a segment.

4.1.2.   The hardness of multi-interval set cover

In this case, each member of the universal set is an interval and each set of covers is also a set of intervals.

Figure 8.   1-Dimensional intervals a, b, c, such that $c \subset (a \cup b)$.

For three intervals a, b, c such that c is a subset of the union of a and b (See Figure 8), we have:

$$\{a\} \cup \{b\} = \{a\} \cup \{b\} \cup \{c\}.$$

Therefore, the set cover of sets of intervals is different from the set cover of sets of points.

Theorem 4.3. The multi-interval set cover is NP-complete.

Proof:

The decision version of multi-interval set cover can be solved by sorting the intervals in the sets and sweeping the intervals in the universe while keeping the last covered point of the interval. So, the problem is in NP. It remains to prove the NP-hardness.

We proceed by a reduction from set cover. Let U be the universal set and $\mathcal{S}_1, \ldots, \mathcal{S}_n$ be the set of sets in the set cover instance. Map the points of U to the interval $[0, n]$ on the real line, where the interval $[i - 1, i]$ corresponds to the i-the member of U. Any set $\mathcal{S}_i$ can then be represented as a set of intervals; see Figure 7 for an illustration. This is an instance of the multi-interval set cover problem with $[0, n]$ as the universe and the intervals of $\mathcal{S}_i, i = 1, \ldots, n$ as the sets of intervals. Conversely, it is easy to see that the optimal solution of this instance of multi-interval set cover results in an optimal solution of the set cover. So, the current multi-interval set cover instance is equivalent to set cover. □

Theorem 4.4. The reduction of Definition 4.3 is approximation-preserving.

Proof:

In the multi-interval set cover, the objective function is the number of segments in the universal set that are covered by the chosen sets. Each member of the universe in the set cover is an interval in the multi-interval set cover in the reduction of Definition 4.3. Therefore, the objective function of set cover is preserved by the reduction. □

Approximating multi-interval set cover

Based on Definition 4.3, the best possible approximation factor for this problem is $\Omega(\log n)$, as well [1]. For low frequency set cover, f-approximation, where f is the maximum frequency of an element is possible [1].

Algorithm 2 Approximate Multi-Interval Set Cover

Input: A set of 1D intervals $Q_i, i = 1, \ldots, n$
Output: A subset of Q
  1: U= the set of disjoint intervals with endpoints from the endpoints of intervals in $Q_i, i = 1, \ldots, n$
  2: S_i= the set of intervals in U that are covered by the intervals of $Q_i, i = 1, \ldots, n$
  3: Solve set cover for $S_i, i = 1, \cdots, n$
  4: **return** the indices of the sets from the previous step.

Theorem 4.5. Algorithm 2 solves the multi-interval set cover problem with the same approximation factor as the set cover.

Proof:

Based on the construction of set U, the covered intervals are the same as that of $\cup_{i=1}^{n} Q_i$, i.e. $\cup_{u \in U} u = \cup_{i=1}^{n} \cup_{q \in Q_i} q$. The construction of S_i also shows that $\cup_{s \in S_i} s = \cup_{q \in Q_i} q$. So, a set cover for $\cup_{u \in U} u$ and $S_i, i = 1, \ldots, n$ is a set cover for $\cup_{i=1}^{n} \cup_{q \in Q_i} q$ and Q_i, for $i = 1, \cdots, n$.

The opposite also holds, since the unions of intervals in each of the discussed sets are equal to their corresponding sets in the other instance. As a result, the objective function, the solution set and the approximation factor of these problems are preserved.   □

Lemma 4.6. Algorithm 1 takes $O(n^3)$ time, if the $O(\log n)$-approximation algorithm [1] is used.

Proof:
The universe is the union of the intervals resulting from cutting each of the n input intervals at the endpoints of other $n - 1$ intervals. So, it has size at most $O(n^2)$. The number of steps of the set cover algorithm is at most n, since at each step of the algorithm at least one set is chosen. Removing the elements of the universe covered at each step takes $O(n^2)$ time. So, the overall time complexity of the algorithm is $O(n^3)$.   □

The time complexity of the algorithm for low frequency (equivalent to low ply in this setting) is the complexity of solving the corresponding linear program [1], therefore, it is different from the time complexity of the $O(\log n)$-approximation algorithm discussed in Definition 4.6.

4.1.3. Approximating the radius of k-center

1-center   In the case where the input is a set of segments and $k = 1$, the problem is equivalent to compute the SED of the segments whose center is restricted to lie on a segment. The SED in this case be determined by the endpoints of the segments.

Algorithm 3 1-Center

Input: A set of segments S
Output: A center from P
  1: **for** $p \in P$ **do**
  2:     $d_p = \max_{s \in P} \max_{q \in s} d(p, q)$
  3: **return** $\arg\min_{p \in P} d_p$.

