



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE  
MÉXICO

**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

**“PROBLEMAS DE PROXIMIDAD SOBRE OBJETOS  
GEOMÉTRICOS EN EL PLANO”**

**T E S I S**

**QUE PARA OBTENER EL GRADO DE:**

**MAESTRO EN CIENCIAS  
(COMPUTACIÓN)**

**P R E S E N T A:**

**LUIS FELIPE BARBA FLORES**

**DIRECTOR DE TESIS:**

**JORGE URRUTIA GALICIA**

**México, D.F.**

**2011.**



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



# Índice general

<b>Prefacio</b>	<b>7</b>
<b>Introducción</b>	<b>11</b>
<b>1. <math>P</math>-círculos y el diagrama de Voronoi del punto más lejano</b>	<b>17</b>
1.1. Nuestro problema . . . . .	17
1.2. El diagrama de Voronoi del punto más lejano . . . . .	18
1.2.1. ¿Qué es el Diagrama de Voronoi del punto más lejano?	19
1.2.2. Calculando el diagrama de Voronoi . . . . .	24
1.3. $P$ -círculos de radio mínimo . . . . .	26
1.3.1. Calculando el $P$ -círculo de radio mínimo . . . . .	29
<b>2. <math>P</math>-círculo de radio mínimo con centro en una línea de consulta</b>	<b>31</b>
2.1. Nuestro problema . . . . .	31
2.2. El $P$ -círculo de radio mínimo con centro en $L$ . . . . .	32
2.2.1. Calculando el $P$ -círculo de radio mínimo con centro sobre una línea recta . . . . .	35
2.3. Encontrando a $C_L$ mediante consultas dinámicas . . . . .	38
2.3.1. Esquema del algoritmo . . . . .	38
2.3.2. $c_L$ y las trayectorias sobre el árbol $\mathcal{V}(P)$ . . . . .	40
2.3.3. Estructura de datos sobre $\mathcal{V}(P)$ . . . . .	44
2.3.4. El algoritmo . . . . .	45
<b>3. Separabilidad por círculos entre polígonos convexos</b>	<b>47</b>
3.1. Nuestro problema . . . . .	47
3.2. Círculos separadores . . . . .	49
3.3. El algoritmo . . . . .	53
3.3.1. Precizando la posición de $c$ . . . . .	56
3.3.2. Algoritmo para encontrar $q_c$ . . . . .	58



# Índice de figuras

1.	El diagrama de Voronoi del punto más cercano del conjunto de gasolineras de la ciudad. . . . .	8
2.	El diagrama de Voronoi del punto más lejano del conjunto clientes $\{c_1, \dots, c_6\}$ . La región del cliente $c_i$ está representada por el polígono $R(c_i)$ , para toda $1 \leq i \leq 6$ . . . . .	9
3.	Un conjunto de antenas retransmisoras de televisión ofreciendo su servicio a un conjunto de clientes. . . . .	12
1.1.	La antena inalámbrica de transmisión radiofónica dando cobertura a un conjunto de casas representadas por este conjunto de puntos. .	18
1.2.	a) El círculo $C_P$ y su centro $c_P$ . b) El diagrama de Voronoi del punto más lejano de $\{p_0, \dots, p_6\}$ . . . . .	19
1.3.	El diagrama de Voronoi de un conjunto de dos puntos. . . . .	20
1.4.	La prueba de la Proposición 1.2.1. . . . .	21
1.5.	El diagrama de Voronoi visto como una gráfica $\mathcal{V}(P)$ , al agregar hojas suficientemente lejos sobre cada arista no acotada. . . . .	22
1.6.	a) El proceso de inserción de $p_i$ . b) El diagrama de Voronoi después de la inserción de $p_i$ . . . . .	25
1.7.	a) Cualquier círculo que pase por un único punto de $P$ , contiene otro círculo que pasa por al menos 2 puntos de $P$ . b) El círculo $C_P$ pasando por tres puntos de $P$ . . . . .	27
1.8.	La prueba de la Proposición 1.3.3. . . . .	28
2.1.	El círculo $C_L$ y su centro $c_L$ sobre la línea de consulta $L$ . . . . .	32
2.2.	Los tres casos de la Proposición 2.2.1. Se puede observar que la proyección ortogonal de $p_L$ sobre $L$ no siempre yace en el interior de $R(p_L)$ . . . . .	34
2.3.	El punto $p_\perp$ es el mínimo de la función $\rho$ sobre la recta $L$ . . . . .	34
2.4.	La prueba de la observación 2.2.3. . . . .	35
2.5.	El accionar del Algoritmo 2.2.1 sobre el punto $p_L$ . . . . .	37
2.6.	Podemos ver que $p^*$ puede ser diferente de $p_L$ si $c_P$ y $p_L$ están en distintos semiplanos inducidos por $L$ . . . . .	39

2.7. Podemos ver la existencia una recta $M$ , paralela a $L$ , suficientemente lejos de $p^*$ , para la cual $c_M$ es la proyección ortogonal de $p^*$ sobre $M$ . . . . .	40
2.8. La trayectoria $\mathcal{T}_x$ corta en dos puntos a la recta $L$ . . . . .	42
2.9. La prueba del Lema 2.3.3. . . . .	43
2.10. La sub-estructura para los vértices $v_0, v_1$ . . . . .	44
3.1. Separabilidad por círculos entre conjuntos de puntos. . . . .	47
3.2. Separabilidad por círculos entre polígonos convexos. . . . .	49
3.3. En esta figura se observa la prueba de la Observación 3.2.2. . . . .	51
3.4. En esta figura se observa la prueba de la Proposición 3.2.3. . . . .	52
3.5. La construcción del punto $s_0$ . . . . .	54
3.6. El segmento $S$ sobre el cual se encuentra $c$ . . . . .	55
3.7. La construcción del conjunto $Q_S$ y el punto $q_c$ . . . . .	56
3.8. La maximalidad de $q_c$ bajo la función $\tau$ . . . . .	57
3.9. Ya que el círculo $C(u)$ intersecta a $Q$ en más de dos puntos, $\tau(x)$ no es unimodal cuando la definimos sobre toda la frontera de $Q$ , sin embargo lo es cuando la definimos sobre $Q_S$ . . . . .	58

# Prefacio

El interés principal de este trabajo es presentar algunos resultados de investigación en el área de Geometría Computacional, por lo que quizá debamos preguntarnos primero ¿Qué es la Geometría Computacional?

Desde un punto de vista general, la Geometría Computacional es un área dentro de las Ciencias de la Computación, en la cual los principales elementos de estudio son los objetos geométricos en un espacio dado, cabe resaltar que los espacios euclidianos son los más estudiados. Esta área surge como parte del diseño y análisis de algoritmos a finales de la década de 1970 y ha sido un área de creciente interés desde entonces, lo cual se ve reflejado en la consagración a nivel internacional de varias revistas en el área, así como en el crecimiento de la comunidad de investigadores activos.

El éxito de esta área de investigación puede ser explicado por la belleza tanto de los problemas que en ella se estudian, como de las soluciones obtenidas. Además, estos problemas no se quedan exclusivamente en la parte teórica, sino que trascienden gracias a sus diversas aplicaciones en áreas como graficación por computadora, sistemas de información geográfica (SIG), robótica, comunicación, redes, etc. Es por esto que muchos de los problemas en Geometría Computacional son tangibles y tienden a tener una motivación real, lo que los hace fáciles de explicar aunque ciertamente no de resolver, como es el caso del siguiente problema.

Imaginemos que nos encontramos conduciendo nuestro auto en una calle y la alerta de gasolina nos avisa que necesitamos cargar el tanque urgentemente. Sabemos que hay muchas gasolineras en la ciudad, pero dada nuestra urgencia queremos ir a la más cercana. Sería útil tener un mapa en el cual pudiéramos buscar dicha estación, un mapa en donde la ciudad estuviera dividida en regiones y por cada región estuviera indicada la gasolinera más cercana: ver Figura 2. Esta subdivisión de la ciudad es llamada el diagrama de Voronoi del punto más cercano, y es uno de los conceptos fundamentales de la Geometría Computacional junto con todas sus variantes.



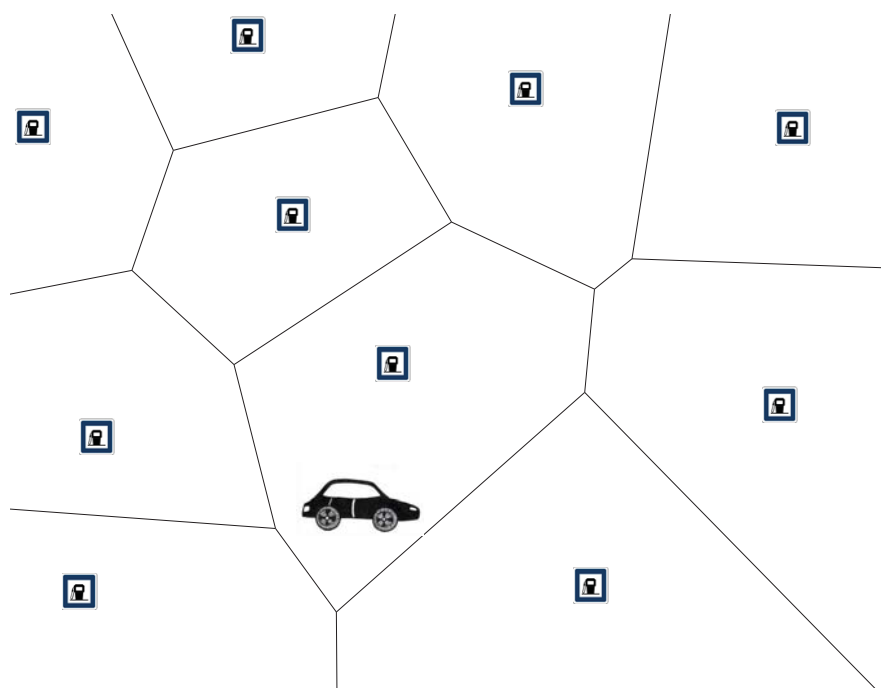


Figura 1: El diagrama de Voronoi del punto más cercano del conjunto de gasolineras de la ciudad.

Pensemos ahora el problema de manera un poco distinta e imaginemos que dentro de la ciudad existe un conjunto de  $m$  sitios en los que se pueden construir nuevas gasolineras. Sin embargo debido al presupuesto sólo  $k$  de ellas se pueden construir. Nos gustaría elegir los mejores  $k$  sitios de construcción, de modo que la gente que se tiene que desplazar una mayor distancia para llegar a una gasolinera, no tenga que hacer un recorrido muy largo. En pocas palabras, queremos minimizar la distancia máxima que existe entre un cliente y su gasolinera más cercana apegándonos al presupuesto. La solución de dicho problema es estudiada en un área de la Geometría Computacional llamada *Ubicación de Servicios*. En general, este tipo de problemas son llamados problemas min-máx y serán abordados a lo largo de este trabajo.

Si pensamos que el presupuesto nos permite construir únicamente una gasolinera, entonces nos interesaría saber quién es el cliente más lejano a cada posible sitio de construcción, ya que sabiendo esto para cada sitio, podríamos elegir aquel en el que se minimice la distancia que recorre el cliente que se desplaza la mayor distancia. Nuevamente nos gustaría tener una subdivisión del plano en regiones, de tal suerte que en cada región estuviera indicado cual es el cliente más lejano. De esta forma, si sabemos que una gasolinera

pertenece a una de estas regiones, implícitamente sabríamos cual es su cliente más lejano y cual es la distancia que debe recorrer dicho cliente para utilizar este servicio. A esta subdivisión del plano se le conoce como el diagrama de Voronoi del punto más lejano del conjunto de clientes y será el pilar principal del trabajo que presentaremos a continuación.

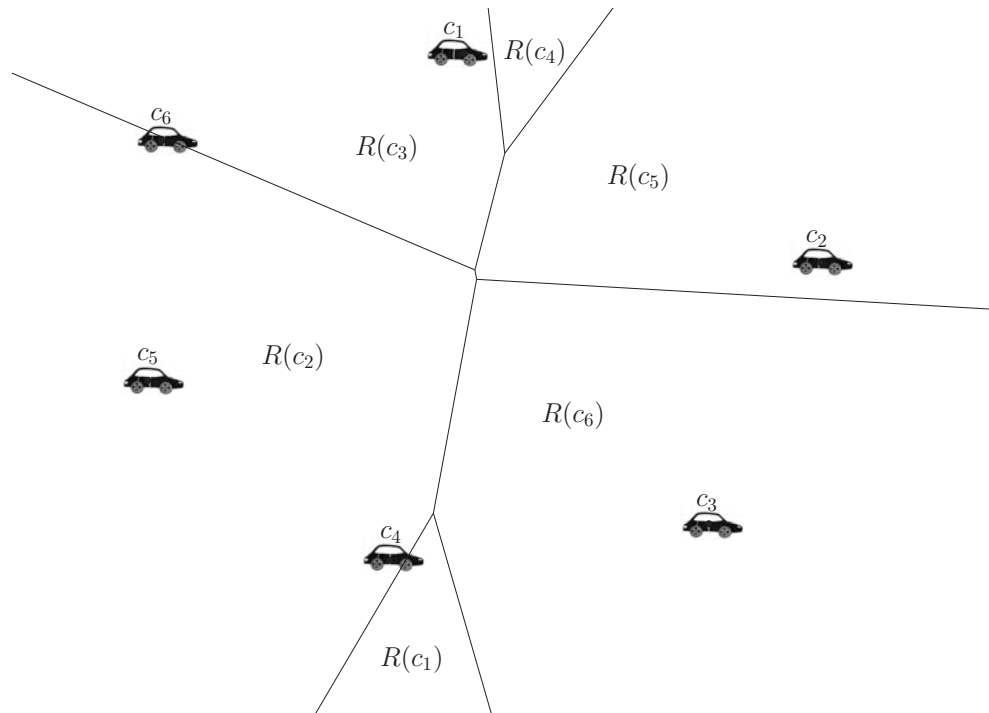


Figura 2: El diagrama de Voronoi del punto más lejano del conjunto clientes  $\{c_1, \dots, c_6\}$ . La región del cliente  $c_i$  está representada por el polígono  $R(c_i)$ , para toda  $1 \leq i \leq 6$ .



# Introducción

En este trabajo estudiaremos variantes de un problema clásico en geometría computacional llamado el problema de *Ubicación de servicios*. En dicho problema se tiene un conjunto de clientes que demandan un servicio (gasolineras, hospitales, tiendas, cobertura telefónica, cobertura televisiva, etc.); dicho servicio se ofrece a través de un conjunto de proveedores que deberán estar situados de manera que todos los clientes tengan acceso al servicio. Existen muchas variantes del problema, sin embargo, la más estudiada es aquella relacionada con encontrar la ubicación óptima para los proveedores, de modo que minimicemos la distancia máxima entre un cliente y su proveedor más cercano.

A lo largo de este trabajo nos enfocaremos en servicios de comunicación inalámbrica y particularmente, pensaremos a nuestros proveedores como un conjunto de antenas capaces de emitir mensajes a un conjunto de clientes con una potencia dada. Este modelo puede ser comparado con el servicio de radio o televisión, el cual se ofrece a través de antenas retransmisoras ubicadas en distintos puntos de una ciudad, y que cuentan con un cierto radio de influencia alrededor de la antena; ver Figura 3.

Este modelo supone que los clientes son únicamente receptores y el objetivo es que cada uno reciba el mensaje de al menos una de las antenas emisoras. El proceso de comunicación que estudiaremos es unidireccional y el objetivo es optimizar la potencia de transmisión de las antenas, ya que las antenas más potentes son más caras y requieren de mayor energía para transmitir.

El modelo se describe formalmente a continuación. Una antena inalámbrica  $\mathcal{A}$  tendrá asignada una posición y una potencia de transmisión que definirá un radio de cobertura  $r$  alrededor de dicha antena, de tal forma que un receptor  $\mathcal{R}$  escucha el mensaje emitido por  $\mathcal{A}$ , si y sólo si  $\mathcal{R}$  se encuentra a distancia a lo más  $r$  de  $\mathcal{A}$ .

Geoméricamente, una antena inalámbrica  $\mathcal{A}$  es modelada por un círculo  $C$ , centrado en la posición de la antena y con un radio  $r$  definido por la potencia de  $\mathcal{A}$ . Un receptor  $\mathcal{R}$ , generalmente será representado por un punto,

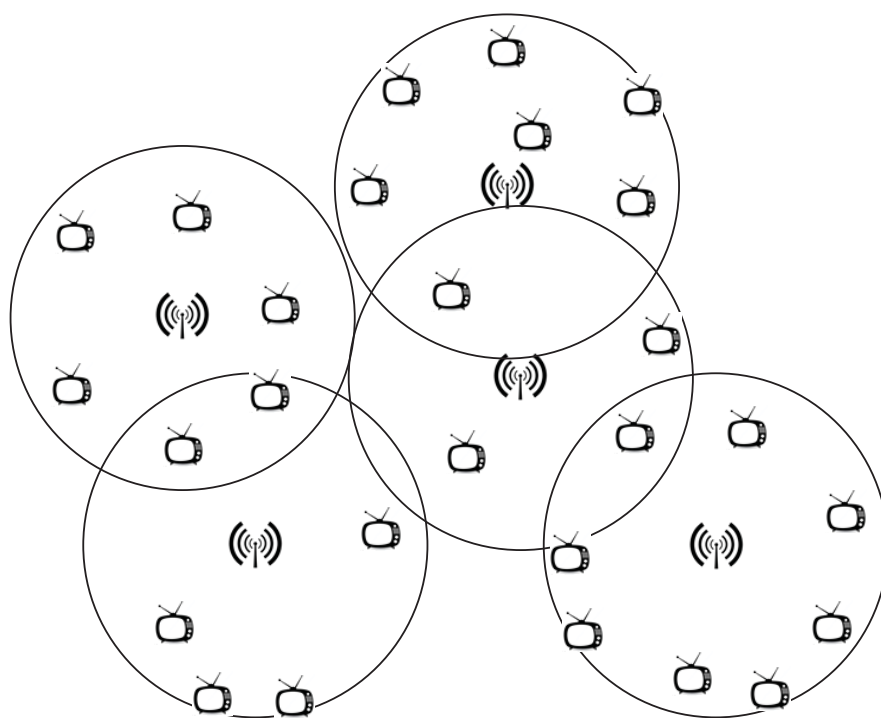


Figura 3: Un conjunto de antenas retransmisoras de televisión ofreciendo su servicio a un conjunto de clientes.

de modo que para que  $\mathcal{R}$  reciba el mensaje emitido por  $\mathcal{A}$ , el punto que lo representa deberá estar dentro del círculo  $C$ . Diremos que un receptor  $\mathcal{R}$  tiene cobertura, si existe al menos una antena  $\mathcal{A}$  tal que  $\mathcal{R}$  recibe los mensajes emitidos por  $\mathcal{A}$ .

### El problema del $k$ -centro

Una vez descrito el modelo, pasaremos a describir el problema de optimización que estudiaremos, dicho problema es conocido como el problema del  $k$ -centro y se plantea de la siguiente manera:

Dados  $P$  un conjunto de  $n$  puntos en el plano representando a los receptores y  $k$  un entero positivo, un  $k$ -centro de  $P$  es un conjunto  $A$  de  $k$  puntos en el plano representando a las antenas, de tal forma que la distancia máxima entre un receptor de  $P$  y su antena más cercana sea la menor posible.

Sea  $A$  un  $k$ -centro de  $P$  y sea  $r$  la distancia máxima entre un receptor y su antena más cercana. Note que si pedimos que todas las antenas tengan la misma potencia, entonces el radio mínimo que una antena necesita cubrir

es  $r$ , ya que de lo contrario, sin importar donde coloquemos las  $k$  antenas, un receptor de  $P$  quedaría siempre sin cobertura. Es por esto que la solución al problema de optimización está dada por un  $k$ -centro  $A$  y la distancia  $r$  asociada a éste, es decir, que buscamos la menor potencia posible que podemos asignar a  $k$  antenas, de manera que, distribuidas de manera óptima en el plano, ofrezcan cobertura a todos los clientes.

