

# Minimizing the Range for $k$ -Covered Paths on Sensor Networks

MANUEL ABELLANAS<sup>1</sup>, ANTONIO LESLIE BAJUELOS<sup>2</sup> AND INÊS MATOS<sup>2,3,\*</sup>

<sup>1</sup>*Departamento de Matemática Aplicada (UPM), Campus de Montegancedo, 28660 Boadilla del Monte, Madrid, Spain*

<sup>2</sup>*Departamento de Matemática (UA), Campus Universitário de Santiago, 3810-193 Aveiro, Portugal*

<sup>3</sup>*Departament de Matemàtica Aplicada II (UPC), Edifici Omega. C. Jordi Girona, 1-3, 08034 Barcelona, Spain*

\*Corresponding author: ipmatos@ua.pt

Coverage problems are a flourishing topic in optimization, thanks to the recent advances in the field of wireless sensor networks. The main coverage issue centres around critical conditions that require reliable monitoring and prohibit failures. This issue can be addressed by maximal-exposure paths, regarding which this article presents new results. Namely, it shows how to minimize the sensing range of a set of sensors in order to ensure the existence of a  $k$ -covered path between two points on a given region. Such a path's coverage depends on  $k \geq 2$ , which is fixed. The three types of regions studied are: a planar graph, the whole plane and a polygonal region.

*Keywords:* geometric optimization; multiple coverage; higher-order Voronoi diagrams; sensor networks

Received 6 May 2011; revised 13 July 2011

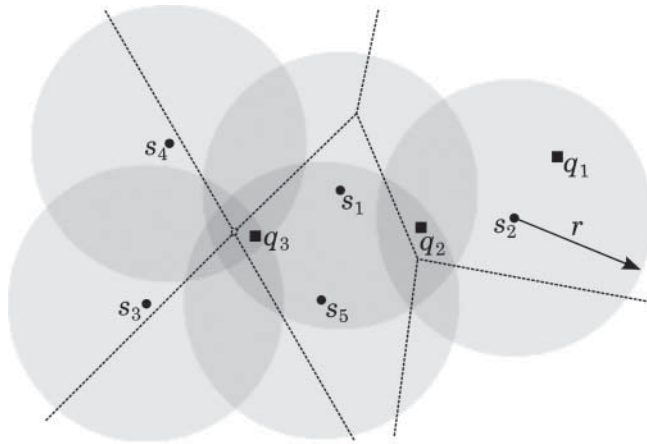
Handling editor: Mostafa Bassiouni

## 1. INTRODUCTION

Let  $S$  be a set of  $n$  points on the plane that represent the location of  $n$  sensors. This discussion follows the widely adopted disc model for sensing—that is, each sensor is only capable of detecting points within distance  $r \in \mathbb{R}^+$  (common to all sensors). All distances are considered to be Euclidean. Further, assume that the communication range of each sensor is at least twice its sensing range so that two sensors can communicate when they are able to detect each other [1, 2]. Formally, the distance between a point  $q$  on the plane and a set  $S$  is defined as the minimum distance between  $q$  and any one point of  $S$ . Point  $q$  is said to be *covered* by a set  $S$  of sensors with sensing range  $r$  if the distance between  $q$  and  $S$  is less than or equal to  $r$ . Therefore,  $q$  is said to be  *$k$ -covered* by  $S$  if it is covered by at least  $k$  sensors of  $S$  (see Fig. 1). The first articles that mention multiple coverage were introduced as Art Gallery problems: Pesant [3] and Belleville *et al.* [4] studied how to find a collection of guards to  $k$ -guard a simple polygon,  $k \in \mathbb{N}$ . A geometric structure that is invariably associated with this topic is the Voronoi diagram of  $S$  [5, 6]. Such diagram divides the plane into several regions by grouping points that have the same closest sensor. Accordingly, the  $k$ th order Voronoi diagram of  $S$ ,  $VD_k(S)$ , groups points on the plane that share the same  $k$

closest sensors,  $k \in \{1, \dots, n-1\}$  (see Fig. 1). This type of diagram is called the *higher-order Voronoi diagram* (note that ‘higher’ is not the space dimension, but the number of points defining each Voronoi region). Higher-order Voronoi diagrams are naturally associated with multiple coverage since a point is  $k$ -covered if it is at least within range of its  $k$  closest sensors.

One way to evaluate the quality of the coverage provided by a particular sensor network is to find minimal- and maximal-exposure paths. The minimal-exposure path can be seen as the worst covered path. Since sensor networks are designed to monitor sensor fields, the minimal-exposure path is a way to measure the ability to move inside the sensor field without being detected. Therefore, it is important to have a sensor network that maximizes such minimum. Anirvan *et al.* [7] further explored minimal-exposure paths and proposed two asymptotically efficient algorithms to compute them. Moreover, they also showed that this path does not necessarily lie along the edges of the Voronoi diagram. On the other hand, the maximal-exposure path corresponds to the most covered path and it was proved to be NP-hard to find by Veltri *et al.* [8]. Fang and Low [9] generalized the problem of finding maximal- and minimal-exposure paths to  $k$ -coverage, when  $k$  is known. By adopting the method of growing discs, they transformed their problem



**FIGURE 1.**  $VD_2(S)$  is shown in a black dashed trace. Since the sensors have sensing range  $r$ , point  $q_1$  is 1-covered, point  $q_2$  is 3-covered and point  $q_3$  is 4-covered.

into finding a series of adjacent discs' borders and proposed a polynomial time algorithm to solve it (unfortunately no explicit complexities were presented).

Given any wireless sensor network, it is clear that larger sensing ranges provide better coverage. However, larger ranges also result in higher costs and shorten the sensors' battery life. Therefore, it is appropriate to develop strategies that optimize the coverage of a given region without compromising the network's lifespan. For example, sensors' battery life can be extended by reducing the devices' sensing and transmission range as much as optimum, thus enhancing the overall user experience [10–12]. There are several techniques that have been developed and tested to tackle this problem. The first work that brought together *ad hoc* sensor networks and computational geometry was solely focused on simple coverage [13, 14], but it was built on by several other authors in the years that followed. Based on the work of Wang *et al.* [1], Zhou *et al.* [15] proposed a centralized greedy algorithm to compute a minimum energy-cost  $k$ -coverage, which is within a factor  $\mathcal{O}(\log n)$  of optimal. Efrat *et al.* [16] also studied the problem of minimizing the number of sensors required to 'well cover' a given region. According to their definition, a point is 'well covered' if it is detected by three sensors that form a triangle containing the point or if it is detected by two sensors that are separated by an angle of at least  $\alpha$ . The first definition is clearly a variation of 3-coverage. As sensor networks are expected to outlast individual sensors, the technique *sleep-wakeup* was introduced as a means to extend the lifetime of sensor networks. To this end, a set of sensors is chosen to be activated while the rest are turned off (asleep). If one of the active sensors stops working, then one or more of the sleeping sensors has to be brought up to replace it and maintain the desired coverage. Kumar *et al.* [17] considered the problem of determining the number of sensors needed to  $k$ -cover a region when sensors were allowed to sleep for most of their lifetime. This idea was then translated to  $k$ -barrier coverage [18, 19], where sleep-wakeup

schedules were proposed to maximize the network's lifespan. Their algorithms work for sensing regions that are not modelled as discs and even allow a different sensing region for each sensor. Moreover, the authors solved the same problem when sensors have a distinct lifespan. Interestingly, for this type of coverage, sensors that have the same lifespan reduce the sensor network's lifetime by two-thirds. Zhou *et al.* [15] focused on selecting a minimum number of sensors to  $k$ -cover a region, provided that the communication graph induced by such set of sensors was connected. To this end, they presented a greedy algorithm that delivers a near-optimal solution, as well as a distributed version. Huang *et al.* [20] combined coverage and connectivity as a generalization of their previous work. They proposed theorems to determine the levels of coverage and connectivity of a sensor network, and so made it possible to extend the network's lifetime by setting the required coverage while turning off some sensors or by adjusting their ranges.

Gupta *et al.* [21] took a different approach to reduce sensor networks' costs. They used the ability of the network to provide multiple coverage to choose a small subset of sensors sufficient to process queries and then re-organized the network using this set. For this, they introduced the concept of *connected sensor cover* and proposed a centralized algorithm to construct a topology involving a near-optimal connected sensor cover. This algorithm is approximated because the problem is NP-hard, which is then reflected in the problem of finding a  $k$ -connected sensor cover. We further explored how to extend a sensor network's lifetime by minimizing the sensors' range to 2-cover a path on the plane [22] or within a polygonal region [23]. In the latter it is also shown how to calculate the sensors' minimum range in order to fully 2-cover a polygonal region. Both results will be extended to  $k$ -coverage in the following. Working in the same direction but also combining uncertainty in sensor positions, Vu and Zheng [24] addressed the problem of minimizing the sensors' range while ensuring the  $k$ -coverage of a point or a polygonal region. This article will be further discussed in Section 2.3.

Our contribution is also an attempt at extending the life of a sensor network. In this way, the objective is to minimize the sensors' range in order to assure the existence of a  $k$ -covered path between two points on a given region. The quality of the path's coverage depends on  $k \geq 2$ , which is known. The present research is focused on maximal-exposure paths within three types of regions: a planar graph, the whole plane and a polygonal region. Each of these regions will be analysed in detail in the following section. This paper concludes with Section 3, which discusses some results and possible further research.

