Topological stability of kinetic k-centers

Citation for published version (APA):
van der Hoog, I., van Kreveld, M., Meulemans, W., Verbeek, K., & Wulms, J. (2019). Topological stability of kinetic k-centers. In S. Nakano, G. K. Das, P. S. Mandal, & K. Mukhopadhyaya (Eds.), *WALCOM: Algorithms and Computation - 13th International Conference, WALCOM 2019, Proceedings* (pp. 43-55). (Lecture Notes in Computer Science; Vol. 11355). Springer. https://doi.org/10.1007/978-3-030-10564-8_4

DOI:
10.1007/978-3-030-10564-8_4

Document status and date:
Published: 01/01/2019

Document Version:
Accepted manuscript including changes made at the peer-review stage

Topological stability of kinetic k-centers*

Ivor Hoog v.d.[1], Marc van Kreveld[1], Wouter Meulemans[2], Kevin Verbeek[2], and Jules Wulms[2]

[1] Dept. of Information and Computing Sciences, Utrecht University, The Netherlands
{i.d.vanderhoog,m.j.vankreveld}@uu.nl
[2] Dept. of Mathematics and Computer Science, TU Eindhoven, The Netherlands
{w.meulemans,k.a.b.verbeek,j.j.h.m.wulms}@tue.nl

Abstract. We study the k-center problem in a kinetic setting: given a set of continuously moving points P in the plane, determine a set of k (moving) disks that cover P at every time step, such that the disks are as small as possible at any point in time. Whereas the optimal solution over time may exhibit discontinuous changes, many practical applications require the solution to be *stable*: the disks must move smoothly over time. Existing results on this problem require the disks to move with a bounded speed, but this model is very hard to work with. Hence, the results are limited and offer little theoretical insight. Instead, we study the *topological stability* of k-centers. Topological stability was recently introduced and simply requires the solution to change continuously, but may do so arbitrarily fast. We prove upper and lower bounds on the ratio between the radii of an optimal but unstable solution and the radii of a topologically stable solution—the topological stability ratio—considering various metrics and various optimization criteria. For $k = 2$ we provide tight bounds, and for small $k > 2$ we can obtain nontrivial lower and upper bounds. Finally, we provide an algorithm to compute the topological stability ratio in polynomial time for constant k.

Keywords: Stability analysis · time-varying data · facility location.

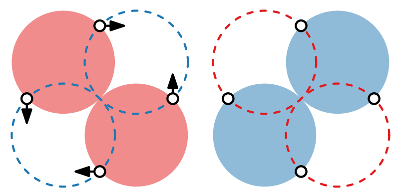
1 Introduction

The *k-center problem* or *facility location problem* asks for a set of k disks that cover a given set of n points in the plane, such that the radii of the disks are as small as possible. The problem can be interpreted as placing a set of k facilities (e.g. stores) such that the distance from every point (e.g. client) to the closest facility is minimized. Since the introduction of the k-center problem by Sylvester [20] in 1857, the problem has been widely studied and has found many applications in practice. Although the k-center problem is NP-hard if k is part of

* W. Meulemans and J. Wulms are (partially) supported by the Netherlands eScience Center (NLeSC) under grant number 027.015.G02. K. Verbeek is supported by the Netherlands Organisation for Scientific Research (NWO) under project no. 639.021.541. A full version of the paper can be found at https://arxiv.org/abs/1810.00794

the input [15], efficient algorithms have been developed for small fixed k. Using rectilinear distance, the problem can be solved in $O(n)$ time [6,13,19] for $k = 2, 3$ and in $O(n \log n)$ time [17,18] for $k = 4, 5$. The problem becomes harder in Euclidean distance, and the currently best known algorithm for Euclidean 2-centers runs in $O(n \log^2 n (\log \log n)^2)$ time [3].

In recent decades there has been an increased interest, especially in the computational geometry community, to study problems for which the input points are moving, including the k-center problem. These problems are typically studied in the framework of *kinetic data structures* [1], where the goal is to efficiently maintain the (optimal) solution to the problem as the points are moving. The kinetic version of the k-center problem also finds a lot of practical applications in, for example, mobile networks and robotics.



A number of kinetic data structures have been developed for maintaining (approximate) k-centers [4,9,10,11], but in a kinetic setting another important aspect starts playing a role: *stability*. In many practical applications, e.g., if the disks are represented physically, or if the disks are used for visualization, the disks should move smoothly as the points are moving smoothly. As the optimal k-center may exhibit discontinuous changes as points move (see figure), we need to resort to approximations to guarantee stability.

