Separated Red Blue Center Clustering

Marzieh Eskandari
Department of Computer Science, Alzahra University, Tehran, Iran

Bhavika Khare
Department of Computer Science, University of Memphis, TN, USA

Nirman Kumar
Department of Computer Science, University of Memphis, TN, USA

Abstract

We study a generalization of k-center clustering, first introduced by Kavand et. al., where instead of one set of centers, we have two types of centers, p red and q blue, and where each red center is at least α distant from each blue center. The goal is to minimize the covering radius. We provide an approximation algorithm for this problem, and a polynomial-time algorithm for the constrained problem, where all the centers must lie on a line ℓ.

2012 ACM Subject Classification Mathematics of computing → Approximation algorithms; Theory of computation → Facility location and clustering; Theory of computation → Computational geometry

Keywords and phrases Algorithms, Facility Location, Clustering, Approximation Algorithms, Computational Geometry

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2021.41

Related Version *Full Version*: https://arxiv.org/abs/2107.07914

1 Introduction

The k-center problem is a well-known geometric location problem, where we are given a set P of n points in a metric space and a positive integer k, and the task is to find k balls of minimum radius whose union covers P. This problem can be used to model the following facility location scenario. Suppose we want to open k facilities (such as supermarkets) to serve the customers in a city. It is common to assume that a customer shops at the facility closest to their residence. Thus, we want to locate k locations to open the facilities, so that the maximum distance between a customer and their nearest facility is minimized. The problem was first shown to be NP-hard by Megiddo and Supowit [12] for Euclidean spaces. We consider a variation of this classic problem where instead of just one set of centers, we consider two sets of centers, one of size p, and the other of size q, but with the constraint that each center of the first set is separated by a distance of at least some given α from each center of the second set. This follows from a more practical facility location scenario, where we want to open two types of facilities (say "Costco's" and "Sam's club"). Each facility type wants to cover all the customers within the minimum possible distance (similar to the k-center clustering objective), but the facilities want to be separated from each other to avoid crowding or getting unfavorably affected by competition from the other.

The k-center problem has a long history. In 1857 Sylvester [15] presented the 1-center problem for the first time, and Megiddo [11] gave a linear time algorithm for solving this problem, also known as the minimum enclosing ball problem, in 1983, using linear programming. Hwang et al. [9] showed that the Euclidean k-center problem in the plane can be solved in $n^{O(\sqrt{k})}$. Agarwal and Procopiuc [1] presented an $n^{O(k^{1-1/d})}$-time algorithm for solving the k-center problem in $\mathbb{R}^d$ and a $(1+\epsilon)$-approximation algorithm with running time $O(n \log k) + (k/\epsilon)^{O(k^{1-1/d})}$.

Due to the importance of this problem, many researchers have considered variations of the basic problem to model different situations. Brass et al. studied the constrained version of the k-center problem in which the centers are constrained to be co-linear [4], also considered previously for $k = 1$ by Megiddo [11]. They gave an $O(n \log^2 n)$-time algorithm when the line is fixed in advance. Also, they solved the general case where the line has an arbitrary orientation in $O(n^4 \log^2 n)$ expected time. They presented an application of the constrained k-center in wireless network design: For a given set of n sensors (which are modeled as points), we want to locate k base servers (centers of balls) for receiving the signal from the sensors. The servers should be connected to a power line, so they have to lie on a straight line which models the power line. Other variations have also been considered [8, 2, 3] for $k = 1$. For $k \geq 2$, variants have been studied as this has applications to the placement of base stations in wireless sensor networks [5, 13, 14].

Hwang et al. [9] studied a variant somewhat opposite to our variant. In their variant, for a given constant $0 \leq \alpha \leq 1$, the α-connected two-center problem is to find two balls of minimum radius r whose union covers the points, and the distance of the two centers is at most $2(1 - \alpha)r$, i.e., any two of those balls intersect such that each ball penetrates the other to a distance of at least $2\alpha r$. They presented an $O(n^2 \log^2 n)$ expected-time algorithm.

The variant we consider was first considered by Kavand et. al. [10]. They termed it as the $(n, 1, 1, \alpha)$-center problem. They aimed to find two balls each of which covers the entire point set, the radius of the bigger one is minimized, and the distance of the two centers is at least α. They presented an $O(n \log n)$-time algorithm for this problem and a linear time algorithm for its constrained version using the furthest point Voronoi diagram.

This paper considers the generalization of the problem defined by [10], and we denote it by $(n, p \wedge q, \alpha)$ problem. We explain our choice of notation, particularly the $\wedge$ sign, in Section 2. For a given set P of n points in a metric space and integers $p, q \geq 1$, we want to find $p + q$ balls of two different types, called **red** and **blue** with the minimum radius such that P is covered by the p red balls and also covered by the second type of q blue balls, and the distance of the centers of each red ball from the centers of the blue balls is at least α. In addition to one example mentioned before, another motivating application of the $(n, p \wedge q, \alpha)$ problem would be to locate p police stations and q hospitals in an area such that the distance between each police station and a hospital is not smaller than a predefined distance α for the convenience of patients. By locating hospitals and police stations at an admissible distance from each other, patients stay away from crowd and noise while the clients have access to hospitals and police stations that are close enough to them. Moreover, it is obviously desirable that the maximum distance between a client and its nearest police station as well as its nearest hospital is minimized. In addition to this general problem we also consider the constrained version due to its applicability in many situations, where the centers are constrained to lie on a given line.

Paper organization. In Section 2, we present the formal problem statement and the definitions required in the sequel. In Section 3, we present an $O(1)$ factor approximation algorithm for the problem in Euclidean spaces. Then, in Section 4, we present a polynomial-time algorithm for the constrained problem. We conclude in Section 5.

2 Problem and Definitions

Let $\mathcal{M}$ denote a metric space. Let $\mathsf{dist}(p, \mathsf{q})$ denote the distance between points p, q in $\mathcal{M}$. For a point $x \in \mathcal{M}$ and a number $r \geq 0$ the ball $\mathsf{B}(x, r)$ is the set of points with distance at most r from x, i.e., $\mathsf{B}(x, r) = \{p \in \mathcal{M} | \mathsf{dist}(x, p) \leq r\}$ is the *closed* ball of radius r with center x.