## 2. MINIMIZING THE SENSING RANGE FOR $K$ -COVERAGE

As previously explained, the minimum sensing range that covers  $q$  is the shortest distance between  $q$  and set  $S$  of sensors. Let

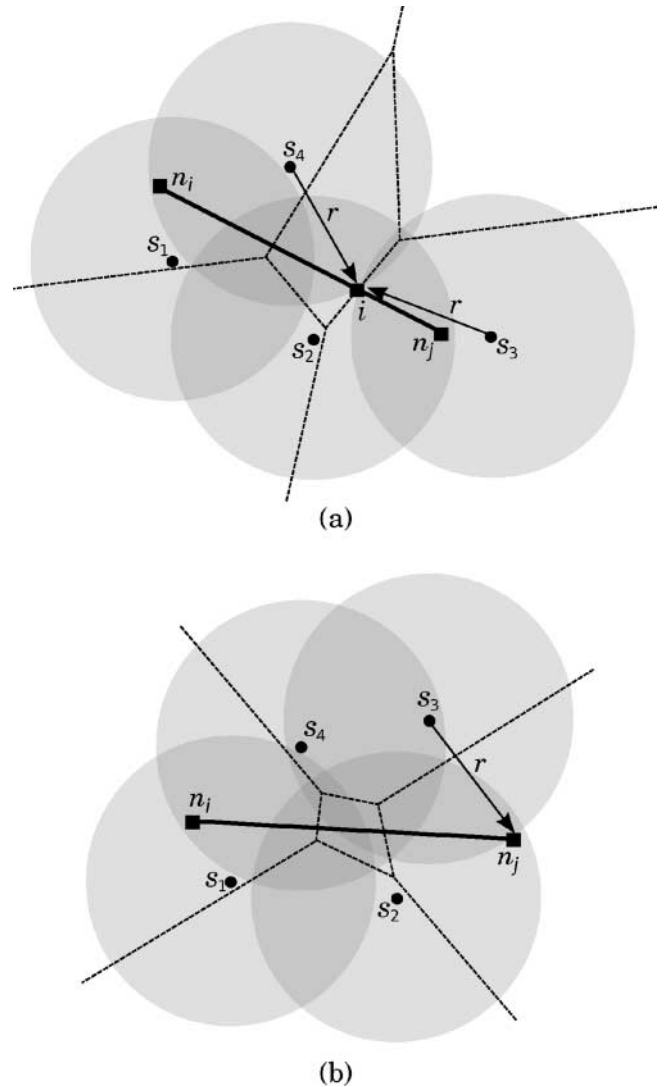
$MR_S^k(x)$  denote the minimum sensing range of  $S$  that assures that an object  $x$  is  $k$ -covered. In the following discussion, object  $x$  is often a path between two points and its desired coverage is given by  $k \geq 2$ , which is a known value. A path that is  $k$ -covered by  $S$  is called a  $k$ -covered path, or a  $k$ -path. The main goal of this section is to find the minimum sensing range of  $S$  such that a  $k$ -path exists on a planar graph in Section 2.1, on the plane in Section 2.2 and finally on a polygonal region in Section 2.4. This last subsection will be preceded by Section 2.3, where it is explained how to calculate the minimum sensing range to fully  $k$ -cover a polygonal region.

### 2.1. Minimizing the sensing range for $k$ -coverage: path on a planar graph

Let  $G$  be a connected planar geometric graph. Suppose the edges of  $G$  represent streets/roads and its nodes represent reachable locations using those streets. Consequently,  $G$  can be regarded as a model of a street network and a path on such graph may be seen as journey from one location to another. Let a path between two nodes  $n_i$  and  $n_j$  of  $G$  be denoted by  $P(n_i, n_j)$ . Using the notation previously introduced and knowing the value of  $k \geq 2$ , the objective of this section is to calculate  $MR_S^k(P(n_i, n_j))$ —that is, the minimum sensing range of  $S$  that assures the existence of a  $k$ -path between  $n_i$  and  $n_j$  on  $G$ . The solution to this problem is presented in the following and is divided into two phases: preprocess and solution.

In the preprocess, graph  $G$  is transformed into edge-weighted graph  $G_w$ . In so doing, the weight of the edges must be properly chosen to suit the problem. Therefore, if  $e$  is an edge of  $G$ , then an appropriate weight for  $e$  is the range  $MR_S^k(e)$ —that is, the minimum sensing range of  $S$  that  $k$ -covers  $e$ . Thus, the weight of  $e$  is calculated as the distance between a point of  $e$  and its  $k$ th closest sensor. To find this particular point, assume that the sensors' range is large enough to  $k$ -cover the whole edge. Now suppose that this range is slowly reduced until there is one point  $i$  on  $e$  that is no longer  $k$ -covered. As a result, the edge is no longer  $k$ -covered for such range nor will it be unless the sensors' range increases. Consequently, the minimum range that  $k$ -covers  $i$  is exactly the minimum range that  $k$ -covers the whole edge  $e$ . Note that point  $i$  is either a point interior to  $e$  or an endpoint (see Fig. 2). Furthermore, if  $i$  is not an endpoint of  $e$  then it has to be an intersection point between  $e$  and the  $k$ th order Voronoi diagram of  $S$ ,  $VD_k(S)$ . Consequently,  $MR_S^k(e) = \max\{MR_S^k(i) : i \in I\}$ , where  $I$  is the set of both endpoints of  $e$  plus the intersection points between  $e$  and  $VD_k(S)$ .

If  $w(e)$  is the weight of edge  $e$ , then  $G$  can be converted into edge-weighted graph  $G_w$  by assigning  $w(e) = MR_S^k(e)$  to each edge  $e$  of  $G$ . Observe that  $MR_S^k(G_w) = \max\{w(e) : e \in G_w\}$ . The following proposition states the complexity of this preprocess.



**FIGURE 2.** Edge  $\overline{n_i n_j}$  is 2-covered by  $S$  with minimum range  $r$  and  $VD_2(S)$  is represented by a dashed line. (a)  $MR_S^2(\overline{n_i n_j}) = d(i, s_3) = d(i, s_4) = r$ . (b)  $MR_S^2(\overline{n_i n_j}) = d(n_i, s_3) = r$ .

**PROPOSITION 2.1.** Let  $S$  be a set of  $n$  sensors and  $G$  a connected geometric planar graph with  $m$  edges. Given  $k \geq 2$ , the weight of every edge of  $G$  can be calculated in  $\mathcal{O}(k^2 n \log n + mk(n - k))$  time and  $\mathcal{O}(k^2(n - k) + mk(n - k))$  space. This is also the complexity of finding  $MR_S^k(G)$ .

*Proof.* Computing  $VD_k(S)$  takes  $\mathcal{O}(k^2 n \log n)$  time since  $S$  is a set of  $n$  sensors on the plane, and the resulting diagram has  $\mathcal{O}(k(n - k))$  regions, edges and vertices [25]. Therefore, set  $I_e = \{e\} \cap VD_k(S)$  plus the endpoints of  $e$  is found in  $\mathcal{O}(k(n - k))$  time for each edge  $e \in G$ . Calculating  $MR_S^k(e) = \max\{MR_S^k(i) : i \in I_e\}$  for every edge of  $G$  takes  $\mathcal{O}(mk(n - k))$  time since  $G$  has  $m$  edges. Consequently, the minimum range of  $S$  to  $k$ -cover  $G$  can be found in  $\mathcal{O}(k^2 n \log n + mk(n - k))$  time. Regarding the space complexity, this procedure takes  $\mathcal{O}(k^2(n - k) + mk(n - k))$  space [25].  $\square$

It is apparent that the minimum range of  $S$  that assures the existence of a  $k$ -path between two nodes of  $G_w$  is given by the weight of the heaviest edge of such path. The following proposition shows that it is only necessary to consider the edges of a minimum spanning tree (MST) of  $G_w$ .

**PROPOSITION 2.2.** *Let  $G_w$  be an edge-weighted connected graph. For each path on  $G_w$ , assume that the path's weight is given by the weight of its heaviest edge. Then the path on an MST of  $G_w$  connecting any pair of nodes of  $G_w$  is a minimum-weight path between such pair.*

*Proof.* Let  $G_w$  be an edge-weighted connected graph,  $n_i$  and  $n_j$  two of its nodes and  $T_w$  an MST of  $G_w$ . Suppose that  $P(n_i, n_j)$  is the only path on  $T_w$  connecting nodes  $n_i$  and  $n_j$  and  $e$  is its heaviest edge. Consequently,  $P(n_i, n_j)$  has weight  $w(e)$ . Now suppose that path  $P^*(n_i, n_j)$  on  $G_w$  is a minimum-weight path connecting  $n_i$  and  $n_j$ . Its weight is given by  $e^*$ , its heaviest edge, and so  $w(e^*) < w(e)$ . Since  $P(n_i, n_j)$  is heavier than  $P^*(n_i, n_j)$ , the edge  $e$  is not an edge of  $P^*(n_i, n_j)$ . If the two paths are united, then a cycle is created. Such cycle contains  $e$ , which clearly is its heaviest edge. But this contradicts the hypothesis, since the heaviest edge of a cycle in  $G_w$  cannot be on an MST of  $G_w$ . Therefore, the path connecting two nodes on an MST of  $G_w$  is a minimum-weight path between those nodes of  $G_w$ .  $\square$

The result given earlier suggests an algorithm to calculate  $\text{MR}_S^k(P(n_i, n_j))$  since paths are unique between two nodes of an MST (see Fig. 3a). It suffices to build  $T_w$ , an MST of  $G_w$ , and find a path there between nodes  $n_i$  and  $n_j$ . Such a path is a minimum-weight path and therefore  $\text{MR}_S^k(P(n_i, n_j))$  is given by the weight of its heaviest edge.

In Fig. 3b, there is an example of a path on an MST connecting nodes  $n_1$  and  $n_2$ . Note that path  $P(n_1, n_2)$  traverses nodes

$n_{10}$ ,  $n_6$  and  $n_7$ . If the sensors' sensing range is larger than  $38 = \max\{33, 32, 28, 38\}$ , then  $P(n_1, n_2)$  is a path 2-covered by  $S$ . If this range is exactly 38, then  $P(n_1, n_2)$  is 2-covered with minimum sensing range, and so  $\text{MR}_S^2(P(n_1, n_2)) = 38$ . The following result states the temporal complexity of the algorithm discussed earlier.

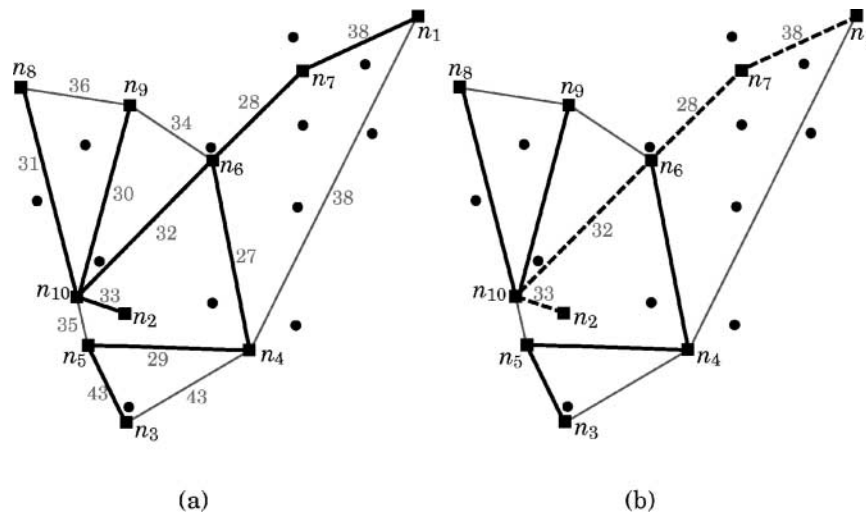
**THEOREM 2.1.** *Let  $S$  be a set of  $n$  sensors,  $G_w$  an edge-weighted planar graph with  $m$  edges and  $T_w$  an MST of  $G_w$ . Given two nodes  $n_i$  and  $n_j$  of  $G_w$ , a path between  $n_i$  and  $n_j$  and  $\text{MR}_S^k(P(n_i, n_j))$  can be found on  $T_w$  in  $\mathcal{O}(m)$  time.*

*Proof.* An MST of  $G_w$ ,  $T_w = (N, B)$ , can be found in  $\mathcal{O}(m)$  time since  $G_w$  is a planar graph [26]. A path  $P(n_i, n_j)$  on  $T_w$  can be found using the DFS algorithm [27], which runs in  $\mathcal{O}(|B|)$  time. According to Proposition 2.2,  $P(n_i, n_j)$  is a minimum-weight path between  $n_i$  and  $n_j$  on  $G_w$ . Since  $\text{MR}_S^k(P(n_i, n_j))$  is given by the weight of  $P(n_i, n_j)$ , then it can be calculated in time proportional to the number of edges of  $P(n_i, n_j)$ . Overall, this procedure takes  $\mathcal{O}(m)$  time.  $\square$

Note that the previous algorithm works for any connected geometric graph, but the temporal and space complexities require the appropriate adjustments if the graph is not planar. Furthermore, once the requested path is found, it is clear how to choose the sensors of  $S$  that contribute to  $k$ -cover such portion of the graph and turn off the rest to save power.