The natural and most intuitive way to enforce stability is as follows. We assume that the points are moving at unit speed (at most), and bound the speed of the disks. Durocher and Kirkpatrick [8] consider this type of stability for Euclidean 2-centers and show that an approximation ratio of $8/\pi \approx 2.55$ can be maintained when the disks can move with speed $8/\pi + 1$. For k-centers with $k > 2$, no approximation factor can be guaranteed with disks of any bounded speed [7]. Similarly, in the black-box KDS model, de Berg et al. [2] show an approximation ratio of 2.29 for Euclidean 2-centers with maximum speed $4\sqrt{2}$.

However, this natural approach to stability is typically hard to work with and difficult to analyze. This is caused by the fact that several aspects are influencing what can be achieved with solutions that move with bounded speed:

1. How is the quality of the solution influenced by enforcing continuous motion?
2. How "far" apart are combinatorially different optimal (or approximate) solutions, that is, how long does it take to change one solution into another?
3. How often can optimal (or approximate) solutions change their combinatorial structure?

Ideally we would use a direct approach and design an algorithm that (roughly) keeps track of the optimal solution and tries to stay as close as possible while adhering to the speed constraints. However, especially the latter two aspects make this direct approach hard to analyze. It is therefore no surprise that most (if not all) approaches to stable solutions are indirect: defining a different structure that is stable in nature and that provides an approximation to what we really want to compute. Although interesting in their own right, such indirect approaches have several drawbacks: (1) techniques do not easily extend to other problems, (2) it is hard to perform better (or near-optimal) for instances where the optimal solution is already fairly stable, and (3) these approaches do not offer much theoretical insight in how optimal solutions (or, by extension, approximate solutions) behave as the points are moving. To gain a better theoretical insight in stability, we need to look at the aspects listed above, ideally in isolation.

Recently, Meulemans et al. [16] introduced a new framework for algorithm stability. This framework includes the natural approach to stability described above (called *Lipschitz stability* in [16]), but it also includes the definition of *topological stability.* An algorithm is topologically stable if its output behaves continuously as the input is changing. The topological stability ratio of a problem is then defined as the optimal approximation ratio of any algorithm that is topologically stable. A more formal definition is given below.

Due to the fact that it allows arbitrary speed, topological stability is mostly interesting from a theoretical point of view: it provides insight into the interplay between problem instances, solutions, and the optimization function; an insight that is invaluable for the development of stable algorithms. Nonetheless, topological stability has practical uses: an example of a very fast and stable change in a visualization can be found when opening a minimized application in most operating systems. The transition starts with the application having a very small size, even as small as a point. The application quickly grows to its intended size in a very smooth and fluid way, to help the user grasp what is happening.

k-center problem. An instance of the k-center problem arises from three choices to obtain variants of the problem: the number k of covering shapes, the geometry of the covering shapes and the criterion that measures solution quality. In this paper, we consider covering shapes in the *Euclidean* model, where the covering shapes are disks. The radius of a covering shape is the distance from its center to its boundary under L_2. Furthermore, the quality of a solution is the maximum radius of its covering shapes, the optimization criterion is to minimize this maximum radius. To refer to this problem, we use the notation k-EC-minmax.

Topological stability. Let us now interpret topological stability, as proposed in [16], for the k-centers problem. Let $\mathcal{I}$ denote the input space of n (stationary) points in $\mathbb{R}^2$ and $\mathcal{S}^k$ the solution space of all configurations of k disks or squares of varying radii. Let Π denote the k-center problem with minmax criterion $f\colon \mathcal{I}\times \mathcal{S}^k \to \mathbb{R}$. We call a solution in $\mathcal{S}^k$ valid for an instance in $\mathcal{I}$ if it covers all points of the instance. An optimal algorithm OPT maps an instance of $\mathcal{I}$ to a solution in $\mathcal{S}^k$ that is valid and minimizes f.

To define instances on moving points and move towards stability, we capture the continuous motion of points in a topology $\mathcal{T}_\mathcal{I}$; an instance of moving points is then a path $\pi\colon [0,1] \to \mathcal{I}$ through $\mathcal{T}_\mathcal{I}$. Similarly, we capture the continuity of solutions in a topology $\mathcal{T}^k_\mathcal{S}$, of k disks or squares with continuously moving centers and radii. A *topologically stable* algorithm $\mathcal{A}$ maps a path π in $\mathcal{T}_\mathcal{I}$ to a path in $\mathcal{T}_{\mathcal{S}}^k$.[3] We use $\mathcal{A}(\pi, t)$ to denote the solution in $\mathcal{S}^k$ defined by $\mathcal{A}$ for the points at time t. The stability ratio of the problem Π is now the ratio between the best stable algorithm and the optimal (possibly nonstable) solution:

$$\rho_{\mathrm{TS}}(\Pi, \mathcal{T}_{\mathcal{I}}, \mathcal{T}_{\mathcal{S}}^k) = \inf_{\mathcal{A}} \sup_{\pi \in \mathcal{T}_{\mathcal{I}}} \sup_{t \in [0,1]} \frac{f(\pi(t), \mathcal{A}(\pi, t))}{f(\pi(t), \mathrm{OPT}(\pi(t)))}$$

where the infimum is taken over all topologically stable algorithms that give valid solutions. For a minimization problem ρ_{TS} is at least 1; lower values indicate better stability.