Computing the distance from a segment to $n - 1$ other segment takes $O(n)$ time. Algorithm 3 for each input segment, computes its distance to all other segments, so, the total running time of this algorithm is $O(n^2)$.

Now, we compute a set of candidates for approximations of the radius of the k-center of a set of segments. These values are used in the latter sections in a parametric pruning algorithm to find the corresponding set of centers. Note that ϵ needs to be smaller than $d(s_i, s_j)/2$, for any $s_i, s_j \in S$. Computing the smallest pairwise distance takes $O(n \log n)$ time by constructing a Voronoi diagram for segments (Line 1).

Theorem 4.7. Algorithm 4 finds a set that solving the k-center problem on that gives a $(1 + \epsilon)$-approximation for the radius of the k-center of a set of segments.

Algorithm 4 Finding r

Input: A set of segments S, an integer k, a constant ϵ
Output: A set of radius R
  1: Discretize the segments of S with chunks of radius ϵ_0
  2: $R=$ the pairwise distances between any pair of vertices from the previous step.
  3: **return** R

Proof:
The distance between an optimal center and a member of its cluster is modified by at most 2ϵ by discretization, assuming that ϵ is smaller than the minimum of R. The optimal cost is the distance between an endpoint of a segment with a point on another segment. Therefore, after discretization, this distance is at most multiplied by $1 + 2\epsilon$. Scaling ϵ by a constant factor concludes the proof. □

Lemma 4.8. Algorithm 4 takes $O(\frac{1}{\epsilon^4})$ time.

Proof:
The number of points after discretization is $O(\frac{1}{\epsilon^2})$. Taking pairs of these distances can be done in $\binom{O(\frac{1}{\epsilon^2})}{2} = O(\frac{1}{\epsilon^4})$ ways. Computing the distance between a pair takes $O(1)$ time. So, the time complexity of the algorithm is $O(\frac{1}{\epsilon^4})$. □

Approximate k-center

We discretize the segments to find the candidates for r to use the set cover algorithm.

Algorithm 5 Approximate k-Center of Segments

Input: A set of segments S, an integer k, a constant ϵ
Output: A set of k segments as centers C
  1: $R =$ Run Algorithm 4.
  2: **for** $r \in R$ **do**
  3:    $C =$ Run Algorithm 1 using the set cover algorithm of [1].
  4: **return** C

Theorem 4.9. Algorithm 5 gives a solution with $O(k \log n)$ centers and radius $r(1 + \epsilon)$ for k-center of segments.

Proof:
Using Definition 4.5, the $O(\log n)$-approximation for set cover can be used to solve the multi-interval set cover with the same approximation factor. This means that Algorithm 5 finds at most $O(k \log n)$ centers, according toDefinition 4.2. The set of radii from Algorithm 4 contain a $(1+\epsilon)$-approximation for the optimal radius as proved in Definition 4.7. □

Theorem 4.10. Algorithm 5 takes $O(\frac{n^3}{\epsilon^4})$ time.

Proof:
The time complexity of the algorithm is the time complexity of Algorithm 4 which is $O(\frac{1}{\epsilon^4})$ based on Definition 4.8, plus $|R| = O(\frac{1}{\epsilon^4})$ times the complexity of Algorithm 1 using the geometric set cover, which is $O(n^3)$. So, the algorithm takes $O(\frac{n^3}{\epsilon^4})$ time. □

4.2. Minimum cost k-center

In the case where $k = 1$, the objective is to find a segment s_i such that the maximum distance of any point on any other segment to s_i is minimized. This problem is equivalent to compute a radius r for which the Minkowski sum of a segment with the disk of radius r intersects the other segments.

to do: add the Minkowski sum step to algorithm 6

Algorithm 6 Minimum 1-Center of Segments

Input: A set of segments P
Output: A center from P
 1: **for** $p \in P$ **do**
 2:      $d_p = \max_{s \in P} \min_{s \in q} d(p, q)$
 3: **return** $\arg\min_{p \in P} d_p$.

Theorem 4.11. Algorithm 6 finds a minimum 1-center in $O(n^2)$ time.

Proof:
The time required for computing the radius r for which the Minkowski sum of a segment p with the disk of radius r intersects all segments is $O(n)$. So, the overall time complexity of the algorithm is $O(n^2)$. □

We modify the algorithms for k-center of segments (Algorithm 5) to solve the minimum k-center of segments problem.

As before, in Algorithm 7, $\oplus$ denotes the Minkowski sum.

