Locating a Communication Path in a Competitive Scenario

M. Abellanas* J.M. Díaz-Báñez† P. Pérez-Lantero † I. Ventura †

December 29, 2010

Abstract

Consider a set of receptors belonging to two competitive telecommunication firms, the blue firm and the red firm. The receptors are represented as points in the plane, b are blue and belong to the blue firm and r are red and belong to the red firm. The blue firm has an emitting device represented as a point that moves along a path sending information to blue receptors as follows: At any time, the device sends information to all blue receptors covered by the largest disk centered at it and containing no red receptor. In this scenario, we study two optimization problems. The first problem is to compute a path $\mathcal{P}$, such that the number of blue receptors served by a moving device is maximized. In particular, we give efficient algorithms when $\mathcal{P}$ is a straight line, an anchored half-line, and an axis-parallel double ray. As a second task, we study the problem of removing the minimum number of red receptors in such a way there exists a straight line path $\mathcal{P}$ so that if the device moves along $\mathcal{P}$ all blue receptors are served. We prove geometrical properties of an optimal straight line and propose efficient algorithms depending on the degrees of freedom of the line.

Keywords: Extensive Facility Location, Geometric Optimization, Algorithms, Competitive Location.

*Dep. Matemática Aplicada, Universidad Politécnica de Madrid, mabellanas@fi.upm.es, partially supported by grant MTM2008-05043.

†Dep. Matemática Aplicada II, Universidad de Sevilla, {dbanez,pplantero,iventura}@us.es, partially supported by grant MTM2009-08625. phone: 34-95-4482157; fax: 34-95-4486165.

1 Introduction

Suppose that we are in the following scenario. We have two competitive telecommunication firms, the blue firm and the red firm, each of them with a set of receptors installed in fixed sites in the plane. The blue firm needs to install a communication path with the aim of transmitting information to its receptors so that no receptor of the red firm is aware of such information. The communication process is performed by an emitting device which moves continuously along all the path. The device works every time and sends information to all the receptors located in the largest circular region centered at it that does not contain any receptor of the red firm.

In a more realistic model, an energy saving can be done by reducing the emission to a discrete number of points on the path. In fact, given the optimal path we can easily find a set of emitting points giving the maximum covering area. In Figure 1, the maximum covering area corresponding to a fixed straight line is showed. It is attained by emitting from the ending points of the path and the intersection points between the path and the edges of the Voronoi diagram generated by the red receptors.

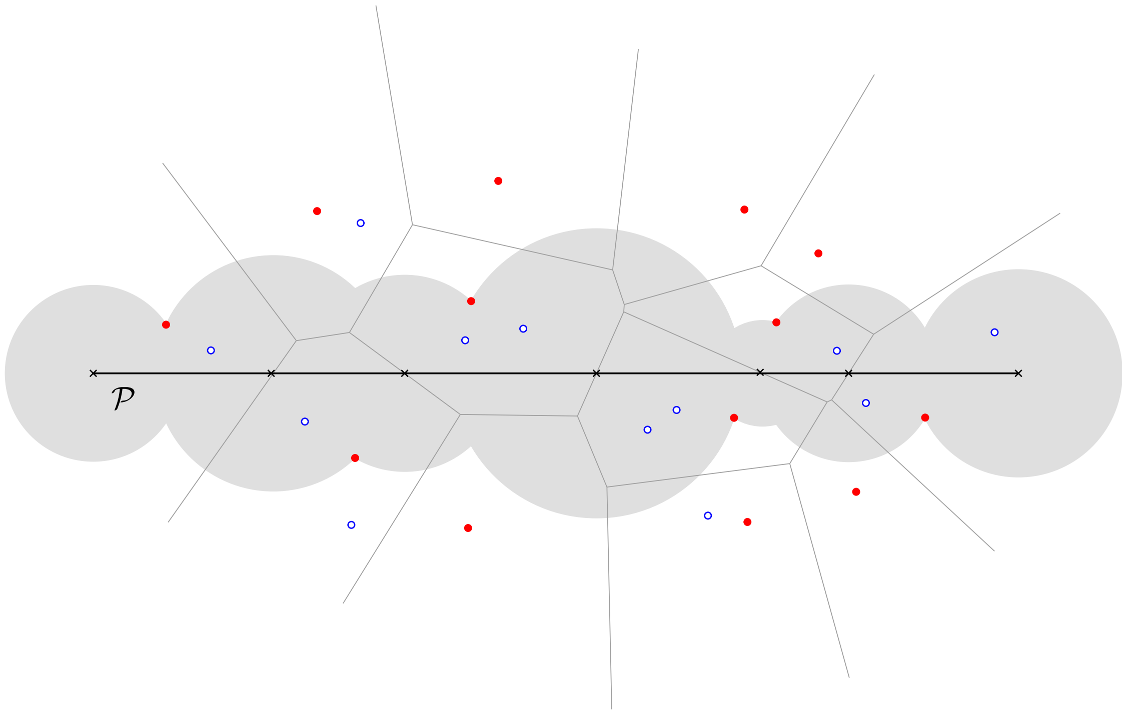

Figure 1: The widest communication corridor. Red points are represented as solid dots and blue points as tiny circles. The device only emits from a discrete point set and the widest communication area is covered. Observe that three blue points are not served.