Contributions. In this paper we study the topological stability of the k-center problem. Although the obtained solutions are arguably not stable, since they can move with arbitrary speed, we believe that analysis of the topological stability ratio offers deeper insights into the kinetic k-center problem, and by extension, the quality of truly stable k-centers.

In Section 2, we prove various bounds on the topological stability for this problem. The ratio is $\sqrt{2}$ for $k = 2$; for arbitrary k, we prove an upper bound of 2 and a lower bound that converges to 2 as k tends to infinity. For small k, we show an upper bound strictly below 2 as well. In Section 3, we provide an algorithm to compute the cost of enforcing topological stability for an instance of moving points in polynomial time for constant k. Some proofs in the upcoming sections are sketched or omitted, while the details are described in the full version.

2 Bounds on topological stability

As illustrated above, some point sets have more than one optimal solution. If we can transform an optimal solution into another, by growing the covering disks or squares at most (or at least) a factor r, we immediately obtain an upper bound (or respectively a lower bound) of r on the topological stability. To analyze topological stability of k-center, we therefore start with an input instance for which there is more than one optimal solution, and continuously transform one optimal solution into another. This transformation allows the centers to move along a continuous path, while their radii can grow and shrink. At any point during this transformation, the intermediate solution should cover all points of the input. The maximum approximation ratio r that we need for such a transformation, gives a bound on the topological stability of k-center. We can simply consider the input to be static during the transformation, since for topological stability the solution can move arbitrarily fast. We start by introducing some tools to help us model and reason about these transformations.

[3] Whereas [16] assumes the black-box model, we allow omniscient algorithms, knowing the trajectories of the moving points beforehand. That is, the algorithm may use knowledge of future positions to improve on stability. This gives more power to stable algorithms, potentially decreasing the ratio. However, our bounds do not use this and thus are also bounds under the black-box model.

2-colored intersection graphs. Consider a point set P and two sets of k convex shapes (disks, squares, ...), such that each set covers all points in P: we use R to denote the one set (red) and B to denote the other set (blue). We now define the *2-colored intersection* graph $G_{R,B} = (V, E)$: each vertex represents a shape ($V = R \cup B$) and is either red or blue; E contains an edge for each pair of differently colored, intersecting shapes. A 2-colored intersection graph always contains equally many red nodes as blue nodes. Both colors in a 2-colored intersection graph must cover all points: there may be points only in the area of intersection between a blue and red shape. In the remainder, we use intersection graph to refer to 2-colored intersection graphs.

Lemma 1. *Consider two sets R and B of k convex translates each covering a point set P. If intersection graph $G_{R,B}$ is a forest, then R can morph onto B without increasing the shape size, while covering all points in P.*

Proof (sketch). We can always find a red leaf by a counting argument, which can then morph freely onto its blue neighbor. This removes these two nodes from $G_{R,B}$; repeating this argument gives a morph from R onto B. □

Without loss of generality, we assume here that the disks all have the same radius. We first need a few results on (static) intersection graphs, to argue later about topological stability.

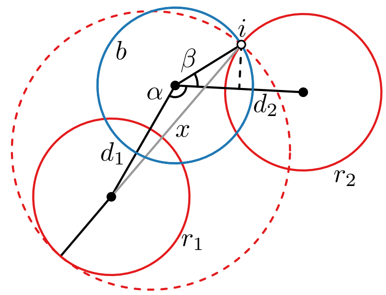
Lemma 2. *Let R and B to be optimal solutions to a point set P for k-EC-minmax. Assume the intersection graph $G_{R,B}$ has a 4-cycle with a red degree-2 vertex. To transform R in such a way that $G_{R,B}$ misses one edge of the 4-cycle, while covering the area initially covered by both sets, it is sufficient to increase the disk radius of a red disk by a factor $\sqrt{2}$.*

Proof (sketch). To morph from R to B, a red disk r_1 has to grow to cover the intersection of an adjacent blue disk b with the other (red) neighbor r_2 of b. This allows r_2 to freely morph to the next adjacent blue disk, after which the intersection graph no longer has the 4-cycle.