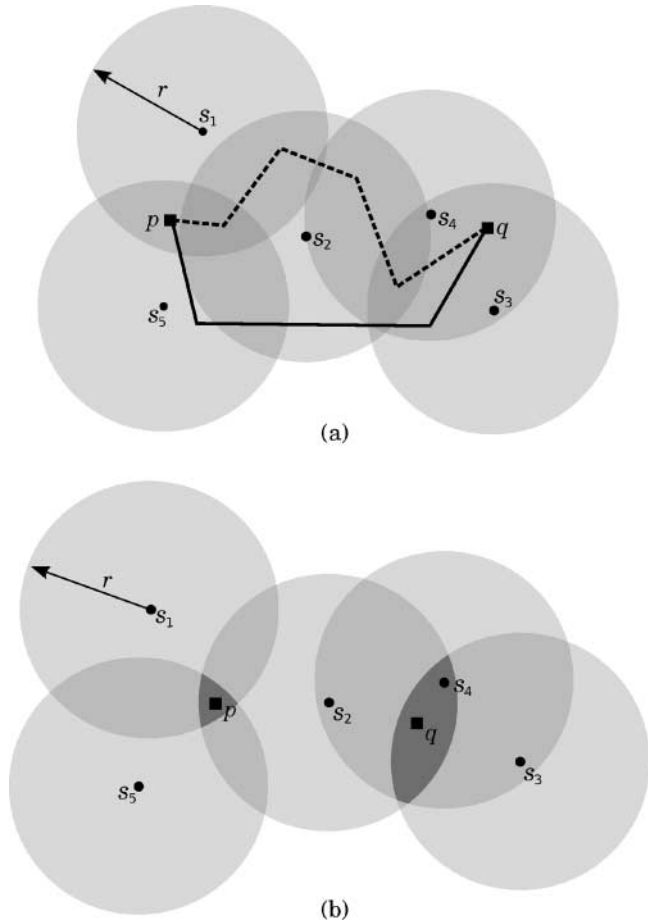
## 2.2. Minimizing the sensing range for $k$ -coverage: path on the plane

Given  $k \geq 2$  and two points  $p$  and  $q$  on the plane, the objective of the following discussion is to calculate the minimum sensing range of  $S$  that allows the existence of a  $k$ -path between  $p$  and  $q$ . As mentioned before, such a range is denoted by  $\text{MR}_S^k(P(p, q))$ .



**FIGURE 3.** (a) An MST of the edge-weighted graph is shown in a darker line. (b) The dashed path connecting  $n_1$  to  $n_2$  on the tree only is 2-covered if the sensors' range is at least  $\max\{33, 32, 28, 38\} = 38$ .



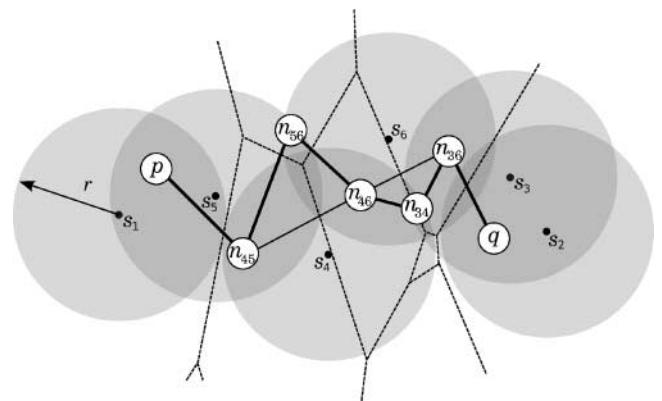


**FIGURE 4.** A set of five sensors with range  $r$ . (a) The path represented by a dashed line is a 2-path connecting  $p$  to  $q$ , whilst the one represented by a solid line is not. (b) Two connected components of the union of 3-blobs are shown in darker grey. Only 1-covered paths between  $p$  and  $q$  are possible if  $S$  has range  $r$ .

In the example in Fig. 4a, the black path connecting  $p$  and  $q$  is not a 2-path since some of its points are only covered by one sensor. On the other hand, the dashed path represents a 2-covered path between  $p$  and  $q$ . As in the previous section, the solution to this problem is divided into two phases. The first is focused on the decision problem: decide if a given sensing range  $r \in \mathbb{R}^+$  of  $S$  is enough to  $k$ -cover a path between  $p$  and  $q$ . The solution to this problem is the basis of the algorithm to calculate  $MR_S^k(P(p, q))$ .

Let  $D_S = \{D(s_i, r) : s_i \in S\}$  be the set of discs centred at sensors of  $S$  of radius  $r$ . Set  $D_S$  is the union of the points on the plane that are at least 1-covered by  $S$ . The points that are at least 2-covered exist on the intersections between each pair of discs. Accordingly, the set of  $k$ -covered points is the union of the intersections between each set of  $k$  discs. If each intersection between  $k$  discs of  $D_S$  is a  $k$ -blob, then the set of  $k$ -covered points is the union of these  $k$ -blobs. Therefore, a  $k$ -covered path

between  $p$  and  $q$  only exists if  $p$  and  $q$  are on the same connected component of  $k$ -blobs (see Fig. 4b). Hence the overlapping of discs of  $D_S$  is the key to finding a  $k$ -path between  $p$  and  $q$ . Consider the intersection graph of the set of  $k$ -blobs, that is, the graph where each node represents a  $k$ -blob and two nodes are connected if the respective  $k$ -blobs intersect each other. If  $r$  is large enough, then the intersection of any  $k$  discs of  $D_S$  is not empty and therefore the intersection graph is a complete graph. If point  $p$  is assigned to node  $n_p$  that corresponds to the  $k$ -blob containing  $p$  and  $q$  to the node  $n_q$ , then the existence of a  $k$ -path between  $p$  and  $q$  depends on whether  $n_p$  and  $n_q$  lie in the same connected component of this graph. However, this is a highly complex graph and so a subgraph  $G$  will be used instead. Graph  $G$  plays the same role as the intersection graph, but is at most quadratic in size and can be constructed by means of the higher-order Voronoi diagram of  $S$ . Each Voronoi region of  $VD_k(S)$  is defined by a set of  $k$  sensors, so if the sensing range  $r$  is enough for them to create a non-empty  $k$ -blob, add a node to graph  $G$  on such  $k$ -blob (see Fig. 5). Two nodes of  $G$  are connected if the corresponding Voronoi regions are neighbours and the two  $k$ -blobs they define intersect each other. Once  $G$  is built, the decision problem can be solved. First, it is important to check if both  $p$  and  $q$  are inside the  $k$ -blob formed by their closest  $k$  sensors, because otherwise there is not a  $k$ -path between them. If both points are  $k$ -covered, then assign them to the nodes that correspond to the Voronoi regions of  $VD_k(S)$  they belong to. Starting at one of these nodes, traverse  $G$  using the DFS algorithm [27]. If the other node is found, then both points belong to the same connected component of  $k$ -blobs and there is a  $k$ -path between them. In Fig. 5, there is an example of graph  $G$  and a path between  $p$  and  $q$ . For that set of sensors with sensing range  $r$ , a 2-covered path connecting  $p$  to  $q$  exists since both points lie in the same connected component of 2-blobs. The following result proves the temporal and space complexities of this decision algorithm.



**FIGURE 5.** Set  $S$  is represented by dots, graph  $G$  by a solid line and  $VD_2(S)$  by a dashed line. The darker edges of  $G$  form a path between  $p$  and  $q$ .

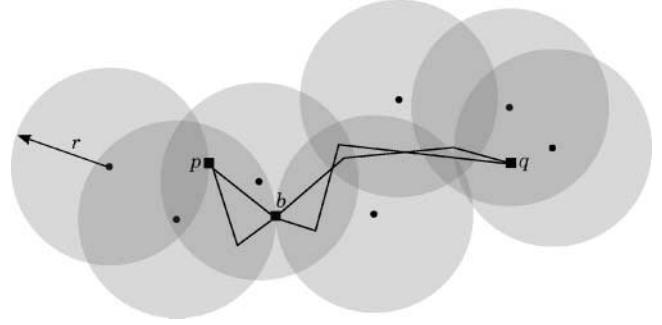
**THEOREM 2.2.** *Given a set  $S$  of  $n$  sensors with sensing range  $r \in \mathbb{R}^+$ ,  $k \geq 2$  and two points  $p$  and  $q$  on the plane, deciding if there is a  $k$ -covered path between  $p$  and  $q$  takes  $\mathcal{O}(k^2 n \log n)$  time and  $\mathcal{O}(k^2(n-k))$  space.*

*Proof.* Constructing  $\text{VD}_k(S)$  takes  $\mathcal{O}(k^2 n \log n)$  time since  $S$  is a set of  $n$  sensors on the plane [25]. Graph  $G$  is a subgraph of the dual graph of  $\text{VD}_k(S)$  and therefore has size  $\mathcal{O}(k(n-k))$  [25]. The construction of  $G$  takes  $\mathcal{O}(k(n-k))$  time because the algorithm performs a constant number of operations per edge of the Voronoi diagram. In the worst case, the DFS algorithm has to visit every node of  $G$  twice, and so its complexity depends on the number of edges and nodes of  $G$ . Consequently, the DFS algorithm runs in  $\mathcal{O}(k(n-k))$  time as well [27]. In conclusion, the decision algorithm runs in  $\mathcal{O}(k^2 n \log n)$  time. Regarding the space complexity,  $\text{VD}_k(S)$  can be stored in  $\mathcal{O}(k^2(n-k))$  space [25].  $\square$

This decision algorithm is a key method to solve several problems associated with  $k$ -coverage. As a result, it will be often applied in the remainder of this paper. It is left to explain how to calculate  $\text{MR}_S^k(P(p, q))$ . Note that for  $|S| \geq k$ , this problem always has a solution because, if  $r$  is large enough, all discs of  $D_S$  contain  $p$  and  $q$ . If  $p$  or  $q$  is a sensor, then it is considered that the sensor covers itself. As previously noted, both points have to be  $k$ -covered in order for a  $k$ -path to exist between them. Consequently, if  $s_p$  is the  $k$ th closest sensor to  $p$  and  $s_q$  the  $k$ th closest sensor to  $q$ , then  $\text{MR}_S^k(P(p, q)) \geq \max\{d(s_p, p), d(s_q, q)\}$ . Furthermore, if a  $k$ -path exists between  $p$  and  $q$  when the sensing range is  $r = \max\{d(p, s_p), d(q, s_q)\}$ , then clearly  $\text{MR}_S^k(P(p, q)) = r$ .