Moreover, a communication path always exists passing through all receptors of the blue firm and through no receptor of the red one. With that path, all blue receptors receive the information in some moment. Nevertheless, in general this path could be too expensive and the transmission path is thus constrained to have some specific shape. In this case, it might be not possible to guarantee that all blue receptors receive the information. Under this observation, we arrive to the facility location problem of locating the communication path, subject to some specific shape, in such a way that the number of blue receptors receiving the information is maximized.

In order to formulate the problems in a geometric setting, we represent the receptors of the blue firm by a set B of b blue points in the plane, and the receptors of the red firm by a set R of r red points. Let $n = r + b$. For each point p of the plane let $D(p)$ be the largest open disk centered at p that does not contain any element of R. Given a path $\mathcal{P}$, let $Z(\mathcal{P})$ denote $\bigcup_{p \in \mathcal{P}} D(p)$. We suppose that the blue firm needs to install the communication path, thus the maximum covering location problem can be formulated as follows:

The Communication Path Problem (CP-problem): *Given R and B, find a path $\mathcal{P}$ so that $|B \cap Z(\mathcal{P})|$ is maximized, subject to $\mathcal{P}$ has some fixed shape.*

Once the communication path has been installed, either some receptors of the blue firm never receive the information or the red firm install new receptors in order to block the total communication of the blue firm. In these situations, a natural problem to be considered by the blue firm is to deactivate the minimum number of receptors belonging to the red firm in such a way that all its own receptors receive the information. Based on this observation, we also consider the next problem:

The Full Communication Fixed-Path Problem (FCFP-problem): *Given R, B, and a fixed path $\mathcal{P}$, remove the minimum number of elements of R in such a way $B \subset Z(\mathcal{P})$.*

A more general version of the FCFP-problem is to find the minimum number of red receptors to be deactivated so that a communication path subject to some fixed shape and serving all blue receptors exists. This results into the next optimization problem:

The Full Communication Path Problem (FCP-problem): *Given R and B, remove the minimum number of elements of R so that there exists a path $\mathcal{P}$ subject to some fixed shape, satisfying $B \subset Z(\mathcal{P})$.*

Overview: In this paper we introduce new problems in the continuous facility location area [21, 12] where the new facility is a structure with a pre-fixed shape in which a device moves covering some kind of clients in a competitive scenario. Let us show the environment where the problem lies.

Given a set of fixed points (demand points), the goal of classical location problems is to find one or several points to optimize one or several possible constrained objective functions. Objective functions usually depend on the interactions among demand points and new facilities. When new facilities cannot be represented as points but some kind of dimensional sets, *extensive facility location* problems arise. These problems consist in choosing an element in a class of (geometric) sets, representing the candidate facilities, that best fits the set of demand points according to specified criteria. In particular, location of straight lines, line segments, hyper-planes, spheres, and some types of polygonal curves have been studied for these problems. An overview about the more general case of locating any kind of dimensional facility has been given in [11]. See also [24, 6] and the references therein. The novelty of our problems lies on the interaction between the facility (the path) and the demand points (blue receptors): The covering is made by a point (device) moving along the path. The problems have been introduced in a competitive scenario and the maximization of the profit of a company (the blue firm) is addressed. See [14, 22] for a comprehensive survey on competitive location problems.

On the other hand, our problems are related to red-blue separation problems in Computational Geometry. The CP-problem introduces a new concept of red-blue separation in the plane. We can state that a set B of blue points is *Communication-Path-separable*, *CP-separable*, from a set R of red points if there exists a path $\mathcal{P}$ (having a pre-established shape) such that $B \subset Z(\mathcal{P})$. Observe that if we consider that $\mathcal{P}$ is a point instead of a path in the CP-problem, then the problem is equivalent to finding a point p (i.e. $\mathcal{P} = p$) such that $|B \cap D(p)|$ is maximum. This problem was addressed in [3]. Red-blue separation problems have been widely studied in Machine Learning [7, 17] and Computational Geometry [23].

Additionally, the FCP-problem can be seen as a red-blue separation problem with *outliers* by means of removing the minimum number of red points (outliers) so that the blue points are CP-separable from the remaining red points. Studies about red-blue separation with outliers can be found in [18, 10].

Our results: In this paper we efficiently solve the CP-problem when the path $\mathcal{P}$ is restricted to a straight line, a half-line with the origin anchored at a fixed point, or an axis-parallel double ray. In the first case we give an $O(r^2 \log r + r^2 \log^2 b + rb)$-time algorithm, and algorithms running in $O(r \log b + r \log r + rb)$ and $O(r^2 b^2)$ time for the second and third cases, respectively. The FCFP-problem is solved when the path $\mathcal{P}$ is a straight line giving a quadratic-time algorithm. We solve the problem by reducing our problem to find a convex region containing the maximum number of red points and no blue point. This problem is named *largest monochromatic island problem* in [5]. Finally, we also study the FCP-problem when the path is restricted to a straight line. When the path is an oriented line we give an algorithm running in $O(r^2(r+b)^2)$ time, and when it is not oriented an $O(r^3 b(r+b)^2)$-time algorithm is presented.

We should say that it was decided to express the time complexity of some algorithms in terms of r and b. It allows us to know better the time complexity when the points are balanced or imbalanced according to the color. In applications, r is usually much smaller than b and one would like to have an algorithm whose performance depends on both r and b. For example, the algorithm of Subsection 2.2 that solves the CP-problem when the path is a line, has time complexity $O(r^2 \log r + r^2 \log^2 b + rb)$ which is $O(n^2 \log^2 n)$ if $r = b = n/2$. But, it is essentially linear if r is a constant.