Let D be the distance from point in an intersection of r_1 with b and the furthest point in an intersection of r_2 with an adjacent blue disk. One can conclude from the triangle inequality that for any pair of optimal solutions R and B that form a 4-cycle, D is at most $\sqrt{2}$ times the radius of a disk in R or B. □

Lemma 3. *Let R and B to be optimal solutions to a point set P for k-EC-minmax. Assume the intersection graph $G_{R,B}$ has only degree-2 vertices. To transform the disks of R onto B, while covering the area initially covered by both sets, it is sufficient to increase the disk radius by a factor $\left(1 + \sqrt{1 + 8\cos^2(\frac{\pi}{2k})}\right)/2$.*

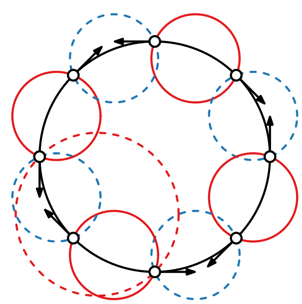
Proof (sketch). To morph from R to B, a red disk r_1 has to grow to cover the intersection of an adjacent blue disk b with the other (red) neighbor r_2 of b (see dashed red disk in figure). We grow r_1 to fully cover its initial disk and the intersection between b and r_2. As a result, we now have to consider only r_1, b, r_2 without concerning ourselves with the other neighbor of r_1 or r_2.



We then assume that r_1 is the red disk that has to grow the least of all red disks in the instance. This allows us to make assumptions on the distances d_1 and d_2 between the center points of r_1 to b and from b to r_2 respectively and the angle α between them at b. In the worst case the instance is symmetric so that $d_1 = d_2 = d$ and $\alpha = \frac{\pi(k-1)}{k}$. Furthermore, for angle β we show that $\cos(\beta) = d/2$ and $\alpha + 2\beta = \pi$ in the worst case. We can finally find the radius of the dashed red disk by calculating x using the Law of Cosines: $x^2 = d^2 + 1^2 - 2d\cos(\alpha + \beta)$. □

Lemma 4. *Let R and B to be optimal solutions to a point set P for k-EC-minmax. Assume the intersection graph $G_{R,B}$ has only degree-2 vertices. To transform the disks of R onto B, while covering the area initially covered by both sets, it may be necessary to increase the disk radius by a factor $2\sin(\frac{\pi(k-1)}{2k})$.*

Proof. Consider a point set of $2k$ points, positioned such that they are the corners of a regular $2k$-gon with unit radius, i.e., equidistantly spread along the boundary of a unit circle. There are now exactly two optimal solutions (see figure).



To morph from R to B, one of the red disks r_1 has to grow to cover the intersection of an adjacent blue disk b with the other (red) neighbor r_2 of b (see dashed red disk in the figure). Since the points are at equal distance from each other on a unit circle, they are the vertices of a regular $2k$-gon. The diameter of the disks in our optimal solution equals the length of a side of this regular $2k$-gon. This means that a red disk has to grow such that its diameter is equal to the distance between a vertex of the $2k$-gon and a second-order neighbor. Hence, r_1 has to grow to with a factor $2\sin(\frac{\pi(k-1)}{2k})$. □

We are now ready to prove bounds on the topological stability of kinetic k-center. The upcoming sequence of lemmata establishes the following theorem.

Theorem 1. *For k-EC-minmax, we obtain the following bounds:*

- $\rho_{\mathrm{TS}}(2\text{-}EC\text{-}minmax, \mathcal{T}_{\mathcal{I}}, \mathcal{T}_{\mathcal{S}}^2) = \sqrt{2}$
- $\sqrt{3} \leq \rho_{\mathrm{TS}}(3\text{-}EC\text{-}minmax, \mathcal{T}_{\mathcal{I}}, \mathcal{T}_{\mathcal{S}}^3) \leq \left(1 + \sqrt{7}\right)/2$
- $2\sin(\frac{\pi(k-1)}{2k}) \leq \rho_{\mathrm{TS}}(k\text{-}EC\text{-}minmax, \mathcal{T}_{\mathcal{I}}, \mathcal{T}_{\mathcal{S}}^k) \leq 2$ *for* $k > 3$.

Lemma 5. $\rho_{\mathrm{TS}}(k\text{-}EC\text{-}minmax, \mathcal{T}_{\mathcal{I}}, \mathcal{T}_{\mathcal{S}}^k) \leq 2$ *for* $k \geq 2$.