In the α-separated red-blue clustering problem we are given a set P with n points in some metric space $\mathcal{M}$, integers $p > 0, q > 0$, and a real number $\alpha > 0$. For a given number $r \geq 0$, p points $c_1, \ldots, c_p$ in $\mathcal{M}$ (with possibly repeating points) called the red centers and q points $d_1, \ldots, d_q$ in $\mathcal{M}$ (with possibly repeating points) called the blue centers are said to be a **feasible solution** for the problem, with **radius of covering** r if they satisfy,

- **Covering constraints:** The union of the balls $\bigcup_{i=1}^{p} \mathsf{B}(c_i, r)$ (called the red balls) covers P, and the union of the balls $\bigcup_{j=1}^{q} \mathsf{B}(d_j, r)$ (called the blue balls) covers P.
- **Separation constraint:** For each $1 \leq i \leq p, 1 \leq j \leq q$ we have $\mathsf{dist}(c_i, d_j) \geq \alpha$, i.e., the red and blue centers are separated by at least a distance of α.

If there exists a feasible solution for a certain value of r, such an r is said to be feasible for the problem. The goal of the problem is to find the minimum possible value of r that is feasible.

We denote this problem as the $(n, p \wedge q, \alpha)$-problem. The $\wedge$ in the notation stresses the fact that *both* the red balls *and* the blue balls cover P. Let $r_{p\wedge q,\alpha}(P)$ denote the optimal radius for this problem. When P, p, q, α are clear from context sometimes we will also denote this by r^*. Also, let $r_k(P)$ denote the optimal k-center clustering radius, for all $k \geq 1$. To be clear, the centers in the k-center clustering problem can be any points in $\mathcal{M}$, not necessarily belonging to P. If that is the requirement, the problem is the *discrete* k-center clustering problem.

For this paper, we will always be concerned with $\mathcal{M} = \mathbb{R}^d$, but we will use the notations as defined above without qualifying the metric space. We let $P = \{p_1, \ldots, p_n\}$ where $p_i = (p_{i1}, p_{i2}, \ldots, p_{id})$. We also consider the *constrained* α-separated red-blue clustering problem (when $\mathcal{M} = \mathbb{R}^d$) we are given a line ℓ and all the red and blue centers are constrained to lie on the line ℓ. Without loss of generality, we will assume that ℓ is the x-axis since this can be achieved by an appropriate affine transformation of space. Moreover, we will use the same notation for the optimal radii and centers. For the constrained problem we need some additional definitions and notations. We assume that no two points in P have the same distance from ℓ. (This general position assumption can however be removed.) For each point p_i, we consider the set of points on the line (x-axis) such that the ball of radius r centered at one of those points can cover p_i. This is the intersection of $\mathsf{B}(p_i, r)$ with the x-axis, see Figure 1. Assuming this intersection is not empty, let the interval be $I_i(r) = [a_i(r), b_i(r)]$. Denote the set of all intervals as $\mathcal{I}(r) = \{I_1(r), \ldots, I_n(r)\}$ where we assume that the numbering is in the sorted order of intervals: those with earlier left endpoints are before, and for the same left endpoints the ones with earlier right endpoint occurs earlier in the order. Notice that feasibility of radius r means that there exist two hitting sets for the set of intervals $\mathcal{I}(r)$, the red centers and the blue centers such that they satisfy the separation constraint.

The interval endpoints $a_i(r), b_i(r)$ can be computed by solving the equation,

$$(x - p_{i1})^2 + p_{i2}^2 + \ldots + p_{id}^2 = r^2,$$

for x. Thus, they are given by $a_i(r) = p_{i1} - \sqrt{r^2 - \sum_{j=2}^{d} p_{ij}^2}$, and $b_i(r) = p_{i1} + \sqrt{r^2 - \sum_{j=2}^{d} p_{ij}^2}$. It is easy to see that for the range of r where the intersection is non-empty, $a_i(r)$ is a strictly decreasing function of r and $b_i(r)$ is a strictly increasing function of r.

Model of computation. We remark that our model of computation is the Real RAM model, where the usual arithmetic operations are assumed to take $O(1)$ time.

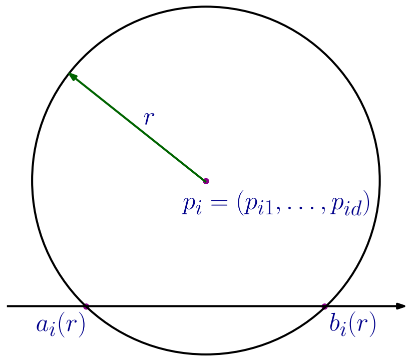

Figure 1 The functions $a_i(r), b_i(r)$.

3 Approximation algorithms for the $(n, p \wedge q, \alpha)$ problem in $\mathbb{R}^d$

The $(n, p \wedge q, \alpha)$ problem is NP-hard when p, q are part of the input, since the k center problem clearly reduces to the $(n, p \wedge q, \alpha)$ problem when $\alpha = 0$ and $p = q = k$. Here we show an approximation algorithm for the problem as well as one with a better approximation factor for the constrained problem. We assume that $p \leq q$, wlog.

Here we show that there is a constant factor algorithm for the $(n, p \wedge q, \alpha)$ problem in $\mathbb{R}^d$. We need a few preliminary results.

▶ **Lemma 1.** *Suppose that $r \geq \alpha/2$ is a number such that there are points $x_1, \ldots, x_p$ satisfying $P \subseteq \bigcup_{i=1}^{p} \mathsf{B}(x_i, r)$. Then, there are $(p+q)$ points $c_1, \ldots, c_p, d_1, \ldots, d_q$ all such that the following are met,*

(I) Separation constraint: $\mathsf{dist}(c_i, d_j) \geq \alpha$ *for all i, j, and,*

(II) Covering constraints: $P \subseteq \bigcup_{i=1}^{p} \mathsf{B}(c_i, 7r)$*, and,* $P \subseteq \bigcup_{j=1}^{q} \mathsf{B}(d_j, 7r)$*.*

Proof. First, from the points $x_1, \ldots, x_p$ we choose a maximal subset of them such that the distance between each pair of them is at least $4r$. This can be done by a simple scooping algorithm that starts with x_1 as first point, then throws away all points x_i (for $i > 1$) with $\mathsf{dist}(x_1, x_i) < 4r$, then choose any one of the remaining points and proceed analogously.

Suppose after this (with some renaming) the points that remain are, $x_1, \ldots, x_t$, where $1 \leq t \leq p$. Then, one can easily show that, $P \subseteq \bigcup_{i=1}^{t} \mathsf{B}(x_i, 5r)$.