El problema del  $k$ -centro ha sido estudiado extensamente, Megiddo y Supowit probaron que el problema es NP-completo [27], más aún, Feder y Greene probaron que es NP-completo aproximararlo por un factor menor a 1.82 [18]. Sin embargo, existen varios algoritmos con factor de aproximación 2 y uno de ellos puede ser encontrado en [20]. Por otra parte, el problema del  $k$ -centro discreto, en el cual un conjunto  $M$  de posibles ubicaciones para las antenas está dado desde un inicio, es también NP-completo [31], no obstante es  $1 + \varepsilon$  aproximable para cualquier  $\varepsilon > 0$  [4]. Note que si  $k$  es constante con respecto a  $n$ , entonces el problema del  $k$ -centro discreto se puede resolver en tiempo polinomial, revisando los  $O(n^k)$  posibles subconjuntos de  $k$  elementos de  $M$  y determinando cual es el óptimo. Para el caso cuando  $k = 2$ , existe un algoritmo determinista de complejidad  $O(n \log^2 n \log^2 \log n)$  para el problema del 2-centro [7], dicho algoritmo es comparable con el algoritmo aleatorio para el mismo problema propuesto por Eppstein en [16], el cual tiene un tiempo esperado de ejecución de  $O(n \log^2 n)$ . Vale la pena resaltar que el caso  $k = 2$ , es el único caso en que se conocen algoritmos para resolver el problema del  $k$ -centro en menos de  $O(n^k)$ .

Para el caso  $k = 1$ , el problema del  $k$ -centro es equivalente a encontrar el círculo de radio mínimo que contenga a todos los receptores de  $P$ , visto de otra forma, se busca encontrar la posición de la antena en el plano que requiera la menor potencia para cubrir a todos los elementos de  $P$ .

El problema de encontrar el círculo de radio mínimo que contiene a un conjunto  $P$  de  $n$  puntos dados fue propuesto originalmente por Sylvester en 1857 [34]. Por simplicidad llamaremos  $P$ -círculo a cualquier círculo que contenga al conjunto  $P$ . Un algoritmo de  $O(n^2)$  fue presentado por Elzinga y Hearn [15] para encontrar el  $P$ -círculo de radio mínimo. Más adelante, Preparata [29] y Shamos y Hoey [33] presentaron independientemente dos algoritmos para resolver el problema en tiempo  $O(n \log n)$ . Lee [24] propuso el diagrama de Voronoi del punto más lejano, con el cual también fue posible encontrar el centro del  $P$ -círculo de radio mínimo en tiempo  $O(n \log n)$ . Finalmente, Megiddo [25] propuso un algoritmo óptimo de tiempo  $O(n)$  para resolver el problema del 1-centro en el plano utilizando “prune and search”. Más aún, el problema del 1-centro se puede resolver en tiempo  $O(n)$  en el peor caso en cualquier dimensión fija  $d$  [8][12]. Sin embargo debido a la complejidad de la implementación de dichos algoritmos, varios algoritmos aleatorios

de fácil implementación fueron propuestos posteriormente con un tiempo de ejecución esperado de  $O(n)$  [9][35].

Nuevas variantes del problema del 1-centro se han presentado recientemente, en este trabajo nos enfocaremos en aquellas en las que se busca encontrar el centro del  $P$ -círculo de radio mínimo bajo ciertas restricciones geométricas. Algunas de estas variantes se presentan a continuación:

Megiddo estudió el problema de encontrar el  $P$ -círculo de radio mínimo cuyo centro debe pertenecer a una línea recta dada, y propuso un algoritmo óptimo de tiempo  $O(n)$  [25] para resolver el problema. Hurtado, Sacristan y Toussaint presentaron un algoritmo de tiempo  $O(n + m)$  para encontrar el  $P$ -círculo de radio mínimo cuyo centro está restringido a cumplir  $m$  desigualdades lineales [17]. En particular, esto resuelve el problema el  $P$ -círculo de radio mínimo con centro en un polígono convexo de tamaño  $m$ . Bose y Toussaint consideraron la generalización de este problema, restringiendo el centro del  $P$ -círculo al interior de un polígono simple de tamaño  $m$ . Ellos propusieron un algoritmo de tiempo  $O((n + m) \log(n + m) + k)$ , donde  $k$  es el número de intersecciones del polígono con el diagrama de Voronoi del punto más lejano de  $P$  [6].

Posteriormente, Das, Karmakar, Nandy y Roy propusieron el problema de consulta, en el cual se permite preprocesar al conjunto  $P$  en una estructura de datos, de modo que pueda responder a consultas de la siguiente forma: Dada una línea recta  $L$ , encontrar el círculo de radio mínimo con centro en  $L$  de manera eficiente. Ellos presentaron un algoritmo con tiempo de preprocesado  $O(n \log n)$ , espacio lineal y que responde a las consultas en tiempo  $O(\log^2 n)$  [32]. Posteriormente Das, Karmakar y Roy mejoraron el tiempo de la consulta a  $O(\log n)$  utilizando espacio y preprocesado  $O(n^2)$  [21]. En este trabajo estudiamos el mismo problema y presentamos un algoritmo óptimo. En dicho algoritmo proponemos un preprocesado de tiempo  $O(n \log n)$  utilizando espacio lineal, que permite responder a las consultas en tiempo  $O(\log n)$ . Aunque cabe mencionar que Bose, Langerman y Roy propusieron otro algoritmo y estructura de datos completamente distintos que alcanzan el mismo tiempo de preprocesado y de consulta [30].

## Separación por círculos

El segundo problema que trataremos en este trabajo está relacionado con el concepto de separación geométrica. Formalmente definiremos  $\Phi$  como una familia de curvas en el plano tal que, para cada  $\phi \in \Phi$ ,  $\mathbb{R}^2 - \phi$  tiene exactamente dos componentes conexas  $\Pi_1(\phi)$ ,  $\Pi_2(\phi)$ . Decimos entonces que dos subconjuntos  $P, Q$  de  $\mathbb{R}^2$  son separables por  $\Phi$ , si y sólo si existe un  $\phi \in \Phi$ , tal que el interior de  $P$  está contenido en  $\Pi_1(\phi)$  y el interior de  $Q$

está contenido en  $\Pi_2(\phi)$ .

El problema de separabilidad geométrica ha atraído recientemente el interés de la comunidad científica, con  $\Phi$  considerada regularmente como la familia de líneas, círculos o polígonos simples en el plano. Si  $P$  y  $Q$  son conjuntos finitos de puntos en el plano, entonces el problema de separación por líneas se puede resolver usando programación lineal en tiempo  $O(n)$  [26].

El problema de encontrar un polígono simple separador con el menor número de vértices entre dos conjuntos finitos de puntos fue estudiado por Edelsbrunner y Preparata en [14]. Aggarwal, Booth, O'Rourke, Suri y Yap estudiaron el problema de encontrar el polígono separador de área mínima que se encuentra entre las fronteras de dos polígonos convexos anidados [2]. Das y Joseph extendieron el problema a dimensiones mayores y probaron que el problema de calcular el poliedro separador, que tenga el mínimo número de caras posible, para dos poliedros convexos anidados es NP-completo [10].

El estudio de la separabilidad por círculos se extendió a partir del descubrimiento de sus muy variadas aplicaciones en el área de reconocimiento de patrones y procesamiento de imagen [22] [19]. En dimensiones mayores, Lay [23] introdujo la idea de transformar una instancia del problema de separabilidad por esferas en  $\mathbb{R}^d$ , en un problema de separación lineal en  $\mathbb{R}^{d+1}$  al utilizar la proyección estereográfica. Dados  $P, Q$  subconjuntos de  $\mathbb{R}^2$ , el problema de optimización más estudiado es el de encontrar el círculo de radio mínimo que separe a  $P$  de  $Q$ . Anderson y Kim [22] presentaron un algoritmo de tiempo cuadrático para resolver este problema entre dos conjuntos finitos de puntos. Bhattacharya [3] mejoró el tiempo de ejecución a  $O(n \log n)$ , calculando las regiones dónde podrían encontrarse centrados todos los posibles círculos separadores entre ambos conjuntos. Finalmente Kosaraju, Megiddo y O'Rourke [28] propusieron un algoritmo óptimo, encontrando en tiempo  $O(n)$  el círculo separador de radio mínimo, y en tiempo  $O(n \log n)$  el círculo separador de radio máximo entre dos conjuntos finitos de puntos. Ellos utilizaron transformaciones parabólicas para obtener una instancia de un problema de minimización cuadrática, convexa en tres dimensiones. Así mismo Boissonnat, Czyzowicz, Devillers y Yvinec [5] propusieron un algoritmo lineal de decisión, que verifica si dos conjuntos de puntos son o no, separables por círculos.

En este trabajo consideramos la versión en línea del problema de separación por círculos, en el cual nos permitimos preprocesar un polígono  $P$  con  $n$  vértices, para construir una estructura de datos que soporta consultas de la siguiente forma:

Dado un polígono convexo  $Q$  como la lista de sus  $m$  vértices, encontrar eficientemente el círculo de radio mínimo que contiene a  $P$  y cuyo interior no interseca a  $Q$ . La estructura de datos propuesta en este trabajo requiere



de tiempo  $O(n)$  para su construcción, utiliza espacio lineal y responde a las consultas en tiempo  $O(\log n \log m)$ . Cabe destacar que resolver este problema sin hacer uso del preprocesado requiere de tiempo  $O(n)$  [28].

# Capítulo 1

## $P$ -círculos y el diagrama de Voronoi del punto más lejano

### 1.1. Nuestro problema

Sea  $P$  un conjunto de  $n$  casas representadas por  $n$  puntos en el plano. Nuestro objetivo es ofrecer cobertura radiofónica a cada casa utilizando una antena inalámbrica. La potencia de transmisión de la antena define un radio de cobertura  $r$  alrededor del punto de emisión, tal que toda casa que tenga cobertura se encuentra a distancia a lo más  $r$  de la antena. Es claro que si aumentamos la potencia de nuestra antena, aumentará su consumo de energía y por ende también su costo de operación. Buscaremos siempre ofrecer el servicio de la manera más barata, por lo que nuestro objetivo es encontrar la posición y potencia óptimas para la antena, de manera que todas las casas se encuentren dentro de su radio de cobertura, ver Figura 1.1.

En este trabajo, un círculo es el conjunto de puntos a distancia a lo más  $r$  de un punto en el plano,  $r \geq 0$ . Claramente nuestra antena, junto con su radio de cobertura, definen un círculo en el plano. El problema de cobertura óptima se traduce entonces en encontrar el círculo de radio mínimo que contenga los  $n$  puntos de  $P$ .

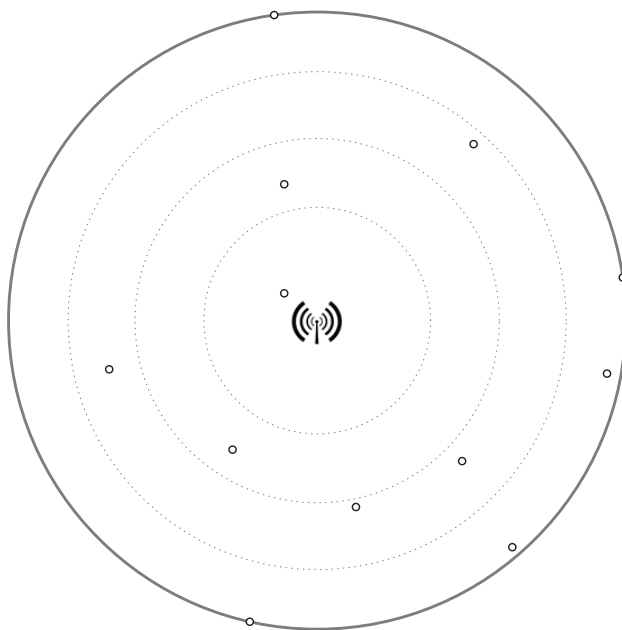


Figura 1.1: La antena inalámbrica de transmisión radiofónica dando cobertura a un conjunto de casas representadas por este conjunto de puntos.

## 1.2. El diagrama de Voronoi del punto más lejano

Sea  $P$  un conjunto de  $n$  puntos en el plano y sea  $C$  un círculo. Diremos que  $C$  es un  $P$ -círculo si contiene a todos los puntos de  $P$ .

Sea  $C_P$  el  $P$ -círculo de radio mínimo y sea  $c_P$  su centro. Es fácil ver que la circunferencia de  $C_P$  debe pasar por al menos un punto de  $P$ , de lo contrario siempre podríamos reducir el radio del círculo de manera que todos los puntos de  $P$  sigan en el interior. Sea  $p$  un punto de  $P$  sobre la frontera del círculo  $C_P$ . Note que la distancia de  $c_P$  a cualquier otro punto de  $P$  es a lo más  $d(c_P, p)$ , donde  $d(*, *)$  denota la distancia euclidiana entre dos objetos geométricos en el plano. Visto de otra forma,  $p$  es uno de los puntos más lejanos de  $c_P$  de entre los elementos del conjunto  $P$ ; ver Figura 1.2(a). Este concepto de lejanía nos lleva a pensar en las siguientes preguntas. ¿Cuáles serán los puntos en el plano para los que  $p$  es el elemento más lejano dentro del conjunto  $P$ ? ¿Se podrá dar una caracterización de este conjunto de puntos? El diagrama de Voronoi del punto más lejano de  $P$  es el objeto geométrico que da respuesta a estas y otras preguntas.

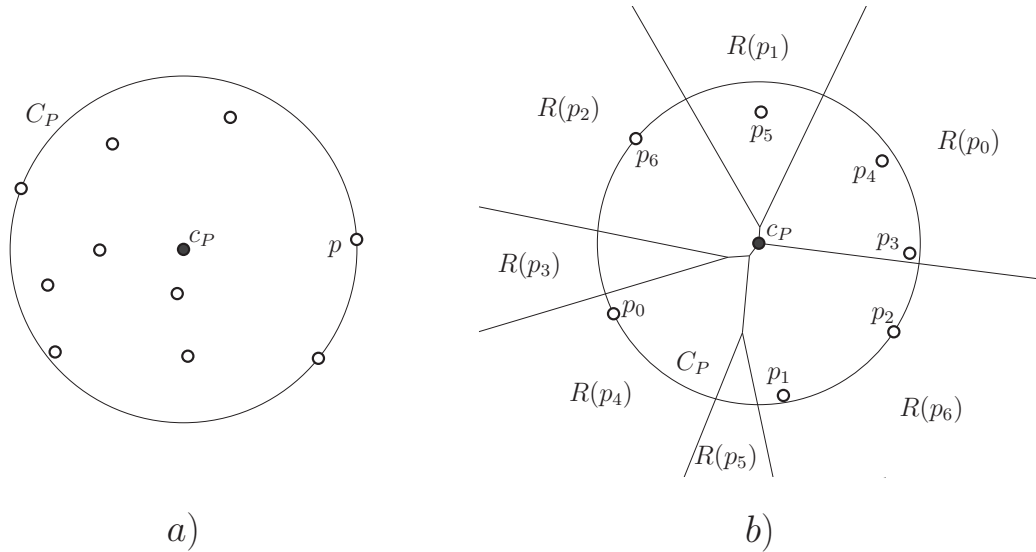


Figura 1.2: a) El círculo  $C_P$  y su centro  $c_P$ . b) El diagrama de Voronoi del punto más lejano de  $\{p_0, \dots, p_6\}$ .

### 1.2.1. ¿Qué es el Diagrama de Voronoi del punto más lejano?

Sea  $P = \{p_1, \dots, p_n\}$  un conjunto de  $n$  puntos en el plano. El diagrama de Voronoi del punto más lejano de  $P$  o simplemente diagrama de Voronoi de  $P$ <sup>1</sup>, es una división del plano en  $n$  regiones  $R(p_1), \dots, R(p_n)$ , una por cada punto de  $P$ , con la propiedad de que un punto  $x$  yace en la región  $R(p_i)$  si y sólo si  $d(p_i, x) \geq d(p_j, x)$ ,  $1 \leq i, j \leq n$ .

Si suponemos que  $P$  consta únicamente de dos puntos  $p, q$ , entonces la bisectriz de  $p$  y  $q$  divide el plano en dos semiplanos cerrados que definen las regiones de Voronoi  $R(p)$  y  $R(q)$ ; ver Figura 1.3.

Sea  $\Pi_j$  el semiplano cerrado inducido por la bisectriz de  $p_i$  y  $p_j$  que contiene a  $p_j$ . Otra forma de definir a  $R(p_i)$  es la siguiente:

**Definición 1.** Sea  $p_i$  un punto en  $P$ ,

$$R(p_i) = \bigcap_{j \neq i} \Pi_j.$$

<sup>1</sup>Cabe resaltar que nombre corto de *diagrama de Voronoi* es comúnmente usado en la literatura para referirse al diagrama de Voronoi del punto más cercano, sin embargo en este trabajo sólo utilizaremos el diagrama del punto más lejano por lo que no habrá confusiones.

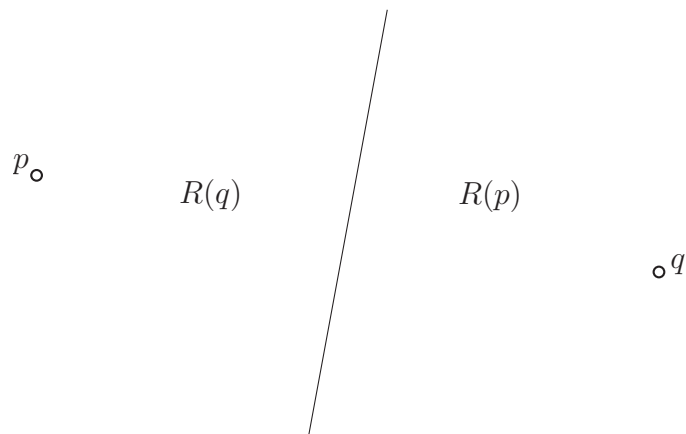


Figura 1.3: El diagrama de Voronoi de un conjunto de dos puntos.

Es fácil verificar que ambas definiciones de  $R(p_i)$  son equivalentes. Note que si  $x \in \Pi_j$ , entonces  $x$  está más lejos de  $p_i$  que de  $p_j$ . Como  $x$  pertenece a cada  $\Pi_j$  para todo  $j \neq i$ , entonces  $x$  está más lejos de  $p_i$  que de cualquier otro punto de  $P$ .

De la definición anterior podemos deducir varias cosas, en primer lugar observemos que la intersección que define a  $R(p_i)$  puede ser vacía. Note también que  $R(p_i)$  es una región convexa, ya que se define como la intersección finita de conjuntos convexos. Otra propiedad que se deriva, es que toda arista en la frontera de  $R(p_i)$  está definida a partir de la bisectriz de dos puntos de  $P$ .

Formalmente definimos al diagrama de Voronoi del punto más lejano de  $P$ , denotado por  $V_P$  como sigue:

$$V_P = \{R(p) : p \in P\}.$$

Un ejemplo del diagrama  $V_P$  se puede observar en la Figura 1.2(b). Veremos a continuación varias propiedades del diagrama de Voronoi.

**Proposición 1.2.1.** *Para cualquier punto  $x \in \mathbb{R}^2$ , el punto de  $P$  más lejano de  $x$  yace sobre la frontera del cierre convexo de  $P$ .*

*Demostración.* Un resultado bien conocido dice que un punto está en la frontera de un conjunto convexo, si y sólo si existe una recta soporte que pasa por ese punto. Donde una recta soporte se define como una línea recta que intersecta al conjunto, pero que lo deja completamente contenido en uno de los dos semiplanos cerrados que induce.