Algorithm 7 Reduction from Minimum k-Center of Segments to Multi-Interval Set Cover
---
Input: A set of segments S, an integer k, a constant r
Output: k segments as centers
 1: **for** $s_i \in S$ **do**
 2:      **for** $s_j \in S$ **do**
 3:          **if** $s_i \oplus$ disk of radius r intersects with s_j **then**
 4:              $Q_j \leftarrow Q_j \cup \{s_j\}$.
 5: $C=$ multi-interval set cover with $\{Q_i\}_{i=1}^n$ as sets and S as the universal set.
 6: return C

Algorithm 8 Approximate Minimum k-Center of Segments

Input: A set of segments S, an integer k, a constant ϵ
Output: A set of k segments as centers C
  1: $R = $ Run Algorithm 4.
  2: **for** $r \in R$ **do**
  3:     $C = $ Run Algorithm 7 using the set cover algorithm of [1].
  4: **return** C

Theorem 4.12. Algorithm 8 gives a solution with $O(k \log n)$ centers and radius $r(1 + \epsilon)$ for k-center of segments.

Proof:
The proof is similar to Definition 4.9, the only difference is that it is enough if one point of a segment is within distance r of a segment, which was modeled in Algorithm 7.   □

Remark. We note that since a segment can be considered as a polygon with zero area, the presented algorithms in the next section can be applied to solve the MinMax segment clustering problem in the AGU model. The difference is that the centers are segments in the MinMax segment clustering, while they are points in the domain-restricted k-center of polygons.

5. Domain-restricted k-center of polygons

5.1. Maximum k-center

For the maximum k-center of polygons for $k = 1$, the center of SED of the vertices might not fall inside any of the polygons. We give a 2-approximation algorithm for this problem (Algorithm 9).

Algorithm 9 1-center

Input: A set of polygons P
Output: A center from P
  1: $c = $ the center of the SED of points of P and the vertices of the polygons in P
  2: **return** the nearest neighbor of c in P

Theorem 5.1. The approximation factor of Algorithm 9 is 2.

Proof:
By definition, the radius of the smallest enclosing disk (r) is a lower bound for 1-cluster. If the center of the smallest enclosing disk falls inside the polygon, that is the optimal solution. The distance from any point to the center of the smallest enclosing disk is at most its radius. Let o denote the center of the optimal 1-center and p be the nearest neighbor (point) of c in P. Then, using triangle inequality,

$$d(p, o) \leq d(o, c) + d(c, p) \leq 2r.$$ □

Theorem 5.2. The time complexity of Algorithm 9 is $O(n)$ expected time and $O(n \log n)$ worst-case time.

Proof:
Computing the nearest neighbor of a point to a set of segments takes $O(n)$ time. Also, the SED can be computed in expected linear time and worst-case $O(n \log n)$ time, so the total time complexity is $O(n)$ expected time and $O(n \log n)$ worst-case time. □

5.1.1. Maximum k-center of convex polygons

If we solve the problem with a naive grid algorithm, we may miss the covering of the sharp corners of the boundary of the polygons. To treat these cases, we first compute the Minkowski sum of a polygon with a square of side length ϵ, and then consider the input on a grid with cell length ϵ. This guarantees at least one vertex is chosen for the sharp corners. See Figure 9 for an illustration. But the rounded polygons on the grid may impose some centers which do not lie on the polygons. To avoid this, we map the vertices of the rounded polygons to the closest point of the polygon to satisfy the domain-restriction assumption (Line 11 and 12 of Algorithm 10). We note that ϵ needs to be smaller than $\frac{r_i}{k}$, where r_i denotes the radius of the SEC of the vertices of P_i, for $i = 1, \ldots, n$.

Figure 9. A polygon P_i and the result of the Minkowski sum of P_i and a square of side length ϵ. The cross points denote the contributed points by P_i on the discretization.

We show that applying any k-center algorithm on the union of the grid points lying within the polygons and the mapped points of the grid to the boundary of the polygons (in sharp corners) solves our problem. If we use the presented algorithm of [1], we come up with a $2 + \epsilon$ approximation.

Theorem 5.3. The approximation factor of Algorithm 10 is $2 + \epsilon$, for fixed aspect ratio and ϵ and its time complexity is $O(\sum_{i=1}^{n} |P_i| + \frac{1}{\epsilon^2})$.