Let us concentrate in the case  $\text{MR}_S^k(P(p, q)) > \max\{d(s_p, p), d(s_q, q)\}$ . If the sensing range is  $r = \max\{d(s_p, p), d(s_q, q)\}$ , then  $p$  and  $q$  lie in different connected components of the union of  $k$ -blobs. As range increases, the discs of  $D_S$  expand accordingly and will eventually unite  $p$  and  $q$  under the same connected component of  $k$ -blobs. The moment these two connected components intersect each other defines  $\text{MR}_S^k(P(p, q))$ . The intersection points between the components at this specific range are called *bottleneck-points* for  $k$ -paths between  $p$  and  $q$  (see Fig. 6). Note that in degenerated cases, there might be more than one bottleneck-point for  $k$ -paths between  $p$  and  $q$ . For the purposes of this paper, degenerate input data will not be considered. Let  $b_S^k(p, q)$  denote the bottleneck-point for  $k$ -paths between  $p$  and  $q$ . If the sensing range is precisely  $\text{MR}_S^k(P(p, q))$  then every  $k$ -path connecting  $p$  and  $q$  crosses  $b_S^k(p, q)$ , since  $b_S^k(p, q)$  is the only point connecting the component of  $k$ -blobs that contains  $p$  to the one containing  $q$ . Bottleneck-points may be classified into two categories according to the following definition.

**DEFINITION 1.** *Let  $p$  and  $q$  be two points on the plane,  $k \geq 2$  and  $S$  a set of sensors with range  $r = \text{MR}_S^k(P(p, q))$ . Point  $b = b_S^k(p, q)$  is said to be a type I bottleneck-point if there are exactly  $k-1$  sensors such that  $b$  lies inside each disc*



**FIGURE 6.** The sensing range is  $r = \text{MR}_S^k(P(p, q))$ . The connected component of 2-blobs that contains  $p$  meets the component containing  $q$  at point  $b$ . Two possible 2-paths connecting  $p$  to  $q$  are represented by a solid trace.

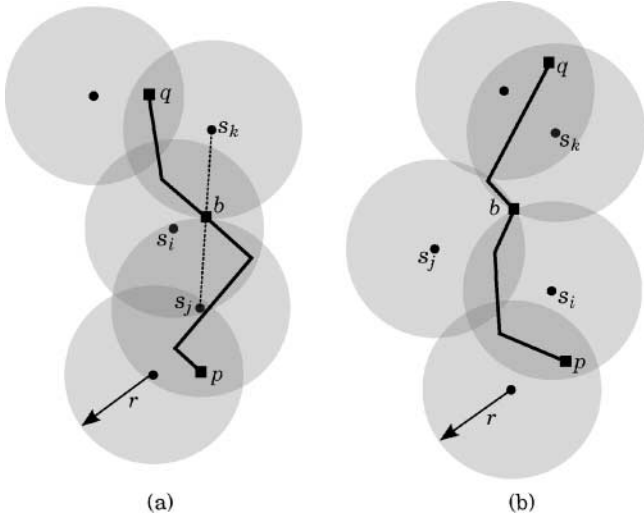
of  $\{D(s_i, r) : s_i \in \{s_{j_1}, \dots, s_{j_{k-1}}\}\}$ , and two more so that  $D(s_{j_1}, r) \cap \dots \cap D(s_{j_{k+1}}, r) = \{b\}$ . Otherwise  $b$  is a type II bottleneck-point if there are exactly  $k-2$  sensors such that  $b$  lies inside each disc of  $\{D(s_i, r) : s_i \in \{s_{j_1}, \dots, s_{j_{k-2}}\}\}$ , and three more so that  $D(s_{j_1}, r) \cap \dots \cap D(s_{j_{k+1}}, r) = \{b\}$ .

**PROPOSITION 2.3.** *Let  $p$  and  $q$  be two points on the plane,  $k \geq 2$  and  $S$  a set of sensors. Then the bottleneck-point for  $k$ -paths between  $p$  and  $q$  can only be of type I or II.*

*Proof.* As the sensors' range increases, the two connected components of the union of  $k$ -blobs (one containing  $p$  and the other containing  $q$ ) eventually unite at  $b_S^k(p, q)$ . If this union is made through the intersection of two  $k$ -blobs, then the intersection point is a type I bottleneck-point. Otherwise, the union is made through the intersection of three  $k$ -blobs at a time and that intersection point is a type II bottleneck-point. Recall that no degenerate cases are considered.  $\square$

Both types of bottleneck-points can be seen in Fig. 7. According to Definition 1, a type I bottleneck-point for  $k$ -paths between  $p$  and  $q$  lies on an edge of  $\text{VD}_k(S)$ . Suppose such an edge is part of the perpendicular bisector between sensors  $s_j$  and  $s_k$ , denoted by  $\text{PB}(s_j, s_k)$ . Then the bottleneck-point is the midpoint of the segment  $\overline{s_j s_k}$  (see Fig. 7a). The following proposition establishes a connection between bottleneck-points and higher-order Voronoi diagrams.

**PROPOSITION 2.4.** *Let  $p$  and  $q$  be two points on the plane,  $k \geq 2$  and  $S$  a set of sensors with range  $r = \text{MR}_S^k(P(p, q))$ . If  $b = b_S^k(p, q)$  is a type I bottleneck-point, suppose  $s_{j_1}$  and  $s_{j_2}$  are the two sensors that cover  $b$  but  $D(s_{j_1}, r)$  and  $D(s_{j_2}, r)$  do not contain  $b$ . Then  $b$  is on the edge of  $\text{VD}_k(S)$  that is part of  $\text{PB}(s_{j_1}, s_{j_2})$ . If  $b$  is a type II bottleneck-point, suppose  $s_{j_1}, s_{j_2}$  and  $s_{j_3}$  are the three sensors whose discs of radius  $r$  do not contain  $b$ . Then  $b$  is the vertex of  $\text{VD}_k(S)$  defined by  $s_{j_1}, s_{j_2}$  and  $s_{j_3}$ .*



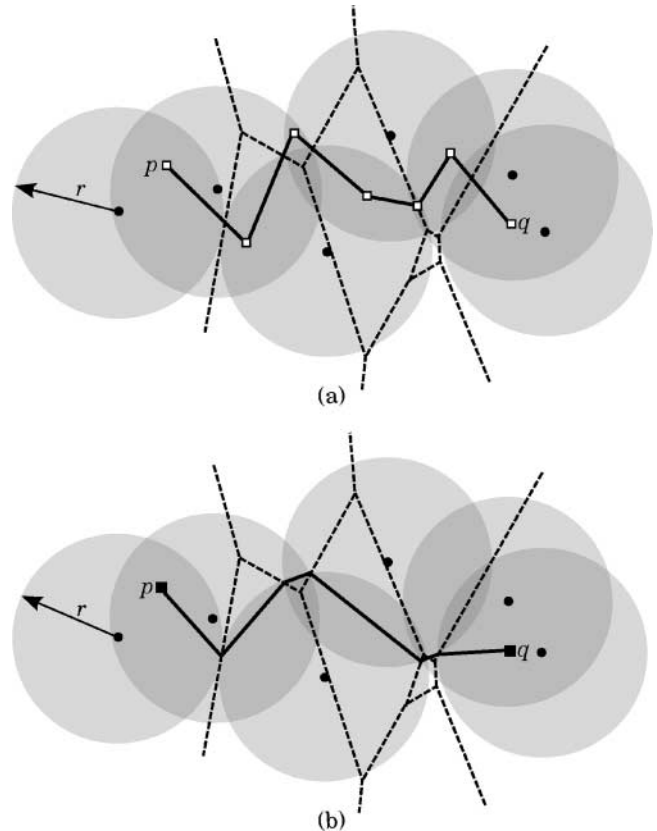
**FIGURE 7.** The sensing range is  $r = \text{MR}_S^2(P(p, q))$  and a 2-path connecting  $p$  to  $q$  is represented by a solid line. (a) Point  $b$  is a type I bottleneck-point. (b) Point  $b$  is a type II bottleneck-point.

*Proof.* According to the definition of type I bottleneck-points,  $b$  is interior to a set of  $k - 1$  discs centred at  $S$  of radius  $r$  but there are two other sensors,  $s_{j_1}$  and  $s_{j_2}$ , whose range is exactly the distance between them and  $b$ . Therefore, discs  $D(s_{j_1}, r)$  and  $D(s_{j_2}, r)$  meet at  $b$  and so  $b \in \text{PB}(s_{j_1}, s_{j_2})$ . In fact, this perpendicular bisector is an edge of  $\text{VD}_k(S)$  because  $b$  is exclusively covered by its closest  $k + 1$  sensors,  $\{s_{j_1}, s_{j_2}, \dots, s_{j_{k+1}}\}$ . Since two of them are at the same distance from  $b$ ,  $b$  lies on the edge that separates the  $k$ th order Voronoi region of  $\{s_{j_1}, s_{j_3}, \dots, s_{j_{k+1}}\}$  from the  $k$ th order Voronoi region of  $\{s_{j_2}, s_{j_3}, \dots, s_{j_{k+1}}\}$ . Moreover,  $b$  is the closest point that is simultaneously covered by  $s_{j_1}$  and  $s_{j_2}$ , which makes it the midpoint of  $\overline{s_{j_1}s_{j_2}}$ . That is,  $b$  is the intersection point between  $\text{PB}(s_{j_1}, s_{j_2})$  and  $\overline{s_{j_1}s_{j_2}}$ . The proof is similar for type II bottleneck-points. According to its definition,  $b$  is exclusively covered by its closest  $k + 1$  sensors, but three of them,  $\{s_{j_1}, s_{j_2}, s_{j_3}\}$ , are at the same distance from  $b$ . Consequently, the three discs centred at these three sensors of radius  $r$  meet simultaneously at  $b$ . That is,  $b$  is the intersection point between  $\text{PB}(s_{j_1}, s_{j_2})$ ,  $\text{PB}(s_{j_1}, s_{j_3})$  and  $\text{PB}(s_{j_2}, s_{j_3})$ . Hence  $b$  is a vertex of  $\text{VD}_k(S)$ .  $\square$

This proposition proposes to focus the search for candidates to bottleneck-points on edges and vertices of  $\text{VD}_k(S)$ —more precisely, on the vertices of  $\text{VD}_k(S)$  and the points on edges of  $\text{VD}_k(S)$  that are crossed by the line segment connecting the two sensors responsible for that edge. For every such candidate, calculate the minimum range needed to  $k$ -cover it. Then sort these ranges into a list in ascending order. Afterwards, determine  $s_p$  and  $s_q$ , the  $k$ th closest sensor of  $S$  to  $p$  and  $q$ , respectively, to calculate  $r_0 = \max\{d(s_p, p), d(s_q, q)\}$ . Range  $r_0$  is the starting point for the list of ranges since  $\text{MR}_S^k(P(p, q)) \geq r_0$ . Note that if  $r_0$  is the largest element of said list, then  $\text{MR}_S^k(P(p, q)) = r_0$ .