Proof. Consider a point in time t where there are two optimal solutions; let R denote the solution that matches the optimal solution at $t-\varepsilon$ and B the solution at $t+\varepsilon$ for arbitrarily small $\varepsilon > 0$. Let C be the maximum radius of the disks in R and in B. Furthermore, let intersection graph $G_{R,B}$ describe the above situation. First we make a maximal matching between red and blue vertices that are adjacent in $G_{R,B}$, implying a matching between a number of red and blue disks. The intersection graph of the remaining red and blue disks has no edges, and we match these red and blue disks in any way.

We find a bound on the topological stability as follows. All the red disks that are matched to blue disks they already intersect grow to overlap their initial disk and the matched blue disk. Now the remaining red disks can safely move to the blue disks they are matched to, and adjust their radii to fully cover the blue disks. Finally, all red disks shrink to match the size of the blue disk they overlap to finish the morph (since each blue disk is now fully covered by the red disk that eventually morphs to be its equal). When all the red disks are overlapping blue disks, the maximum of their radii is at most $2C$, since the radius of each red disk grows by at most the radius of the blue disk it is matched to. □

Lemma 6. $\rho_{\mathrm{TS}}(k\text{-}EC\text{-}minmax, \mathcal{T}_{\mathcal{I}}, \mathcal{T}_{\mathcal{S}}^{k}) \geq 2\sin(\frac{\pi(k-1)}{2k})$ *for* $k \geq 2$.

Proof (sketch). The bound follows from Lemma 4, if we can show that a set of *moving* points that actually force this swap to happen. We let points moving on tangents of the circle defining the $2k$ points, to arrive at this situation at a time t, while ensuring that a swap before or after t would be more costly. □

Lemma 7. $\rho_{\mathrm{TS}}(2\text{-}EC\text{-}minmax, \mathcal{T}_{\mathcal{I}}, \mathcal{T}_{\mathcal{S}}^{2}) = \sqrt{2}$.

Proof. The lower bound follows directly from Lemma 6 by using $k = 2$. For the upper bound, consider a point in time t where there are two optimal solutions; let R denote the solution that matches the optimal solution at $t - \varepsilon$ and B the solution at $t + \varepsilon$ for arbitrarily small $\varepsilon > 0$. If $G_{R,B}$ is a forest, Lemma 1 applies and we do not need to increase the maximum radius during the morph. If $G_{R,B}$ contains a cycle, the entire graph must be a 4-cycle. Lemma 2 gives an upper bound of $\sqrt{2}$ for transforming the intersection graph $G_{R,B}$ to no longer have this 4-cycle, resulting in a tree. Finally, we can morph R into B without further increasing the maximum radius using Lemma 1. □

Lemma 8. $\sqrt{3} \leq \rho_{\mathrm{TS}}(3\text{-}EC\text{-}minmax, \mathcal{T}_{\mathcal{I}}, \mathcal{T}_{\mathcal{S}}^{3}) \leq \left(1 + \sqrt{7}\right)/2$.

Proof (sketch). A case distinction can be made on how the intersection graph looks: If the intersection graph is a forest or there is a 6-cycle, we can respectively use Lemma 1 or Lemma 3 for the upper and Lemma 4 for the lower bound. However, in the remaining cases we carefully analyze how cycles can be broken until Lemma 1 can be applied. □

The above proof shows the strengths and weaknesses of the earlier lemmata. While in many cases we can get close to tight bounds, dealing with high degree vertices in the intersection graph requires additional analysis. Furthermore, in general we cannot upper bound the approximation factor needed for stable solutions with bounded speed [7], but Theorem 1 can act as a lower bound for such bounded speed solutions.

3 Algorithms for *k*-center on moving points

Topological stability captures the worst-case penalty that arises from making transitions in a solution continuous. In this section we are interested in the corresponding algorithmic problems that result in instance optimal penalties: how efficiently can we compute the (unstable) k-center for an instance with n moving points, and how efficiently can we compute the stable k-center? When we combine these two algorithms, we can determine for any instance how large the penalty is when we want to solve a given instance in a topologically stable way.

The second algorithm gives us a topologically stable solution to a particular instance of k-center. This solution can be used in a practical application requiring stability, for example as a stable visualization of k disks covering the moving points at all time. Since we are dealing with topological stability, the solution can sometimes move at arbitrary speeds. However, in many practical cases, we can alter the solution in a way that bounds the speed of the solution and makes the quality of the k-center only slightly worse.