Proof:
Since $\epsilon \leq r_i$, at least one vertex of a grid with cell length ϵ falls inside M_i. Therefore, all the points of P_i are covered by disks of radius ϵ centered at the vertices of the grid. Mapping a grid vertex (x) to its nearest neighbor in P_i (y) still covers the same area of P_i, but with radius 2ϵ. So, a clustering of Y with radius r, is a clustering of X with radius $r + \epsilon$, and a clustering of P_i with radius $r + 2\epsilon$. Let r^* denote the optimal radius of the k-center of $\cup_{i=1}^{n} P_i$. Since Y is a subset of the points of polygons,

---

Algorithm 10 k-Center of Convex Polygons

Input: A set of polygons P, an integer k, a constant ϵ
Output: A set of k centers inside P

  1: **for** $i = 1, \ldots, n$ **do**
  2:      M_i = the Minkowski sum of P_i with a disk of radius ϵ.
  3: build a grid of cell length ϵ
  4: X = compute the vertices of the grid inside the shapes in M
  5: $S = \emptyset$
  6: **for** $x \in X$ **do**
  7:      **if** $x \notin P$ **then**
  8:          **for** $M_i \in M, x \in M_i$ **do**
  9:              $Y = \emptyset$
10:              y = the nearest neighbor of x in P_i.
11:              $Y \leftarrow Y \cup \{y\}$
12:      $S \leftarrow S \cup Y$.
13: C = an approximate k-center of $S \cup X$
14: return C

---

the radius of k-center of Y is less than or equal to the k-center radius of $\cup_{i=1}^{n} P_i$, i.e. $r \leq r^*$. The approximation factor of the algorithm is therefore:

$$\frac{r + 2\epsilon}{r^*} \leq 1 + \frac{2\epsilon}{r^*} \leq 1 + 2\epsilon.$$

For the smallest possible r_i, $i = 1, \ldots, n$ we have $\frac{r_i}{k} \leq 2r^*$ which is because each P_i can be covered by multiple centers, and the radius of the clusters are at least $\frac{r_i}{2k}$. Knowing $\epsilon \leq \min_i \frac{r_i}{k}$, we prove the right side of the inequality. Since we compute a 2-approximation of the k-center of Y, the overall approximation factor is $2(1 + 2\epsilon) = 2 + 4\epsilon$.

Computing the circumscribed circle of P_i takes $O(|P_i|)$ time. So, computing the minimum of radii of these circles takes $O(\sum_{i=1}^{n} |P_i|)$. The time complexity of computing a k-center for the set of grid points is $O(k|Y|) = O(k|X|)$. Also, we have that: $|X| = \sum_{i=1}^{n} \frac{\text{Area}(P_i)}{\epsilon^2}$, where $\text{Area}(P_i)$ denotes the area of polygon P_i. So, the total time complexity of the algorithm is: $O(\sum_{i=1}^{n} |P_i| + \sum_{i=1}^{n} \frac{\text{Area}(P_i)}{\epsilon^2})$. By scaling the bounding box of the input to have a unit area, i.e. $\sum_{i=1}^{n} \text{Area}(P_i) = 1$, and adjusting the value of ϵ to be $\epsilon'/4$, where ϵ' is the input value for ϵ, the bounds in the statement of the theorem are achieved. $\qquad\square$

5.1.2. Maximum k-center of abitrary polygons

For a set of arbitrary polygons, we first compute a triangulation of P_i for $i = 1, \ldots, n$. Let T denote the set of all triangles of P_i for $i = 1, \ldots, n$. We apply Algorithm 10 on the set T, and compute a discrete set of $\Theta(nk)$ points. We show that applying a k-center algorithm with (e.g., with approximation factor 2) on the discrete set gives a $6 + \epsilon$ approximation of the optimal solution.

Algorithm 11 Approximation Algorithm for the k-Center of arbitrary polygons

---

Input: A set of polygons P, an integer k
Output: A set of centers from P
 1: T= the set of triangles in a triangulation of the input polygons
 2: X= the output of Algorithm 10 on the set T
 3: C = approximate k-center of X
 4: C' = the closest point of each center in C to a point of a polygon of P
 5: return C

Theorem 5.4. The approximation factor of Algorithm 11 is $6 + \epsilon$.

Proof:
Let $\mathcal{R}(Z)$ denote the optimal radius of the k-center of the points inside a set of shapes Z. If Z is a set of points, $\mathcal{R}(Z)$ is the optimal radius of the points in Z. If Z is a polygon, $\mathcal{R}(Z)$ denote the radius of the k-center of the points inside Z. If Z is a set of polygons, $\mathcal{R}(Z)$ denote the optimal radius of the k-center of all the points inside the polygons Z. The optimal centers of the k-center of a convex polygon lie inside it, since otherwise there is a center inside it with a lower or equal radius. The set of points of each triangle $T_i \in T$ is a subset of the points of P, so, they are covered by the centers of P with radius at most $\mathcal{R}(P)$. Based on the optimality of $\mathcal{R}(T_i)$, we have:

$$\mathcal{R}(T_i) \leq \mathcal{R}(P).$$

Taking the maximum over all such triangles gives the following bound:

$$\max_{\forall T_i \in T} \mathcal{R}(T_i) \leq \mathcal{R}(P).$$

Similarly, the centers of the k-center of T_i's lie inside P, so, $X \subseteq P$ and we have $\mathcal{R}(X) \leq \mathcal{R}(P)$.