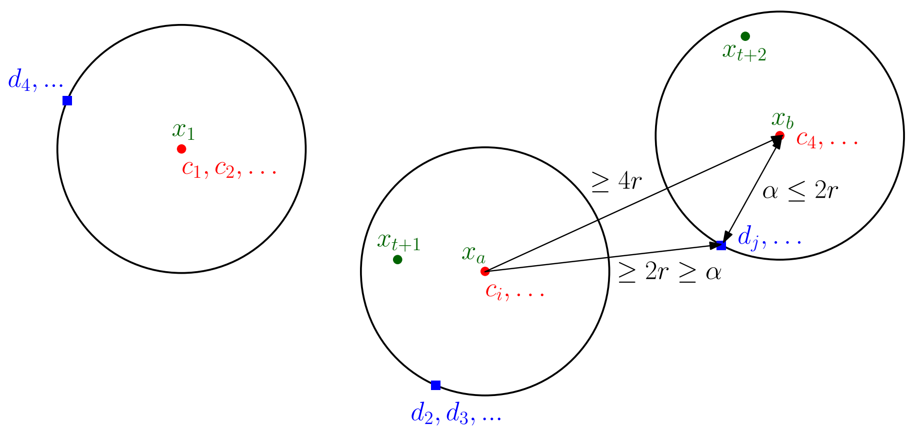

Figure 2 Illustration for proof of Lemma 1.

Now, we choose the p red centers $c_1, \ldots, c_p$ at the points $x_1, \ldots, x_t$ such that each of them is chosen. Notice that this is possible since $t \leq p$. If $t < p$, some may be co-located at one x_i, though. Then, let $d_1, \ldots, d_q$ be any points on the surface of those balls (i.e., on the spheres). Since $t \leq q$ we have enough points to hit all the balls. If necessary we can co-locate some points d_j to hit the target number q. See Figure 2.

The covering constraints are met since as remarked above, the balls of radius $5r$ around all c_i (i.e., around all x_i) covers P. Similarly, by the triangle inequality and because $\alpha \leq 2r$, the balls of radius $7r$ around the d_j cover P.

To see the separation constraint, let c_i, d_j be any red and blue centers as defined above. Suppose c_i is located at the center x_a where $1 \leq a \leq t$, and d_j is located on the surface of the ball $\mathsf{B}(x_b, \alpha)$ where $1 \leq b \leq t$. If $a = b$, then since c_i is at the center and d_j on the surface of the ball $\mathsf{B}(x_a, \alpha)$ their distance is exactly α. If, on the other hand, $a \neq b$, then by the triangle inequality we have that, $\mathsf{dist}(x_a, d_j) + \mathsf{dist}(d_j, x_b) \geq \mathsf{dist}(x_a, x_b) \geq 4r$. Now, $c_i = x_a$, and $\mathsf{dist}(d_j, x_b) = \alpha \leq 2r$, and so, $\mathsf{dist}(c_i, d_j) \geq 4r - 2r = 2r \geq \alpha$, as desired. ◀

▶ **Lemma 2.** *We have that, $r_{p \wedge q, \alpha}(P) \geq \alpha/2$.*

Proof. Consider any point p_l. This point is in some red ball $\mathsf{B}(c_i, r)$ and in some blue ball $\mathsf{B}(d_j, r)$. Thus, by the triangle inequality, $\mathsf{dist}(c_i, d_j) \leq \mathsf{dist}(c_i, p_l) + \mathsf{dist}(p_l, d_j) \leq 2r$. On the other hand, $\alpha \leq \mathsf{dist}(c_i, d_j)$. Thus, $\alpha \leq 2r$ and the claim follows. ◀

Observe that since $p \leq q$, $r_p(P) \geq r_q(P)$.

▶ **Lemma 3.** *We have that, $r_{p \wedge q, \alpha}(P) \geq r_p(P)$.*

Proof. Consider a feasible solution with the radius $r^* = r_{p \wedge q, \alpha}(P)$. The p red balls cover P with radius r^*. Thus, $r^* \geq r_p(P)$, since by definition, $r_p(P)$ is the minimum p-center clustering radius. ◀

Now, for the $O(1)$ approximation to $r_{p \wedge q, \alpha}(P)$ notice that we can easily compute by adapting Gonzalez's algorithm [7] a 2-approximation to the p-center clustering problem, i.e., to $r_p(P)$. (This is standard and well-known so we omit the details.) In other words we have now computed, p centers $x_1, \ldots, x_p$ all in P, and a radius $r \leq 2r_p(P)$ such that balls $\mathsf{B}(x_i, r)$ cover P. We now consider the radius $r' = max(r, \alpha/2)$, and clearly the balls $\mathsf{B}(x_i, r')$ also define such a covering. Now, using Lemma 1 we can find a feasible solution with radius at most $7r' \leq 14r$. Thus, we have shown the following theorem.

▶ **Theorem 4.** *Let r^* be the optimal radius for the α-separated red blue clustering problem on an n point set P with parameters $p, q \geq 1$. Then, we can compute in polynomial-time, a feasible solution where the covering radius is at most $14r^*$.*

This approximation factor can be improved for the constrained problem, i.e., where all the centers are constrained to be on a fixed line ℓ. The details can be found in the full version of this paper [6].

4 Polynomial-time algorithm for the constrained problem

Our basic approach will be to do a binary search for the optimal radius. We first present an algorithm to decide if a given value of the radius r is feasible. Then, we present an algorithm to determine a finite set of values such that the optimal radius must be within that set. Then, a binary search using the feasibility testing algorithm gives us the optimal radius.

4.1 Deciding feasibility for given radius r

Given a radius $r > 0$ we give a polynomial-time method to decide if there is a solution to the constrained $(n, p \wedge q, \alpha)$ problem with covering radius r. The algorithm is a dynamic programming algorithm. One of the challenges encountered is that the centers can be anywhere on the line, and thus a naive implementation of dynamic programming does not work since there are not finitely many sub-problems. As such, we first show how we can compute a finite set $\mathcal{C}(r)$ of $O(n^2)$ points such that if r is feasible, one can find a feasible solution with covering radius r with centers belonging to $\mathcal{C}(r)$.