The paper is organized as follows. The CP-problem, the FCFP-problem, and the FCP-problem are studied respectively in sections 2, 3, and 4. In section 5, conclusions and further research are stated.

2 The CP-problem

In this section we solve the CP-problem when the path is a straight line, a half-line with the origin anchored, or an axis-parallel double ray. Before that, we give some useful properties in the next subsection.

2.1 Properties

Let d denote the Euclidean distance. Given a set of points X in the plane and a point $x \in X$, let $V(X)$ be the Voronoi diagram of X, and $VR(x, X)$ be the Voronoi region of x in $V(X)$. For each blue point p, let V_p denote $VR(p, R \cup \{p\})$.

Lemma 2.1 *Given R, B, and a path $\mathcal{P}$, a blue point p is contained in $Z(\mathcal{P})$ if and only if $\mathcal{P}$ intersects with the interior of V_p.*

Proof. Notice from the Voronoi diagram properties that V_p consists of the points of the plane that are closer from p than from any point in R. Thus the lemma follows. □

It is clear to see that in all cases the CP-problem is equivalent to finding a path $\mathcal{P}$ intersecting the interiors of the maximum number of the convex regions V_p, $p \in B$. We assume w.l.o.g. that each V_p is a closed convex polygon. We follow now with properties about the intersection of those regions.

Lemma 2.2 *Let p and q be two blue points. The boundaries of V_p and V_q intersect at most twice, and furthermore they intersect at points of the bisector of p and q.*

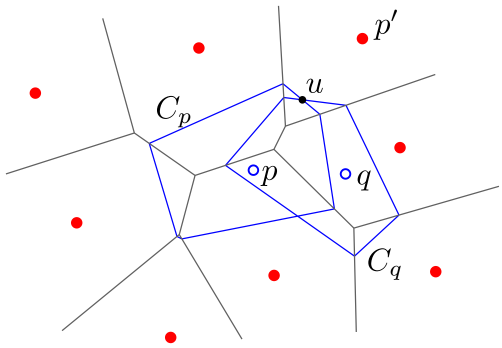

Figure 2: Given two blue points p and q, V_p and V_q intersect at most twice.

Proof. Let u be one point in the intersection of the boundaries of V_p and V_q, as shown in Figure 2. Let $p' \in R$ be the point such that $u \in VR(p', R)$. We have that $d(u, p) = d(u, p')$ and $d(u, q) = d(u, p')$. Therefore, $d(u, p) = d(u, q)$ and u lies on the bisector of p and q. Thus all points in the intersection of the boundaries of V_p and V_q belong to the bisector of p and q. Since the Voronoi regions are convex regions, V_p and V_q intersect at most twice. □

Corollary 2.1 *Given two blue points p and q, V_p and V_q have non-empty intersection if and only if p and q are neighbours in $V(R \cup \{p, q\})$.*

We show now how to compute the set $\{V_p \mid p \in B\}$ in $O(r \log r + rb)$ time. In a first step we compute $V(R)$ in $O(r \log r)$ time by using [20]. In a second step, for each blue point p, we first insert p in $V(R)$ computing V_p in $O(r)$ time, and after that remove p from $V(R)$. The second step can be done in $O(rb)$ time [16].

Since for each blue point p the number of vertices of V_p is at most r [20], the total space used by the set $\{V_p \mid p \in B\}$ is $O(rb)$.

2.2 $\mathcal{P}$ is a straight line

Given a set S of geometric objects in the plane, a *stabber* is a straight line which intersects every element of S. The stabber problem is to determine whether or not a straight line exists that intersects every object in S. In [4], the stabber problem of n polygons was studied and solved in $O(nN + N \log N)$ time, where N is the total number of vertices of the convex hulls of these n polygons. In [8], the stabber problem of n disjoint convex polygons is solved in $O(N + n \log n)$ time. Furthermore, a stabber of n axis-parallel rectangles can be found in $O(n)$ time by using a linear programming approach [13].

Given a set of convex polygons in the plane, the *Maximum Stabbing Problem* is to find a straight line intersecting the maximum number of the polygons. A randomized approximation algorithm to this maximum stabbing problem can be derived from [1]. We can see that the maximum stabbing problem for a set of convex polygons is 3SUM-hard [15]. Indeed, it is enough to consider a point a special case of a convex polygon and use a reduction from the 3POINTS-IN-LINE-problem [15].

The CP-problem when the path is a straight line is quite similar to the maximum stabbing problem over a set of convex polygons. The difference is that in this case the line must intersect the interior of the polygons. Due to the hardness of the maximum stabbing problem, it is unlikely that a faster than quadratic-time algorithm for our problem exists.

In what follows we will show our algorithm to solve the CP-problem. Given a polygon V_p, let I_p be the open orthogonal projection (i.e. an open segment or interval) of V_p in the x-axis. The algorithm is based on the following simple remark: Given two blue points p and q, we have a vertical line intersecting the interior of both V_p and V_q if and only if the intersection of I_p and I_q is not empty.