3.1 An unstable Euclidean *k*-center algorithm

Let P be a set of n points moving in the plane, each represented by a constant-degree algebraic function that maps time to the plane. We denote the point set at time t as $P(t)$ and we want to find the optimal set of k minimum covering disks that cover $P(t)$, denoted $\mathcal{B}^*(t)$. Observe that we can define $\mathcal{B}^*$ as the Cartesian product of k triples, pairs, and singletons of distinct points from the set $P(t)$. Not every triple is always relevant: if the circumcircle of the three points is not the boundary of the smallest covering disk, then the triple is irrelevant at that time. This formalization allows us to define what we call *candidate* k-centers.

Definition 1. *Any set of k disks $D_1, \ldots, D_k$ where each disk is the minimum covering disk of one, two or three points in $P(t)$ is called a **candidate k-center** and is denoted $\mathcal{B}(t)$. A candidate k-center is **valid** if the union of its disks cover all points of $P(t)$.*

This definition allows us to rephrase the goal of the algorithm: for each time t we want to compute the smallest value $C(t)$, such that there exists a valid candidate k-center $\mathcal{B}(t)$ where $C(t)$ is the cost in the minmax model. We can see these costs changing over time as curves in a graph that maps time to radii. There are $O(n^3)$ such curves. Using an analysis of the arrangement, lower envelope computation [12], and static k-center algorithms [5,14], we can show:

Theorem 2. *Given a set of n points whose positions in the plane are determined by constant-degree algebraic functions, the minmax Euclidean k-center problem can be solved in $O(n^{2k+5})$ or $n^{O(\sqrt{k})}$ time.*

3.2 A stable Euclidean *k*-center algorithm

Intuitively, the unstable algorithm finds the lower envelope of all the *valid* radii by traversing the arrangement of all valid radii over time. At each time t a minimal enclosing disk D_1 (defined by a set of at most three points) in the set of optimal disks $\mathcal{B}^*(t)$ needs to be replaced with a new disk D_2, we "hop" from our previous curve to the curve corresponding to D_2. If we require that the algorithm is topologically stable these hops have a cost associated with them.

We first show how to model and compute the cost $C(t)$ of a topological transition between any two k-centers at a fixed time t. We then extend this approach to work over time. Let t be a moment in time where we want to go from one k-center $\mathcal{B}_1$ to another candidate k-center $\mathcal{B}_2$. The transition can happen at infinite speed but must be continuous. We denote the infinitesimal time frame around t in which we do the transition as $[0, T]$. We extend the concept of a k-center with a corresponding partition of P over the disks in the k-center:

Definition 2 (Disk set). *For each disk D_i of a candidate k-center $\mathcal{B}$ for $P(t)$ we define its **disk set** $P_i \subseteq P(t) \cap D_i$ as the subset of points assigned to D_i. A candidate k-center $\mathcal{B}$ with disk sets $P_1, \ldots, P_k$ is **valid** if the disk sets partition $P(t)$. We say $\mathcal{B}$ is **valid** if there exist disk sets $P_1, \ldots, P_k$ such that $\mathcal{B}$ with disk sets $P_1, \ldots, P_k$ is valid.*

k-centers with disk sets will change in the time interval $[0, T]$ while the points $P(t)$ do not move. In essence the time t is equivalent to the whole interval $[0, T]$. For ease of understanding we use t' to denote any time in the interval $[0, T]$. Observe that our definition for a topologically stable algorithm leads to an intuitive way of recognizing a stable transition:

Lemma 9. *A transition from one candidate k-center $\mathcal{B}_1(t)$ to another candidate k-center $\mathcal{B}_2(t)$ in the time interval $[0, T]$ is **topologically stable** if and only if the change of the disks' centers and radii is continuous over $[0, T]$ and at each time $t' \in [0, T]$, $\mathcal{B}(t')$ is **valid**.*

Proof. Note that by definition the disks must be transformed continuously and that all the points in $P(t)$ are covered in $[0, T]$ precisely when a valid candidate k-center exists. □

Now that we can recognize a topologically stable transition, we can reason about what such a transition looks like:

Lemma 10. *Any topologically stable transition from one k-center $\mathcal{B}_1(t)$ to another k-center $\mathcal{B}_2(t)$ in the timespan $[0, T]$ that minimizes $C(t)$ (the largest occurring* minmax *over $[0, T]$) can be obtained by a sequence of events where in each event, occurring at a time $t' \in [0, T]$, a disk $D_i \in \mathcal{B}(t')$ adds a point to P_i and a disk $D_j \in \mathcal{B}(t')$ removes a point from P_j. We call this **transferring**.*

Proof. The proof is by construction. Assume that we have a transition from $\mathcal{B}_1(t)$ to $\mathcal{B}_2(t)$ and the transition that minimizes the maximum of all radii contains simultaneous continuous movement. Let this transition take place in $[0, T]$.