Sea  $p$  uno de los puntos de  $P$  más lejanos de  $x$ , y sea  $C_x$  el círculo con centro en  $x$  y radio  $d(x, p)$ . Es claro que  $C_x$  es un  $P$ -círculo, por consiguiente la tangente a  $C_x$  por el punto  $p$  es una recta soporte del cierre convexo de  $P$ , el resultado se sigue; ver Figura 1.4.  $\square$

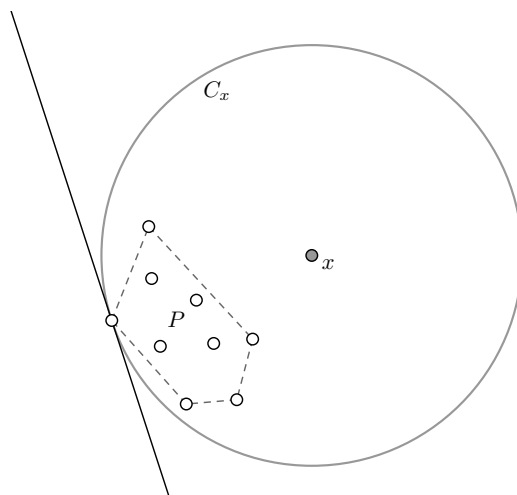


Figura 1.4: La prueba de la Proposición 1.2.1.

De la observación anterior podemos concluir que la región de Voronoi de un punto  $p \in P$  es no vacía, si y sólo si  $p$  forma parte de la frontera del cierre convexo de  $P$ . Por ende, podemos olvidarnos de los puntos interiores y suponer sin pérdida de generalidad que  $P$  es un conjunto de puntos en posición convexa.

**Observación 1.2.2.** *Sea  $p$  en  $P$  y sea  $x$  un punto en el plano. Si el círculo con centro en  $x$  y radio  $d(x, p)$  es un  $P$ -círculo, entonces el punto  $x$  pertenece a  $R(p)$ .*

La observación anterior se sigue directamente de la definición y con ella podemos deducir lo siguiente:

**Proposición 1.2.3.** *Si  $p$  es un punto en la frontera del cierre convexo de  $P$ , entonces la región de Voronoi  $R(p)$  es un polígono convexo no acotado.*

*Demostración.* Sea  $x$  un punto en la frontera de  $R(p)$ , y sea  $\ell_{x,p}$  la línea recta que pasa por  $x$  y  $p$ . Consideremos la semirecta  $\lambda$  contenida en  $\ell_{x,p}$  con ápice  $x$  y que no contiene a  $p$ . Como  $x \in R(p)$ , entonces el círculo  $C_x$  con centro en  $x$  y radio  $d(x, p)$  es un  $P$ -círculo. Sea  $y$  un punto en  $\lambda$ , y sea  $C_y$  el círculo

con centro en  $y$  y radio  $d(y, p)$ . Es claro que  $C_y$  es un  $P$ -círculo pues contiene a  $C_x$ . Usando la Observación 1.2.2 tenemos que  $y \in R(p)$  y por lo tanto  $R(p)$  es un conjunto convexo no acotado.  $\square$

Al ser  $R(p)$  un polígono convexo no acotado, podemos concluir que su frontera contiene exactamente dos semirrectas, nos referimos a estas semirrectas como las aristas no acotadas de  $R(p)$ .

Note que cada región de Voronoi es un polígono definido por un conjunto de vértices y aristas. Por ende nos interesará pensar en el diagrama de Voronoi como una gráfica, sin embargo hay que tener cuidado ya que  $V_P$  contiene un número lineal de aristas no acotadas. Es por esto que para poder referirnos al diagrama de Voronoi como una gráfica, agregaremos hojas  $\eta_1, \dots, \eta_h$  sobre cada una de las aristas no acotadas de  $V_P$  suficientemente lejos, como se observa en la Figura 1.5. Obtendremos así una gráfica geométrica plana sobre el conjunto de vértices de las regiones de  $V_P$ , a la cual denotaremos a lo largo de este trabajo por  $\mathcal{V}(P)$ .

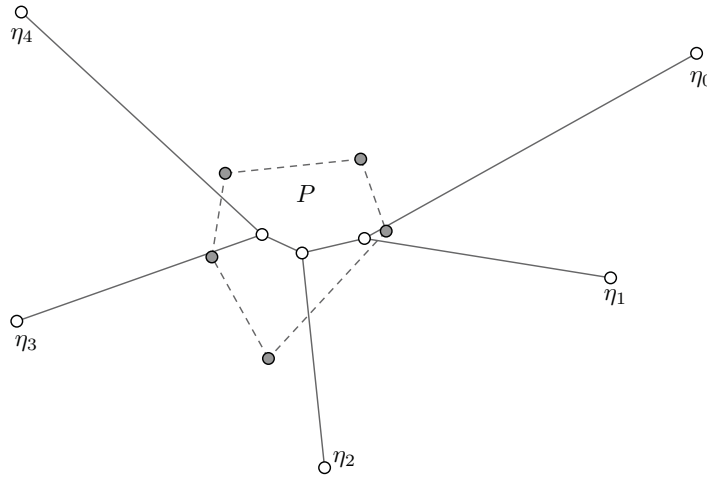


Figura 1.5: El diagrama de Voronoi visto como una gráfica  $\mathcal{V}(P)$ , al agregar hojas suficientemente lejos sobre cada arista no acotada.

**Proposición 1.2.4.** *Los vértices y aristas de  $\mathcal{V}(P)$  forman una gráfica acíclica y conexa.*

*Demostración.* Ya que el diagrama de Voronoi es una división del plano, podemos concluir que la gráfica formada por los vértices y aristas del diagrama es conexa. Además, no puede tener ciclos puesto que ello implicaría la existencia de una región acotada.  $\square$

Gracias a la proposición anterior sabemos que  $\mathcal{V}(P)$  es un árbol, sin embargo no conocemos cuál el número de aristas o vértices que tiene, es por esto que el siguiente resultado será útil más adelante.

**Lema 1.2.5.** *Si  $P$  es un conjunto de  $n$  puntos en el plano en posición convexa, tal que no contenga cuatro puntos cocirculares, entonces el árbol  $\mathcal{V}(P)$  tiene exactamente  $2n - 3$  aristas.*

*Demostración.* Denotaremos por  $m_e, m_v, m_f$  al número de aristas, vértices y caras respectivamente que tiene el diagrama de Voronoi de  $P$ . Sea  $\mathcal{V}'(P)$  la gráfica obtenida de  $\mathcal{V}(P)$  al contraer las  $n$  hojas  $\eta_1, \dots, \eta_n$  a un sólo vértice  $v_\infty$ . Es claro que  $\mathcal{V}'(P)$  es una gráfica plana encajable en la esfera.

Observe que  $m_f = n$ , ya que el número de regiones del diagrama de Voronoi corresponde biunívocamente con el número de puntos de  $P$ . Usamos la característica de Euler para obtener la siguiente igualdad:

$$(m_v + 1) - m_e + m_f = 2. \quad (1.1)$$

Un resultado bien conocido en Teoría de gráficas dice que

$$\sum_{v \in V} \delta(v) = 2|E|.$$

Es decir que la suma de los grados de los vértices es igual al doble del número de aristas de la gráfica. Cada vértice de  $\mathcal{V}'(P)$  distinto de  $v_\infty$  tiene grado tres, puesto que no hay cuatro puntos cocirculares, además por construcción  $\delta(v_\infty) = n$ , por ende:

$$3m_v + n = \sum_{v \in \mathcal{V}'(P)} \delta(v) = 2m_e. \quad (1.2)$$

Resolviendo el sistema de ecuaciones definido por (1.1) y (1.2) podemos concluir lo siguiente:

$$m_e = 2n - 3.$$

Además, por construcción el número de aristas de  $\mathcal{V}'(P)$  es igual al de  $\mathcal{V}(P)$ , por lo que nuestro resultado se sigue.  $\square$

Note que si no podemos garantizar que no haya cuatro puntos cocirculares, entonces lo único que podemos afirmar es que  $m_e \leq 2n - 3$ .



### 1.2.2. Calculando el diagrama de Voronoi

Sea  $P$  un conjunto de  $n$  puntos en el plano, en esta sección describiremos algunos métodos para calcular el diagrama de Voronoi del punto más lejano de  $P$ .

Note que de la Definición 1 podemos obtener un algoritmo trivial de tiempo  $O(n^2 \log n)$  para calcular a  $V_P$ . Esto puede lograrse al construir la región de Voronoi de cada punto de  $P$  a partir de la intersección de  $n-1$  semiplanos, sin embargo veremos que este algoritmo dista mucho de ser óptimo. Describiremos a continuación un algoritmo aleatorio para obtener el diagrama de Voronoi del punto más lejano de  $P$  en tiempo esperado  $O(n \log n)$  [11].

Como primer paso obtenemos el cierre convexo del conjunto  $P$  en tiempo  $O(n \log n)$  y consideremos los  $h$  puntos de  $P$  que forman parte de la frontera del cierre convexo en el orden de las manecillas del reloj.

Dado un punto  $p$  en la frontera del cierre convexo de  $P$ , definimos al punto siguiente, denotado por  $cw(p)$ , y al anterior de  $p$ , denotado por  $ccw(p)$ , como sus vecinos sobre el cierre convexo en la dirección de las manecillas del reloj y en contra de éstas respectivamente.

Ordenamos aleatoriamente los  $h$  puntos de la frontera del cierre convexo de  $P$  y nos olvidamos de los puntos interiores. Supongamos que el orden aleatorio obtenido es el siguiente:  $p_1, p_2, \dots, p_h$ . Removemos uno por uno de estos puntos, comenzando por  $p_h$  y continuamos en orden hasta quedarnos únicamente con los puntos  $p_1, p_2$  y  $p_3$ . Justo antes de remover a  $p_i$ ,  $4 \leq i \leq h$ , le agregamos a este punto un apuntador a  $cw(p_i)$  y otro a  $ccw(p_i)$ . Note que una vez que un punto es removido, no puede ser el vecino anterior ni siguiente de ningún otro punto removido posteriormente.

Una vez terminado el proceso, usamos el algoritmo extraído de la definición 1, para calcular en tiempo constante el diagrama de Voronoi del punto más lejano de los puntos  $p_1, p_2$  y  $p_3$ . Calcularemos el diagrama de Voronoi incrementalmente al insertar los vértices  $p_4, \dots, p_h$  en ese orden. Supongamos que tenemos calculado el diagrama de Voronoi de los puntos  $p_1, p_2, \dots, p_{i-1}$ , describimos a detalle el proceso de inserción del punto  $p_i$  en el diagrama.

La región de Voronoi de  $p_i$  se insertará “entre” las regiones de  $cw(p_i)$  y  $ccw(p_i)$ , de hecho, antes de la inserción de  $p_i$ , las regiones de Voronoi de  $cw(p_i)$  y  $ccw(p_i)$  son vecinas en el diagrama de los puntos  $\{p_1, \dots, p_{i-1}\}$ . Por ende, estas regiones están separadas por una semirrecta contenida en la bisectriz de  $cw(p_i)$  y  $ccw(p_i)$ , semirrecta a la cual supondremos que  $cw(p_i)$  tiene un apuntador; ver Figura 1.6(a).

La bisectriz de  $p_i$  y  $ccw(p_i)$ , denotada por  $\ell$ , contiene una semirrecta  $\lambda$  que formará parte de la frontera de  $R(p_i)$ . Note que antes de la inserción de  $p_i$ ,  $\lambda$  está contenida en la región de Voronoi de  $ccw(p_i)$ . El algoritmo recorre en

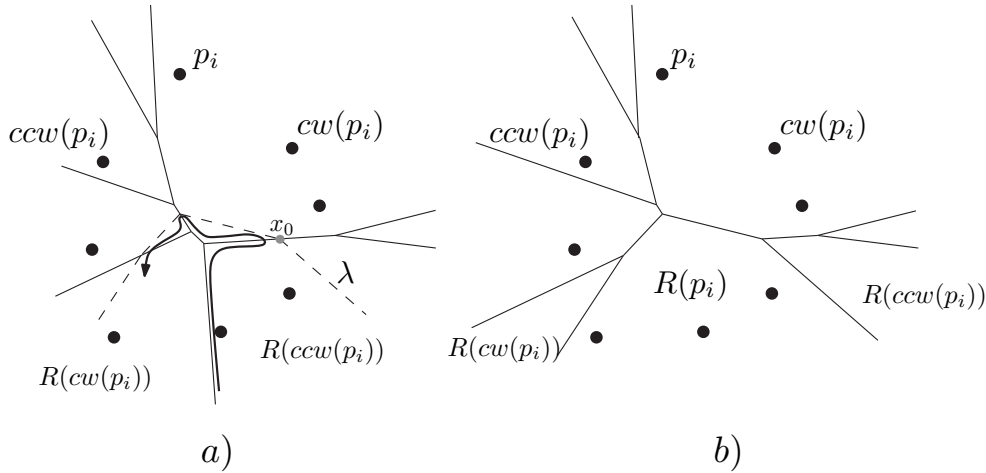


Figura 1.6: a) El proceso de inserción de  $p_i$ . b) El diagrama de Voronoi después de la inserción de  $p_i$ .

orden las aristas de la frontera de  $R(ccw(p_i))$ , en el sentido de las manecillas del reloj, hasta encontrar aquella arista que intersecta a la recta  $\ell$  en un punto  $x_0$ ;  $\lambda$  será entonces la semirrecta con apice  $x_0$  contenida en  $\ell \cap R(ccw(p_i))$ ; ver Figura 1.6(a).

Del otro lado de la arista donde yace  $x_0$ , se encuentra la región de Voronoi de otro punto de  $P$ , digamos  $p_j$ ,  $j \in \{1, \dots, i-1\}$ . La bisectriz de  $p_j$  y  $p_i$ , denotada por  $\ell_j$ , definirá otra arista de la región de Voronoi de  $p_i$ . Note que  $x_0 \in \ell_j$ , puesto que  $x_0$  está en la frontera tanto de  $R(p_j)$  como de  $R(ccw(p_i))$ . Recorremos en orden las aristas de  $R(p_j)$  en el orden de las manecillas del reloj, desde  $x_0$ , hasta encontrar otra arista en la frontera de  $R(p_j)$  que intersecte a  $\ell_j$  en un punto  $x_1$ . Note que  $x_1$  está en la frontera de dos regiones de Voronoi y por lo tanto, del otro lado de la arista que contiene a  $x_1$ , está la región de Voronoi de otro punto de  $P$ .

Este proceso continúa hasta encontrar una arista que intersecte la frontera de la región de Voronoi de  $ccw(p_i)$ , momento en el cual la bisectriz de  $p_i$  y  $ccw(p_i)$  define una nueva semirrecta, con lo cual finaliza el algoritmo; ver Figura 1.6. Note que el algoritmo construye la región de Voronoi de  $p_i$  en el orden opuesto a las manecillas del reloj y actualiza las regiones de Voronoi visitadas durante la inserción.

**Teorema 1.2.6.** *Dado un conjunto  $P$  de  $n$  puntos en el plano, el diagrama de Voronoi del punto más lejano de  $P$  se puede calcular en tiempo esperado  $O(n \log n)$ .*

*Demostración.* Encontrar los  $h$  puntos del cierre convexo de  $P$  en el orden de las manecillas del reloj requiere de tiempo  $O(n \log n)$ . Sin embargo, calcular el diagrama de Voronoi del punto más lejano de estos  $h$  puntos, se puede hacer en tiempo esperado  $O(h)$ . Para mostrar esto, nos fijamos en el momento justo después de la inserción del punto  $p_i$ . Note que si la región de Voronoi de  $p_i$  consta de  $k$  aristas, entonces en el proceso de inserción de  $p_i$ , se atravesaron exactamente  $k$  regiones de Voronoi asociadas a  $k$  puntos de  $\{p_1, p_2, \dots, p_{i-1}\}$ ; llamemos a esos puntos  $p'_1, p'_2, \dots, p'_k$ .

Es fácil ver que las aristas que se atravesaron durante esta construcción, son exactamente aquellas que pertenecen al diagrama de Voronoi de  $\{p'_1, p'_2, \dots, p'_k\}$ ; ver Figura 1.6(a). Por ende, en la inserción de  $p_i$ , se visitaron a lo más  $2k - 3$  aristas distintas, sin embargo, cada arista pudo ser visitada dos veces. Por consiguiente, en la construcción de la región  $R(p_i)$  se visitaron a lo más  $4k - 6$  aristas. Obteniendo así una complejidad de  $O(k)$  para la construcción de la región de Voronoi de  $p_i$ .

Note que el diagrama de Voronoi de los puntos  $\{p_1, p_2, \dots, p_{i-1}\}$  tiene exactamente  $2i - 3$  aristas por el Lema 1.2.5, además, cada arista estará compartida por dos regiones. Por lo tanto, el número de aristas en promedio de cada una de las  $i - 1$  regiones de Voronoi de los puntos  $\{p_1, p_2, \dots, p_{i-1}\}$  es menor que cuatro. Y ya que cada punto de  $P$  tiene la misma probabilidad de ser el  $i$ -ésimo insertado, entonces podemos afirmar que el tamaño esperado de cada región de Voronoi es menor que cuatro. Por consiguiente el tiempo esperado de la inserción del punto  $p_i$  es  $O(1)$  para todo  $i$ . Obteniendo así un tiempo esperado de  $O(h)$  para la construcción del diagrama de Voronoi del punto más lejano.  $\square$

Recordemos que si tenemos los puntos en posición convexa, entonces nuestro algoritmo termina en tiempo lineal esperado. De hecho, existe otro algoritmo que calcula el diagrama de Voronoi del punto más lejano de un conjunto de  $n$  puntos en posición convexa, en tiempo  $O(n)$  [1].

Como cada arista de  $\mathcal{V}(P)$  yace sobre la bisectriz de un par de puntos, supondremos de aquí en adelante que cada arista del diagrama de Voronoi tiene apuntadores a los dos puntos de  $P$  que la definen.

### 1.3. $P$ -círculos de radio mínimo

En esta sección analizaremos la relación que existe entre los  $P$ -círculos de radio mínimo y el diagrama de Voronoi del punto más lejano. Finalmente caracterizaremos al  $P$ -círculo de radio mínimo y daremos un algoritmo para encontrar su centro y el valor de su radio. Recordemos que  $C_P$  denota al  $P$ -círculo de radio mínimo y  $c_P$  a su centro.

**Observación 1.3.1.**  $C_P$  contiene al menos dos puntos de  $P$  en su circunferencia.

La observación anterior se puede apreciar en la Figura 1.7(a). De hecho, si suponemos que no hay cuatro puntos cocirculares en  $P$ , entonces  $C_P$  contiene a lo más tres puntos de  $P$  en su circunferencia. La observación anterior tiene trivialmente la siguiente implicación sobre la ubicación de  $c_P$ .

**Proposición 1.3.2.** El punto  $c_P$  es un vértice de  $\mathcal{V}(P)$  o yace en el interior de una de sus aristas.

*Demostración.* Es claro que si la circunferencia de  $C_P$  pasa por dos puntos  $p_1, p_2$ , entonces  $c_P$  se encuentra sobre la bisectriz de  $p_1, p_2$  y pertenece tanto a  $R(p_1)$  como a  $R(p_2)$ . Por ende  $c_P$  yace sobre una arista de  $\mathcal{V}(P)$ . Un argumento similar se utiliza para el caso dónde  $C_P$  pasa por tres o más puntos de  $P$ , caso en el cual  $c_P$  corresponde a un vértice de  $\mathcal{V}(P)$ ; ver Figura 1.7(b).  $\square$

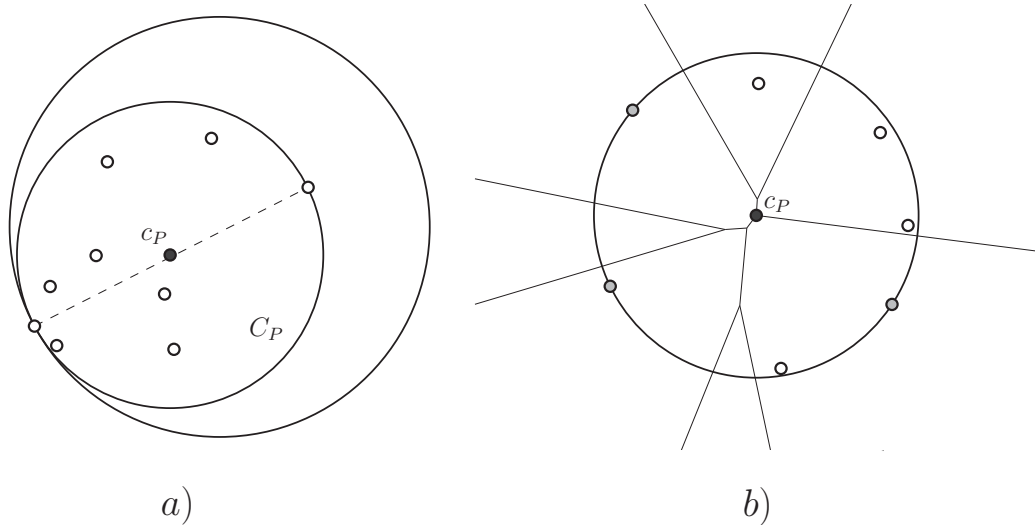


Figura 1.7: a) Cualquier círculo que pase por un único punto de  $P$ , contiene otro círculo que pasa por al menos 2 puntos de  $P$ . b) El círculo  $C_P$  pasando por tres puntos de  $P$ .