Running Definition 5.1 with approximation factor $2 + \epsilon$ on the triangles, gives the set X. To guarantee the centers chosen by the algorithm lie inside the input polygonal domain, we compute a k-center on set X, where only points of X can be chosen as centers. The approximation factor of this problem with the k-center using arbitrary points of the plane as centers is 2, and $\mathcal{R}(X) \leq \mathcal{R}(T)$. Using a 2-approximation k-center algorithm on X, results in a 4-approximation, and since the approximation factor of Algorithm 10 is $2 + \epsilon$, Algorithm 11, and applying the triangle inequality gives a $(6 + \epsilon)$-approximation for $\mathcal{R}(P)$:

$$\max_{T_i \in T} \mathcal{R}(T_i) + 2\mathcal{R}(X) \leq (2 + \epsilon)\mathcal{R}(P) + 2(2\mathcal{R}(P)) = (6 + \epsilon)\mathcal{R}(P). \qquad \square$$

5.2. Minimum cost k-center

In Algorithm 12, first we build a grid similar to the maximum k-center (Algorithm 10) with resolution $\epsilon = \min(\epsilon, \min_{p,q \in P} \min_{u \in p, v \in q} d(u, v))$, then we use colorful k-center to find a k-center that covers at least one point from each polygon. The limitation is that colorful k-center has only been solved for a constant number of colors in polynomial time [35], which limits the number of input polygons to a constant.

Algorithm 12 Approximate Minimum k-Center of Polygons

Input: A constant size set of polygons P, an integer k, a constant $\epsilon > 0$
Output: A set of centers

1: Build a grid of cell length ϵ
2: $M_i = P_i \oplus$ disk of radius ϵ
3: $T_i =$ the closest point of $\cup_j P_j$ to the vertices of the grid inside M_i, for $P_i \in P$
4: Color the points in T_i, for $P_i \in P$ with color i.
5: $c =$ compute the colorful k-center of the colored points.
6: **return** c

Theorem 5.5. Algorithm 12 is a $O(1)$-approximation for minimum k-center.

Proof:
To guarantee that the vertices of the grid cover the area of each polygon with distance at most $O(\epsilon)$, we compute the grid vertices inside the Minkowski sum of each polygon with the disk of radius ϵ. To satisfy the constraint that the centers must be a subset of input polygons, we replace these points with their nearest neighbors in the input polygons. Computing the solution on the vertices of a grid adds a $1 + \epsilon$ factor to the approximation. The approximation factor of colorful k-center is $O(1)$ for a constant number of colors. □

6. Experimental studies

We implement our algorithm for maximum k-center in the aggregated uncertainty model (Algorithm 11) and compare the results to the k-center of points ([44]) on a big network data-set [45, 46], for $k = 20$ and $\epsilon = 5$, where ϵ is the grid cell length.

The approximation factors of the k-center algorithms for big data is 4 for the metric case [44], and $2 + \epsilon$ for the doubling metrics, including low-dimensional Euclidean space [47, 48]. Since we want a small summary size, we use the 4-approximation algorithm with a summary of size $O(kL)$ instead of the $(2 + \epsilon)$-approximation algorithms with summaries of size $\omega(\frac{kL}{\epsilon^2})$, where L is the number of partitions. Algorithm 11 creates a summary of size $O(\frac{L}{\epsilon^2})$, which is still feasible in practice. Note that ϵ is about $1/10$ times the radius in our experiment and halving the value of ϵ can multiply the size of the summary by at most the doubling dimension of the Euclidean plane, which is 5.

The data-set used in our experiments is the time and location information of check-ins made by users of the Brightkite social network. It has 58,228 users and 4,491,143 check-ins of these users over the period of Apr. 2008 - Oct. 2010. It is available as a part of the SNAP network data-sets [46]. The number of users with at least one check-in the aforementioned period of time is 51,685. Each user had at most 325,821 check-ins.

In then implementation of Algorithm 11, we also use a test data set, that contains the grid points which lie inside the convex polygons. We compare the output of our algorithm on both the test data set and the convex polygons itself.

We consider each check-in data as a point in 2D space. We use the maximum k-center algorithm of polygons to cluster the users into k groups. To summarize the data of each user, we compute their convex hull. As the test data-set, we use the vertices of the grid of cell-length ϵ that fall inside each convex hull as possible future locations of that user. After computing the convex-hulls, the total number of vertices was 189,355, which is an almost 23.7% data-compression ratio.