4.1.1 Candidate points for the centers

Consider all the end-points of the intervals in $\mathcal{I}(r)$, i.e., $a_i(r), b_i(r)$, and consider the sorted order of them. Assume that in sorted order the points are renamed to $x_1, x_2, \ldots, x_{2n}$ (possibly some of them are co-located). As remarked before, the feasibility problem is equivalent to finding two hitting sets for the intervals in $\mathcal{I}(r)$ that satisfy the separation constraints. As in standard in such hitting set problems, we look at the *faces* of the arrangement of the intervals. Some of these faces might be open intervals, or half-open intervals, or even singleton points. Notice that all faces are disjoint by definition. It is easy to see that if F is such a face, then a point in the closure $\bar{F}$ of F will hit at least the same intervals as points in F hit. To avoid confusion when we refer to faces vs. their closures, in the remaining discussion we will always say face F for the original face and closure face $\bar{F}$ when referring to one of the closures (even though they may be the same set of points).

We explain now, why considering face closures is valid. Suppose a certain center belongs to a face F and suppose it is allowable to choose **any** point close enough to one of its boundary points, such that the separation constraints are met. Then, it is also valid to choose it at the boundary point and respecting the separation constraint since the separation constraint is that distance between the red and blue centers is $\geq \alpha$ as opposed to a strict inequality $> \alpha$. Therefore, it is valid to replace faces with their closures.

To compute all the closures of the faces in the arrangement of the $\mathcal{I}(r)$, we sort the x_i and we retain all the consecutive intervals $[x_i, x_{i+1}]$ that do not lie outside any of the intervals $I_i(r)$. This can be done by a simple line sweep algorithm. Notice that there are only $O(n)$ such closure faces.

Next, we define a sequence for each such closure face. Consider such a closure face, $[x_i, x_{i+1}]$ that is the **starting closure face** for this sequence. Consider the sequence, $x_i, x_i + \alpha, x_i + 2\alpha, \ldots$. We only want to retain, for each closure face, (at most) the first three points that hit the closure face. So, given any starting closure face $[x_i, x_{i+1}]$ there are only $O(n)$ points in this sequence since it is bounded by the number of closure faces (in fact beyond the starting closure face $[x_i, x_{i+1}]$) times three. Let the sequence of points that result due to starting closure face $[x_j, x_{j+1}]$ be denoted by SEQ_j. We have the following lemma,

▶ **Lemma 5.** *For each starting closure face, $[x_i, x_{i+1}]$ the associated sequence SEQ_i can be computed in $O(n)$ time.*

Proof. We consider each closure face and compute the points of this sequence that possibly lie in this closure face. Consider such a closure face $[x_j, x_{j+1}]$. For a member of SEQ_i to lie in this closure face there is an integer k such that $x_j \leq x_i + k\alpha \leq x_{j+1}$. This is equivalent to, $\frac{x_j - x_i}{\alpha} \leq k \leq \frac{x_{j+1} - x_i}{\alpha}$. Thus, to find the first three points of SEQ_i hitting the closure face, we only need to find the three smallest integers in the interval, $[\frac{x_j - x_i}{\alpha}, \frac{x_{j+1} - x_i}{\alpha}]$, if there are such. This can be done in $O(1)$ time per closure face. Since there are $O(n)$ closure faces, the entire sequence can be constructed in $O(n)$ time. ◀

Computing such sequence for each closure face as starting closure face leads to a total of $O(n^2)$ points, and can be computed in $O(n^2)$ time overall by the previous lemma. The set $\mathcal{C}(r)$ is the set of the points in all these sequences. Let the sorted order of points in this set be denoted by $\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathsf{c}_m$ where $m = O(n^2)$. Thus $\mathcal{C}(r) = \{\mathsf{c}_1, \ldots, \mathsf{c}_m\}$. For any such point c_k denote by $\mathsf{s}(k)$ the first index j such that $\mathsf{c}_j - \mathsf{c}_k \geq \alpha$. If there is no such point, let $\mathsf{s}(k) = m + 1$. Notice that we can compute $\mathsf{s}(k)$ by a successor query in $O(\log n)$ time if the set $\mathcal{C}(r)$ is sorted. The following lemma says that we can assume that the centers (of both colors) are in $\mathcal{C}(r)$.

▶ **Lemma 6.** *Suppose that the constrained $(n, p \wedge q, \alpha)$ problem has a feasible solution with covering radius r. Then there is a solution with all centers belonging to $\mathcal{C}(r)$.*

Proof. Consider a feasible solution with p red centers and q blue centers. First, we remark that we can assume, wlog, that in any face F there are at most two points, one red and one blue. This is true because having more than one red or more than one blue point in a face does not affect the covering constraints, as each point in a face hits (i.e., belongs to) the exact same set of intervals by definition. Thus, we may assume that there are at most $T \leq (p + q)$ such centers, since we might need to throw away some of them when two centers of the same color belong to one face. Let the T centers be $u_1, \ldots, u_T$ where any of them can be red or blue. We show how to construct iteratively another feasible solution where all the centers are in $\mathcal{C}(r)$.

We will proceed face by face, and consider all the centers within the face. We know that there are at most two centers within a face. Moreover, if there are two they are of different colors. Our basic strategy is to move the first center left till we can, while remaining within the closure of the face, without violating any separation constraint. If there is only one point in a face, we are done. Otherwise, once the position of the first point is fixed, the second point can be similarly moved left until its position is determined within $\mathcal{C}(r)$. Let the sorted order of faces be $F_1, F_2, \ldots, F_N$ where $N = O(n)$.

We construct a new sequence $v_1, \ldots, v_T$ where v_i is assigned color of u_i and is obtained by shifting u_i to the left (so that it will lie in $\mathcal{C}(r)$, but never leaving closure of the face it belongs to). We prove the following claim by induction, which implies the claim that all the v_i belong to $\mathcal{C}(r)$.

▷ Claim 7. For each $k \geq 1$, all the points u_i belonging to face F_k are mapped to points v_i (belonging to $\bar{F}_k$) such that, the at most two such v_i, lie on consecutive points of the same sequence SEQ_j for some j.

Consider the base case $k = 1$. If there are no points in F_1, the claim holds vacuously. If there is only one point in F_1, slide it left until it hits the boundary of F_1. This does not violate any constraints. The claim holds true trivially. Suppose there are two points in F_1. Now, $\bar{F}_1 = [x_1, x_2]$ and after sliding the first point left till it coincides with x_1, and thus in SEQ_1, the second point clearly satisfies $u_2 - x_1 \geq \alpha$ since even before the sliding the inequality was satisfied. Notice that the points are of different color. Thus we can place the second point v_2 at $x_1 + \alpha \in \mathsf{SEQ}_1$ and they are consecutive points of SEQ_1.