Presentamos algunas definiciones que encontraremos útiles más adelante.

**Definición 2.** Sea  $x \in \mathbb{R}^2$ ,  $C(x)$  se define como el  $P$ -círculo de radio mínimo con centro en  $x$ .

Sea  $x \in \mathbb{R}^2$ , note que si  $x$  pertenece a la región de Voronoi de un punto  $p \in P$ , entonces el radio de  $C(x)$  está dado por la distancia entre  $x$  y  $p$ . De aquí la importancia del diagrama de Voronoi del punto más lejano siempre que hablemos de  $P$ -círculos de radio mínimo. Veamos ahora algunas propiedades del radio de los  $P$ -círculos.

**Definición 3.** Sea  $\rho : \mathbb{R}^2 \rightarrow \mathbb{R}$ , tal que  $\rho(x)$  es el radio de  $C(x)$ ,  $x \in \mathbb{R}^2$ .

**Proposición 1.3.3.** Sean  $x, y$  dos puntos en el plano. Para todo  $z \in [x, y]$  se cumple que  $C(z) \subseteq C(x) \cup C(y)$ . Más aún,  $\rho(z) \leq \max\{\rho(x), \rho(y)\}$ .

*Demostración.* Sean  $a$  y  $b$  los dos puntos de intersección entre las circunferencias de  $C(x)$  y  $C(y)$ . Para cualquier  $z \in [x, y]$ , defina

$$r = d(a, z) = d(b, z)$$

y sea  $C_r(z)$  el círculo de radio  $r$  con centro en  $z$ . Es claro que  $C_r(z)$  contiene a  $C(x) \cap C(y)$  y por ende contiene también a  $P$ , de aquí que  $\rho(z) \leq r$ . Además es claro que  $C_r(z) \subseteq C(x) \cup C(y)$  por construcción; ver Figura 1.8. Por consiguiente  $C(z) \subseteq C(x) \cup C(y)$ .  $\square$

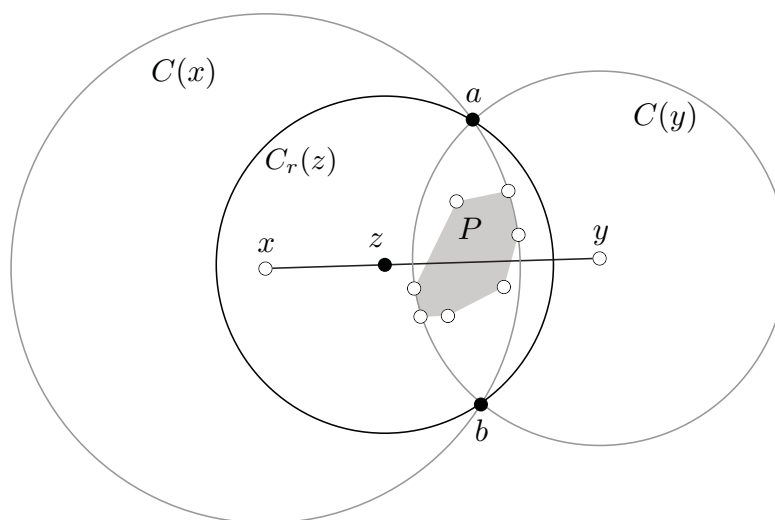


Figura 1.8: La prueba de la Proposición 1.3.3.

Veremos a continuación una caracterización del punto  $c_P$ .

**Proposición 1.3.4.** El punto  $c_P$  es el único mínimo local de la función  $\rho$  en el plano.

*Demostración.* Supongamos que existen dos mínimos locales  $x_0, x_1 \in \mathbb{R}^2$  de la función  $\rho$ , sin pérdida de generalidad, supongamos que  $\rho(x_0) \geq \rho(x_1)$ . Sea  $z$  un punto en el segmento  $[x_0, x_1]$  a distancia epsilon de  $x_0$ . Por la Proposición 1.3.3,  $C(z) \subseteq C(x_0) \cup C(x_1)$ . Por ende  $C(z)$  es un  $P$ -círculo que tiene radio estrictamente menor que  $C(x_0)$ , lo cual implica que  $x_0$  no es un mínimo local de  $\rho$ . Contradicción que viene de suponer que existe más de un mínimo local de la función  $\rho$ .  $\square$

La proposición anterior implica trivialmente que el  $P$ -círculo de radio mínimo es único.

**Proposición 1.3.5.** *Sea  $uv$  una arista de  $\mathcal{V}(P)$  contenida en la bisectriz de los puntos  $p_0, p_1 \in P$ . Si  $[u, v] \cap [p_0, p_1] \neq \emptyset$ , entonces  $c_P = [u, v] \cap [p_0, p_1]$*

*Demostración.* Sea  $x \in [u, v] \cap [p_0, p_1]$ . Note que  $[u, v] = R(p_0) \cap R(p_1)$ , por ende  $\rho(x) = d(x, p_0) = d(x, p_1)$ . Si movemos a  $x$  en cualquier dirección una epsilon distancia, entonces  $x$  se alejará de  $p_0$  o de  $p_1$ , por lo que el radio de  $C(x)$  aumentará necesariamente. Por consiguiente  $x$  es un mínimo local de la función  $\rho$ , lo cual usando la Proposición 1.3.4 implica que  $x = c_P$ .  $\square$

Analizaremos ahora el comportamiento de la función  $\rho$  sobre los puntos de  $\mathcal{V}(P)$ .

**Teorema 1.3.6.** *Sea  $x$  un punto sobre una arista de  $\mathcal{V}(P)$  y sea  $T$  una trayectoria sobre el árbol  $\mathcal{V}(P)$  que una al punto  $c_P$  con  $x$ . Se cumple que  $\rho$  es una función estrictamente monótona sobre los puntos de  $T$  con mínimo en  $c_P$ .*

*Demostración.* Sea  $uv$  una arista de  $T$  y supongamos que está contenida en la bisectriz de dos puntos  $p_0, p_1 \in P$ . Sean  $\alpha, \beta$  dos puntos sobre el segmento  $[u, v]$ , es claro que  $\rho(\alpha) = d(\alpha, p_0) = d(\alpha, p_1)$  pues  $uv$  es la frontera entre las regiones  $R(p_0)$  y  $R(p_1)$ , análogamente  $\rho(\beta) = d(\beta, p_0)$ . Note que los triángulos  $\Delta(p_0 p_1 \alpha), \Delta(p_0 p_1 \beta)$  son isósceles y comparten la misma base. Por ende si  $\alpha$  está más lejos de la base del triángulo que  $\beta$ , entonces  $\rho(\alpha) > \rho(\beta)$ . Por la Proposición 1.3.5, sabemos que  $c_P$  no pertenece al interior de ninguna arista de  $T$ , por consiguiente  $\rho$  es una función monótona sobre los puntos de la arista  $uv$ . Repitiendo el argumento recursivamente para todas las aristas de  $T$  se sigue el resultado.  $\square$

### 1.3.1. Calculando el $P$ -círculo de radio mínimo

Concluimos este capítulo presentando un algoritmo de tiempo  $O(n)$  para encontrar al punto  $c_P$  una vez calculado el diagrama de Voronoi del punto más

lejano de  $P$ . Existe otro algoritmo propuesto por Megiddo [25], que encuentra al punto  $c_P$  en tiempo lineal, sin la necesidad de calcular el diagrama de Voronoi. Sin embargo, el objetivo de este capítulo es mostrar la relación que existe entre el diagrama de Voronoi y los  $P$ -círculos de radio mínimo.

Note que como  $C_P$  pasa por al menos dos puntos de  $P$ , entonces  $c_P$  yace sobre la bisectriz de dos puntos de  $P$ . Podríamos pensar ingenuamente, que con un algoritmo que busque a  $c_P$  en cada una de las bisectrices definidas por pares de puntos de  $P$  nos es suficiente. Sin embargo el número de bisectrices a revisar es exactamente  $\binom{n}{2}$ , por lo que nuestro algoritmo terminaría en al menos tiempo  $O(n^2)$ ; veremos que podemos hacerlo mucho mejor. Pensemos a las aristas del árbol  $\mathcal{V}(P)$  como segmentos cerrados, por ende, gracias a la Proposición 1.3.2,  $c_P$  yace sobre una arista de  $\mathcal{V}(P)$ . Nuestro algoritmo revisa cada arista del árbol y determina si  $c_P$  pertenece o no a esa arista como sigue. Sea  $uv$  una arista de  $\mathcal{V}(P)$  contenida en la bisectriz de  $p_0, p_1$ , determinamos en tiempo constante si  $[u, v] \cap [p_0, p_1] \neq \emptyset$ , en cuyo caso por la Proposición 1.3.5, concluimos que  $c_P = [u, v] \cap [p_0, p_1]$ . El algoritmo anterior termina en tiempo lineal ya que por el Lema 1.2.5, el árbol  $\mathcal{V}(P)$  tiene  $O(n)$  aristas.

## Capítulo 2

# $P$ -círculo de radio mínimo con centro en una línea de consulta

### 2.1. Nuestro problema

Sea  $P$  un conjunto de  $n$  puntos en el plano representando a  $n$  sensores y sea  $L$  una línea recta representando la vía de un tren. Consideremos una antena inalámbrica  $\mathcal{A}$  montada sobre un tren que viaja a lo largo de la vía representada por  $L$ . El objetivo de la antena es emitir un mensaje durante el trayecto en el tren, de manera que todos los sensores lo reciban.

Por motivos de seguridad, la antena  $\mathcal{A}$  puede enviar un único mensaje durante el recorrido del tren, más aún buscaremos la potencia óptima de transmisión, no sólo para minimizar los costos, sino que además no queremos que el mensaje se difunda a una gran distancia, ya que podría ser interceptado y se podría hacer un mal uso de la información.

Nuestro objetivo será entonces el de encontrar el momento justo durante el recorrido para emitir el mensaje, de manera que cada uno de los  $n$  sensores reciba el mensaje y la potencia utilizada sea la mínima posible. Recordemos que la potencia asignada a nuestra antena define un radio de transmisión  $r$  alrededor de la posición de  $\mathcal{A}$ , por ende al momento de la transmisión, todo sensor deberá estar a distancia a lo más  $r$  de  $\mathcal{A}$  para recibir el mensaje.

Recordemos que nuestra antena, junto con su radio de transmisión, se puede representar geoméricamente por un círculo con centro en la posición de  $\mathcal{A}$  y radio  $r$ . Por ende la solución a nuestro problema se puede ver geoméricamente como el  $P$ -círculo de radio mínimo con centro sobre la recta  $L$ , ya que éste determina el lugar y la potencia óptimas de transmisión de la antena  $\mathcal{A}$ ; ver Figura 2.1.



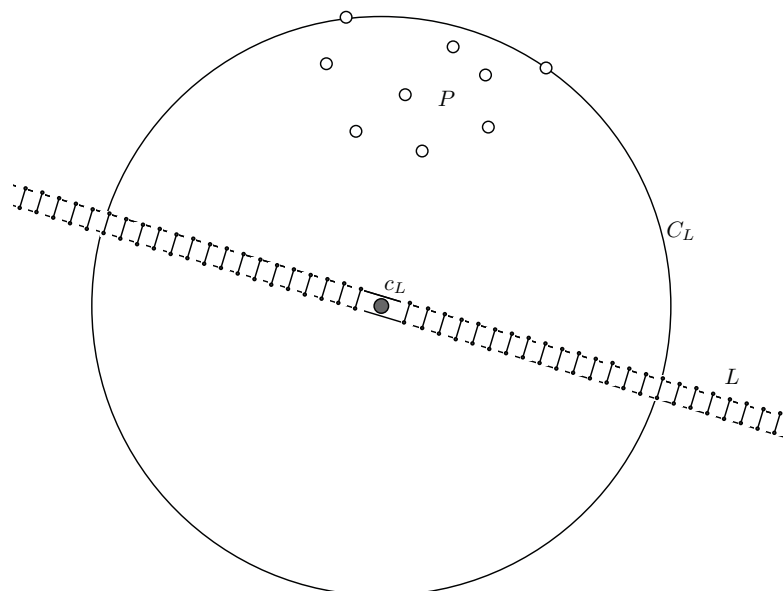


Figura 2.1: El círculo  $C_L$  y su centro  $c_L$  sobre la línea de consulta  $L$ .

Dada una línea recta  $L$ , definimos a  $C_L$  como el  $P$ -círculo de radio mínimo con centro sobre la recta  $L$  y a  $c_L$  como su centro; nuestro problema se traduce entonces en encontrar el centro y radio de  $C_L$  de manera eficiente.

En este capítulo presentaremos una caracterización del punto  $c_L$  y mostraremos la estrecha relación que guarda con el diagrama de Voronoi del punto más lejano de  $P$ . Esto nos permitirá generar un algoritmo de tiempo lineal para determinar la ubicación de  $c_L$  una vez calculado el diagrama de Voronoi. Finalmente, consideraremos una variante del problema, en dónde el conjunto de puntos  $P$  está fijo mientras la recta  $L$  varía. Propondremos entonces una estructura de datos sobre  $\mathcal{V}(P)$  y un algoritmo que nos permitirán encontrar eficientemente al punto  $c_L$  dada cualquier línea recta  $L$ .

## 2.2. El $P$ -círculo de radio mínimo con centro en $L$

Sea  $P$  un conjunto de  $n$  puntos en el plano, tales que no existen cuatro puntos cocirculares, y sea  $L$  un línea recta. Siendo que nuestro objetivo es encontrar el  $P$ -círculo de radio mínimo con centro sobre la recta  $L$ , podemos suponer que  $P$  es un conjunto de puntos en posición convexa, ya que cualquier

círculo que contenga al cierre convexo de  $P$ , contiene también a todos los puntos interiores del conjunto.

**Definición 4.** *Dado un punto  $p$  de  $P$ , denotamos por  $p_{\perp}$  al punto definido por la proyección ortogonal de  $p$  sobre la recta  $L$ .*

Sea  $p_L$  el punto de  $P$  más lejano a la recta  $L$ . La siguiente observación caracteriza las tres ubicaciones posibles de  $c_L$  en relación al diagrama de Voronoi del punto más lejano y se puede observar a detalle en la Figura 2.2.

**Proposición 2.2.1.** *Se cumple uno de los siguientes casos:*

1.  $C_L$  pasa por un único punto  $p$  de  $P$ , si y sólo si  $p = p_L$  y  $p_{\perp}$  está en el interior de  $R(p_L)$ , en cuyo caso  $c_L = p_{\perp}$  es el centro de  $C_L$ .
2.  $C_L$  pasa por dos puntos  $p_0, p_1$  de  $P$ , si y sólo si  $c_L$  es la intersección de  $L$  con una arista  $e$  de  $\mathcal{V}(P)$ , y  $e$  está contenida en la bisectriz de  $p_0, p_1$ .
3.  $C_L$  pasa por tres puntos de  $P$ , si y sólo si  $c_L$  es un vértice de  $\mathcal{V}(P)$  que se encuentra sobre la recta  $L$ .

*Demostración.* El inciso (2), (3) y el el regreso del inciso (1) se siguen de la definición del diagrama de Voronoi del punto más lejano. Sin embargo, la ida del inciso (1) no es trivial y la prueba se muestra a continuación.

Si  $C_L$  pasa por un único punto  $p$  de  $P$ , por definición  $c_L$  pertenece al interior de la región de Voronoi  $R(p)$ . Procedamos por contradicción y supongamos que  $c_L \neq p_{\perp}$ . Por consiguiente el triángulo rectángulo  $\Delta(pp_{\perp}c_L)$  tiene como hipotenusa al segmento  $[p, c_L]$  que es un radio de  $C_L$ . Ahora, como  $c_L$  está en el interior de  $R(p)$ , existe una  $\varepsilon > 0$ , tal que si movemos a  $c_L$  una  $\varepsilon$  distancia sobre  $L$  en la dirección de  $p_{\perp}$ , tendremos que  $c_L$  sigue perteneciendo a  $R(p)$  y la hipotenusa del triángulo  $\Delta(pp_{\perp}c_L)$  se redujo. Con esto encontraríamos un  $P$ -círculo de radio menor que el de  $C_L$  con centro en la recta  $L$  lo cual es una contradicción. Por lo tanto, si  $c_L$  pertenece a  $R(p)$ ,  $c_L$  es la proyección ortogonal de  $p$  sobre  $L$ . Es decir que  $C_L$  es el círculo con centro en  $p_{\perp}$  y radio  $d(p, L)$ . Es fácil ver ahora que  $p$  tiene que ser  $p_L$ , ya que  $d(p_L, L) \geq d(q, L)$  para todo  $q \in P$ .

□

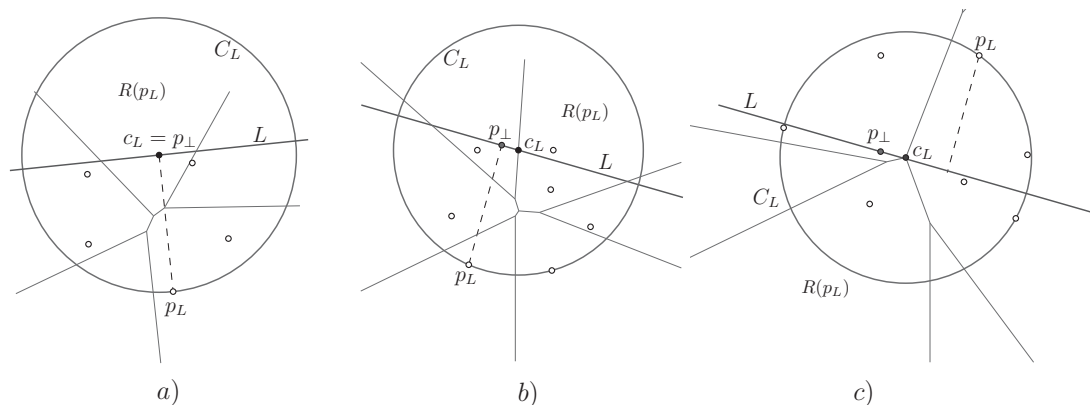


Figura 2.2: Los tres casos de la Proposición 2.2.1. Se puede observar que la proyección ortogonal de  $p_L$  sobre  $L$  no siempre yace en el interior de  $R(p_L)$ .

La proposición siguiente nos da un criterio para determinar la ubicación de  $c_L$ .

**Proposición 2.2.2.** *Sea  $p$  un punto de  $P$  y sea  $p_\perp$  la proyección ortogonal de  $p$  sobre la línea recta  $L$ . Si  $p_\perp \in R(p)$ , entonces  $c_L = p_\perp$ .*

*Demostración.* Si  $p_\perp \in R(p)$ , entonces  $C(p_\perp)$  es un  $P$ -círculo tal que su circunferencia pasa por  $p$ . Si tomamos cualquier otro punto  $x \in L$ , como el triángulo  $\Delta xp_\perp p$  es rectángulo con hipotenusa  $[x, p]$ , entonces tenemos que  $d(x, p) > d(p_\perp, p)$ . Por lo tanto  $\rho(x) \geq d(x, p) \geq d(p_\perp, p) = \rho(p_\perp)$ , para todo  $x \in L$ . Concluimos entonces que entre los puntos de  $L$ ,  $\rho$  alcanza su mínimo en  $p_\perp$  por lo que  $c_L = p_\perp$ ; la proposición se observa en la Figura 2.3.  $\square$

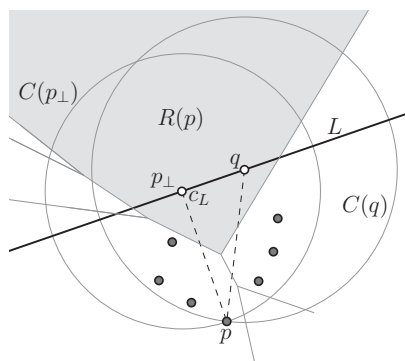


Figura 2.3: El punto  $p_\perp$  es el mínimo de la función  $\rho$  sobre la recta  $L$ .