Figure 10. The approximate k-center of vertices using the 4-approximation k-center algorithm for points [44] for $k = 20$ (blue). The red points are the summary of points for $k = 20$ after the first round, which are the union of k-centers of the partitions.

In Figure 10, the summary and the centers of the algorithm of [44] are shown in red and blue, respectively. Figure 11 shows the centers of Algorithm 11 and the summary of the algorithm of [44]. In Figure 12, the output of Algorithm 11 for clustering the convex polygons is depicted. The red points are the vertices of the grid inside the convex hull of the convex polygons, and the blue points are the centers of the algorithm. The result of the clustering of the points of the grid (test data set) with the centers computed by the algorithm presented in [44] is illustrated in Figure 13.

Figure 11.   The approximate k-center of polygons computed by sampling equidistant points (grid vertices) inside the shapes and then clustering them Algorithm 11, for $k = 20$ and $\epsilon = 5$ (blue). The red points are the summary of 4-approximation k-center of points [44] for $k = 20$ and show the centers computed by this algorithm are good centers for the summary points computed in the first round of [44].

In the figures, only the grid points that lie inside the convex polygons are shown (in red). The computed centers are shown in blue. See Table 3 for the summary of the results.

Table 3.   The summary of the experimental results.

k	ϵ	data-set	alg	summary size	radius
20	-	input	[44]	580,327	$r = 49.7757$
	$\epsilon = 5$	input	Algorithm 11	135,890	$r = 51.76$
	-	test	[44]	580,327	$r = 65.1846$
	$\epsilon = 5$	test	Algorithm 11	135,890	$r = 51.76$

Figure 12. The approximate k-center of polygons by sampling equidistant points (grid vertices) inside the shapes and then clustering them Algorithm 11, for $k = 20$ and $\epsilon = 5$ (blue). The red points are the summary of Algorithm 11 for $\epsilon = 5$ that are shown in the figure as the representatives of the input shapes which show the polygonal shape of the input.

In our experiments, the radius of covering the input points are almost the same, which implies the effective sampling of our algorithm in practice. However, the set of vertices of a grid with cell-length ϵ inside each convex polygon has also been tested for the centers computed by the algorithm of [44], which require a slightly larger radius than the centers computed with our algorithm.

7.   Conclusions and open problems

In this paper, we introduced a new clustering problem so-called MinMax segments clustering. Our algorithms for solving this problem are mostly bicriteria approximations. Extending our model to other shapes such as polygons and disks remains open, as well as improving the approximation factors of our algorithms. Solving these problems in the presence of outliers can also be interesting.

Figure 13.   The approximate k-center of points computed using the 4-approximation algorithm for k-center of points [44], for $k = 20$ and $\epsilon = 5$ (blue). The red points are the summary of Algorithm 11 for $\epsilon = 5$ and approximately represent the input shape. Note that the blue points are the centers of the disks that cover the red points, and the clustering radius is the maximum distance between a red point and its nearest blue point.

Also, we introduced the multi-interval set cover and proved it is NP-hard, and gave approximation algorithms for it, which can be of independent interest.

We generalized the region-based uncertainty model to allow multiple points from each region, to allow multiple centers to be chosen from the same region. Assuming the input points are in the region-based uncertainty model with polygonal regions, we gave bicriteria approximation algorithms for the domain-restricted k-center of polygons in this model.

Further directions of research about aggregated uncertainty models can be adding weights to points, that are useful in problems where the number of points matters, such as k-means and capacitated clustering.

Acknowledgement

References

[1] Vazirani VV. Approximation algorithms. Springer Science & Business Media, 2013. doi:10.1007/978-3-662-04565-7.

[2] Feder T, Greene D. Optimal algorithms for approximate clustering. In: Proc. 20th Annu. ACM Sympos. Theory Comput. ACM, 1988 pp. 434–444. doi:10.1145/62212.62255.

[3] Salesin D, Stolfi J, Guibas L. Epsilon geometry: building robust algorithms from imprecise computations. In: Proc. 5th Annu. ACM Sympos. Comput. Geom. ACM, 1989 pp. 208–217. doi:10.1145/73833.73857.

[4] Löffler M. Data imprecision in computational geometry. Ph.D. thesis, Utrecht Univesity, 2009.

[5] Cormode G, McGregor A. Approximation algorithms for clustering uncertain data. In: Proc. 27th ACM SIGMOD-SIGACT-SIGAI Sympos. Princ. Database Syst. ACM, 2008 pp. 191–200. doi:10.1145/1376916.1376944.

[6] Suri S, Verbeek K, Yıldız H. On the most likely convex hull of uncertain points. In: Proc. 21st Annu. European Sympos. Algorithms. Springer, 2013 pp. 791–802. doi:10.1007/978-3-642-40450-4_67.