Suppose that the claim is true up-to k. We now consider the case $k + 1$. Again, as before if there are no points in F_{k+1} the claim holds vacuously. If there is only one point, we slide it left to the first point in $\mathcal{C}(r)$ which is allowable for it. The meaning of allowable is the following. Suppose this point is v_s. Then, if v_{s-1}, which is in a previous face, is of the same color as v_s, then v_s can be anywhere within its face. If it is of different color, then v_s has to be at least at $v_{s-1} + \alpha$. Since the starting point of the closure $\bar{F}_{k+1}$ is in $\mathcal{C}(r)$, as is $v_{s-1} + \alpha$

if it lies in $\bar{F}_{k+1}$, while sliding left we will hit a point in $\mathcal{C}(r)$ eventually and we stop there. Now consider, the case where there are two points u_i, u_{i+1} in F_{k+1}. After v_i has been placed at its position in $\mathcal{C}(r)$ as outlined for the case of single point in F_{k+1}, suppose it belongs to SEQ_j for some j. Clearly it is at most the second point, of SEQ_j in $\bar{F}_{k+1}$ as the second point is already α ahead of the first point of SEQ_j in $\bar{F}_{k+1}$. Now, $u_{i+1} - v_i \geq \alpha$. Thus, we can place the second point of F_{k+1} at the next point of SEQ_j in $\bar{F}_{k+1}$, which exists in $\bar{F}_{k+1}$ since the next point is $v_i + \alpha$ which is in $\bar{F}_{k+1}$ by assumption. We observe that all claims hold true. ◀

4.1.2 The dynamic programming algorithm

The dynamic programming algorithm computes two tables, $\text{CanCover}^{\text{R}}[\mathcal{I}_1, \mathcal{I}_2, a, b, k]$, and $\text{CanCover}^{\text{B}}[\mathcal{I}_1, \mathcal{I}_2, a, b, k]$ of Boolean $\textbf{True}, \textbf{False}$. Here, $\mathcal{I}_1, \mathcal{I}_2$ are prefixes of intervals of $\mathcal{C}(r)$ (when they are ordered by their left and right end-points), and $0 \leq a \leq p, 0 \leq b \leq q$ are integers, and $1 \leq k \leq (m+1)$ is also an integer. The table entry $\text{CanCover}^{\text{R}}[\mathcal{I}_1, \mathcal{I}_2, a, b, k]$ is **True**, if there is a hitting set consisting of (at most) a red points that hit the intervals in $\mathcal{I}_1$, (at most) b blue points that hit the intervals in $\mathcal{I}_2$ and with the constraint that the first point to be possibly put is red and at c_k if $k < m$. Here $k > m$ represents that there is no where to really put the first red point. Notice that the separation constraint between red/blue points must be met. Similarly the table entry $\text{CanCover}^{\text{B}}[\mathcal{I}_1, \mathcal{I}_2, a, b, k]$ is true if the first point is blue and at c_k (for $k \leq m$). Assuming that the above tables have been computed, we can answer whether the radius r is feasible by computing the following expression, where we denote $\mathcal{I}(r)$ by $\mathcal{I}$ for brevity,

$$\begin{gathered}\text{CanCover}^{\text{R}}[\mathcal{I}, \mathcal{I}, p, q, 1] \vee \ldots \vee \text{CanCover}^{\text{R}}[\mathcal{I}, \mathcal{I}, p, q, m] \\ \bigvee \\ \text{CanCover}^{\text{B}}[\mathcal{I}, \mathcal{I}, p, q, 1] \vee \ldots \vee \text{CanCover}^{\text{B}}[\mathcal{I}, \mathcal{I}, p, q, m].\end{gathered}$$

In the above expression, we try to hit all the intervals in $\mathcal{I}$ by both red and blue points and we try all possible starting locations and color for the first point. We know that the centers can be assumed to belong to $\mathcal{C}(r) = \{\mathsf{c}_1, \ldots, \mathsf{c}_m\}$.

Now we present the recursive definition of the algorithm to fill the tables. We only present the definitions for $\text{CanCover}^{\text{R}}[\cdot]$ but there is an entirely similar definition for $\text{CanCover}^{\text{B}}[\cdot]$ with the roles of red and blue interchanged.

$$\text{CanCover}^{\text{R}}[\mathcal{I}_1, \mathcal{I}_2, a, b, k] =$$
$$\begin{cases} \textbf{False} & \text{if } (\mathcal{I}_1 \neq \emptyset \wedge a = 0) \vee (\mathcal{I}_2 \neq \emptyset \wedge b = 0) \vee (\mathcal{I}_1 \cup \mathcal{I}_2 \neq \emptyset \wedge k > m), \\ \textbf{True} & \text{if } (\mathcal{I}_1 = \emptyset \wedge \mathcal{I}_2 = \emptyset), \\ \textbf{False} & \text{if there exists an interval in } \mathcal{I}_1 \cup \mathcal{I}_2 \text{ ending before } \mathsf{c}_k, \\ B_a \vee B_b & \text{otherwise.} \end{cases}$$

The first case means that if there are not any red centers to put while some unhit intervals remain in $\mathcal{I}_1$, or not any blue centers to put but unhit intervals in $\mathcal{I}_2$, or if we have already passed over all centers ($k > m$) but any unhit red or blue intervals remain, we return false. The next case means that if that all intervals have been hit already we should return true. The penultimate case means that if the first point c_k is so far ahead that at least one interval in $\mathcal{I}_1 \cup \mathcal{I}_2$ ends before it, then there can be no solution. This is true because any later points, red or blue, will only be ahead of c_k and thus the ended interval cannot be hit. The last case uses Boolean variables B_a, B_b that are defined as follows, and they also capture the main recursive cases. As required by the definition of the function, we must put the next center as red and at c_k. This would cause some intervals in $\mathcal{I}_1$ to be hit by c_k. We remove those intervals from $\mathcal{I}_1$. Let $\mathcal{I}_1'$ be the intervals in $\mathcal{I}_1$ not hit by c_k. It is easy to see that if $\mathcal{I}_1$ is a prefix of $\mathcal{I}$, then so is $\mathcal{I}_1'$. The definitions of B_a, B_b are as follows.

$$B_a \leftarrow (\mathcal{I}_1 = \emptyset) \vee \bigvee_{j=k+1}^{m} \text{CanCover}^{\text{R}}[\mathcal{I}_1', \mathcal{I}_2, a-1, b, j]$$