Presentamos a continuación una observación que será utilizada de manera recurrente a lo largo de este trabajo.

**Observación 2.2.3.** *Si  $p$  es un punto en el cierre convexo de  $P$ , entonces toda semirrecta con ápice  $p$  corta en a lo más un punto a la frontera de  $R(p)$ .*

*Demostración.* Sean  $\ell_1, \ell_2$  las bisectrices de los dos segmentos que inciden en  $p$  en el cierre convexo de  $P$ , note que  $\ell_1$  y  $\ell_2$  contienen a las dos semirrectas que forman parte de la frontera de  $R(p)$ . Además,  $\ell_1, \ell_2$  forman una doble cuña en el plano, dejando a  $p$  en un cuadrante y a  $R(p)$  en el opuesto. Por lo tanto, toda semirrecta con ápice  $p$  corta en a lo más un punto a  $\ell_i$ ,  $i = 1, 2$ , esto aunado a la convexidad de  $R(p)$  nos permite concluir nuestro resultado; ver Figura 2.4. □

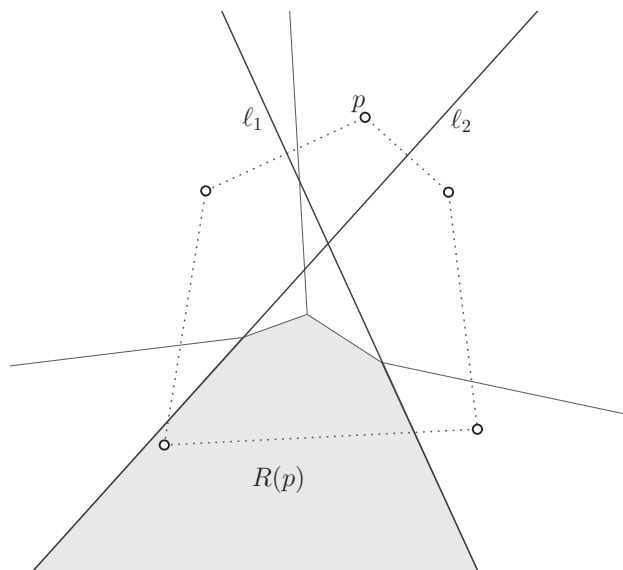


Figura 2.4: La prueba de la observación 2.2.3.

### 2.2.1. Calculando el $P$ -círculo de radio mínimo con centro sobre una línea recta

Sea  $P$  un conjunto de  $n$  puntos en el plano, sea  $L$  una línea recta y sea  $p_L$  el punto de  $P$  más lejano a la recta  $L$ . Propondremos un algoritmo de tiempo lineal para encontrar la posición de  $c_L$  una vez calculado el diagrama de Voronoi del punto más lejano de  $P$ . Cabe resaltar que existe un algoritmo de tiempo lineal propuesto por Megiddo en [25], para encontrar al punto  $c_L$  sin

la necesidad de calcular el diagrama de Voronoi. Sin embargo, las variantes del problema que introduciremos en lo sucesivo utilizarán al diagrama de Voronoi como parte central. Por consiguiente el algoritmo aquí propuesto, nos dará mucha intuición y herramientas que usaremos más adelante.

Como primer paso del algoritmo construimos el diagrama de Voronoi del punto más lejano de  $P$ . Si los puntos del conjunto  $P$  están en posición general, podemos construir a  $\mathcal{V}(P)$  en tiempo  $O(n \log n)$ . Sin embargo, si conocemos el cierre convexo de  $P$ , entonces podemos construir a  $\mathcal{V}(P)$  en tiempo  $O(n)$  [1]. Sin pérdida de generalidad, supondremos de aquí en adelante que los puntos de  $P$  están en posición convexa, ya que en el proceso de construcción del diagrama de Voronoi obtenemos el cierre convexo de  $P$ .

Recordemos que por la Proposición 2.2.1, tenemos tres casos posibles para la ubicación de  $c_L$  en relación a  $\mathcal{V}(P)$ . En el primero  $c_L$  pertenece al interior de la región de Voronoi de  $p_L$ , mientras que en el segundo y tercero  $c_L$  yace sobre una arista del árbol  $\mathcal{V}(P)$ . El algoritmo propuesto consta de dos fases. En la primera llevamos a cabo un proceso de evaluación de tiempo  $O(\log n)$  para determinar si  $c_L$  pertenece al interior de la región de Voronoi de  $p_L$ . Si  $c_L$  pertenece a  $R(p_L)$ , entonces gracias a la Proposición 2.2.1 podemos garantizar que el punto  $c_L$  está definido por la proyección ortogonal de  $p_L$  sobre la recta  $L$ . De lo contrario, en una segunda fase, el algoritmo revisará cada arista de  $\mathcal{V}(P)$  hasta encontrar aquella que contenga a  $c_L$ .

El proceso de evaluación de la primera fase se describe detalladamente a continuación.

**Algoritmo 2.2.1.** *Dado un punto  $p \in P$ . Algoritmo para determinar si  $c_L$  pertenece o no al interior de la región  $R(p)$ .*

Recordemos que  $p_\perp$  denota el punto definido por la proyección ortogonal de  $p$  sobre la recta  $L$ . Revisaremos si  $p_\perp$  se encuentra dentro de  $R(p)$ , ya que de ser así, la Proposición 2.2.2 nos garantiza que  $c_L = p_\perp$ .

Sea  $L_\perp$  la línea recta que contiene a los puntos  $p, p_\perp$  y sea  $s$  el punto de intersección de la frontera de  $R(p)$  con  $L_\perp$ , si éste existe. Note que podemos calcular a  $s$  en tiempo  $O(\log n)$  ya que  $R(p)$  es un polígono convexo, note también que  $p_\perp$  está en el interior de  $R(p)$ , si y sólo si el punto  $s$  está entre  $p$  y  $p_\perp$  sobre la recta  $L_\perp$ .

Si  $p_\perp$  yace dentro de  $R(p)$ , entonces reportamos que  $c_L = p_\perp$ , de lo contrario concluimos que  $c_L$  no pertenece a  $R(p)$ .

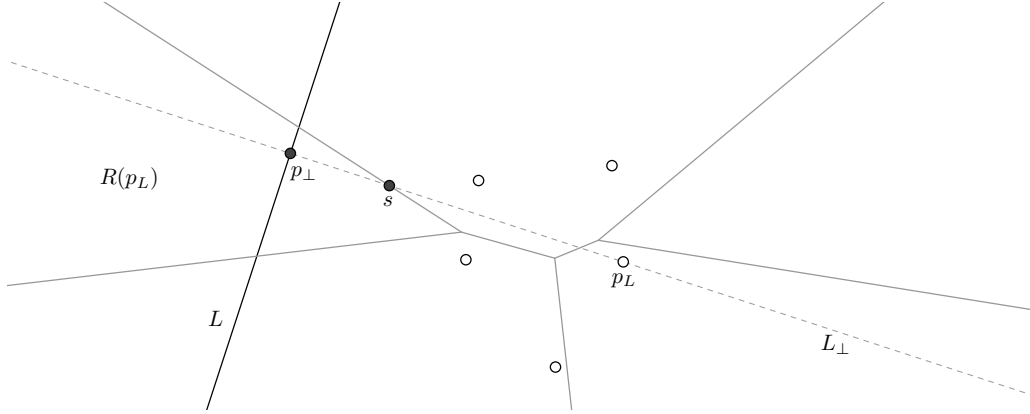


Figura 2.5: El accionar del Algoritmo 2.2.1 sobre el punto  $p_L$ .

Note que podemos encontrar a  $p_L$  en tiempo logarítmico llevando a cabo una búsqueda binaria sobre los vértices del cierre convexo de  $P$ , por lo que podemos aplicar el Algoritmo 2.2.1 a  $p_L$  y obtener en tiempo logarítmico uno de los dos siguientes casos.

Si encontramos a  $c_L$  dentro de  $R(p_L)$ , entonces terminamos y reportamos al círculo con centro en  $c_L$  y radio  $d(c_L, p_L)$ . De lo contrario  $c_L$  pertenece a una arista de  $\mathcal{V}(P)$  y como el Lema 1.2.5 garantiza que el número de intersecciones de  $\mathcal{V}(P)$  con la recta  $L$  es lineal, podemos revisar cada intersección y quedarnos con aquella que tenga el mínimo valor bajo la función  $\rho$ . El algoritmo se explica a detalle a continuación.

Sea  $e$  una arista de  $\mathcal{V}(P)$  contenida en la bisectriz de los puntos  $p_0, p_1$  de  $P$  y sea  $x_e$  la intersección de  $e$  con la recta  $L$  si ésta existe. Como supusimos que  $e$  tiene apuntadores tanto a  $p_0$  como a  $p_1$ , entonces en tiempo constante podemos determinar el valor de  $\rho(x_e) = d(x_e, p_0) = d(x_e, p_1)$ .

Ejecutando esta rutina de tiempo  $O(1)$  en cada arista de  $\mathcal{V}(P)$ , podemos encontrar una arista  $a$  de  $\mathcal{V}(P)$ , tal que  $\rho(x_a) = \min\{\rho(x_e) : e \in \mathcal{V}(P)\}$ . Claramente  $c_L = x_a$ , por lo que el algoritmo termina en tiempo  $O(n)$  después de haber calculado el diagrama de Voronoi del punto más lejano de  $P$ .

## 2.3. Encontrando a $C_L$ mediante consultas dinámicas

Sea  $P$  un conjunto de  $n$  puntos en el plano. Proponemos una nueva variante al problema en la cual nos permitimos realizar un preprocesado sobre el conjunto  $P$ , de manera que podamos responder a la siguiente consulta eficientemente:

Dada una línea recta  $L$ , encontrar el  $P$ -círculo de radio mínimo con centro sobre la recta  $L$ .

Note que la diferencia principal con el caso estático mostrado en la sección anterior, es que ahora  $P$  es un conjunto de puntos fijo mientras que  $L$  varía en cada consulta. Podemos encontrar varios artículos en la literatura que abordan este problema, el primero fue presentado por Das, Karmakar, Nandy y Roy, quienes propusieron un algoritmo con tiempo de preprocesado  $O(n \log n)$  y espacio lineal que responde a las consultas en tiempo  $O(\log^2 n)$  [32]. Posteriormente Das, Karmakar y Roy mejoraron el tiempo de la consulta al óptimo  $O(\log n)$ , sin embargo aumentaron el espacio y preprocesado a  $O(n^2)$  [21]. En este capítulo presentamos un nuevo algoritmo para responder a la consulta en el tiempo óptimo  $O(\log n)$ , utilizando un preprocesado de tiempo  $O(n \log n)$  y espacio  $O(n)$ . Aunque cabe mencionar que Bose, Langerman y Roy estudiaron el mismo problema en [30]. Ellos propusieron un algoritmo y estructura de datos completamente distintos, obteniendo el mismo tiempo de preprocesado y de consulta que nosotros.

### 2.3.1. Esquema del algoritmo

Retomando la construcción del punto  $p_L$  y el accionar del Algoritmo 2.2.1 descrito en la sección anterior, podemos determinar, en tiempo  $O(\log n)$ , si  $c_L$  pertenece al interior de la región de Voronoi de  $p_L$  o yace en una de las aristas de  $\mathcal{V}(P)$ . Como en el primer caso el Algoritmo 2.2.1 resuelve el problema al reportar al punto  $c_L$ , supondremos de aquí en adelante que  $c_L$  yace sobre una arista del árbol  $\mathcal{V}(P)$ . Nuestro objetivo es entonces el de encontrar esta arista, sin tener que probar todas las aristas de  $\mathcal{V}(P)$ .

Dada una línea recta  $K$ , sea  $p_K$  el punto de  $P$  más lejano a  $K$  y sea  $c_K$  el centro del  $P$ -círculo de radio mínimo con centro sobre la recta  $K$ . Definimos también a  $p^*$  como el punto de  $P$  más lejano a  $L$ , que se encuentre en el semiplano inducido por  $L$  que contiene a  $c_P$ ; ver Figura 2.6. Note que  $p^*$  no necesariamente es el punto de  $P$  más lejano a  $L$  como se observa en la figura, aunque en muchos de los casos  $p_L$  y  $p^*$  coincidirán.

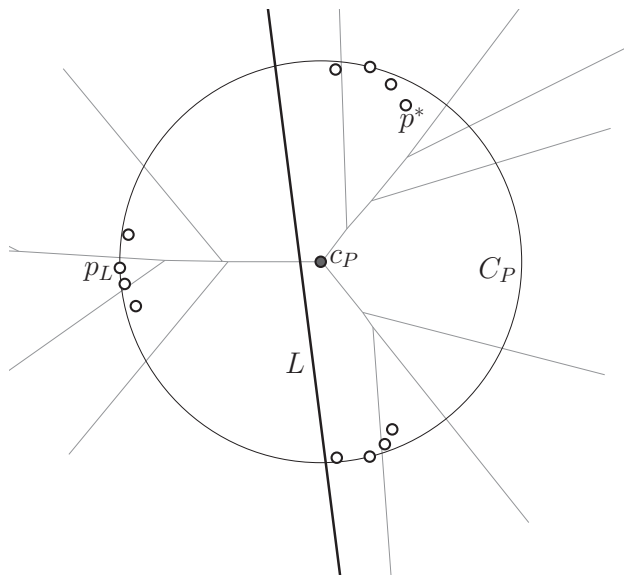


Figura 2.6: Podemos ver que  $p^*$  puede ser diferente de  $p_L$  si  $c_P$  y  $p_L$  están en distintos semiplanos inducidos por  $L$ .

Los pasos a seguir por nuestro algoritmo se describen a grandes rasgos a continuación.

1. Primero encontraremos una línea recta  $M$  paralela a  $L$ , tal que  $c_M \in R(p^*)$ ; ver Figura 2.7. Además pediremos que  $c_M$  y  $c_P$  se encuentren en distintos semiplanos inducidos por  $L$ .
2. Como  $c_M \in R(p^*)$ , usando el Algoritmo 2.2.1, podemos calcular la posición del punto  $c_M$  en tiempo logarítmico.
3. Finalmente tomaremos una línea recta  $K$  paralela a  $L$ , y la trasladaremos continuamente desde la posición de la recta  $M$  hasta la posición de la recta  $L$ , sin perder la ubicación del punto  $c_K$  en el proceso.

En la siguiente sección daremos las ideas y justificaciones formales detrás de la intuición que acabamos de presentar.



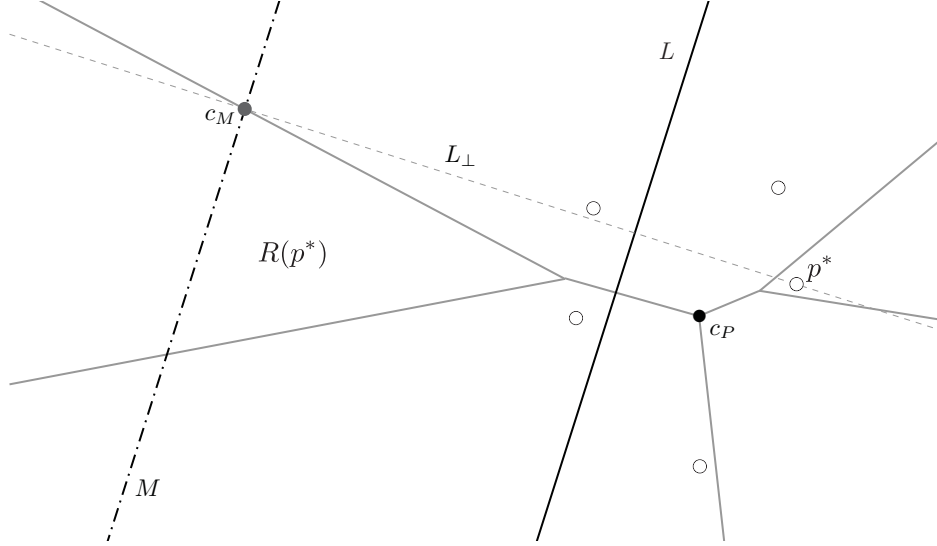


Figura 2.7: Podemos ver la existencia una recta  $M$ , paralela a  $L$ , suficientemente lejos de  $p^*$ , para la cual  $c_M$  es la proyección ortogonal de  $p^*$  sobre  $M$ .

### 2.3.2. $c_L$ y las trayectorias sobre el árbol $\mathcal{V}(P)$

Sea  $P$  un conjunto de  $n$  puntos en el plano y sea  $L$  una línea recta.

**Proposición 2.3.1.** *Si  $p^*$  es el punto de  $P$  más lejano a  $L$  en el semiplano inducido por  $L$  que contiene a  $c_P$ , entonces existe una línea recta  $M$  paralela a  $L$ , tal que  $c_M$  es la proyección ortogonal de  $p^*$  sobre  $M$ ,  $c_M \in R(p^*)$  y  $d(M, p^*) \geq d(L, p^*)$ .*

*Demostración.* Sea  $L_\perp$  la línea recta perpendicular a  $L$  que pasa por  $p^*$  y sea  $\lambda$  la semirrecta contenida en  $L_\perp$  con ápice  $p^*$  que corta a  $L$ ; ver Figura 2.7.

Definimos la función  $c_\lambda : \mathbb{R}^+ \rightarrow \lambda$ , tal que  $c_\lambda(x)$  es el punto sobre  $\lambda$  a distancia  $x$  de  $p^*$ . Para todo  $x \in \mathbb{R}^+$ , denotamos por  $C_\lambda(x)$  al círculo con centro en  $c_\lambda(x)$  y radio  $x$ . Es fácil ver que si  $x$  tiende a infinito, entonces  $C_\lambda(x)$  tiende a una línea recta  $\ell$  paralela a  $L$  que contiene a  $p^*$ . Más aún,  $\ell$  es una recta soporte del cierre convexo de  $P$ , por ende podemos ver a  $\ell$  como un  $P$ -círculo con centro en el infinito.

Concluimos entonces que debe existir  $x_0 > 0$  tal que  $C_\lambda(x_0)$  sea un  $P$ -círculo, por lo tanto, como  $p^*$  está en la circunferencia de  $C_\lambda(x_0)$ , tenemos que  $c_\lambda(x_0) \in R(p^*)$ . Ahora bien, si definimos a  $M$  como la recta paralela a  $L$  que pasa por  $c_\lambda(x_0)$ , entonces  $c_M = c_\lambda(x_0)$  por la Proposición 2.2.2; ver Figura 2.7. Además, es claro por construcción que  $d(M, p^*) \geq d(L, p^*)$ .  $\square$

La siguiente definición nos permitirá trabajar con las trayectorias dentro del árbol  $\mathcal{V}(P)$  de manera más intuitiva.

**Definición 5.** *Sea  $x$  un punto sobre una arista de  $\mathcal{V}(P)$ . Denotamos por  $\mathcal{T}_x$  a la trayectoria contenida en  $\mathcal{V}(P)$  que une a  $c_P$  con el punto  $x$ .*

Sea  $M$  la línea recta paralela a  $L$  existente como consecuencia de la Proposición 2.3.1, y supongamos sin pérdida de generalidad que  $M$  es la línea recta más cercana a  $p^*$  con estas propiedades. Es claro entonces que  $c_M$  es un punto sobre la frontera de  $R(p^*)$ ; ver Figura 2.7. La idea es entonces regresar la recta  $M$  hasta  $L$  sin perder la ubicación de  $c_M$  en el proceso, para lo cual encontraremos útil el siguiente teorema.