Hence, we only need to compute a point of maximum depth in the arrangement induced by the the set of intervals $\{I_p \mid p \in B\}$, in order to find a vertical line ℓ that intersects the interiors of the maximum number of elements in the set $\{V_p \mid p \in B\}$. Notice that, if we rotate the x-axis from $\theta = 0$ to $\theta = \pi$, one of the lines ℓ will be a solution of our problem. Then the idea of the algorithm is to rotate the coordinate axes from $\theta = 0$ to $\theta = \pi$ maintaining the vertical line ℓ.

During the rotation we consider the moments (events) in which two endpoints of different intervals in $\{I_p \mid p \in B\}$ coincide. Notice that an event occurs when the x-axis is perpendicular to a common tangent to two elements of $\{V_p \mid p \in B\}$.

It is easy to see that there are $O(b^2)$ events because two convex polygons have at most four common tangents. The common tangents for disjoint convex polygons can be computed in logarithmic time [19]. Moreover, in the case where we have two convex polygons P and Q, the boundaries of which intersect twice at most, the authors show that the common tangents can be calculated in $O(\log(|P| + |Q|) \log k)$ time, where $k = \min\{|P \cap Q|, |P \cup Q|\}$ and $|X|$ denotes the number of vertices of X.

By Lemma 2.2, we can obtain all common tangents in $O(b^2 \log^2 r)$ time in the worst case, and order them by slope in $O(b^2 \log b)$ time. We present now a dynamic data structure that allows to efficiently maintain the depth in $\{I_p \mid p \in B\}$ when θ goes from 0 to π.

Let $\theta = 0$ and let $x_1 < x_2 < \ldots < x_{2b}$ be the list of the endpoints of the intervals I_p, $p \in B$. We consider the open intervals $(x_1, x_2), (x_2, x_3), \ldots, (x_{2b-1}, x_{2b})$, and in linear time we assign a weight to each of them. The weight of (x_i, x_{i+1}), $i = 1, \ldots, 2b - 1$, is equal to the number of elements in $\{I_p \mid p \in B\}$ that contains (x_i, x_{i+1}).

We store the previous intervals with their weights in the leaves of a balanced binary tree T so that the leftmost leaf of T corresponds to the leftmost interval, and so on. Every internal node v stores the interval of maximum weight (and its weight) among the intervals stored at the leaves of the subtree rooted at v. Hence the root of T stores the depth of $\{I_p \mid p \in B\}$.

At each event, we can update T in $O(\log b)$ time as follows. Suppose that x_i and x_{i+1} become equal and (x_i, x_{i+1}) has weight w. We remove (x_i, x_{i+1}) from T and insert (x_{i+1}, x_i). One can clearly see that the weight of (x_{i+1}, x_i) is equal to w, $w - 2$, or $w + 2$.

Then we obtain the following result,

Theorem 2.1 *The CP-problem can be solved in $O(b^2 \log b + b^2 \log^2 r + rb)$ time and $O(rb + b^2)$ space when the path is a straight line.*

2.3 $\mathcal{P}$ is a half-line with the origin anchored

Here we study the CP-problem when the path is a half-line emanating from a fixed point u. Notice that if we do not consider that the origin of the path is fixed, then the problem is equivalent to that in which the path is a straight line (see Section 2.2).

Give a point u, the problem consists in finding a half-line h emanating from u that intersects the maximum number of the interiors of V_p, $p \in B$. We take a circle $\mathcal{C}$ enclosing all elements of $\{V_p \mid p \in B\}$, and for each blue point p we consider the arc α_p equal to the projection of the interior of V_p on $\mathcal{C}$ with respect to u. Notice that if u is contained in the interior of V_p, then α_p is equal to all $\mathcal{C}$. Given a blue point p, α_p can be found in $O(\log r)$ time. Take note that h intersects the interior of V_p, $p \in B$, if and only if it intersects α_p. Then h is determined by a point in $\mathcal{C}$ covered by the maximum number of elements in $\{\alpha_p \mid p \in B\}$. Such a point can be found in $O(b \log b)$ time by using the radial order of the endpoints of the arcs $\{\alpha_p \mid p \in B\}$. Therefore, the following result is obtained:

Theorem 2.2 *The CP-problem can be solved in $O(b \log b + rb)$ time and $O(rb)$ space when the path is a half-line with the origin anchored at a fixed point.*

2.4 $\mathcal{P}$ is an axis-parallel double ray

An axis-parallel double ray $\mathcal{P}$ consists of two half-lines, one horizontal and the other one vertical, which emanate from a common point u. We say that u is the corner of $\mathcal{P}$ and consider w.l.o.g. that u is the leftmost (resp. topmost) point of the horizontal (resp. vertical) half-line.

We note that $\mathcal{P}$ intersects the interior of a convex polygon V_p, $p \in B$, if and only if the corner of $\mathcal{P}$ belongs to the interior of the open region H_p depicted in Figure 3 a) in color gray. Then, our problem is equivalent to finding a point u of the plane covered by the maximum number of regions H_p, $p \in B$.

In order to find such a point, we decompose each region H_p into a set T_p of trapezoids as shown in Figure 3 b) and reduce the problem to finding a point u of maximum depth in the arrangement $\mathcal{A}(G)$ induced by the set $G := \bigcup_{p \in B} T_p$. Notice that the elements of G are either rectangles or rectangular trapezoids as shown Figure 3 c). Furthermore, G contains $O(rb)$ trapezoids in the worst case. We say that $\mathcal{A}(G)$ is the *trapezoidal arrangement* of $R \cup B$.