This assignment ensures that if $\mathcal{I}_1 = \emptyset$, then we never really try to put any red point. If not, then we try all possibilities for the next position of the red point. Notice that putting another red point at c_k is not necessary so we start with the remaining positions and go up to m. The coverage requirements for blue points and their numbers remain unchanged. The red number decreases by 1. The Boolean B_b has the following definition,

$$B_b \leftarrow (\mathcal{I}_2 = \emptyset) \vee \bigvee_{j=\mathsf{s}(k)}^{m} \text{CanCover}^{\text{B}}[\mathcal{I}_1', \mathcal{I}_2, a-1, b, j].$$

This is because, if the next point (after the current red one) is to be blue, it can only be at index $\mathsf{s}(k)$ or later. Thus we look-up $\text{CanCover}^{\text{B}}[\mathcal{I}_1', \mathcal{I}_2, a-1, b, j]$ for all such possible j. The first check $\mathcal{I}_2 = \emptyset$ means that if the blue intervals have already been hit, we do not need to put any blue point later. Both the tables $\text{CanCover}^{\text{R}}[\cdot], \text{CanCover}^{\text{B}}[\cdot]$ are filled simultaneously by first filling in the entries fitting the base cases, and then traversing them in order of increasing a, increasing b, decreasing k, and increasing $\mathcal{I}_1, \mathcal{I}_2$ (i.e., the smaller prefixes come earlier). It is easy to see that the traversal order meets the dependencies as written in the recursive definitions.

Analysis. First, observe that computing the candidate centers can be done in $O(n^2)$ time as implied by Lemma 5 and the following discussion. Moreover, the successor points $\mathsf{s}(k)$ can all be computed in total $O(n^2 \log n)$ time by first sorting $\mathcal{C}(r)$ and then followed by successor queries. The time however is dominated by the main dynamic programming algorithm. Observe that there are $O(n)$ prefixes, and $m = O(n^2)$ possible center locations. Thus there are in total $O(n^4 pq)$ entries to be filled. Except for the base cases, filling in an entry requires looking up $O(n^2)$ previous entries, as well as some computation such as finding which intervals are not hit by the current point. Such queries can be handled easily for all the intervals say in $\mathcal{I}_1$ wrt the point c_k in $O(n)$ time. Thus for a particular table entry, we require $O(n^2)$ time. Overall we will take $O(n^6 pq)$ time. We get the following theorem,

▶ **Theorem 8.** *For the constrained $(n, p \wedge q, \alpha)$ problem where the centers are constrained to lie on the x-axis, given a radius r, it can be decided if r is feasible in time $T_{DP}(n, p, q) = O(n^6 pq)$. Moreover, if r is feasible, a feasible solution with covering radius r can also be computed in the same time.*

To justify the comment about the feasible solution, note that by standard dynamic programming techniques, we can also remember while computing the table entries the solution, and it can be output at the end.

4.2 Candidate values for r

In this section, we will find a discrete candidate set for the optimal radii that facilitates a polynomial-time algorithm for solving the constrained $(n, p \wedge q, \alpha)$ problem as presented in Section 4.3. For this purpose, we need to determine some properties of an optimal solution.

First, we define a standard form solution and describe an easy approach to convert a feasible solution to the standard form. Then we present a lemma for proving a property of an optimal solution. Finally, we compute a finite candidate set for optimal radii.

Let $U = \{u_1, u_2, \ldots, u_{p+q}\}$ be a feasible solution with covering radius r. The closure face that contains u_i is denoted by $[x_{i,1}(r), x_{i,2}(r)]$, where $x_{i,1}(r)$ and $x_{i,2}(r)$ are the endpoints of some intervals (i.e., $a_j(r)$ or $b_j(r)$). Since u_is are on the x-axis, be a slight abuse of notation, we let u_i denotes the x-coordinate of point u_i. For a given feasible solution $U = \{u_1, u_2, \ldots, u_{p+q}\}$, its **standard** form has two following properties:

1. If u_1 and u_{p+q} are on the endpoints.
2. Any two consecutive same color centers are on the endpoints.

Converting a given solution to standard form. If u_1 (resp. u_{p+q}) is not on an endpoint, we move it to the left (resp. right) to hit $x_{1,1}(r)$ (resp. $x_{p+q,2}(r)$). For every pair of two consecutive same color centers u_i and u_{i+1}, $1 \leq i \leq p+q-1$, if u_i is not on an endpoint, we move it to the right to hit $x_{i,2}(r)$ and if u_{i+1} is not on an endpoint we move it to the left to hit $x_{i+1,1}(r)$.

Clearly, the standard form solution as constructed above satisfies the covering and separation constraints. Let $S(k, j)$ denote a sequences of $j + 1$ consecutive centers in U starting from u_k, i.e., $(u_k, u_{k+1}, \ldots, u_{k+j})$. A sequence $S(k, j)$ is called **alternate** if for all $i, k \leq i \leq k + j - 1$, u_i and u_{i+1} have different colors and centers u_k and u_{k+j} are on the endpoints and the other centers of the sequence are not on the endpoints (such a center is called **internal**).

Now note that if U is a standard solution, then the consecutive red-blue centers can be clustered in some alternate sequences. These alternate sequences can be provided by scanning the centers from left to right and clustering a couple of consecutive blue-red centers between two endpoints that include a center. To this end, we have the following simple approach:

Clustering centers in alternate sequences. Let u_{cur} be the first center that has not been visited yet. At the beginning, $u_{cur} = u_1$. Let u_i be the next closest different color center to u_{cur}. All centers from u_{cur} to u_{i-1} should be on the endpoints since they are the same color. We can construct the next sequence from $k = i - 1$, i.e., we add u_{i-1} and u_i to a sequence. There are two events:

Event 1: u_i is on an endpoint, so the sequence is completed. If there are any unvisited centers, we continue scanning the centers by starting from u_i, i.e., we mark u_i as unvisited, set $u_{cur} = u_i$, and proceed as before until there is no unvisited center.

Event 2: u_i is not on an endpoint. Consider u_{i+1}. u_{i+1} and u_i should have different colors since U is standard. We add u_{i+1} to the sequence. If there are any unvisited centers, consider u_{i+2} and check Events 1 or 2 for $i = i + 2$.

Note that some of the centers may belong to two alternate sequences (e.g., a center on an endpoint with different color adjacent centers) and some of them may not be in a sequence (e.g., a center with same-color adjacent centers).