**Lema 2.3.2.** *Si  $K$  es una recta paralela a  $L$  tal que*

$$d(M, p^*) \geq d(K, p^*) \geq d(L, p^*),$$

*entonces  $c_K$  es un punto sobre una arista de  $\mathcal{T}_{c_M}$ .*

*Demostración.* Denotaremos por  $\tau$  a la curva continua en el plano, formada por la unión de las aristas de la trayectoria  $\mathcal{T}_{c_M}$ . Note que si movemos continuamente a la recta  $K$ , desde  $M$  hasta  $L$ , sin modificar su pendiente, entonces el punto  $c_K$  se mueve describiendo una curva continua  $\gamma$ . Afirmamos que  $\gamma$  está contenida en la curva  $\tau$ .

Procedemos por contradicción. Supongamos que existe una línea recta  $K_0$ , tal que  $d(M, p^*) \geq d(K_0, p^*) \geq d(L, p^*)$  y  $c_{K_0} \notin \tau$ . Como  $c_M \in \gamma \cap \tau$ , entonces podemos definir a  $K'$  como la paralela a  $L$  más cercana a  $p^*$ , tal que  $c_{K'} \in \gamma \cap \tau$  y  $d(M, p^*) \geq d(K', p^*) > d(K_0, p^*)$ .

Sea  $\varepsilon > 0$  y sea  $K_\varepsilon$  la línea recta que representa a  $K'$  desplazada hacia  $p^*$  una  $\varepsilon$  distancia. Es claro que  $c_{K_\varepsilon} \notin \tau$ , por consiguiente está en el interior de una región de Voronoi  $R(q)$ , para algún punto  $q$  de  $P$ . Sin embargo, por el inciso (1) de la Proposición 2.2.1, tenemos que esto ocurre si y sólo si  $q$  es el punto más lejano de  $K_\varepsilon$ . Por lo tanto forzosamente  $q = p^*$ ; más aún,  $c_{K_\varepsilon}$  tiene que ser la proyección de  $p^*$  sobre  $K_\varepsilon$  y  $c_{K_\varepsilon} \in R(p^*)$ . Note que además  $d(K_\varepsilon, p^*) < d(M, p^*)$ , lo cual es una contradicción, pues  $M$  era la recta con estas propiedades más cercana a  $p^*$ .

Por lo tanto, para toda línea recta  $K$  paralela a  $L$ , tal que

$$d(M, c) \geq d(K, c) \geq d(L, c),$$

se cumple que  $c_K \in \tau$  y por ende  $c_K$  pertenece a una arista de  $\mathcal{T}_{c_M}$ . □

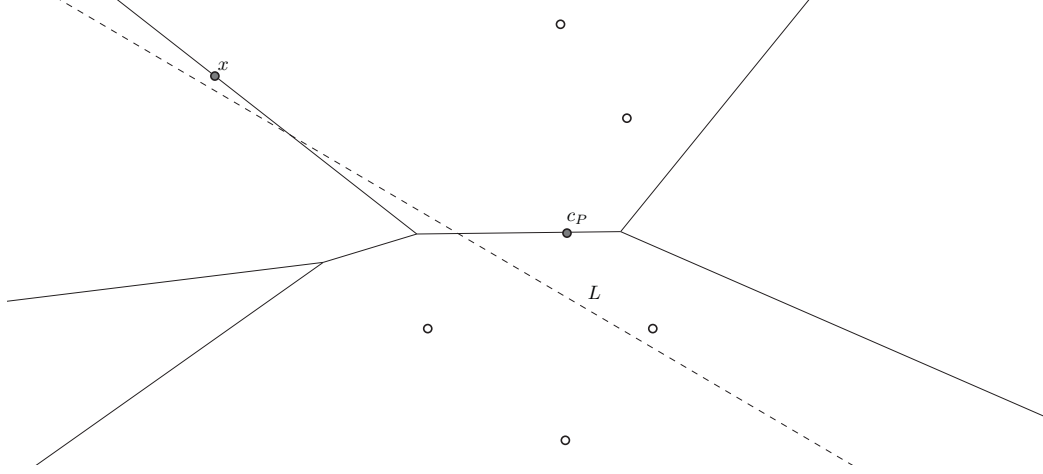


Figura 2.8: La trayectoria  $\mathcal{T}_x$  corta en dos puntos a la recta  $L$ .

Sea  $x$  un punto sobre una arista de  $\mathcal{V}(P)$ . En la Figura 2.8 podemos observar como la recta  $L$  puede tener varias intersecciones con  $\mathcal{T}_x$ , sin embargo, gracias al siguiente resultado podemos garantizar que la intersección de  $\mathcal{T}_{c_M}$  con  $L$  es única, lo cual será determinante en el algoritmo que propondremos más adelante.

**Lema 2.3.3.** *La trayectoria  $\mathcal{T}_{c_M}$  corta en un único punto a la recta  $L$ .*

*Demostración.* Denotaremos por  $\tau$  a la curva continua en el plano, formada por la unión de las aristas de la trayectoria  $\mathcal{T}_{c_M}$ . Recordemos que si trasladamos a la recta  $M$  hacia  $p^*$  de manera continua, entonces  $c_M$  describe una curva continua  $\gamma$  en el plano.

Procedemos por contradicción, supongamos que  $L$  corta a  $\tau$  en al menos dos puntos. Por construcción sabemos que  $M$  corta en un sólo punto a  $\tau$ , por ende, entre  $L$  y  $M$  podemos definir una línea recta  $K$  paralela a  $L$ , tal que  $K$  sea la paralela más lejana a  $p^*$  que corta en al menos dos puntos a  $\tau$ . Note que cualquier recta  $N$  paralela a  $L$  tal que  $d(M, p^*) \geq d(N, p^*) > d(K, p^*)$ , corta en un sólo punto a  $\tau$  y dicha intersección define al punto  $c_N$ ; ver Figura 2.9.

Sean  $x_0, \dots, x_m$  los puntos de intersección de  $K$  con  $\tau$ , y supongamos que están ordenados de mayor a menor con respecto a su profundidad en el árbol  $\mathcal{V}(P)$ . Recordemos que por el Teorema 1.3.6,  $c_K = x_m$ , pues  $\rho(x_m) \leq \rho(x_i)$ ,  $i \in \{0, \dots, m\}$ . Sin embargo esto representa una discontinuidad de la curva  $\gamma$ , pues  $x_0$  es un punto de acumulación de  $\gamma$ . Contradicción que viene de suponer que  $L$  corta a  $\tau$  en más de un punto.  $\square$

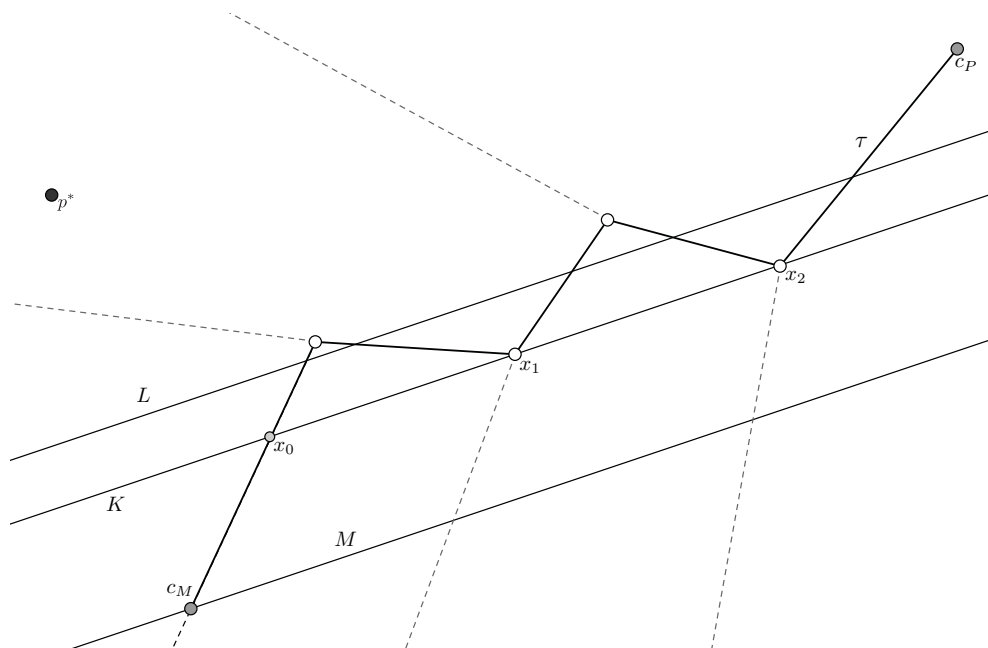


Figura 2.9: La prueba del Lema 2.3.3.

**Teorema 2.3.4.** *Dada una línea recta  $L$ , si  $M$  es la paralela a  $L$  más cercana a  $p^*$ , existente como consecuencia de la Proposición 2.3.1, entonces  $c_L$  es el único punto de intersección entre la trayectoria  $\mathcal{T}_{c_M}$  y la recta  $L$ .*

*Demostración.* Por definición, sabemos que  $p^*$  y  $c_P$  se encuentran en el mismo semiplano definido por  $L$ , por ende al construir a la recta  $M$  alejándonos del punto  $p^*$ , garantizamos que la recta  $M$  y el punto  $c_P$  se encuentran en distintos semiplanos definidos por  $L$ . Por consiguiente la trayectoria  $\mathcal{T}_{c_M}$  corta a la recta  $L$  y lo hace en un único punto gracias al Lema 2.3.3. Más aún, el Lemma 2.3.2 garantiza que  $c_L$  es exactamente el punto de intersección entre la trayectoria  $\mathcal{T}_{c_M}$  y la recta  $L$ .  $\square$

### 2.3.3. Estructura de datos sobre $\mathcal{V}(P)$

Gracias al Teorema 2.3.4, podemos garantizar que el punto  $c_L$  está sobre una arista de  $\mathcal{T}_{c_M}$ . Sin embargo esta trayectoria podría ser, en principio, cualquier trayectoria contenida en  $\mathcal{V}(P)$ . Nuestro objetivo será entonces desarrollar una estructura de datos, que nos permita hacer búsquedas eficientes sobre las aristas de cualquier trayectoria en el árbol  $\mathcal{V}(P)$ .

Describiremos una estructura de datos simple que requiere de tiempo y espacio  $O(n \log n)$  para su construcción, y que permite realizar búsquedas binarias sobre cualquier trayectoria del árbol  $\mathcal{V}(P)$  en tiempo  $O(\log n)$ . Sin embargo, cabe destacar que existe otra estructura de datos más ingeniosa y compleja, propuesta por Roy, Karmakar, Das y Nandy [32], que requiere de tiempo y espacio  $O(n)$  para su construcción, y que permite realizar búsquedas en el mismo tiempo de ejecución.

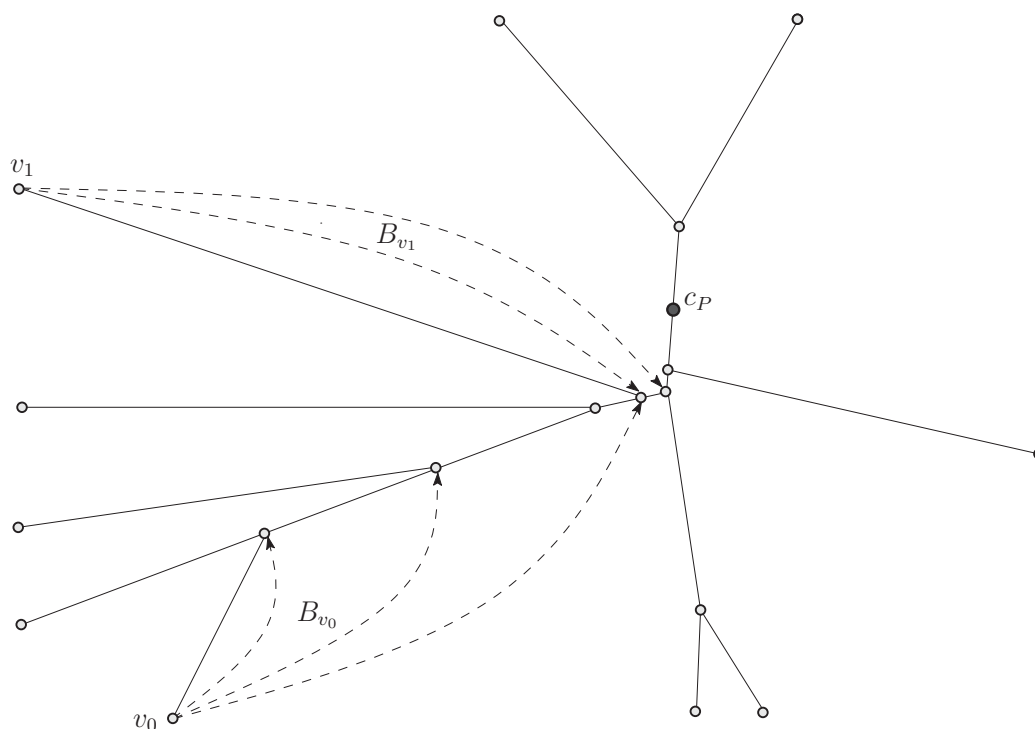


Figura 2.10: La sub-estructura para los vértices  $v_0, v_1$ .

Para la construcción de la estructura de datos, supongamos que el árbol  $\mathcal{V}(P)$  ya ha sido calculado; para mayor detalle revisar el Capítulo 1. Realizaremos una Búsqueda en Profundidad (BP) sobre la gráfica  $\mathcal{V}(P)$  partiendo del vértice raíz  $c_P$ . La única diferencia es que usaremos un arreglo de tamaño

$n$  en vez de una pila durante la BP, haremos esto ya que necesitaremos acceso aleatorio a los elementos de la pila.

Es claro que cuando la BP alcance a un vértice  $v$  a profundidad  $\lambda$ , la pila (emulada por un arreglo) constará exactamente de los  $\lambda$  vértices de la trayectoria  $\mathcal{T}_v$ . En ese momento asociaremos al vértice  $v$  una sub-estructura  $B_v$ , que consiste de un arreglo de tamaño  $\lceil \log \lambda \rceil$  con apuntadores a los elementos de la pila en las posiciones  $\lceil \frac{\lambda}{2} \rceil, \lceil \frac{3\lambda}{4} \rceil, \lceil \frac{7\lambda}{8} \rceil, \dots$ . Note que estos apuntadores corresponden exactamente a los vértices en  $\mathcal{T}_v$  a profundidades  $\lceil \frac{\lambda}{2} \rceil, \lceil \frac{3\lambda}{4} \rceil, \lceil \frac{7\lambda}{8} \rceil, \dots$  en el árbol  $\mathcal{V}(P)$ ; ver Figura 2.10.

La construcción de la sub-estructura  $B_v$  requiere entonces de tiempo y espacio  $O(\log n)$ , por lo que al asociar esta estructura a cada vértice de  $\mathcal{V}(P)$  durante la BP, construiremos una estructura de datos de tamaño  $O(n \log n)$ . Más aún, el tiempo requerido para construir esta estructura, incluyendo el tiempo de construcción del diagrama de Voronoi del punto más lejano es  $O(n \log n)$ .

### 2.3.4. El algoritmo

Sean  $P$  un conjunto de  $n$  puntos en el plano,  $L$  una línea recta y  $p_L$  el punto de  $P$  más lejano a la recta  $L$ . Supongamos que la estructura de datos descrita en la sección anterior ya fue construida.

Si el Algoritmo 2.2.1 aplicado a  $p_L$  encuentra a  $c_L$  dentro de  $R(p_L)$ , entonces terminamos y reportamos al círculo con centro en  $c_L$  y radio  $d(c_L, p_L)$ . De lo contrario sabemos que  $c_L$  pertenece a una arista de  $\mathcal{V}(P)$ .

Sea  $p^*$  el punto de  $P$  más lejano a la línea recta  $L$  en el semiplano definido por  $L$  que contiene a  $c_P$ , y sea  $p_\perp$  la proyección ortogonal de  $p^*$  sobre la recta  $L$ .

Si  $L_\perp$  es la línea recta perpendicular a  $L$  que pasa por  $p^*$ , entonces gracias a la Proposición 2.3.1 podemos garantizar que  $L_\perp \cap R(p^*) \neq \emptyset$ . Sea entonces  $s_0$  el punto de intersección de la frontera de  $R(p^*)$  con  $L_\perp$ , esta intersección es única por la Observación 2.2.3 y se puede calcular en tiempo logarítmico usando una búsqueda binaria.

Supongamos que  $s_0$  yace sobre una arista  $e = uv$  del árbol  $\mathcal{V}(P)$ , tal que la profundidad de  $u$  en el árbol es mayor que la de  $v$ . Consideremos la trayectoria  $\mathcal{T}_u = \{c_P = v_\lambda, \dots, v_2, v_1 = v, v_0 = u\}$ , por el Teorema 2.3.4 podemos garantizar que  $c_L$  yace sobre una arista de  $\mathcal{T}_u$ . Nuestro objetivo entonces será el de hacer una búsqueda binaria por  $c_L$  sobre los vértices de  $\mathcal{T}_u$ . Dicho de otra forma, buscamos el único punto de intersección entre  $\mathcal{T}_u$  y  $L$ . Procedemos de la siguiente manera.

**Algoritmo 2.3.1.** *Algoritmo para encontrar a  $c_L$  sobre la trayectoria  $\mathcal{T}_u$ .*

1. Sea  $x = v_\lambda$  el nodo de partida y sea  $i = 1$ .
2. Si  $x \in L$ , entonces  $c_L = x$  y el algoritmo termina.
3. Sea  $y$  el  $i$ -ésimo elemento de  $B_x$ .
4. Si  $y \in L$ , entonces  $c_L = y$  y el algoritmo termina.
5. Si  $x$  y  $y$  son vértices consecutivos en la trayectoria  $\mathcal{T}_u$ , entonces  $c_L = [x, y] \cap L$  y el algoritmo termina.
6. Determinamos si  $x$  y  $y$  yacen o no del mismo lado de  $L$ .
7. Si yacen en lados distintos, entonces  $i = i + 1$  y regresamos al paso 2.
8. Si yacen del mismo lado, entonces  $x = y$ ,  $i = 1$  y regresamos al paso 2.

**Teorema 2.3.5.** *El algoritmo 2.3.1 termina en tiempo  $O(\log n)$  y encuentra el centro del  $P$ -círculo de radio mínimo con centro sobre la recta  $L$ .*

*Demostración.* Note que en cada ronda del Algoritmo 2.3.1 se descarta una fracción lineal de los vértices en  $\mathcal{T}_u$ , por consiguiente podemos garantizar que el número de rondas en toda ejecución es  $O(\log n)$ . Es claro entonces que nuestro algoritmo termina en tiempo  $O(\log n)$ , ya que todos los pasos requieren de tiempo constante.

El algoritmo anterior realiza una búsqueda binaria sobre los vértices de  $\mathcal{T}_u$ , hasta encontrar ya sea un vértice de  $\mathcal{T}_u$  que esté en  $L$ , en cuyo caso la búsqueda concluye; o dos vértices consecutivos  $x, y$  tales que el segmento que los une corte a la recta  $L$ . En el segundo caso, gracias al Lema 2.3.3, garantizamos que la búsqueda binaria encuentra al único segmento de  $\mathcal{T}_u$  que intersecta a  $L$  y por el Teorema 2.3.4, concluimos que  $c_L = [x, y] \cap L$ .  $\square$

Una vez encontrado  $c_L$  sobre una arista  $e = xy$ , como  $e$  tiene un apuntador a los dos puntos de  $P$  cuyas regiones de Voronoi separa, calcular el radio de  $C_L$  requiere de tiempo constante.

Recapitulando, el preprocesado de nuestro algoritmo utiliza  $O(n \log n)$  tiempo y espacio para generar una estructura de datos sobre  $\mathcal{V}(P)$ . Estructura con la cual, dada cualquier recta  $L$ , podemos encontrar el centro y radio de  $C_L$  en tiempo  $O(\log n)$ . Recordemos que el espacio puede ser mejorado a  $O(n)$  si utilizamos la estructura de datos propuesta por Roy, Karmakar, Das y Nandy en [32].