[7] Agarwal PK, Procopiuc CM, Varadarajan KR. Approximation algorithms for a k-line center. *Algorithmica*, 2005. **42**(3-4):221–230. doi:10.1007/s00453-005-1166-x.

[8] Sadhu S, Roy S, Nandy SC, Roy S. Linear time algorithm to cover and hit a set of line segments optimally by two axis-parallel squares. *Theoret. Comput. Sci.*, 2019. **769**:63–74. doi:10.1016/j.tcs.2018.10.013.

[9] Du H, Xu Y. An approximation algorithm for k-center problem on a convex polygon. *J. Comb. Optim.*, 2014. **27**(3):504–518. doi:10.1007/s10878-012-9532-5.

[10] Basappa M, Jallu RK, Das GK. Constrained k-center problem on a convex polygon. In: Proc. 15th Int. Conf. on Comput. Sci. Appl. Springer, 2015 pp. 209–222. doi:10.1007/978-3-319-21407-8_16.

[11] Edalat A, Lieutier A, Kashe E. The convex hull in a new model of computation. In: Proc. 13th Canad. Conf. Computat. Geom. 2001.

[12] Löffler M, van Kreveld M. Largest and smallest convex hulls for imprecise points. *Algorithmica*, 2010. **56**(2):235–269. doi:10.1007/s00453-008-9174-2.

[13] Löffler M, van Kreveld M. Largest bounding box, smallest diameter, and related problems on imprecise points. *Comput. Geom.*, 2010. **43**(4):419–433. doi:10.1016/j.comgeo.2009.03.007.

[14] Keikha V, Löffler M, Mohades A. Largest and Smallest Area Triangles on Imprecise Points. *arXiv preprint arXiv:1712.08911*, 2017. doi:10.1016/j.comgeo.2020.101742.

[15] Mukhopadhyay A, Kumar C, Greene E, Bhattacharya B. On intersecting a set of parallel line segments with a convex polygon of minimum area. *Inform. Process. Lett.*, 2008. **105**(2):58–64. doi:10.1016/j.ipl.2007.08.029.

[16] Rappaport D. Minimum polygon transversals of line segments. *Int. J. of Comput. Geom. Appl.*, 1995. **5**(03):243–256.

[17] Boissonnat JD, Lazard S. Convex hulls of bounded curvature. 1996.

[18] Huang L, Li J. Stochastic k-center and j-flat-center problems. In: Proc. 28th ACM-SIAM Sympos. Discrete Algorithms. SIAM, 2017 pp. 110–129. doi:10.1137/1.9781611974782.8.

[19] Wang H, Zhang J. One-dimensional k-center on uncertain data. *Theoret. Comput. Sci.*, 2015. **602**:114–124. doi:10.1007/978-3-319-08783-2_10.

[20] Wang H, Zhang J. Computing the Rectilinear Center of Uncertain Points in the Plane. *Int. J. of Comput. Geom. Appl.*, 2018. **28**(03):271–288. doi:10.1142/S0218195918500073.

[21] Khelifa B, Haffaf H, Madjid M, Llewellyn-Jones D. Monitoring connectivity in wireless sensor networks. In: 2009 IEEE Symposium on Computers and Communications. IEEE, 2009 pp. 507–512. doi:10.1109/ISCC.2009.5202236.

[22] Kloder S, Hutchinson S. Barrier coverage for variable bounded-range line-of-sight guards. In: Proceedings 2007 IEEE International Conference on Robotics and Automation. IEEE, 2007 pp. 391–396. doi:10.1109/ROBOT.2007.363818.

[23] Bopardikar SD, Bullo F, Hespanha JP. On discrete-time pursuit-evasion games with sensing limitations. *IEEE Transactions on Robotics*, 2008. **24**(6):1429–1439. doi:10.1109/TRO.2008.2006721.

[24] Gonzalez TF. Handbook of approximation algorithms and metaheuristics. Chapman and Hall/CRC, 2007. ISBN:978-1-58488-550-4.

[25] Megiddo N. Linear-time algorithms for linear programming in R^3 and related problems. *SIAM J. Comput.*, 1983. **12**(4):759–776.

[26] Abellanas M, Hurtado F, Ramos PA. Structural tolerance and Delaunay triangulation. *Inform. Process. Lett.*, 1999. **71**(5-6):221–227. doi:10.1016/S0020-0190(99)00107-6.

[27] Agarwal PK, Kumar N, Sintos S, Suri S. Range-max queries on uncertain data. *J. Comput. Systems Sci.*, 2018. **94**:118–134. doi:10.1016/j.jcss.2017.09.006.