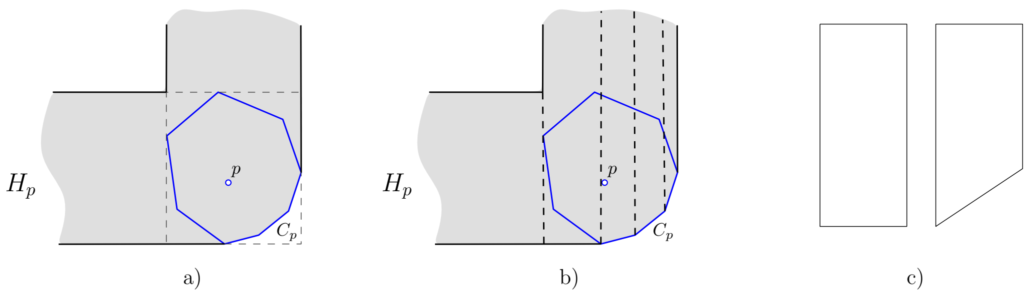

Figure 3: a) The region H_p corresponding to the convex polygon V_p. b) The decomposition of H_p in trapezoids. c) The two possible shapes of the elements of G.

A point of maximum depth in $\mathcal{A}(G)$ can be found in quadratic time, that is, in $O(N^2)$ time where N is number of elements of G, by generating all $\mathcal{A}(G)$. This approach can be used when there is not a good distribution of red and blue points. In that case we could expect $\mathcal{A}(G)$ to have quadratic complexity, and thus the approach sounds reasonable. Otherwise, if red and blue points are well distributed as often happens in practice, the number of vertices in $\mathcal{A}(G)$, denoted by k, is less than quadratic. We can then apply the next approach whose time complexity depends on k and is $O(N \log N + k \log N)$.

We sweep all elements of G from top to bottom with a horizontal line ℓ. During the sweep we maintain the set I of the segments corresponding to the intersection of ℓ with each element of G. The set I is stored in a dynamic balanced tree T able to report the depth of I in $O(\log N)$ time. See the tree used in [2] in the sequential algorithm to compute the depth of an arrangement of axis-parallel rectangles.

The events in which I changes are when the line ℓ meets: 1) a top side of an element of G, 2) a bottommost point (a vertex or side) of an element of G, and 3) a point in the set of intersections points of the set of vertical sides and non-axis-parallel bottom sides, of all elements of G.

In Case 1) a new segment is inserted in T in $O(\log N)$ time. In case 2) a segment is removed from T in $O(\log N)$ time. Finally, in Case 3), two coinciding endpoints of different segments are exchanged in T, also in $O(\log N)$ time. The events in Case 3) can be detected by using a standard algorithm of segment intersection [9].

In total there are $2N + O(k)$ events, and since each of them can be processed in $O(\log N)$ time, the following result is thus obtained:

Theorem 2.3 *When the path is an axis-parallel double ray, the CP-problem can be solved either in $O(N^2)$ or in $O(N \log N + k \log N)$ time, where N is the number of trapezoids, and k the number of vertices, in the trapezoidal arrangement of $R \cup B$. The value of k does not depend on N and both are $O(rb)$ in the worst case.*

3 The FCFP-problem

We start with a different interpretation of the FCFP-problem. Given a path $\mathcal{P}$, let us consider a closed interval $I \subseteq \mathbb{R}$ and a continuous bijective function $m : I \to \mathcal{P}$. Given a point $p \in R \cup B$ and a value $x \in I$, let $d_p(x) = d(p, m(x))$. The set $\Gamma := \{d_p(x) \mid p \in R \cup B\}$ is a collection of $r + b$ continuous totally-defined functions in the interval I. We say that a graph of a function $d_p(x)$ is red (resp. blue) if and only if p is a red (resp. blue) point.

Let $\mathcal{A}(\Gamma)$ be the arrangement induced by the graphs of the elements of Γ. We say that a blue graph $f \in \mathcal{A}(\Gamma)$ *is visible* if there is a point $u \in f$ strictly below of all red graphs, or equally if f intersects the lower envelope of the red graphs (see Figure 4). It is straightforward to see that a blue graph $d_p(x)$ is visible if and only if $p \in Z(\mathcal{P})$. Therefore the FCFP-problem can be reformulated as follows:

Remove the minimum number of red graphs of Γ so that all blue graphs are visible in $\mathcal{A}(\Gamma)$.

By using the interpretation above, we now solve the FCFP-problem when the path $\mathcal{P}$ is a straight line ℓ. We assume w.l.o.g. that ℓ is the x-axis. In this case, I is $\mathbb{R}$, $m(x) = (x, 0)$, and the elements of Γ are convex functions fulfilling that all of them have the expression $\sqrt{x^2 + ax + c}$ and the graphs of every pair intersect in at most one point. Then Γ is a set of pseudo-lines and can be mapped to a set of lines Γ' as follows: $\sqrt{x^2 + ax + c} \mapsto ax + c$.