To determine $C(t)$ we only need to look at times $t' \in [0,T]$ where a disk $D_i \in \mathcal{B}$ adds a new point p to its disk set P_i and another disk D_j removes it from P_j. Only at t' must both disks contain p; before t' disk D_j may be smaller and after t' disk D_i may be smaller.

We claim that for any optimal simultaneous continuous movement of cost $C(t)$, we can discretize the movement into a sequence of events with cost no larger than $C(t)$. We do so recursively: If the movement is continuous then there exists a $t_0 \in [0,T]$ as the first time a disk $D_i \in \mathcal{B}$ adds a point to P_i. At t_0, D_i has to contain both P_i and p and must have a certain size d. All the other disks $D_j \in \mathcal{B}$ with $j \neq i$ only have to contain the points in P_j so they have optimal size if they have not moved from time 0. In other words, it is optimal to first let D_i obtain p in an event and to then continue the transition from $[t_0, T]$. This allows us to discretize the simultaneous movement into sequential events. □

Corollary 1. *Any topologically stable transition from one k-center $\mathcal{B}_1(t)$ to another k-center $\mathcal{B}_2(t)$ in the timespan $[0,T]$ that minimizes $C(t)$ (the largest occurring* minmax *over $[0,T]$) can be obtained by a sequence of events where in each event the following happens:*

A disk $D_i \in \mathcal{B}_1(t)$ that was defined by one, two or three points in $P(t)$ is now defined by a new set of points in $P(t)$ where the two sets differ in only one element.

*With every event, P_i must be updated with a corresponding insert and/or delete. We call these events a **swap** because we intuitively swap out of the at most three defining elements.*

The cost of a single stable transition. Corollary 1 allows us to model a stable transition as a sequence of swaps but how do we find the optimal sequence of swaps? A single minimal covering disk at time t is defined by at most three unique elements from $P(t)$ so there are at most $O(n^3)$ subsets of $P(t)$ that could define one disk of a k-center. Let these $O(n^3)$ sets be the vertices in a graph G. We create an edge between two vertices v_i and v_j if we can transition from one disk to the other with a single swap and that transition is topologically stable. Each vertex is incident to only a constant number of edges (apart from degenerate cases) because during a swap the disk D_i corresponding to v_i can only add one element to P_i. Moreover, the radius of the disk is maximal on vertices in G and not on edges. The graph G has $O(n^3)$ complexity and takes $O(n^4)$ time to construct.

This graph provides a framework to trace the radius of the transition from a single disk to another disk. However, we want to transition from one k-center to another. We use the previous graph to construct a new graph G^k where each vertex w_i represents a set of k disks: a candidate k-center $\mathcal{B}_i$. We again create an edge between vertices w_i and w_j if we can go from the candidate k-center $\mathcal{B}_i$ to $\mathcal{B}_j$ in a single swap. With a similar argument as above, each vertex is only connected to $O(k)$ edges. The graph thus has $O(n^{3k})$ complexity and can be constructed in $O(n^{3k+1})$ time. Each vertex w_i gets assigned the cost of the k-center $\mathcal{B}_i$ where the cost is ∞ if $\mathcal{B}_i$ is invalid.

Any connected path in this graph from w_i to w_j without vertices with cost ∞ represents a stable transition from w_i to w_j by Corollary 1, where the cost of the path (transition) is the maximum value of the vertices on the path. We can now find the optimal sequence of swaps to transition from any vertex w_i to w_j by finding the cheapest path in this graph in $O(n^{3k} \log n)$ time, which is dominated by the $O(n^{3k+1})$ time it takes to construct the graph.

Maintaining the cost of a flip. For a single point in time we can now determine the cost of a topologically stable transition from a k-center $\mathcal{B}_i$ to $\mathcal{B}_j$ in $O(n^{3k+1})$ time. If we want to maintain the cost $C(t)$ for all times t, the costs of the vertices in the graph change over time. If we plot the changes of these costs over time, the graph consists of monotonously increasing or decreasing segments, separated by moments in time where two radii of disks are equal. These $O(n^{3k})$ events also contain all events where the structure of our graph G^k changes *and* all the moments where a vertex in our graph becomes *invalid* and thus gets cost ∞. The result of these observations is that we have a $O(n^{3k})$ size graph, with $O(n^{3k})$ relevant changes where with each change we need $O(n)$ time to restore the graph. This leads to an algorithm which can determine the cost of a topologically stable movement in $O(n^{6k+1})$ time.

Theorem 3. *Given a set of n points whose positions in the plane are determined by constant-degree algebraic functions, the stable minmax Euclidean k-center problem can be solved by an algorithm that runs in $O(n^{6k+1})$ time.*

If we run the unstable and stable algorithms on the moving points, we obtain two functions that map time to a cost. The maximum over time of the ratio of the cost is the stability ratio of the instance, obtained in the same running time.