## Capítulo 3

# Separabilidad por círculos entre polígonos convexos

### 3.1. Nuestro problema

En esta capítulo plantearemos una variante del problema de transmisión de mensajes utilizando antenas inalámbricas.

Sean  $X, Y$  dos conjuntos de puntos en el plano con  $n, m$  puntos respectivamente que representan dos grupos de espías, el primero de aliados y el segundo de enemigos. Sea  $\mathcal{A}$  una antena inalámbrica con la cual queremos enviar un mensaje en una sola transmisión a todos los espías aliados, pero sin que éste sea interceptado por los enemigos. La potencia de la antena  $\mathcal{A}$  define un radio de transmisión  $r$  alrededor del punto de emisión, tal que todo espía a distancia a lo más  $r$  de  $\mathcal{A}$  en el momento de la transmisión recibirá el mensaje.

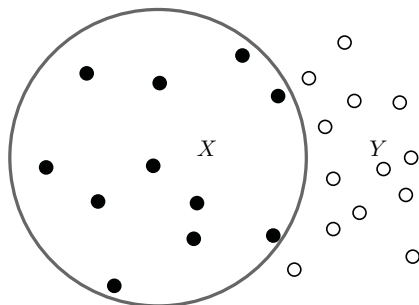


Figura 3.1: Separabilidad por círculos entre conjuntos de puntos.



Nuestro objetivo será encontrar la ubicación y potencia óptimas para la antena  $\mathcal{A}$ , de modo que transmita el mensaje a todos los espías en  $X$ , usando la menor potencia posible, y de manera que ningún espía en  $Y$  reciba el mensaje. Este problema puede verse entonces como un problema de seguridad en el que los mensajes no deben ser interceptados por el enemigo.

Note que la solución al problema, en caso de existir, se puede representar geoméricamente a través de el círculo  $C$  de radio mínimo que contiene a todo punto de  $X$ , pero que no contiene a ningún punto de  $Y$ ; ver Figura 3.1. Dicho de otra forma, podemos referirnos a  $C$  como un círculo que separa a los conjuntos de puntos  $X$  y  $Y$ . Note que  $C$  divide al plano en dos regiones, el interior del círculo y el exterior del mismo, de manera que  $X$  y  $Y$  deberán pertenecer a distintas regiones.

El problema de separabilidad por círculos entre conjuntos de puntos ha sido bien estudiado y es un problema clásico en la Geometría Computacional. La solución óptima para conjuntos discretos fue presentada por Kosaraju, Megiddo y O'Rourke, quienes propusieron un algoritmo óptimo de tiempo lineal, para encontrar el círculo de radio mínimo que separe a dos conjuntos finitos de puntos cualesquiera [28]. Este resultado fue extendido en el mismo artículo para encontrar el círculo separador de radio mínimo entre dos polígonos simples.

En este trabajo nos interesará la siguiente variante del problema de separación por círculos. Sean  $P, Q$  dos polígonos convexos con  $n, m$  vértices respectivamente. Imaginemos que  $P$  representa un conjunto de torretas situadas en el perímetro de defensa de una ciudad, y  $Q$  representa un vehículo enemigo en las inmediaciones de la ciudad.

Buscaremos entonces colocar nuestra antena en la posición óptima y asignarle la menor potencia posible, de manera que toda torreta en el perímetro de defensa reciba los mensajes de la antena, pero del tal forma que en ningún momento el vehículo enemigo  $Q$  pueda interceptar algún mensaje. En este caso la posición óptima de la antena, junto con su potencia, se pueden representar por el círculo de radio mínimo que contiene a  $P$  y tal que su interior no intersecta al polígono  $Q$ ; ver Figura 3.2. Note que en nuestro problema el vehículo representado por el polígono  $Q$  no está fijo, sino que tiene la capacidad de cambiar de posición en las inmediaciones de la ciudad; sin embargo el conjunto  $P$  está fijo, puesto que las torretas no cambian de posición. Nuestro objetivo será encontrar la posición óptima de la antena en relación a la posición de  $Q$  sin importar si éste cambia de posición.

Para facilitar la notación, diremos que  $P$  y  $Q$  son separables por círculos si existe un círculo separador  $C$ , tal que su disco cerrado contiene a  $P$ , y el interior de  $C$  no intersecta al polígono  $Q$ . En este capítulo formalizamos este problema de separabilidad por círculos de la siguiente manera: Sea  $P$  un

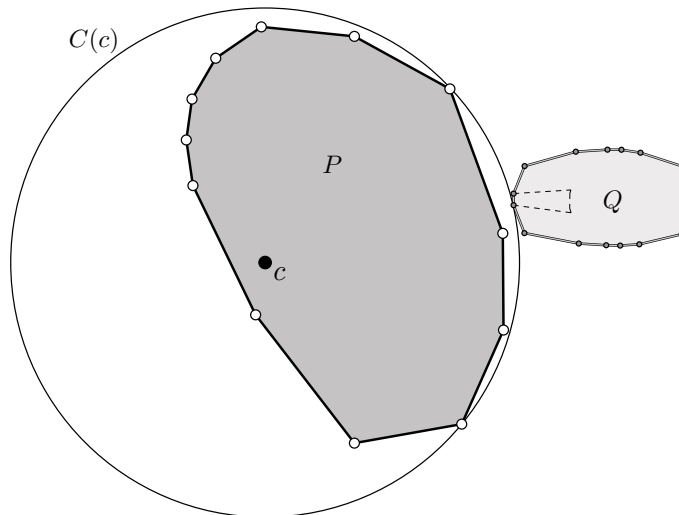


Figura 3.2: Separabilidad por círculos entre polígonos convexos.

polígono convexo con  $n$  vértices, buscaremos preprocesar al polígono  $P$  para obtener una estructura de datos que soporte consultas de la siguiente forma: Dado un polígono convexo  $Q$  como la lista de sus  $m$  vértices, encontrar el círculo separador de radio mínimo de manera eficiente.

Propondremos un algoritmo que realizará un preprocesado de tiempo lineal sobre los vértices de  $P$ , lo que nos permitirá encontrar el círculo separador de radio mínimo en tiempo  $O(\log n \log m)$  dado cualquier polígono convexo  $Q$ .

### 3.2. Círculos separadores

Sean  $P, Q$  dos polígonos convexos, cada uno representado por una lista ordenada de vértices (de acuerdo a su posición sobre el cierre convexo de  $P$  y  $Q$  respectivamente). Para facilitar la explicación, supongamos que los vértices de  $P$  y  $Q$  están en posición general y supongamos también que no hay cuatro puntos cocirculares en los vértices de  $P$ .

**Definición 6.** Diremos que un círculo separa a  $P$  de  $Q$ , si su disco cerrado contiene a  $P$  y su interior no intersecta a  $Q$ . Llamaremos separador, a cualquier círculo que separe a  $P$  de  $Q$ .

Si pensamos a  $P$  como su conjunto de vértices, entonces podemos retomar las definiciones de capítulos anteriores para conjuntos de puntos. En

particular  $\mathcal{V}(P)$  denotará al diagrama de Voronoi del punto más lejano de los vértices de  $P$ , así como un  $P$ -círculo será aquel que contiene a todos los vértices de  $P$ .

Recordemos que  $C_P$  denota al  $P$ -círculo de radio mínimo y  $c_P$  a su centro. Es claro que si  $Q$  y  $C_P$  tienen interiores ajenos, entonces  $C_P$  es un círculo que separa trivialmente a  $P$  de  $Q$ . Es claro también que si los interiores de  $P$  y  $Q$  no son ajenos, entonces no existe un círculo separador, por ende supondremos de aquí en adelante que  $d(P, Q) > 0$ .

Note que todo  $P$ -círculo contenido en un círculo separador es también un círculo separador. Es por esto que ambos conceptos guardan una estrecha relación y por ende retomaremos muchas propiedades de los  $P$ -círculos vistas en capítulos anteriores.

**Proposición 3.2.1.** *El círculo separador de radio mínimo es único.*

*Demostración.* Procedamos por contradicción y supongamos que existen  $C, C'$  dos círculos separadores de radio mínimo con centros  $c, c'$  respectivamente; recordemos que en particular  $C, C'$  son también  $P$ -círculos. Sea  $x$  un punto en el intervalo abierto  $(c, c')$ , por la Proposición 1.3.3 sabemos que  $C(x)$  es un  $P$ -círculo contenido en  $C \cup C'$  y por ende  $C(x)$  es también un círculo separador. Más aún,  $\rho(x) < \rho(c) = \rho(c')$  por lo que  $C(x)$  es un círculo separador de radio menor que  $C, C'$  lo cual es una contradicción; concluimos entonces nuestro resultado. □

Gracias a la proposición anterior podemos utilizar la siguiente notación.

**Definición 7.** *Sea  $c$  el centro del círculo separador de radio mínimo.*

La observación siguiente nos permitirá acotar la búsqueda por  $c$ , a un conjunto de puntos del que conocemos muchas de sus propiedades.

**Observación 3.2.2.** *El punto  $c$  yace sobre una arista de  $\mathcal{V}(P)$ .*

*Demostración.* Procedamos por contradicción y supongamos que  $c$  está en el interior de  $R(p)$  para algún vértice  $p$  de  $P$ . Sea  $\ell$  la línea recta que pasa por  $p$  y  $c$ ; note que por la Observación 2.2.3,  $\ell$  corta a la frontera de  $R(p)$  en un sólo punto. Sea entonces  $s$  el punto de intersección de la frontera de  $R(p)$  y  $\ell$ , es claro que  $\rho(s) < \rho(c)$ ; ver Figura 3.3. Además  $C(s)$  es un círculo separador pues está contenido en  $C(c)$ , lo cual es una contradicción pues  $C(c)$  es el círculo separador de radio mínimo. Concluimos entonces que  $c$  pertenece a una arista de  $\mathcal{V}(P)$ . □

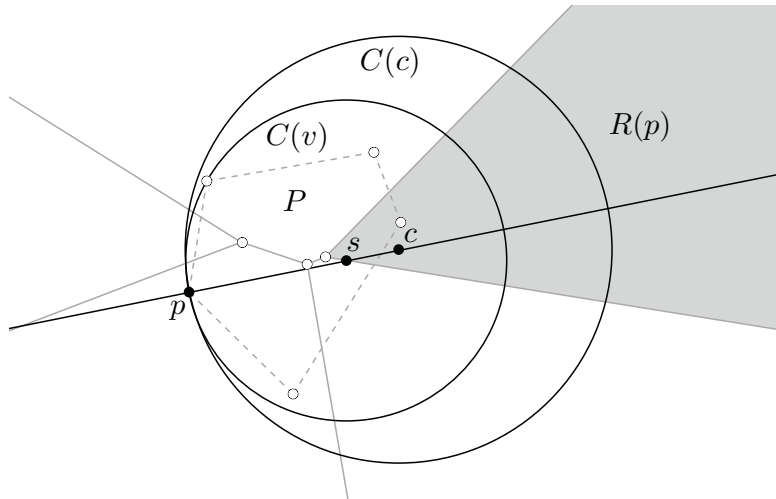


Figura 3.3: En esta figura se observa la prueba de la Observación 3.2.2.

Presentamos a continuación una serie de propiedades de los círculos separadores que nos permitirán determinar más adelante la posición de  $c$ .

**Proposición 3.2.3.** *Sea  $p$  un vértice de  $P$  y sean  $x, y$  dos puntos sobre la frontera de  $R(p)$ , tales que  $C(x), C(y)$  son círculos separadores. Si  $z$  es el primer ancestro común tanto de  $x$  como de  $y$  en el árbol  $\mathcal{V}(P)$  enraizado en  $c_P$ , entonces  $C(z)$  es un círculo separador y  $\rho(z) \leq \min\{\rho(x), \rho(y)\}$ .*

*Demostración.* Supongamos que  $y \notin \mathcal{T}_x, x \notin \mathcal{T}_y$ , de lo contrario el resultado se obtiene trivialmente. Note entonces que las trayectorias que unen a  $x$  y  $y$  con  $z$  tienen interiores ajenos.

Sea  $\ell_{z,p}$  la línea recta que pasa por  $z$  y  $p$ ; es claro que esta recta deja a  $x$  y  $y$  en semiplanos distintos. Sea  $z'$  la intersección de  $\ell_{z,p}$  con  $[x, y]$ ; por la Observación 1.3.3 sabemos que  $C(z') \subseteq C(x) \cup C(y)$ . Como  $z', z, p$  son colineales, entonces  $C(z) \subseteq C(z')$  y por ende  $\rho(z) < \rho(z')$ ; ver Figura 3.4. Finalmente, por transitividad de la contención obtenemos que  $C(z) \subset C(x) \cup C(y)$ , lo cual implica que  $C(z)$  es un círculo separador. Además, al ser  $z$  un ancestro de  $x, y$ , por el Teorema 1.3.6 concluimos que  $\rho(z) < \min\{\rho(x), \rho(y)\}$ .  $\square$

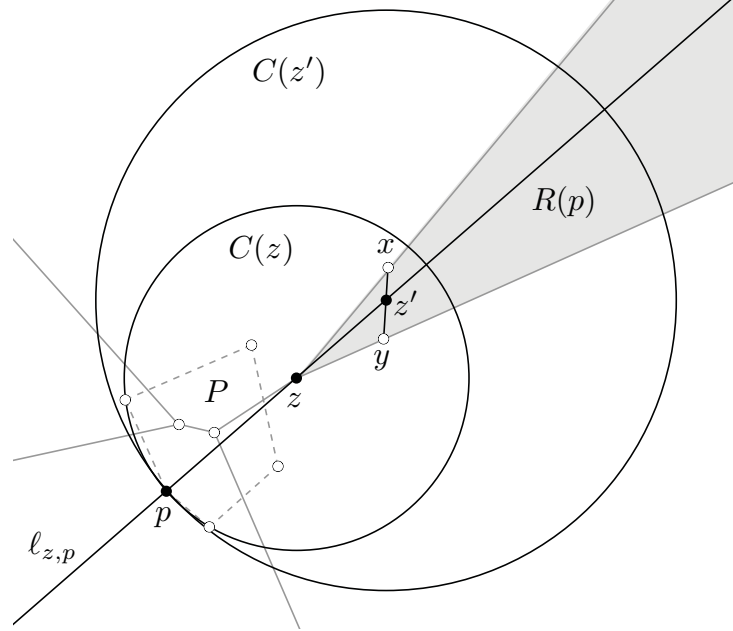


Figura 3.4: En esta figura se observa la prueba de la Proposición 3.2.3.

El resultado anterior se generaliza de la siguiente manera.

**Lema 3.2.4.** Sean  $x, y$  dos puntos sobre  $\mathcal{V}(P)$ , tales que  $C(x)$  y  $C(y)$  son círculos separadores. Si  $z$  es el primer ancestro común tanto de  $x$  como de  $y$  en el árbol enraizado  $\mathcal{V}(P)$ , entonces  $C(z)$  es un círculo separador y  $\rho(z) \leq \min\{\rho(x), \rho(y)\}$ .

*Demostración.* Procedamos por contradicción y supongamos que  $C(z)$  no es un círculo separador. Sea  $w_x$  un punto sobre  $\mathcal{T}_x$  tal que

$$\rho(w_x) = \min\{\rho(w) \mid w \in \mathcal{T}_x \text{ y } C(w) \text{ es un círculo separador}\},$$

es claro entonces que  $w_x \neq z$ . Consideremos las intersecciones del segmento cerrado  $[w_x, y]$  con  $\mathcal{V}(P)$  y supongamos que los puntos de intersección son  $w_x = x_0, x_1, \dots, x_k = y$  en ese orden. Sea  $z'$  es el primer ancestro común de  $w_x$  y  $x_1$ . Como  $w_x$  y  $x_1$  pertenecen a la misma región de Voronoi, usando la Proposición 3.2.3, concluimos que  $C(z')$  es un círculo separador de radio menor que  $C(w_x)$ . Note además que  $z' \in \mathcal{T}_x$  por ser ancestro de  $w_x$ , lo cual es una contradicción con la definición de  $w_x$ ; concluimos entonces nuestro resultado.  $\square$

El siguiente resultado nos permitirá determinar la ubicación de  $c$ , a partir de la ubicación del centro de un círculo separador arbitrario.

**Teorema 3.2.5.** *Sea  $s$  un punto sobre  $\mathcal{V}(P)$ . Si  $C(s)$  es un círculo separador, entonces  $c$  pertenece a  $\mathcal{T}_s$ .*

*Demostración.* Sea  $w$  un punto sobre una arista de  $\mathcal{T}_s$ , tal que

$$\rho(w) = \min\{\rho(z) \mid z \in \mathcal{T}_s \text{ y } C(z) \text{ es un círculo separador}\}.$$

Procedamos por contradicción y supongamos que  $c$  no es un punto sobre  $\mathcal{T}_s$ , por lo tanto  $w \neq c$ . Por Lema 3.2.4, si  $z$  es el primer ancestro común tanto de  $c$  como de  $w$ , entonces  $C(z)$  es un círculo separador y  $\rho(z) \leq \rho(c)$ . Además, como  $c \notin \mathcal{T}_w \subseteq \mathcal{T}_s$ , entonces la desigualdad es estricta, lo cual es una contradicción que viene de suponer que  $w \neq c$ ; concluimos entonces nuestro resultado.  $\square$

El teorema anterior nos permite encontrar a  $c$  sobre cualquier trayectoria que una a  $c_P$  con el centro de un círculo separador arbitrario. Por ende nuestro espacio de búsqueda se reduce considerablemente. Además, recordemos que en el capítulo anterior definimos una estructura de datos que nos permite hacer búsquedas binarias sobre las trayectorias en el árbol  $\mathcal{V}(P)$ .

Finalmente mostraremos el siguiente resultado que nos será útil en la búsqueda del punto  $c$ .

**Observación 3.2.6.** *El  $P$ -círculo  $C(c)$  es tangente al polígono  $Q$  en un único punto  $q_c$ .*

*Demostración.* Procedamos por contradicción y supongamos que  $C(c)$  no interseca al conjunto  $Q$ , por ende  $d(c, Q) > \rho(c)$ . Es claro entonces que por continuidad, existe un punto  $x$  a distancia  $\varepsilon$  de  $c$  sobre  $\mathcal{T}_c$ , tal que se sigue cumpliendo que  $d(x, Q) > \rho(c)$  y por Teorema 1.3.6 podemos concluir que  $\rho(x) < \rho(c)$  lo cual es una contradicción a la minimalidad de  $\rho(c)$ .  $\square$

### 3.3. El algoritmo

En esta sección presentamos un algoritmo para encontrar a  $c$ , utilizando la estructura de datos sobre  $\mathcal{V}(P)$  presentada en el capítulo anterior. Nuestro algoritmo construye primeramente un círculo separador con centro  $s_0$ , tal que  $s_0$  yace sobre una arista de  $\mathcal{V}(P)$ . Paso seguido, el algoritmo busca por  $c$  sobre la trayectoria  $\mathcal{T}_{s_0}$  usando una búsqueda binaria. A continuación describimos el algoritmo a detalle.

Inicialmente revisamos si  $C_P$  es o no un círculo separador, de serlo  $C_P$  es el círculo separador de radio mínimo, ejecutar dicha revisión requiere de tiempo  $O(\log m)$  al calcular la distancia de  $c_P$  a  $Q$ . Supondremos de aquí en

adelante que  $C_P$  no es un círculo separador, por lo tanto el círculo separador de radio mínimo es tangente al polígono  $Q$  y procedemos a encontrarlo como sigue.

Calculamos dos puntos  $p_0, q_0$  sobre los polígonos convexos  $P, Q$  respectivamente, tales que  $d(p_0, q_0) = d(P, Q)$ . Dichos puntos se pueden encontrar en tiempo  $O(\log n + \log m)$  utilizando el algoritmo propuesto por Edelsbrunner en [13]. Usando la bisectriz del segmento  $[p_0, q_0]$ , podemos construir una línea recta  $L$  que separe a los cierres convexos de  $P$  y de  $Q$ .

Sea  $p_L$  el vértice de  $P$  más cercano a  $L$  y supongamos que no hay otro vértice en  $P$  a la misma distancia. De lo contrario rotamos ligeramente a  $L$  manteniendo separados a  $P$  y  $Q$ .