Otherwise, perform a binary search on the list of ranges to calculate  $\text{MR}_S^k(P(p, q))$ . In every step of the binary search, decide whether the corresponding range is large enough to allow a  $k$ -covered path connecting  $p$  to  $q$  using the decision algorithm previously described.

Once the binary search is over, the final range is  $\text{MR}_S^k(P(p, q))$  and it is left to compute a  $k$ -path between  $p$  and  $q$ . Consider graph  $G$  built by the decision algorithm for the range  $\text{MR}_S^k(P(p, q))$  (see Fig. 8a). A polygonal  $k$ -path between  $p$  and  $q$  can be constructed from the path on  $G$  that connects nodes  $p$  and  $q$ . For every edge of the path from node  $p$  to  $q$  on  $G$  find a point on its dual Voronoi edge such that it lies in the intersection of the two  $k$ -blobs associated with this edge. As  $k$ -blobs are convex sets, the straight line segment connecting two consecutive such points is entirely contained in one  $k$ -blob. If needed, this path is completed by adding the line segments connecting  $p$  to the first node and  $q$  to the last node of the  $k$ -path. This ensures that these line segments form a  $k$ -covered path from  $p$  to  $q$  (see Fig. 8b). As before, we can turn on only the subset of sensors that are effectively covering this  $k$ -path and shut down the rest. The description of this optimization algorithm concludes with the following theorem.



**FIGURE 8.** The sensing range is  $r = \text{MR}_S^k(P(p, q))$  and  $\text{VD}_2(S)$  is represented by a dashed line. (a) Path between nodes  $p$  and  $q$  on graph  $G$ . (b) A 2-path between points  $p$  and  $q$  on the plane.



**THEOREM 2.3.** *Let  $S$  be a set of  $n$  sensors,  $k \geq 2$  and  $p$  and  $q$  two points on the plane. Calculating  $\text{MR}_S^k(P(p, q))$  and a  $k$ -covered path between  $p$  and  $q$  can be determined in  $\mathcal{O}(k^2 n \log n)$  time and  $\mathcal{O}(k^2(n - k))$  space.*

*Proof.* Computing  $\text{VD}_k(S)$  takes  $\mathcal{O}(k^2 n \log n)$  time since  $S$  is a set of  $n$  sensors on the plane and the resulting diagram has  $\mathcal{O}(k(n - k))$  regions, edges and vertices [25]. Given the diagram's size, searching for possible candidates to bottleneck-points takes  $\mathcal{O}(k(n - k))$  time. According to Theorem 2.2, and having constructed  $\text{VD}_k(S)$ , building graph  $G$  and finding a  $k$ -path (if it exists) takes  $\mathcal{O}(k(n - k))$  time. Since each step of the binary search is performed in  $\mathcal{O}(k(n - k))$  time (the median of the list of ranges can be found in  $\mathcal{O}(k(n - k))$  time as well [28]), the overall time to run the binary search and find a  $k$ -path between  $p$  and  $q$  is  $\mathcal{O}(k(n - k) \log(k(n - k)))$ . Consequently, the whole algorithm runs in  $\mathcal{O}(k^2 n \log n)$  time. Constructing  $\text{VD}_k(S)$  takes  $\mathcal{O}(k^2(n - k))$  space.  $\square$

**THEOREM 2.4.** *Let  $S$  be a set of  $n$  sensors,  $k \geq 2$  and  $p$  and  $q$  two points on the plane. Calculating  $\text{MR}_S^k(P(p, q))$  and a  $k$ -covered path between  $p$  and  $q$  can be determined in  $\Omega(n \log n)$  time.*

*Proof.* This lower bound is achieved by means of a reduction to the Max-Gap problem: given a set of real numbers regarded as points on the  $x$ -axis, find the maximum distance (gap) between any two consecutive points once sorted. Lee and Wu [29] proved that the lower bound for this problem is  $\Omega(n \log n)$  time. Let  $S$  be a set of  $n$  real numbers for the Max-Gap problem. For each number  $s_i$ , consider the point  $(s_i, 0)$  on the  $x$ -axis. Copy every point  $k$  times in order to have  $kn$  points on the  $x$ -axis. Consider  $p = (\min\{s_0, \dots, s_{n-1}\}, 0)$  and  $q = (\max\{s_0, \dots, s_{n-1}\}, 0)$ —that is,  $p$  is the leftmost point on the  $x$ -axis and  $q$  is the rightmost. If  $r = \text{MR}_S^k(P(p, q))$ , then the maximum gap for  $S$  is  $2r$ . As a result, the previous algorithm solves the Max-Gap problem.  $\square$

This last theorem shows that the complexity of the algorithm to calculate  $\text{MR}_S^k(P(p, q))$  (presented in Theorem 2.3) is quite close to the lower bound. As already mentioned, the sensors' distribution might result in degenerate cases, which means that there can be more than one bottleneck-point for  $k$ -paths between two points. Notwithstanding the fact that bottleneck-points are not unique, they can still be found using the previously described technique and remain classified into two categories. The major difference is that  $k$ -covered paths between two points do not have to cross every single bottleneck-point, although they must cross at least one of them. Observe that even though the classification of bottleneck-points does not change in the presence of degenerate cases, type II bottleneck-points can correspond to Voronoi vertices with degree higher than three (in cases where more than three  $k$ -blobs meet at once).

To conclude, it is also possible to find the shortest  $k$ -covered path between two points on the plane. If the sensors' range is  $\text{MR}_S^k(P(p, q))$ , then it suffices to convert the connected component of the union of  $k$ -blobs into a polygon (that might

have holes) using a technique similar to the one used in [30]. Afterwards, the shortest path between  $p$  and  $q$  within such a polygon can be found by applying an algorithm by Kapoor *et al.* [31].

### 2.3. Minimizing the sensing range for $k$ -coverage: polygonal region

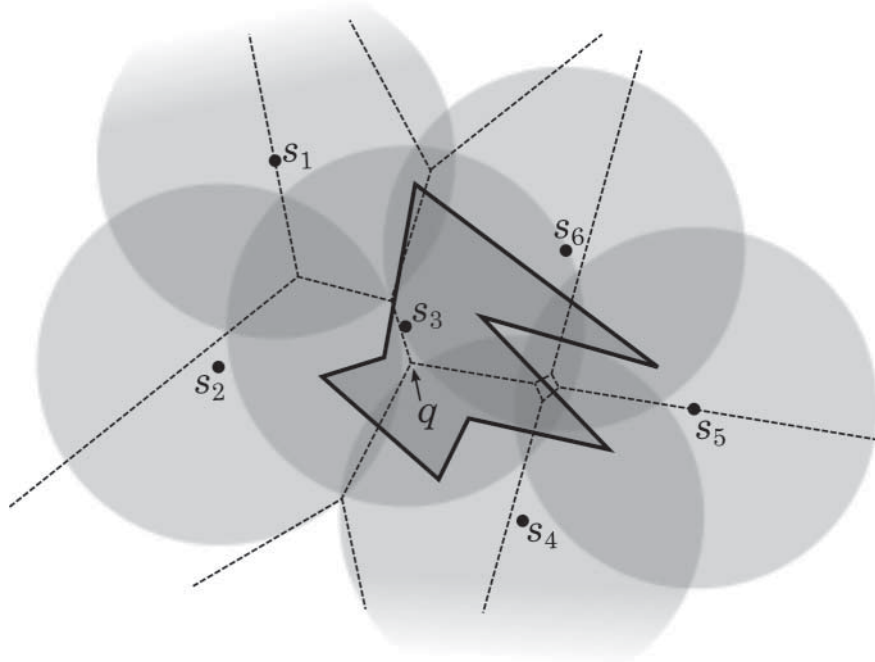
Let  $R$  be a polygonal region (with or without holes) formed by its interior and boundary. Given  $k \geq 2$ , a region is said to be  $k$ -covered by  $S$  if every point on such a region is  $k$ -covered. The algorithm introduced in the following calculates the minimum sensing range to  $k$ -cover  $R$ ,  $\text{MR}_S^k(R)$ . Such a range is the largest distance between a particular point on  $R$  and its  $k$ th closest sensor. To find this point, assume that the sensors' range is sufficiently large to fully  $k$ -cover  $R$ . Now suppose this range is slowly reduced until there is one point  $q$  on  $R$  that no longer is  $k$ -covered. As a result,  $R$  is not  $k$ -covered until point  $q$  is, which means that the minimum range to  $k$ -cover  $q$  is precisely the minimum range to fully  $k$ -cover  $R$ . The following proposition characterizes the location of such point on the polygonal region. Let  $B(R)$  denote the boundary of  $R$ .

**PROPOSITION 2.5.** *Let  $R$  be a polygonal region,  $k \geq 2$ ,  $q$  a point on  $R$  and  $S$  a set of sensors. If  $\text{MR}_S^k(R) = d(q, s_i)$ , for some sensor  $s_i \in S$ , then  $q$  is a vertex of  $R$  or  $q$  is an intersection point between  $B(R)$  and  $\text{VD}_k(S)$  or  $q$  is a vertex of  $\text{VD}_k(S)$  that lies inside  $R$ .*

*Proof.* Let  $q$  be a point on  $R$  be such that  $\text{MR}_S^k(R) = d(q, s_i)$  for some sensor  $s_i \in S$ . Given an edge  $e$  of  $R$ , it was seen in Section 2.1 that  $\text{MR}_S^k(e)$  is calculated using the intersection points between  $e$  and  $\text{VD}_k(S)$  plus the endpoints of  $e$ . Consequently, if  $q \in B(R)$  then  $q$  must be a vertex of  $R$  or an intersection point between  $B(R)$  and  $\text{VD}_k(S)$ . If  $q$  is inside  $R$ , then the situation gets trickier since it does not depend on the shape of  $R$  but on the way  $k$ -blobs interact with each other. As the sensing range increases,  $k$ -blobs expand accordingly and fill the interior of  $R$ . The last interior point to be 2-covered must be a point lying in the last 'hole' (meaning a region of  $R$  not yet  $k$ -covered). These holes are filled when three  $k$ -blobs intersect at a time, since not only the intersection point has to be  $k$ -covered but also its neighbourhood (see Fig. 9). According to Proposition 2.4,  $q$  is a type II bottleneck-point and therefore a vertex of  $\text{VD}_k(S)$ .  $\square$

According to this proposition, there are several candidates on  $R$  for the point that defines  $\text{MR}_S^k(R)$ . Since every point on  $R$  has to be at least  $k$ -covered,  $\text{MR}_S^k(R)$  can be calculated as the minimum range to  $k$ -cover every such candidate. Therefore, every vertex of  $R$  has to be studied, as well as intersection points between  $B(R)$  and  $\text{VD}_k(S)$  and vertices of  $\text{VD}_k(S)$  that fall inside  $R$ . The following theorem is a direct consequence of the method employed to calculate  $\text{MR}_S^k(R)$ .