Now we prove a property of the optimal solutions in standard form for being able to find a discrete candidate set for the optimal radii.

▶ **Lemma 9.** *Let $U = \{u_1, u_2, \ldots, u_{p+q}\}$ be a feasible solution for the constrained $(n, p \wedge q, \alpha)$ problem with radius of covering r. If the distance between any two endpoints of the intervals in $\mathcal{C}(r)$ is not $t\alpha$, where $t \in \mathbb{Z}, 0 \leq t \leq p + q - 1$, then the constrained $(n, p \wedge q, \alpha)$ problem has a feasible solution with radius less than r.*

Proof. We will show that there is a real number $0 < \epsilon < r$ such that the constrained $(n, p \wedge q, \alpha)$ problem has a feasible solution with radius of covering $r - \epsilon$. To this end, we obtain a set of centers, $\bar{U}$, from the given feasible solution U and show that the set of balls centered at the points in $\bar{U}$ with radius $r - \epsilon$ is a feasible solution for the problem. First, we need to modify U to find a feasible solution with the property that any two consecutive blue and red centers are at a distance strictly greater than α (not exactly α). Then we use it for finding a solution, $\bar{U}$, with radius of covering $r - \epsilon$ (that is explained later).

First of all, we convert U to standard form and compute all alternate sequences of the standard solution. Then we use them to find a feasible solution with the property that any two consecutive blue and red centers are at a distance of strictly greater than α. Note that by the Lemma's assumption, each alternate sequence $S(k, j)$ has at least a pair of two consecutive centers at a distance of strictly greater than α (since each distance is at least α and sum of them is not $j\alpha$). But we need to have this strict inequality for all such pairs. So in each sequence $S(k, j)$, if there are two consecutive centers u_{k+i} and u_{k+i+1} at a distance of exactly α, we should perturb the internal centers such that the distance between any two consecutive blue and red centers is strictly greater than α.

For perturbing the internal centers of each alternate sequence $S(k, j)$, if $j = 1$, then $u_{k+1} - u_k > \alpha$, because u_k and u_{k+1} are on the endpoints so $u_{k+1} - u_k \neq \alpha$. If $j > 1$, we proceed by induction on the number of the pairs with the distance of α which is denoted by $n_{k,j}$. For $n_{k,j} = 1$, let $u_{k+i}, u_{k+i+1} \in S(k, j)$ such that $u_{k+i+1} - u_{k+i} = \alpha$. Since $j > 1$, at least one of u_{k+i} and u_{k+i+1} is internal, say u_{k+i}. Since $n_{k,j} = 1$, $u_{k+i} - u_{k+i-1} > \alpha$. So we can shift u_{k+i} toward u_{k+i-1} infinitesimally such that $u_{k+i} - u_{k+i-1} > \alpha$ and we still have $u_{k+i+1} - u_{k+i} > \alpha$. Assume for the induction hypothesis that for all integers $m > 1$, in a sequence $S(k, j)$ with $n_{k,j} < m$, including a pair of consecutive red and blue centers at a distance greater than α, we can perturb the internal centers such that all distances between two consecutive centers are strictly greater than α. Now assume that $n_{k,j} = m > 1$. For some $0 \leq i \leq j - 1$, let $u_{k+i}, u_{k+i+1} \in S(k, j)$ such that $u_{k+i+1} - u_{k+i} = \alpha$. $S(k, j)$ has a pair of two consecutive centers at a distance greater than α. This pair belongs to one of the sequences $S(k, i)$ or $S(k + i + 1, j)$, say $S(k, i)$. It is clear that $n_{k,i} < m$, so by the induction hypothesis, we can perturb the internal centers of $S(k, i)$ such that all distances between two consecutive centers are strictly greater than α. Next, we can move u_{k+i} toward u_{k+i-1} infinitesimally such that $u_{k+i} - u_{k+i-1} > \alpha$ and we now also have $u_{k+i+1} - u_{k+i} > \alpha$. Now we add u_{k+i} to $S(k + i + 1, j)$ to obtain $S(k + i, j)$ in which the distance between centers u_{k+i} and u_{k+i+1} is greater than α. Since $n_{k+i,j} < m$, again by the induction hypothesis, we can perturb the internal centers of $S(k+i, j)$ such that all distances between two consecutive centers are strictly greater than α. It means that $S(k, j)$ no longer contains a consecutive pair with distance α.

Now we can compute $\bar{U}$. Let $0 < \epsilon < r$ be a positive real number to be fixed later. If u_i is on an endpoint, say $x_{i,1}(r)$, let $\bar{u}_i = x_{i,1}(r - \epsilon)$, otherwise, $\bar{u}_i = u_i$. Notice that by our assumptions, there is no solution for $t = 0$ so all endpoints are distinct. As such for an u_i on an endpoint, it is never on two endpoints simultaneously and its movement is unambiguously determined. We will show that there exists an ϵ such that $\bar{U} = \{\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_{p+q}\}$ is a feasible solution with radius of covering $r - \epsilon$, i.e., $\bar{U}$ should satisfy the covering and separation constraints. To this end, firstly, after decreasing r to $r - \epsilon$, the relative order of the endpoints of the intervals should not change, i.e., the displacement of an endpoint of a face F_i should be less than $||F_i||/2$, where $||F_i||$ is the distance between the endpoints of face F_i. Secondly, for satisfying the covering constraint, the internal centers should remain in their faces, i.e., $x_{i,1}(r - \epsilon) < \bar{u}_i < x_{i,2}(r - \epsilon)$. So the displacement of point $x_{i,1}$ (resp. $x_{i,2}$) should be less than $u_i - x_{i,1}(r)$ (resp. $x_{i,2}(r) - u_i$). Finally, for satisfying the separation constraint, in each sequence $S(k,j)$, we should have $\bar{u}_{k+1} - \bar{u}_k \geq \alpha$ and $\bar{u}_{k+j} - \bar{u}_{k+j-1} \geq \alpha$ (note that the distance between two internal centers does not change), i.e., the displacement of the endpoint that contains u_k (resp. u_{k+j}) should be less than $u_{k+1} - u_k - \alpha$ (resp. $u_{k+j} - u_{k+j-1} - \alpha$). Therefore, by choosing real numbers δ_1, δ_2, δ_3 as follows, and $0 < \delta < \min\{\delta_1, \delta_2, \delta_3\}$, because of continuity of the movement of endpoints on line ℓ, we can obtain a positive ϵ such that the displacement of an endpoint becomes at most δ when the radius decreases to $r - \epsilon$.