4 Conclusion

We considered the topological stability of the kinetic k-center problem, in which solutions must change continuously but may do so arbitrarily fast. We proved nontrivial upper bounds for small values of k and presented a general lower bound tending towards 2 for large values of k. We also presented algorithms to compute topologically stable solutions together with the cost of stability for a set of moving points, that is, the growth factor that we need for that particular set of moving points at any point in time. A practical application of these algorithms would be to identify points in time where we could slow down the solution to explicitly show stable transitions between optimal solutions.

Future work. It remains open whether a general upper bound strictly below 2 is achievable for k-EC-minmax. We conjecture that this bound is indeed smaller than 2 for any constant k. For this, we need more insight in how to resolve an intersection graph with more general structures. Our algorithms to compute the cost of stability for an instance have high (albeit polynomial) run-time complexity. Can the results for KDS (e.g. [2]) help us speed up these algorithms? Alternatively, can we approximate the cost of stability more efficiently?

Lipschitz stability requires a bound on the speed at which a solution may change [16]. This stability for $k > 2$ is unbounded, if centers have to move continuously [7]; A potentially interesting variant of the topology is one where a disk may shrink to radius 0, at which point it disappears and may reappear at another location. This alleviates the problem in the example; would it allow us to bound the Lipschitz stability?

References

1. Basch, J., Guibas, L.J., Hershberger, J.: Data structures for mobile data. Journal of Algorithms **31**(1), 1–28 (1999)
2. de Berg, M., Roeloffzen, M., Speckmann, B.: Kinetic 2-centers in the black-box model. In: Proc. 29th Symposium on Computational Geometry. pp. 145–154 (2013)
3. Chan, T.: More planar two-center algorithms. Comp. Geom. **13**(3), 189–198 (1999)
4. Degener, B., Gehweiler, J., Lammersen, C.: Kinetic facility location. Algorithmica **57**(3), 562–584 (2010)
5. Drezner, Z.: On a modified 1-center problem. Mgmt Science **27**, 838–851 (1981)
6. Drezner, Z.: On the rectangular p-center problem. Naval Research Logistics **34**(2), 229–234 (1987)
7. Durocher, S.: Geometric Facility Location under Continuous Motion. Ph.D. thesis, University of British Columbia (2006)
8. Durocher, S., Kirkpatrick, D.: Bounded-velocity approximation of mobile Euclidean 2-centres. International Journal of Computational Geometry & Applications **18**(03), 161–183 (2008)
9. Friedler, S., Mount, D.: Approximation algorithm for the kinetic robust k-center problem. Computational Geometry **43**(6-7), 572–586 (2010)
10. Gao, J., Guibas, L., Hershberger, J., Zhang, L., Zhu, A.: Discrete mobile centers. Discrete & Computational Geometry **30**(1), 45–63 (2003)
11. Gao, J., Guibas, L., Nguyen, A.: Deformable spanners and applications. Computational Geometry: Theory and Applications **35**(1-2), 2–19 (2006)
12. Hershberger, J.: Finding the upper envelope of n line segments in $o(n \log n)$ time. Information Processing Letters **33**(4), 169–174 (1989)
13. Hoffmann, M.: A simple linear algorithm for computing rectangular 3-centers. In: Proc. 11th Canadian Conference on Computational Geometry. pp. 72–75 (1999)
14. Hwang, R., Lee, R., Chang, R.: The slab dividing approach to solve the Euclidean p-center problem. Algorithmica **9**, 122 (1993)
15. Megiddo, N., Supowit, K.: On the complexity of some common geometric location problems. SIAM Journal on Computing **13**(1), 182–196 (1984)
16. Meulemans, W., Speckmann, B., Verbeek, K., Wulms, J.: A framework for algorithm stability. In: Proc. 13th LATIN. pp. 805–819. LNCS 10807 (2018)
17. Nussbaum, D.: Rectilinear p-piercing problems. In: Proc. 1997 Symposium on Symbolic and Algebraic Computation. pp. 316–323 (1997)
18. Segal, M.: On piercing sets of axis-parallel rectangles and rings. In: Proc. 5th European Symposium on Algorithms. pp. 430–442 (1997)
19. Sharir, M., Welzl, E.: Rectilinear and polygonal p-piercing and p-center problems. In: Proc. 12th Symposium on Computational Geometry. pp. 122–132 (1996)
20. Sylvester, J.: A question in the geometry of situation. Quarterly Journal of Mathematics **1**, 79 (1857)