**FIGURE 9.** Region  $R$  is shown in dark trace and  $\text{VD}_2(S)$  is represented by a dashed line. Point  $q$  is the vertex of  $\text{VD}_2(S)$  defined by  $s_2$ ,  $s_3$ ,  $s_4$  and  $s_6$ ,  $\text{MR}_S^2(R) = \text{MR}_S^2(q)$ .

**THEOREM 2.5.** *Given a set  $S$  of  $n$  sensors,  $k \geq 2$  and a polygonal region  $R$  with  $m$  vertices,  $\text{MR}_S^k(R)$  can be calculated in  $\mathcal{O}(mk(n-k) + k^2n \log n)$  time and  $\mathcal{O}(mk(n-k))$  space.*

*Proof.* Computing  $\text{VD}_k(S)$  takes  $\mathcal{O}(k^2n \log n)$  time since  $S$  is a set of  $n$  sensors on the plane and the resulting diagram has  $\mathcal{O}(k(n-k))$  regions, edges and vertices [25]. The cardinality of set  $I = B(R) \cap \text{VD}_k(S)$  is at most  $mk(n-k)$ . Adding all the vertices of  $\text{VD}_k(S)$  that lie inside  $R$  to set  $I$  takes  $\mathcal{O}(mk(n-k))$  time. The region's vertices can be added to  $I$  in  $\mathcal{O}(m)$  time. For each intersection point  $q \in I$ ,  $\text{MR}_S^k(q)$  is calculated in constant time using  $\text{VD}_k(S)$  since it is given by the distance between  $q$  and its  $k$ th closest sensor. The largest of these distances is  $\text{MR}_S^k(R)$ . Overall, the time complexity of this last procedure is  $\mathcal{O}(mk(n-k))$ . Regarding the space complexity, there is the need to store at most  $mk(n-k)$  intersection points while  $\text{VD}_k(S)$  can be stored in  $\mathcal{O}(k^2(n-k))$  space [25].  $\square$

**COROLLARY 2.1.** *If  $R$  is a convex polygonal region with  $m$  vertices, then  $\text{MR}_S^k(R)$  can be calculated in  $\mathcal{O}(k^2n \log n)$  time and  $\mathcal{O}(m + k^2(n-k))$  space.*

Observe that if  $\text{VD}_k(S)$  was previously constructed,  $\text{MR}_S^k(R)$  can be calculated in  $\mathcal{O}(m + k(n-k))$  time for a convex polygonal region  $R$ . The work of Vu and Zheng [24] was previously mentioned as it addressed the problem of minimizing the sensors' range while ensuring the  $k$ -coverage of a point or a polygonal region. In order to work with sensors whose exact location is unknown, they introduced the

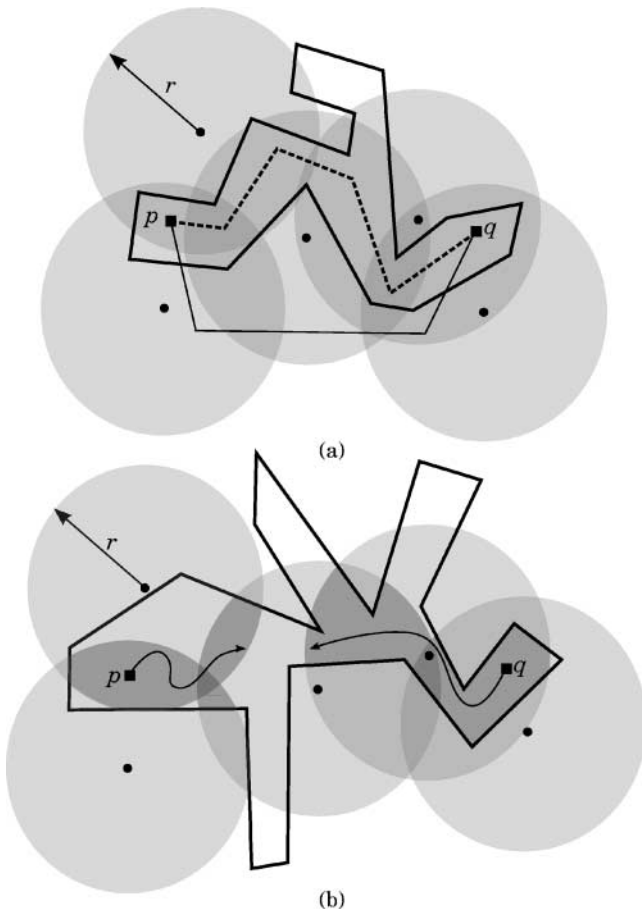
order- $k$  maximum-Voronoi diagram. This diagram is in fact the additively weighted Voronoi diagram of order  $k$  [32]. They calculated the minimum sensing range to  $k$ -cover a polygonal region with  $m$  vertices in  $\mathcal{O}(k^2(n \log n + n[r_{\max}/d_{\min}]) + mNNQ(n, k) + mkn)$  time, where  $d_{\min}$  is the minimum distance between two sensors with uncertain locations,  $r_{\max}$  is the maximum radius of the uncertainty discs and  $NNQ(n, k)$  is the cost of a nearest neighbour query. The time complexity presented in their paper is missing the factor  $\mathcal{O}(mkn)$ , which is the number of intersection points between the additively weighted Voronoi diagram of order  $k$  and the polygonal region. Although both are valid solutions to the same problem, our result is less complex than that presented by Vu and Zheng [24].

#### 2.4. Minimizing the sensing range for $k$ -coverage: path on a polygonal region

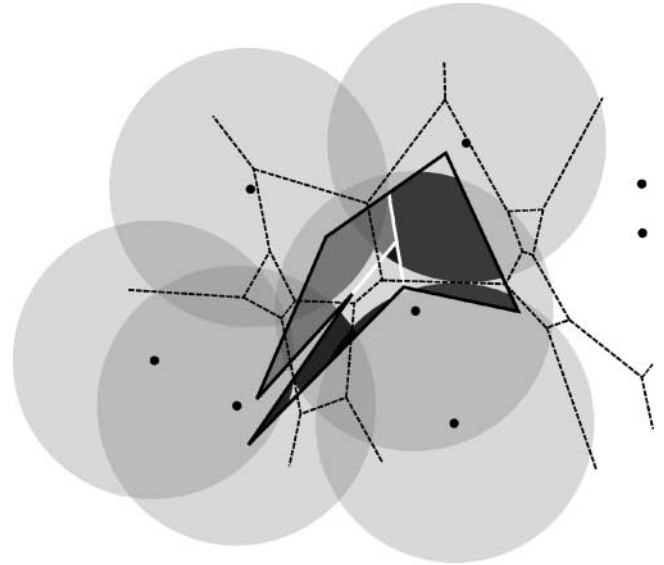
The objective of this final section is to calculate the minimum sensors' range such that there is a  $k$ -covered path within  $R$  connecting two given points  $p$  and  $q$ ,  $k \geq 2$ . Note that it is always assumed that  $p$  and  $q$  are points inside  $R$ . This problem is more restrictive than the previous ones since the space where this path lies is enclosed by  $R$ . The dashed path in Fig. 10a is a 2-path between  $p$  and  $q$ ; not only it has the desired coverage as it exists within  $R$ , which makes it the interesting type of path for this section. In the following, whenever  $k$ -path is mentioned, it is understood as a  $k$ -path within  $R$ . Similarly to what happened in Sections 2.1 and 2.2, the solution to this problem is achieved by

employing the associated decision problem, which is discussed shortly.

Given a set  $S$  of  $n$  sensors with range  $r \in \mathbb{R}^+$  and two points  $p$  and  $q$  on  $R$ , the first problem to tackle is to decide if there is a  $k$ -path connecting  $p$  and  $q$ ,  $k \geq 2$ . As previously mentioned, a  $k$ -path from  $p$  to  $q$  exists if and only if  $p$  and  $q$  lie in the same connected component of the union of  $k$ -blobs enclosed by  $R$  (see Fig. 10b). If it exists, such  $k$ -path only crosses the regions of  $R$  that are  $k$ -covered. With this in mind, let  $A$  be the arrangement of the union of  $k$ -blobs enclosed by  $R$  and intersected by  $\text{VD}_k(S)$ . Observe that if  $R$  is not convex, then there can be more than one face of  $A$  per Voronoi region. For example, there are two non-connected faces of  $A$  defined by the spikes of  $R$  on the lower leftmost Voronoi region in Fig. 11. Consequently, and according to Corollary 2.1, it is easier to work with convex regions since the number of intersection points decreases sharply. Moreover, there is only one face of  $A$  per Voronoi region. Therefore, the first step to solve this problem



**FIGURE 10.** Region  $R$  is shown in a dark trace. (a) The dashed path is a 2-path within  $R$  connecting  $p$  to  $q$ , whilst the solid black is not. (b) It is not possible to find a 2-path within  $R$  between  $p$  and  $q$  because they lie in different connected components of the union of 2-blobs.

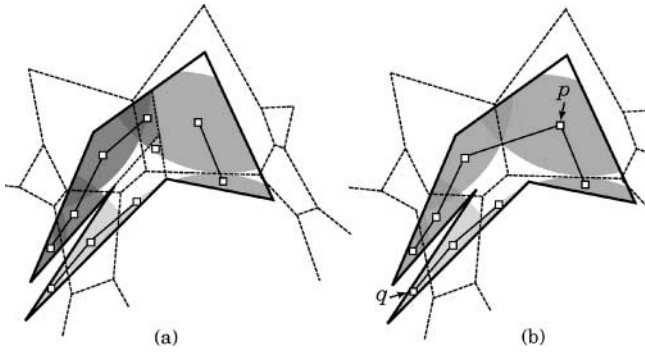


**FIGURE 11.** Polygonal region divided into three convex pieces and the respective arrangements are shown in three different shades of grey.

is to divide  $R$  into convex pieces. This is easily obtained using Steiner points: for each reflex vertex  $v_r \in R$  extend a ray from  $v_r$ , which bisects the internal angle of  $R$  at  $v_r$ , until it reaches  $B(R)$  or a previous ray. It can be shown that if  $R$  has  $j$  reflex vertices, then this set of rays divides  $R$  into  $j + 1$  convex pieces. There are some studies on the optimization of the final number of convex pieces, either using Steiner points [33] or diagonals [34]. Notwithstanding these results, and since it does not worsen the final complexity, the partition technique chosen in the following does not optimize the resulting number of convex pieces. The following procedure constructs an intersection graph  $G_i$  of the arrangement  $A_i$ , which is in fact arrangement  $A$  restricted to convex piece  $R_i \subseteq R$  (see Fig. 11). For each region of  $\text{VD}_k(S)$ , find the  $k$ -blob formed by the  $k$  sensors defining such a region. Then intersect this  $k$ -blob with  $\text{VD}_k(S)$  and  $R_i$ . If the resulting region is not empty, then it is a face of arrangement  $A_i$ . To construct  $G_i$ , add a node to each face of  $A_i$  and connect the nodes whose corresponding faces have a non-empty intersection. The whole arrangement  $A$ , and therefore  $G$ , is constructed by repeating this method to every convex piece  $R_i$  of  $R$ . The temporal complexity to construct the restricted arrangement and corresponding intersection graph is given in the following result.