[28] Xue J, Li Y, Janardan R. On the expected diameter, width, and complexity of a stochastic convex hull. *Comput. Geom.*, 2019. **82**:16–31. 10. doi:1016/j.comgeo.2019.04.002.

[29] Khanban A, Edalat A. Computing Delaunay triangulation with imprecise input data. 2003.

[30] Davari MJ, Edalat A, Lieutier A. The convex hull of finitely generable subsets and its predicate transformer. In: 34th Annu. ACM/IEEE Symp. on Log. in Comput. Sci. IEEE, 2019 pp. 1–14. doi:10.1109/LICS.2019.8785680.

[31] Kazemi MR, Mohades A, Khanteimouri P. On approximability of minimum color-spanning ball in high dimensions. *Discrete Appl. Math.*, 2019. 279:188–191. doi:10.1016/j.dam.2019.10.016.

[32] Abellanas M, Hurtado F, Icking C, Klein R, Langetepe E, Ma L, Palop B, Sacristán V. Smallest color-spanning objects. In: Proc. 9th Annu. European Sympos. Algorithms. Springer, 2001 pp. 278–289. doi:10.1007/3-540-44676-1_23.

[33] Huttenlocher DP, Kedem K, Sharir M. The upper envelope of Voronoi surfaces and its applications. *Discrete Comput. Geom.*, 1993. **9**(3):267–291. doi:10.1007/BF02189323.

[34] Šārîr M, Sharir M, Agarwal PK. Davenport-Schinzel sequences and their geometric applications. Cambridge university press, 1995. ISBN:9780521470254.

[35] Bandyapadhyay S, Inamdar T, Pai S, Varadarajan K. A Constant Approximation for Colorful k-Center. In: Proc. 27th Annu. European Sympos. Algorithms. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019. doi:10.4230/LIPIcs.ESA.2019.12.

[36] Har-Peled S. Geometric Approximation Algorithms. 2005.

[37] Clarkson KL, Varadarajan K. Improved approximation algorithms for geometric set cover. In: Proceedings of the twenty-first annual symposium on Computational geometry. 2005 pp. 135–141. doi:10.1007/s00454-006-1273-8.

[38] Agarwal PK, Pan J. Near-linear algorithms for geometric hitting sets and set covers. In: Proc. 30th Annu. ACM Sympos. Comput. Geom. ACM, 2014 pp. 271–279. doi:10.1145/2582112.2582152.

[39] Agarwal PK, Pan J. Near-linear algorithms for geometric hitting sets and set covers. *Discrete & Computational Geometry*, 2020. **63**(2):460–482. doi:10.1007/s00454-019-00099-6.

[40] Bus N, Garg S, Mustafa NH, Ray S. Tighter estimates for ϵ-nets for disks. *Comput. Geom.*, 2016. **53**:27–35. doi:10.1016/j.comgeo.2015.12.002.

[41] Mustafa NH, Ray S. Improved results on geometric hitting set problems. *Discrete Comput. Geom.*, 2010. **44**(4):883–895. doi:10.1007/s00454-010-9285-9.

[42] Durocher S, Fraser R. Duality for geometric set cover and geometric hitting set problems on pseudodisks. In: Proc. 27th Canad. Conf. Computat. Geom. 2015 pp. 8–16.

[43] Buchin K, Chun J, Löffler M, Markovic A, Meulemans W, Okamoto Y, Shiitada T. Folding free-space diagrams: Computing the Fréchet distance between 1-dimensional curves (multimedia contribution). In: Proc. 33rd Annu. ACM Sympos. Comput. Geom. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.SoCG.2017.64.

[44] Malkomes G, Kusner MJ, Chen W, Weinberger KQ, Moseley B. Fast distributed k-center clustering with outliers on massive data. In: Advances in Neural Information Processing Systems. 2015 pp. 1063–1071. ID:9666048.

[45] Cho E, Myers SA, Leskovec J. Friendship and mobility: user movement in location-based social networks. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. 2011 pp. 1082–1090. doi:10.1145/2020408.2020579.

[46] Leskovec J, Krevl A. SNAP Datasets: Stanford Large Network Dataset Collection. `http://snap.stanford.edu/data`, 2014.

[47] Ceccarello M, Pietracaprina A, Pucci G. Solving kcenter Clustering (with Outliers) in MapReduce and Streaming, almost as Accurately as Sequentially. *Proc. of the VLDB Endow.*, 2019. **12**(7):766–778. doi:10.14778/3317315.3317319.

[48] Aghamolaei S, Ghodsi M. A Composable Coreset for k-Center in Doubling Metrics. In: Conference on Computational Geometry (CCCG 2018). 2018 p. 165. arXiv:1902.01896 [cs.DS].