Based on the above discussion, we have reduced our problem to the following one:

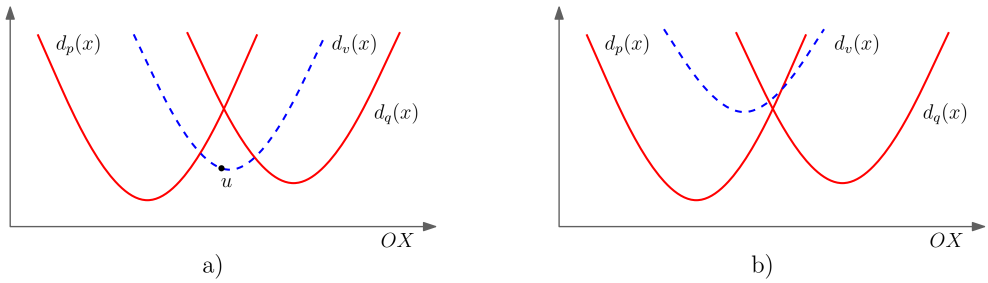

Figure 4: a) The blue graph, drawn with a dashed line, is visible. b) The blue graph is not visible.

Given the set of straight lines Γ', each of them colored red or blue, remove the minimum number of red lines from Γ' so that all blue lines intersect the lower envelope of the remaining red lines.

To solve this new problem we first dualize the lines of Γ' to a bicolored set of points S by using the standard transformation $ax + c \mapsto (a, c)$ [9]. Notice that all blue lines in Γ' intersect the lower envelope of the red lines if and only if in S no blue point is contained in the lower convex hull of the red points. The *lower convex hull* of a point set X, denoted by $LCH(X)$, is the convex hull of $X \cup \{(0, +\infty)\}$ (see Figure 5 a)).

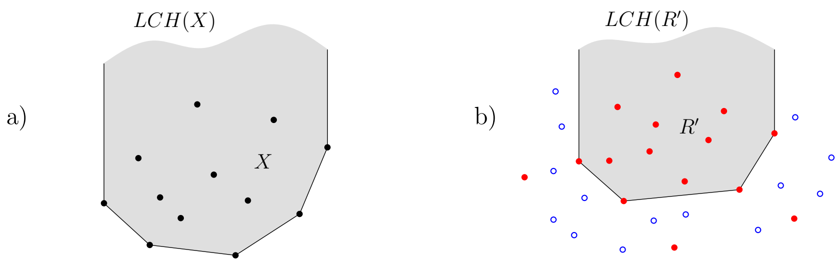

Figure 5: a) A point set X whose lower convex hull $LCH(X)$ is the shaded region. b) The solution of the problem is the subset R' of red points such that $LCH(R')$ contains no blue point.

Therefore, we finally solve the following problem denoted as the Maximum Lower Convex Hull-problem (Refer to Figure 5 b)):

MLCH-problem: Given a bicolored set of points $S = R \cup B$ find a maximum-cardinality subset of R so that its lower convex hull contains no blue points.

Our algorithm is a modification of the algorithm in [5] that can find for $S = R \cup B$ a convex polygon $\mathcal{Q}_S$, with vertices on R, covering the maximum number of red points while avoiding blue points. Its idea is, by using a quadratic time and space preprocessing, to compute in $O(n^2)$ time for each red point p, the polygon $\mathcal{Q}_S$ having p as top vertex. Notice that if we consider a red point p at $(0, +\infty)$, then the polygon $\mathcal{Q}_{S'}$ for the point set $S' = S \cup \{p\}$, subject to p is its top vertex, is a solution of our problem. Therefore, the algorithm in [5] can be used with minor modifications to solve our problem in $O(n^2)$ time, and the following result is thus obtained:

Theorem 3.1 *The FCFP-problem can be solved in $O(n^2)$ time and space when the path is a straight line.*

4 The FCP-problem

Here we study the FCP-problem when the path $\mathcal{P}$ is a straight line. First, in Subsection 4.1, we solve the problem of finding an optimal horizontal line (oriented case). After that, in Subsection 4.2, we address the case of finding an optimal line no matter its direction (unoriented case).

4.1 Oriented case

In what follows, when we refer to a solution to the problem we mean the remaining set of red points after deleting the minimum number of them. We first consider a partition of the y-axis into a finite number of intervals in such a way that the solution for all the horizontal lines that cross a given interval does not change. As we are going to show, the number of such intervals is $O(r^2)$. By using the algorithm of Section 3 that solves the FCFP-problem when the path is a fixed straight line, one can obtain the solution for the oriented case in $O(r^2(r+b)^2)$ time.

We start by introducing some notation. For every pair of red points p and q let $B^+_{p,q}$ (resp. $B^-_{p,q}$) be the set of blue points lying above (resp. below) the segment connecting p and q and whose x-coordinate is in between the x-coordinates of p and q. Let $c^+_{p,q}$ (resp. $c^-_{p,q}$) denote the center of the disk that covers $B^+_{p,q}$ (resp. $B^-_{p,q}$), the boundary of which contains p, q, and an element of $B^+_{p,q}$ (resp. $B^-_{p,q}$). Refer to Figure 6. We will call $c^+_{p,q}$ and $c^-_{p,q}$ the critical points defined by the pair of red points p and q.