**THEOREM 2.6.** *Let  $S$  be a set of  $n$  sensors with range  $r$ ,  $k \geq 2$  and  $R_i$  a convex region with  $m_i$  vertices. If  $\text{VD}_k(S)$  is preprocessed, then both arrangement  $A_i$  and graph  $G_i$  can be constructed in  $\mathcal{O}(k(n - k)m_i)$  time.*

*Proof.* Each  $k$ -blob is intersected at most  $2k$  times by the associated Voronoi region of  $\text{VD}_k(S)$  and the resulting region is still convex. Consequently, the intersection between that



**FIGURE 12.** (a) Each subgraph corresponds to a different convex piece. The subgraph on the leftmost region is disconnected. (b) There is not a 2-path between  $p$  and  $q$  because they lie in different connected components of the graph.

resulting region and  $R_i$  also is convex and can be found in  $\mathcal{O}(m_i + n_i + 2k)$  time,  $n_i$  being the complexity of a  $k$ th order Voronoi region. As the Voronoi diagram has amortized size  $\mathcal{O}(k(n-k))$  [25], arrangement  $A_i$  can be found in  $\mathcal{O}(k(n-k)m_i)$  time. Graph  $G_i$ , the intersection graph of  $A_i$ , has at most  $k(n-k)$  nodes since it may have one node per Voronoi region that intersects  $R_i$  (see Fig. 12a). Two nodes of  $G_i$  are connected if their corresponding faces of  $A_i$  intersect. The vertices of each face of  $A_i$  can be used to check if it intersects other faces of  $A_i$ . It suffices that each vertex is identified as an apex of a  $k$ -blob, a point on  $B(R)$  or a point on a  $k$ th order Voronoi edge. The latter is the important type of vertex for this procedure. Suppose face  $a_j \in A_i$  has a vertex  $v$  that is a point on a Voronoi edge, which means it separates two Voronoi regions and the two corresponding  $k$ -blobs intersect each other. Therefore, to construct  $G_i$  it suffices to connect the nodes corresponding to the faces of  $A_i$  that share at least one vertex on the same  $k$ th order Voronoi edge. Overall, this arrangement and corresponding intersection graph can be constructed in  $\mathcal{O}(k(n-k)m_i)$  time.  $\square$

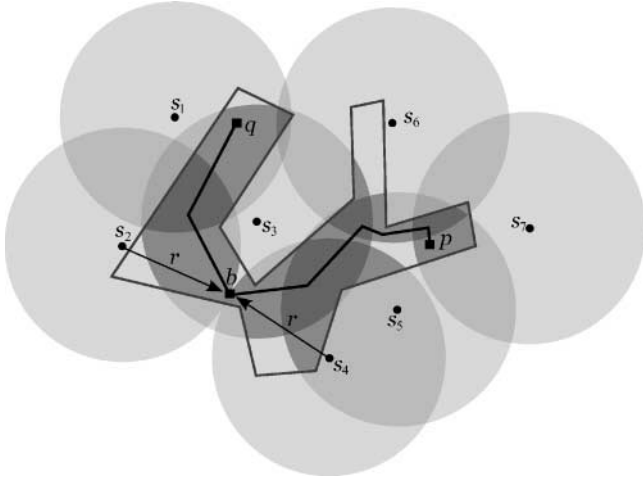
Note that  $G_i$  has as many edges as the  $k$ th order Voronoi diagram and that it may not be connected (see the subgraph on the leftmost region in Fig. 12a). The strategy to decide if there is a  $k$ -path between  $p$  and  $q$  is based on the construction of an intersection graph  $G$  of the whole arrangement  $A$ , which is in fact the union of the subgraphs constructed for each convex piece of  $R$ . It is not hard to see that a node of  $G_i$  is merged with another node of  $G_j$  if the faces of  $A$  they correspond to intersect each other. Once  $G$  is built, point  $p$  is assigned to node  $n_p$  of  $G$  whose corresponding face of  $A$  contains  $p$ . In a similar way, point  $q$  is assigned to node  $n_q$ . Consequently, the existence of a  $k$ -path connecting  $p$  to  $q$  depends on  $n_p$  and  $n_q$  being on the same connected component of  $G$  (see Fig. 12b).

**THEOREM 2.7.** *Let  $S$  be a set of  $n$  sensors with sensing range  $r$ ,  $k \geq 2$  and  $R$  a polygonal region divided into convex pieces*

*by adding  $j$  rays. Let  $M$  be the largest complexity of the convex pieces. If  $\text{VD}_k(S)$  is preprocessed, then deciding if a  $k$ -covered path between points  $p$  and  $q$  exists takes  $\mathcal{O}(jk(n-k)M)$  time.*

*Proof.* Dividing region  $R$  with  $m$  vertices into convex pieces by adding a set  $L$  of  $j$  rays takes  $\mathcal{O}(m + j^2 \log(m/j))$  time [33]. Consequently,  $R$  is divided into  $j + 1$  convex pieces. Supposing each convex piece  $R_i$  has  $m_i$  vertices, assume  $M = \max\{m_1, \dots, m_{j+1}\}$ . According to Theorem 2.6, constructing  $G_i$  for each convex piece takes  $\mathcal{O}(k(n-k)m_i)$  time; so constructing a first version of  $G$  takes  $\mathcal{O}(jk(n-k)M)$  time. For the same reason,  $A$  is also constructed in  $\mathcal{O}(jk(n-k)M)$  time. Analysing the vertices of every face of  $A$  on a convex piece takes  $\mathcal{O}(k(n-k)M)$  time. Consequently, finding the sets of faces that have an edge on the same ray of  $L$  takes  $\mathcal{O}(jk(n-k)M)$  time. Finding the  $k$ -blob that contains a face of  $A$  can be done in constant time. Every ray of  $L$  can intersect  $k(n-k)$  Voronoi regions and consequently  $k(n-k)$  faces of  $A$ . Therefore, each set of faces has cardinality at most  $k(n-k)$ . The vertices of the rays of  $L$  can have degree 3 at most [33]; so the number of nodes of  $G$  to be merged at a time is at most 3, which can be done in constant time. Consequently, the construction of  $G$  is concluded in  $\mathcal{O}(jk(n-k))$  time. Locating  $p$  and  $q$  on  $A$  can be done in  $\mathcal{O}(k(n-k)M)$  time. Traversing  $G$  to decide whether there is a  $k$ -path between points  $p$  and  $q$  takes  $\mathcal{O}(jk(n-k))$  time. Overall, this decision problem can be solved in  $\mathcal{O}(jk(n-k)M)$  time.  $\square$

The strategy to solve the general problem proposed in this section applies the decision algorithm described earlier. Said problem is to calculate the minimum sensing range that ensures the existence of a  $k$ -path on  $R$  between  $p$  and  $q$ . Such a range is denoted by  $\text{MR}_{S,R}^k(P(p, q))$ . As previously mentioned, there is always a solution if  $|S| \geq k$ . Following the idea presented in Section 2.2, locate point  $b$  that is the first intersection between the connected component of  $k$ -blobs containing  $p$  and the one containing  $q$ . As mentioned before, this point can either be on  $B(R)$  or inside  $R$ . The first case was discussed in Proposition 2.5 and so  $b$  can either be a vertex of  $R$  or an intersection point between  $\text{VD}_k(S)$  and  $B(R)$ . The second case is more complicated since now there is no need to completely cover  $R$ . In fact, all that is needed is that  $p$  and  $q$  lie in the same connected component of  $k$ -blobs. Therefore,  $b$  has to be a bottleneck-point for  $k$ -paths between  $p$  and  $q$ —that is,  $b_S^k(p, q)$ . Without loss of generality, degenerate input data are not considered and so bottleneck-points are regarded as unique for every pair of points. According to Proposition 2.4, candidates to bottleneck-points are found on the edges of  $\text{VD}_k(S)$  that are intersected by the line segment joining the two sensors defining such edge and vertices of  $\text{VD}_k(S)$  that lie inside  $R$ . Finally,  $\text{MR}_{S,R}^k(P(p, q))$  is calculated, using a binary search on a list of sorted ranges. Each range of the list is the minimum sensing range needed to  $k$ -cover a candidate and each candidate is either



**FIGURE 13.** The 2-path between  $p$  and  $q$  on  $R$  only exists if the sensors' range is at least  $\text{MR}_{S,R}^2(P(p, q)) = \text{MR}_S^2(b) = r$ . The 2-covered regions of  $R$  are shown in darker grey.

$p, q$ , a vertex of  $R$ , a point of  $\{B(R) \cap \text{VD}_k(S)\}$  or a bottleneck-point. In every step of the binary search, the corresponding range is evaluated to decide whether it is large enough to allow a  $k$ -path between  $p$  and  $q$  within  $R$ . The decision algorithm presented earlier will be used to this end. If the range is indeed large enough, then the search proceeds on the lowest half of the ranges. Otherwise, the search continues on the highest half of the ranges.

In Fig. 13, there is an example of a 2-path between  $p$  and  $q$  within a polygonal region. Any 2-path between these points only exists if the sensors' range is at least  $\text{MR}_{S,R}^2(P(p, q))$ . This range is calculated as the minimum range to 2-cover point  $b = b_S^2(p, q)$ , which is given by  $d(s_2, b) = d(s_4, b)$ . The following theorem concludes this subject.