$$0 < \delta_1 < 1/2 \min_{1 \leq i \leq 2n-1} \{||F_i||\}$$

$$0 < \delta_2 < \min_{\forall S(k,j)} \{ \min_{k+1 \leq i \leq k+j-1} \{u_i - x_{i,1}(r), x_{i,2}(r) - u_i\}\}$$

$$0 < \delta_3 < \min_{\forall S(k,j)} \{u_{k+1} - u_k - \alpha, u_{k+j} - u_{k+j-1} - \alpha\}$$

Consequently, there exists a non-zero $\epsilon > 0$ such that the balls centered at points in $\bar{U}$ with covering radius $r - \epsilon$ is a feasible solution. ◀

By Lemma 9, in the optimal solution, there is at least a pair of two endpoints at distance $t\alpha$, where $t \in \mathbb{Z}, 0 \leq t \leq p + q - 1$. The interval endpoints $a_i(r)$ and $b_i(r)$ are given by $a_i(r) = p_{i1} - \sqrt{r^2 - \sum_{j=2}^{d} p_{ij}^2}$, and $b_i(r) = p_{i1} + \sqrt{r^2 - \sum_{j=2}^{d} p_{ij}^2}$, so, a candidate set for the optimal radius can be computed by solving the following equations for all $1 \leq i, k \leq n$ and $t \in \mathbb{Z}, 0 \leq t \leq p + q - 1$:

$$p_{i1} \pm \sqrt{r^2 - \sum_{j=2}^{d} p_{ij}^2} - p_{k1} \pm \sqrt{r^2 - \sum_{j=2}^{d} p_{kj}^2} = t\alpha,$$

since at least one of those equalities holds true. Due to our general position assumption, i.e., no two points in P have the same distance from ℓ, these equations have a finite number of solutions. This is not too hard to show. See the full version of this paper [6] for the details. (We remark that the general position assumption can be removed. Without the assumption, Lemma 9 needs an amended statement and proof. Due to space constraints, we do not show this here. The amended statement and proof can be found in the full version of this paper [6].)

By solving these equations, we obtain $O(n^2(p+q))$ candidates for the optimal radius and this proves the following lemma.

▶ **Lemma 10.** *There is a set of $O(n^2(p+q))$ numbers, such that the optimal radius $r_{p \wedge q, \alpha}(P)$ is one among them, and this set can be constructed in $O(n^2(p+q))$ time.*

4.3 Main result

By first computing the candidates for r^* and then performing a binary search over them using the feasibility testing algorithm, we can compute the optimal radius. Thus we have the following theorem.

▶ **Theorem 11.** *The constrained $(n, p \wedge q, \alpha)$ problem can be solved in $O(n^2(p + q) + T_{DP}(n, p, q) \log n) = O(n^6 pq \log n)$ time.*

5 Conclusions

Improving the approximation factor of our main approximation algorithm (Theorem 4) and the running time of our polynomial-time algorithm for the constrained problem (Theorem 11) are obvious candidates for problems for future research work. Apart from this, it seems that a multi-color generalization of the k-center problem is worth studying for modeling similar practical applications. Here we want k different colored centers, and balls of each color covering all of P but with the separation constraints more general, i.e., between the centers of colors i, j the distance must be at least some given α_{ij}. It seems that new techniques would be required for this general problem.

References

1. P. K. Agarwal and C. M. Procopiuc. Exact and approximation algorithms for clustering. *Algorithmica*, 33(2):201–226, 2002. doi:10.1007/s00453-001-0110-y.
2. P. Bose, S. Langerman, and S. Roy. Smallest enclosing circle centered on a query line segment. In *Proc. of Can. Conf. on Comp. Geom.*, 2008.
3. P. Bose and G. Toussaint. Computing the constrained euclidean geodesic and link center of a simple polygon with applications. In *Proc. Pacific Graph. Int.*, pages 102–112, 1996. doi:10.1109/CGI.1996.511792.
4. P. Brass, C. Knauer, H.-Suk Na, C.-Su Shin, and A. Vigneron. The aligned k-center problem. *Int. J. Comp. Geom. Appl.*, 21(2):157–178, 2011. doi:10.1142/S0218195911003597.
5. G. Das, S. Roy, S. Das, and S. Nandy. Variations of base-station placement problem on the boundary of a convex region. *Int. J. Found. Comput. Sci.*, 19:405–427, 2008. doi:10.1142/S0129054108005747.
6. M. Eskandari, B. B. Khare, and N. Kumar. Separated red blue center clustering, 2021. arXiv:2107.07914.
7. T. F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Th. Comp. Sc.*, 38:293–306, 1985. doi:10.1016/0304-3975(85)90224-5.
8. F. Hurtado and G. Toussaint. Constrained facility location. *Studies of Location Analysis, Sp. Iss. on Comp. Geom.*, pages 15–17, 2000.
9. R. Z. Hwang, R. C. T. Lee, and R. C. Chang. The slab dividing approach to solve the euclidean p-center problem. *Algorithmica*, 9:1–22, 1993. doi:10.1007/BF01185335.
10. P. Kavand, A. Mohades, and M. Eskandari. $(n, 1, 1, \alpha)$-center problem. *Amirkabir Int. J. of Sc. & Res.*, 2014.
11. N. Megiddo. Linear time algorithms for linear programming in $\mathbb{R}^3$. *SIAM J. Comput.*, 12(4), 1983. doi:10.1137/0212052.
12. N. Megiddo and K. J. Supowit. On the complexity of some common geometric location problems. *SIAM J. Comput.*, 13(1):182–196, 1984. doi:10.1137/0213014.
13. S. Roy, D. Bardhan, and S. Das. Efficient algorithm for placing base stations by avoiding forbidden zone. In *Proc. of the Sec. Int. Conf. Dist. Comp. and Int. Tech.*, pages 105–116, 2005. doi:10.1007/11604655_14.
14. C.-S. Shin, J.-H. Kim, S. K. Kim, and K.-Y. Chwa. Two-center problems for a convex polygon. In *Proc. of the 6th Ann. Euro. Symp. Alg.*, pages 199–210, 1998. doi:10.1007/3-540-68530-8_17.
15. J. J. Sylvester. A question in the geometry of situation. *Quart. J. Math.*, 322(10):79, 1857.