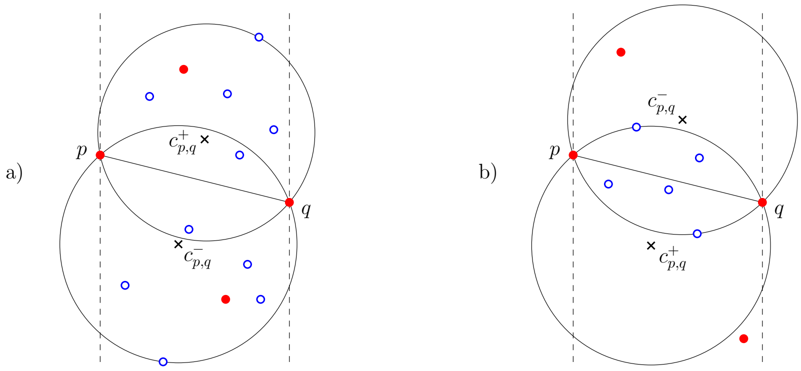

Figure 6: The critical points $c^+_{p,q}$ and $c^-_{p,q}$ defined by the red points p and q. The solution for every line passing in between $c^+_{p,q}$ and $c^-_{p,q}$ never contain both p and q in case a). In case b) it may contain both points.

Lemma 4.1 *For all pairs of red points p and q, the solution for all horizontal lines above (resp. below) the critical point $c^-_{p,q}$ (resp. $c^+_{p,q}$) cannot contain both red points p and q.*

Proof. It is a consequence of the following observation: Every disk centered in such lines and whose boundary contains an element of $B^-_{p,q}$ (resp. $B^+_{p,q}$), covers at least one of the red points p or q. Refer to Figure 6. □

Corollary 4.1 *Two red points p and q never belong to the same solution if the y-coordinate of $c^-_{p,q}$ is less than the y-coordinate of $c^+_{p,q}$.*

Lemma 4.2 *Let C be the set of all critical points $c^+_{p,q}$ and $c^-_{p,q}$. If the points of C are sorted from bottom to top with respect to their y-coordinates, then for every horizontal line in between two consecutive elements in C the solution of the FCFP-problem is the same.*

Proof. Suppose we have a solution $\mathcal{S} \subseteq R$ for a fixed horizontal line. We say that the dual point set is the set of points (or dual points) in which the MLCH-problem is solved (refer to Section 3). If we continuously move the line upwards in a parallel way, the dual points moves vertically. Then $\mathcal{S}$ will be valid until the configuration of the dual point set changes in such a way that we can have a better solution to the MLCH-problem or until we cannot maintain $\mathcal{S}$. A change in the dual point set occurs if and only if three of its points get aligned. This is because only in these cases the solution to the MLCH-problem can change.

Note that three dual points are aligned if and only if the corresponding primal points lie on a circle centered on the path line. In fact, let $y = y_0$ be the path line and let us consider the circle centered at the point (x_0, y_0) with radius r. A point in that circle has coordinates $(x_0 + r\cos\alpha, y_0 + r\sin\alpha)$, and its dual point is $(-2(x_0 + r\cos\alpha), (x_0 + r\cos\alpha)^2 + (y_0 + r\sin\alpha - y_0)^2)$ which lies in the line $y = -x_0 x + r^2 - x_0^2$.

Therefore, changes in the dual configuration will occur when the moving line in the primal space reaches the circumcenter of three points. Nevertheless, only when such circumcenters belong to the set C, the corresponding changes in the dual configuration are relevant for the changes in the solution. In fact, in order to have a change in the configuration of the dual points, allowing to obtain a better solution or giving rise to a worse one, it is necessary that a blue dual point crosses a segment connecting two red points. In the first case allowing at least one of the red points to become part of the better solution and, in the second case forcing at least one of them to exit the solution. This is equivalent to saying that the moving line is crossing a critical point in the primal space. But, for every pair of red points p and q, the only relevant critical points are $c^+_{p,q}$ and $c^-_{p,q}$, because any other critical point corresponding to a blue point in $B^-_{p,q}$ (resp. $B^+_{p,q}$) lies above (resp. below) $c^-_{p,q}$ (resp. $c^+_{p,q}$) and, because of Corollary 4.1, both red points p and q do not belong to the solution when the line crosses such critical points. □

Theorem 4.1 *The FCP-problem can be solved in $O(r^2(r + b)^2)$ time if the path is an oriented straight line.*

Proof. We assume w.l.o.g. that the path is a horizontal line. For each pair of red points p and q, the critical points $c^+_{p,q}$ and $c^-_{p,q}$ can be obtained in $O(b)$ time, thus all of them in $O(r^2 b)$ time. Since there are $O(r^2)$ critical points, sorting them by y-coordinate can be done in $O(r^2 \log r)$ time. Now for each two consecutive critical points in the y-order, the $O((r + b)^2)$-time algorithm for the FCFP-problem can be applied. Hence, we spend $O(r^2(r + b)^2)$ overall time for the oriented case. □

4.2 Unoriented case

Taking into account that critical points do not depend on the orientation of the straight line, if we have a solution for the oriented case and rotate the line, changes in the solution will occur only when some blue points exit the vertical strip determined by two red points. Therefore, for solving the unoriented case, it suffices to solve the oriented one as many times as different orderings can be when projecting orthogonally the red and blue points on a line. Since there could be $O(rb)$ of such orderings, the solution for the unoriented case can be obtained in $O(r^3 b(r + b)^2)$ time. Thus we have:

Theorem 4.2 *The FCP-problem can be solved in $O(r^3 b(r + b)^2)$ time if the path is a straight line.*

5 Concluding remarks

In this paper, a new framework has been introduced in which a collection of new facility location problems arises. Given a set of points on the plane belonging to two different classes, the general problem is to find a path so that when a point moves along the path, it covers the maximum number of points of a class without covering any other point of the other class. Based on geometrical properties of an optimal path, efficient algorithms were proposed when the path is restricted to a straight line, an anchored half-line or an orthogonal double ray. Even more interesting are the problems that arise when we are interested in covering all the points of a class (Full Communication Path). In this case, we have to remove the minimum number of points of the other class so that a full communication is possible.