**THEOREM 2.8.** *Let  $S$  be a set of  $n$  sensors,  $k \geq 2$  and  $R$  a polygonal region with  $m$  vertices divided into convex pieces by adding  $j$  rays. Let  $M$  be the largest complexity of the convex pieces. Given two points  $p$  and  $q$  on  $R$ ,  $\text{MR}_{S,R}^k(P(p, q))$  can be calculated in  $\mathcal{O}(jMk(n-k) \log mn)$  time.*

*Proof.* Computing  $\text{VD}_k(S)$  takes  $\mathcal{O}(k^2n \log n)$  time [25]. Set  $I = \{p, q\} \cup \{B(R) \cap \text{VD}_k(S)\}$  has cardinality  $mk(n-k)$  since  $\text{VD}_k(S)$  has size  $\mathcal{O}(k(n-k))$  and  $R$  has  $m$  vertices. Adding  $m$  vertices to  $I$  plus the  $k(n-k)$  candidates to bottleneck-points for  $k$ -paths between  $p$  and  $q$  takes  $\mathcal{O}(m+k(n-k))$  time. Therefore, set  $I$  can be found in  $\mathcal{O}(mk(n-k))$  total time. This is also the temporal complexity of calculating  $\text{MR}_S^k(i)$  for every  $i \in I$ . Supposing each convex piece has  $m_i$  vertices, assume that  $M = \max\{m_1, \dots, m_{j+1}\}$ . Then according to Theorem 2.7, each step of the binary search runs in  $\mathcal{O}(jk(n-k)M)$  time. Consequently, the binary search runs in  $\mathcal{O}(jk(n-k)M \log mk(n-k)) = \mathcal{O}(jk(n-k)M \log mn)$  time, which is the final complexity.  $\square$

### 3. CLOSING REMARKS AND FUTURE RESEARCH

This paper was dedicated to multiple coverage and intended as a means to present new results involving maximal-exposure paths on different types of regions. Regarding multiple coverage of paths, there is a paper by Mehta *et al.* [35] that was purposely omitted. This paper presented an algorithm to find minimal exposure paths regarding simple coverage and it claimed that such an algorithm is extendable to multiple coverage by applying higher-order Voronoi diagrams. We do not believe this is the case and Fang and Low [9] also pointed out why it does not work.

All four sections presented algorithms to calculate the minimum sensing range to allow the existence of a maximal-exposure path. By minimizing the sensors' range while providing the desired coverage, the network's lifespan is extended and this in turn lowers the associated costs. The most interesting aspect of these algorithms is that they also output the subset of sensors needed to  $k$ -cover such a path and so it is possible to turn off the sensors that are not needed. Although the sensors' range is optimized to ensure the existence of a  $k$ -path on  $R$ , it is never shown how to construct it in Section 2.4. Future research will obviously include the construction of a  $k$ -path within  $R$  and even the construction of the shortest  $k$ -path. It is not clear how to do this efficiently because the union of  $k$ -blobs restricted to  $R$  is not convex and presents an uncommon shape: some edges are arcs while others are line segments. The key question to reduce the remaining complexities is whether the construction of Voronoi diagrams can be avoided. That is, find another way to directly compute the point of the region or path that needs the largest sensing range in order to be  $k$ -covered. Such question is this paper's main unresolved problem. Furthermore, since these problems are based in real-life situations, it is reasonable to consider the generalization of these results to higher dimensions. For example, when using three dimensions, all these results could be directly applied to coverage problems.

### FUNDING

M.A. is co-supported by Projects MTM2008-05043 and HP2008-0060. A.L.B. is supported by FCT through CIDMA of University of Aveiro, and I.M. is supported by the grant SFRH/BPD/66572/2009.

### REFERENCES

- [1] Wang, X., Xing, G., Zhang, Y., Lu, C., Pless, R. and Gill, C. (2003) Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks. *Proc. 1st Int. Conf. Embedded Networked Sensor Systems*, Los Angeles, CA, USA, November 5–7, pp. 28–39. ACM, New York, NY, USA.
- [2] Zhang, H. and Hou, J. (2005) Maintaining sensing coverage and connectivity in large sensor networks. *Ad Hoc Sens. Wirel. Netw.*, **1**, 89–124.



- [3] Pesant, G. (1988) Galleries Require More Sleepy Watchmen:  $k$ -guarding Simple Polygons. In Toussaint, G. (ed.), *Snapshots of Computational and Discrete Geometry*, pp. 145–165. McGill University.
- [4] Belleville, P., Bose, P., Czyzowicz, J., Urrutia, J. and Zaks, J. (1994)  $k$ -guarding Polygons on the Plane. *Proc. 6th Canadian Conf. Computational Geometry*, Saskatchewan, Canada, August, pp. 381–386.
- [5] Aurenhammer, F. and Klein, R. (2000) Voronoi Diagrams. In Sack, J.-R. and Urrutia, J. (eds), *Handbook of Computational Geometry*, pp. 201–290. North-Holland, Amsterdam, The Netherlands.
- [6] Okabe, A., Boots, B., Sugihara, K. and Chiu, S.N. (2000) *Spatial Tessellations—Concepts and Applications of Voronoi Diagrams* (2nd edn), Series in Probability and Statistics. John Wiley & Sons, Inc., New York, NY, USA.
- [7] Anirvan, D., Arijit, B. and Indranil, S. (2006) Optimisation Problems Based on the Maximal Breach Path Measure for Wireless Sensor Network Coverage. In Madria, S., Claypool, K., Kannan, R., Uppuluri, P. and Gore, M. (eds), *Distributed Computing and Internet Technology*, Lecture Notes in Computer Science 4317, pp. 27–40. Springer, Berlin/Heidelberg.
- [8] Veltri, G., Huang, Q., Qu, G. and Potkonjak, M. (2003) Minimal and Maximal Exposure Path Algorithms for Wireless Embedded Sensor Networks. *Proc. 1st Int. Conf. Embedded Networked Sensor Systems*, Los Angeles, CA, USA, November 5–7, pp. 40–50. ACM, New York, NY, USA.
- [9] Fang, C. and Low, C.P. (2007) Redundant Coverage in Wireless Sensor Networks. *Proc. IEEE Int. Conf. Communications*, Glasgow, Scotland, June 24–28, pp. 3535–3540. IEEE.
- [10] Gomez, J., Campbell, A., Naghshineh, M. and Bisdikian, C. (2001) Conserving Transmission Power in Wireless Ad hoc Networks. *Proc. 9th Int. Conf. Network Protocols*, Mission Inn, Riverside, CA, USA, November 11–14, pp. 24–34. IEEE Computer Society, Washington, DC, USA.
- [11] Li, X.-Y., Wan, P.-J. and Frieder, O. (2003) Coverage in wireless ad hoc sensor networks. *IEEE Trans. Comput.*, **52**, 753–763.
- [12] Zhou, Z., Das, S. and Gupta, H. (2009) Variable radii connected sensor cover in sensor networks. *ACM Trans. Sens. Netw.*, **5**, 1–36.
- [13] Meguerdichian, S., Koushanfar, F., Potkonjak, M. and Srivastava, M. (2001) Coverage Problems in Wireless Ad-hoc Sensor Networks. *Proc. 20th Annual Joint Conf. IEEE Computer and Communications Societies*, Anchorage, AK, USA, April 22–26, pp. 1380–1387. IEEE.
- [14] Megerian, S., Koushanfar, F., Potkonjak, M. and Srivastava, M. (2005) Worst and best-case coverage in sensor networks. *IEEE Trans. Mob. Comput.*, **4**, 84–92.
- [15] Zhou, Z., Das, S. and Gupta, H. (2004) Connected  $k$ -Coverage Problem in Sensor Networks. In James, R. (ed.), *Proc. 13th Int. Conf. Computer Communications and Networks*, Chicago, IL, USA, October 11–13, pp. 373–378. ACM.
- [16] Efrat, A., Har-Peled, S. and Mitchell, J. (2005) Approximation Algorithms for Two Optimal Location Problems in Sensor Networks. *Proc. 2nd Int. Conf. Broadband Networks*, Boston, MA, USA, pp. 714–723. IEEE.
- [17] Kumar, S., Lai, T.H. and Balogh, J. (2004) On  $k$ -coverage in a Mostly Sleeping Sensor Network. *Proc. 10th Annual Int. Conf. Mobile Computing and Networking*, Philadelphia, PA, USA, pp. 144–158. ACM, New York, NY, USA.
- [18] Kumar, S., Lai, T.H., Posner, M.E. and Sinha, P. (2007) Optimal Sleep-Wakeup Algorithms for Barriers of Wireless Sensors. *Proc. 4th Int. Conf. Broadband Communications, Networks and Systems*, Raleigh, USA, September 10–14, pp. 327–336. IEEE.
- [19] Kumar, S., Lai, T.H., Posner, M.E. and Sinha, P. (2010) Maximizing the lifetime of a barrier of wireless sensors. *IEEE Trans. Mob. Comput.*, **9**, 1161–1172.
- [20] Huang, C.-F., Tseng, Y.-C. and Wu, H.-L. (2007) Distributed protocols for ensuring both coverage and connectivity of a wireless sensor network. *ACM Trans. Sens. Netw.*, **3**, 5.
- [21] Gupta, H., Zhou, Z., Das, S. and Gu, Q. (2006) Connected sensor cover: self-organization of sensor networks for efficient query execution. *Trans. Netw.*, **14**, 55–67.
- [22] Abellanas, M., Bajuelos, A. and Matos, I. (2009) 2-Covered paths by a set of antennas with minimum power transmission range. *Inf Process Lett*, **109**, 768–773.
- [23] Abellanas, M., Bajuelos, A. and Matos, I. (2009) Optimal 2-coverage of a polygonal region in a sensor network. *Algorithms*, **2**, 1137–1154.
- [24] Vu, K. and Zheng, R. (2011) Robust Coverage under Uncertainty in Wireless Sensor Networks. *Proc. 30th Annual Joint Conf. IEEE Computer and Communications Societies*, Shanghai, China, April 10–15, pp. 2015–2023. IEEE.
- [25] Lee, D. (1982) On  $k$ -nearest neighbor Voronoi diagrams in the plane. *IEEE Trans. Comput.*, **31**, 478–487.
- [26] Matsui, T. (1995) The minimum spanning tree problem on a planar graph. *Discrete Appl Math.*, **58**, 91–94.
- [27] Golomb, S. and Baumert, L. (1965) Backtrack programming. *J. ACM*, **12**, 516–524.
- [28] Blum, M., Floyd, R., Pratt, V., Rivest, R. and Tarjan, R. (1973) Time bounds for selection. *J. Comput. Syst. Sci.*, **7**, 448–461.
- [29] Lee, D. and Wu, Y. (1986) Geometric complexity of some location problem. *Algorithmica*, **1**, 193–211.
- [30] Matos, I. (2009) Limited range coverage problems. PhD Thesis, Universidade de Aveiro.
- [31] Kapoor, S., Maheshwari, S. and Mitchell, J. (1997) An efficient algorithm for euclidean shortest paths among polygonal obstacles in the plane. *Discrete Comput Geom.*, **18**, 377–383.
- [32] Rosenberger, H. (1991) Order- $k$  Voronoi diagrams of sites with additive weights in the plane. *Algorithmica*, **6**, 490–521.
- [33] Chazelle, B. (1980) Computational geometry and convexity. PhD Thesis, Yale University.
- [34] Keil, J. (1985) Decomposing a polygon into simpler components. *SIAM J. Comput.*, **14**, 799–817.
- [35] Mehta, D., Lopez, M. and Lin, L. (2003) Optimal Coverage Paths in Ad-hoc Sensor Networks. *Proc. IEEE Int. Conf. Communications*, Anchorage, Alaska, May 11–15, pp. 507–511. IEEE.