Beyond the possible application (in Robotic, Aeronautics, Wireless Networks, etc.) of the problems studied in this paper and the solutions given, we have introduced a family of geometric problems on bicolored point sets. The CP-problem is a generic problem that can be particularized for many types of paths, different than those studied here. Depending on the path shape, different *communications corridors* can be generated. In Figure 1 a linear communication corridor is generated for a fixed linear path. Thus it gives rise to a family of interesting open problems from the geometrical point of view.

The FCFP-problem also offers a family of open problems by considering other types of paths. For this problem, we established a framework on red and blue curves and reduced the problem to solve a specific case of the *largest monochromatic island problem* [5]. We leave as open problem to study this problem for other paths that induce a set of curves such that every pair of them intersect at most a constant number of times. For example, when the path is a circle, the curves intersect at most twice.

Finally, it would be worthy to study the problems introduced here by using different metrics.

References

[1] P.K. Agarwal, D.Z. Chen, S.K. Ganjugunte, E. Misiolek, M. Sharir, K. Tang (2008) Stabbing Convex Polygons with a Segment or a Polygon. In Proc. of the 16th annual European Symposium on Algorithms.

[2] H. Alt, L. Scharf (2010) Computing the depth of an arrangement of axis-aligned rectangles in parallel, In Proc of the 26th European Workshop on Computational Geometry (EuroCG), 33–36.

[3] B. Aronov, S. Har-Peled (2008) On approximating the depth and related problems. SIAM J. Comput. Vol. 38, pp. 899–921.

[4] D. Avis, M. Doskas (1990) Algorithms for high dimensional stabbing problem. Theoretical Foundations of Computer Graphics and CAD, 199–210.

[5] C. Bautista-Santiago, J.M. Díaz-Báñez, D. Lara, P. Pérez-Lantero, J. Urrutia, I. Ventura (2009) Computing Maximal Islands, In Proc. 25th European Workshop on Computational Geometry, 333–336.

[6] R. Blanquero, E. Carrizosa, P. Hansen (2009) Locating objects in the plane using Global Optimization techniques, Math. Oper. Res. 34, 837–858.

[7] C. Burges (1998) A tutorial on support vector machines for pattern recognition. Data Min. Knowl. Discov. 2, 121–167.

[8] F. Y.L. Chin, F. L. Wang (2002) Efficient algorithms for transversal of disjoint convex polygons. Information Processing Letters 83, 141–144.

[9] M. de Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf (2000) Computational Geometry: algorithms and applications, Springer-Verlag Berlin Heidelberg New York.

[10] C. Cortés, J.M. Díaz-Báñez, P. Pérez-Lantero, C. Seara, J. Urrutia, I. Ventura (2009) Bichromatic separability with two boxes: A general approach, Journal of Algorithms 64(2-3), 79–88.

[11] J.M. Díaz-Báñez, J.A. Mesa, A. Schobel (2004) Continuous location of dimensional structures, Eur. J. Oper. Res., 152, 22–44.

[12] Z. Drezner and H. W. Hamacher (Eds.) (2002) Facility Location: Applications and Theory, Springer, 2002.

[13] H. Edelsbrunner (1985) Finding transversals for sets of simple geometric figures. Theoretical Computer Science 35, 55–69.

[14] H. A. Eiselt, G. Laporte, and J. F. Thisse (1993) Competitive location models: A framework and bibliography, Transportation Science vol. 27, pp. 44-54.

[15] A. Gajentaan, M.H. Overmars (1995) On a class of $O(n^2)$ problems in computational geometry. Computational Geometry Theory and Applications 5, 165–185.

[16] I.G. Gowda, D.G. Kirkpatrick, D.T. Lee, A. Naamad (1983) Dynamic Voronoi Diagrams. IEEE Transactions on Information Theory, 29(5).

[17] J. Han and M. Kamber (2006) Data mining: concepts and techniques, Morgan Kaufmann Publishers, USA.

[18] S. Har-Peled, V. Koltun, (2005) Separability with Outliers. Algorithms and Computation. Lecture Notes in Computer Science, 3827, 28–39.

[19] D. Kirkpatrick, J. Snoeyink (2006) Computing common tangents without a separating line. Lecture Notes in Computer Science, 955/1995, 183–193.

[20] A. Okabe, B. Boots, K. Sugihara, (1992) Spatial Tessellations: Concepts and Applications of Voronoi Diagrams, Prentice Hall.

[21] F. Plastria (1995) Continuous location problems: research, results and questions. In: Drezner, Z. (ed.) Facility location: a survey of applications and methods, pp. 85127. Springer, Berlin.

[22] F. Plastria (2001) Static competitive location: An overview of optimisation approaches, Eur. J. Operations Research, vol. 129, pp. 461–470.

[23] C. Seara. *On geometric separability.* Ph. Thesis. Advisor: F. Hurtado. Universitat Politécnica de Catalunya, Barcelona, 2002.

[24] A. Schobel (1998) Locating Least Distant Lines in the Plane, European Journal of Operational Research, 106(1), 152–159.