Sea  $L_\perp$  la recta perpendicular a  $L$  que pasa por  $p_L$  y sea  $s_0$  la intersección de  $L_\perp$  con la frontera de  $R(p_L)$ ; dicha intersección es única por la Observación 2.2.3 y se puede calcular en tiempo  $O(\log n)$ , pues es la intersección de un polígono convexo con una línea recta. Note que  $d(s_0, p_L)$  define el radio de  $C(s_0)$ , por lo tanto  $C(s_0)$  es un círculo separador; ver Figura 3.5. Note también que por construcción  $s_0$  yace sobre una arista de  $\mathcal{V}(P)$ . Es claro entonces que podemos calcular a  $s_0$  en tiempo  $O(\log n + \log m)$ .

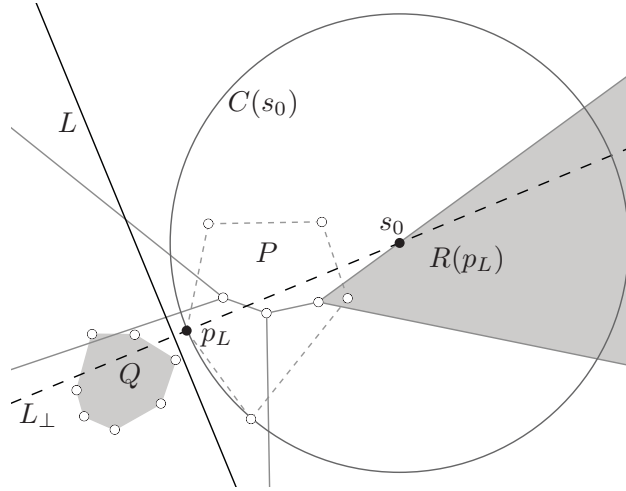


Figura 3.5: La construcción del punto  $s_0$ .

Supongamos que  $s_0$  yace sobre la arista  $xy$  de  $\mathcal{V}(P)$ , y sea

$$\mathcal{T}_x = (c_P = u_0, u_1, \dots, u_{\lambda-1} = y, u_\lambda = x).$$

Por el Teorema 3.2.5 sabemos que  $c$  pertenece a una arista de  $\mathcal{T}_x$ . Usando la estructura de datos propuesta en el capítulo anterior llevamos a cabo la siguiente búsqueda binaria por  $c$  sobre los vértices de  $\mathcal{T}_x$ .

**Algoritmo 3.3.1.** *Algoritmo para encontrar a  $c$  sobre la trayectoria  $\mathcal{T}_x$ .*

1. Sea  $u = u_\lambda$  el nodo de partida y sea  $i = 1$ .
2. Sea  $v$  el vértice al que punta el  $i$ -ésimo apuntador de  $B_u$ .
3. Calculamos  $d(v, Q)$  en tiempo logarítmico [13] y en tiempo constante calculamos el valor de  $\rho(v)$ .
4. Si  $u$  y  $v$  son vértices consecutivos en  $\mathcal{T}_x$ , entonces terminamos y regresamos la arista  $uv$ .
5. Si  $d(v, Q) = \rho(v)$ , entonces  $c = v$  y terminamos.
6. Si  $d(v, Q) < \rho(v)$ , entonces  $i = i + 1$  y regresamos al paso 2.
7. Si  $d(v, Q) > \rho(v)$ , entonces  $u = v$ ,  $i = 1$  y regresamos al paso 2.

El algoritmo anterior concluye y despliega al nodo  $c$  si éste es un vértice de  $\mathcal{T}_x$ . Sin embargo si éste es un punto interior de un segmento, lo cual supondremos de aquí en adelante, entonces el algoritmo despliega dos nodos consecutivos  $u, v$  de  $\mathcal{T}_x$  tales que  $c \in [u, v]$ . Visto de otra forma, tenemos que  $\rho(u) < d(u, Q)$  y  $\rho(v) > d(v, Q)$ , es decir que  $C(u)$  es un círculo separador y  $C(v)$  ya no lo es; ver Figura 3.6.

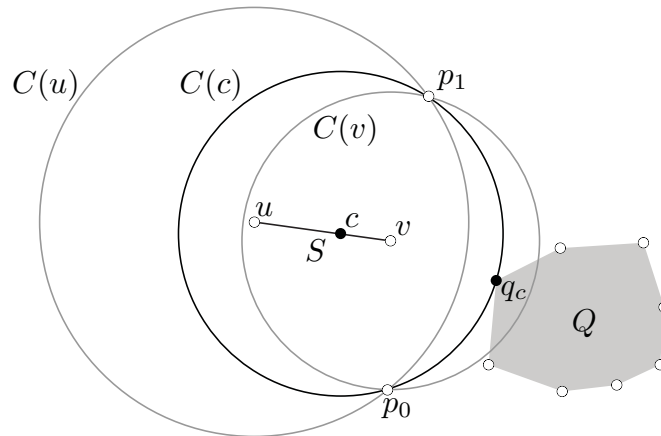


Figura 3.6: El segmento  $S$  sobre el cual se encuentra  $c$ .



Acotar nuestra búsqueda a este segmento requiere entonces de tiempo  $O(\log n \log m)$ ; esto último ya que el Algoritmo 3.3.1 termina en  $O(\log n)$  rondas, y cada paso requiere de tiempo constante a excepción del paso 3 que se ejecuta en tiempo  $O(\log m)$ . En el apartado siguiente describiremos como encontrar a  $c$  una vez que el Algoritmo 3.3.1 encontró un segmento sobre el que yace el centro del círculo separador de radio mínimo.

### 3.3.1. Precisando la posición de $c$

Sea  $S = [u, v]$  el segmento obtenido por el Algoritmo 3.3.1, recordemos que  $c \in S$  y supongamos que  $S$  está contenido en la bisectriz de los vértices  $p_0, p_1$  de  $P$ .

Diremos que un punto  $q$  de la frontera de  $Q$  es visible desde un punto  $x$  en  $S$ , si el segmento abierto  $(x, q)$  no intersecta a  $Q$ . Sea  $Q_S$  el conjunto de puntos de la frontera de  $Q$ , visibles desde cualquier punto de  $S$ . Gracias a la convexidad de  $Q$ ,  $Q_S$  es la intersección de los puntos visibles desde  $u$  y los puntos visibles desde  $v$ , por ende podemos calcularlo en tiempo  $O(\log m)$  usando una búsqueda binaria sobre los vértices de  $Q$ .

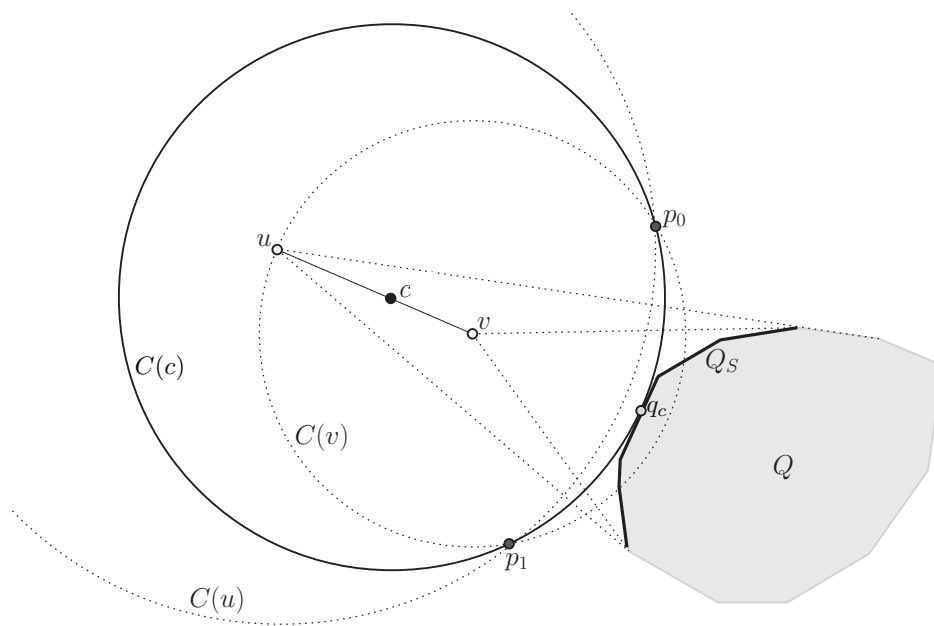


Figura 3.7: La construcción del conjunto  $Q_S$  y el punto  $q_c$ .

Por la Observación 3.2.6,  $C(c)$  es un  $P$ -círculo tangente al polígono  $Q$ , por ende definimos a  $q_c$  como el punto de intersección de  $C(c)$  con  $Q$ . Es fácil ver que se cumple el siguiente resultado; ver Figura 3.7.

**Observación 3.3.1.** *El punto  $q_c$  está en  $Q_S$ , es decir que  $q_c$  es visible desde cualquier punto de  $S$ .*

Quisiéramos asociar un punto de  $S$  a cada punto de  $Q_S$ , de modo que pudiéramos buscar por  $c$  utilizando los vertices de  $Q$  que yacen en  $Q_S$ . Para ello utilizaremos la siguiente definición.

**Definición 8.** *Dados tres puntos  $x, y, z \in \mathbb{R}^2$ , denotamos por  $C(xyx)$  al circuncírculo del triángulo  $\Delta(xyz)$ .*

Note que para cada punto  $q$  en  $Q_S$ , el centro de  $C(p_0qp_1)$  yace sobre el bisector de  $p_0, p_1$ , más aún, note que el radio de  $C(p_0qp_1)$  se incrementa cuando movemos a  $q$  hacia  $q_c$  y decrementa al alejarnos; ver Figura 3.8.

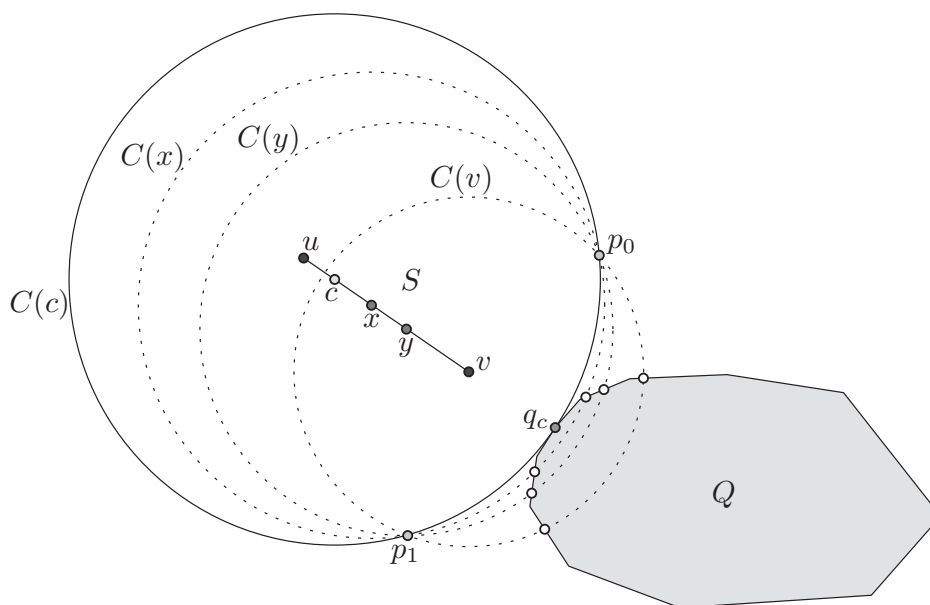


Figura 3.8: La maximalidad de  $q_c$  bajo la función  $\tau$ .

Formalmente, definimos a  $\tau(x)$  como el radio del círculo  $C(p_0xp_1)$ . Es fácil ver que  $\tau(x)$  define una función unimodal sobre los puntos de  $Q_S$ , si los recorremos en el sentido de las manecillas del reloj sobre la frontera de  $Q$ . Note que, si  $w \in [u, c]$ , entonces  $C(w)$  no interseca a  $Q$  ya que  $c$  es el primer punto de  $S$  tal que  $C(c)$  interseca a  $Q$ . Por ende para cada punto  $q \in Q_S$ , el círculo  $C(p_0qp_1)$  tiene su centro sobre en el segmento  $[c, v] \subseteq S$ . Afirmamos que  $\tau$  alcanza su máximo en  $q_c$  ya que  $c$  es el centro de  $C(p_0q_cp_1)$  y  $c$  es máximo local de la función  $\rho$  en el segmento  $[c, v]$ ; ver Figura 3.8.

Este resultado nos permitirá realizar una búsqueda binaria por  $q_c$  en los vértices de  $Q$  que pertenezcan a  $Q_S$ . Note que necesitamos restringir la búsqueda a  $Q_S$ , ya que  $\tau(x)$  no es unimodal sobre todos los puntos de la frontera de  $Q$ ; ver Figura 3.9.

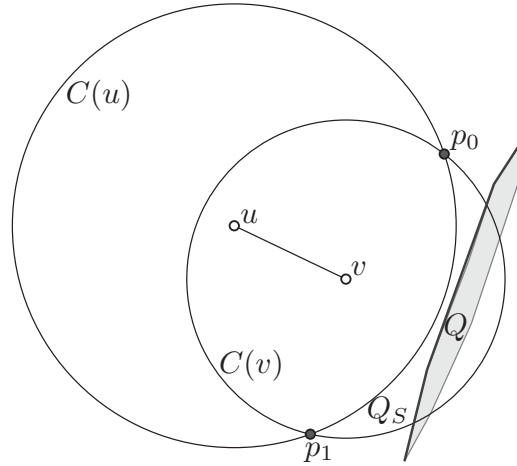


Figura 3.9: Ya que el círculo  $C(u)$  intersecciona a  $Q$  en más de dos puntos,  $\tau(x)$  no es unimodal cuando la definimos sobre toda la frontera de  $Q$ , sin embargo lo es cuando la definimos sobre  $Q_S$ .

### 3.3.2. Algoritmo para encontrar $q_c$

Implementamos el algoritmo para encontrar a  $q_c$  sobre  $Q_S$  como sigue. Es claro que en tiempo  $O(\log m)$  podemos encontrar los extremos de  $Q_S$ , digamos  $q_0$  y  $q_r$ . Sea  $Q_S^* = \{q_0, q_1, \dots, q_r\}$  el conjunto de vértices de  $Q$  que pertenecen a  $Q_S$  y supongamos que se encuentran en ese orden sobre la frontera de  $Q$ . Procedemos ahora a realizar una búsqueda binaria por  $q_c$  sobre los puntos de  $Q_S^*$ .

Inicialmente nuestro espacio de búsqueda será todo el conjunto  $Q_S^*$ . En cada ronda del algoritmo definimos a  $q_i$  como el punto medio del intervalo actual de búsqueda. Calculamos el valor de  $\tau(q_i)$  en tiempo constante y tomamos dos puntos a cada lado de  $q_i$  sobre la frontera de  $Q$  a distancia epsilon. Revisamos si  $q_i$  es o no un máximo local de  $\tau$ , de serlo terminamos y lo desplegamos. De lo contrario, usamos la unimodalidad de  $\tau$  y los puntos a distancia epsilon para averiguar si  $q_c$  está a la izquierda o a la derecha de  $q_i$  en  $Q_S^*$ . Una vez determinado lo anterior redefinimos el intervalo de búsqueda y repetimos recursivamente el proceso. Formalmente describimos el algoritmo como sigue.

**Algoritmo 3.3.2.** *Algoritmo para encontrar a  $q_c$ .*

1. Definimos inicialmente el intervalo de la búsqueda de modo que  $g = 0$  y  $h = r$ .
2. Creamos un apuntador al elemento a la mitad de nuestro intervalo actual de búsqueda, es decir  $i = \lceil \frac{g+h}{2} \rceil$ .
3. Tomamos puntos a distancia  $\varepsilon > 0$  de  $q_i$  hacia  $q_{i+1}$  y  $q_{i-1}$  sobre la frontera de  $Q$ , formalmente  $q_i^+ = q_i + \varepsilon(q_{i+1} - q_i)$  y  $q_i^- = q_i + \varepsilon(q_{i-1} - q_i)$ .
4. Calculamos en tiempo constante los radios de  $C(p_0q_i p_1)$ ,  $C(p_0q_i^+ p_1)$  y  $C(p_0q_i^- p_1)$ .
5. Si  $\tau(q_i) > \max\{\tau(q_i^+), \tau(q_i^-)\}$ , entonces  $q_i$  es un máximo de la función  $\tau$ , por lo tanto  $q_c = q_i$  con lo que termina el algoritmo.
6. Si  $\tau(q_i) < \tau(q_i^+)$ , entonces redefinimos el intervalo de búsqueda al asignar  $g = i$ .
7. Si  $\tau(q_i) < \tau(q_i^-)$ , entonces redefinimos el intervalo de búsqueda al asignar  $h = i$ .
8. Si  $g, h$  son nodos consecutivos, entonces terminar y desplegar el segmento  $H = [q_g, q_h]$ . De lo contrario regresar al paso 2.

Siendo que descartamos la mitad de los puntos en el intervalo de búsqueda en cada ronda, es claro que el algoritmo anterior termina en  $O(\log m)$  rondas. Note también que cada una de las rondas se ejecuta en tiempo constante, por lo tanto el algoritmo anterior termina en tiempo  $O(\log m)$ . Al finalizar, el algoritmo despliega el valor de  $q_c$  si es éste era un vértice de  $Q$ , o un segmento  $H = (q_i, q_{i+1})$  de  $Q_S$ , tal que  $q_c$  pertenece a  $H$ . En el primer caso habremos concluido, pues una vez encontrado  $q_c$ , en tiempo constante podemos determinar la ubicación de  $c$  y por ende encontrar al círculo separador de radio mínimo.

En el segundo caso nuestro problema se reduce a encontrar un punto  $c \in S$  tal que  $d(c, p_0) = d(c, H)$ . Este caso se resuelve mediante un sistema de dos ecuaciones con dos incógnitas como sigue. Supongamos que la ecuación de la recta que contiene a  $H$  es  $Ax + By + C = 0$  y supongamos también que la ecuación de la recta que contiene a  $S$  es  $A'x + B'y + C' = 0$ . Por lo tanto, el sistema de ecuaciones siguiente, cuya solución se puede encontrar en tiempo constante, nos permite precisar la posición del punto  $q_c$ .

$$\sqrt{(c[x] - p_0[x])^2 + (c[y] - p_0[y])^2} = \frac{|Ac[x] + Bc[y] + C|}{\sqrt{A^2 + B^2}} \quad (3.1)$$

$$A'c[x] + B'c[y] + C' = 0 \quad (3.2)$$

En resumen, el algoritmo encargado de la localización de  $q_c$  se lleva a cabo en tiempo  $O(\log m)$ . Dándonos así una complejidad total del algoritmo de consulta de  $O(\log n \log m)$ .

# Bibliografía

- [1] A. Aggarwal, L. J. Guibas, J. Saxe, and P. W. Shor. A linear-time algorithm for computing the voronoi diagram of a convex polygon. *Discrete Comput. Geom.*, 4:591–604, September 1989.
- [2] Alok Aggarwal, Heather Booth, Joseph O'Rourke, Subhash Suri, and Chee K. Yap. Finding minimal convex nested polygons. In *Proceedings of the first annual symposium on Computational geometry*, SCG '85, pages 296–304, New York, NY, USA, 1985. ACM.
- [3] B. K. Bhattacharya. Circular separability of planar point sets. *Computational Morphology*, pages 25–39, 1988.
- [4] Vittorio Bilò, Ioannis Caragiannis, Christos Kaklamanis, and Panagiotis Kanellopoulos. Geometric clustering to minimize the sum of cluster sizes. In Gerth Brodal and Stefano Leonardi, editors, *Algorithms – ESA 2005*, volume 3669 of *Lecture Notes in Computer Science*, pages 460–471. Springer Berlin / Heidelberg, 2005.
- [5] Jean-Daniel Boissonnat, Jurek Czyzowicz, Olivier Devillers, and Mariette Yvinec. Circular separability of polygon. In *SODA '95: Proceedings of the sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 273–281, Philadelphia, PA, USA, 1995. Society for Industrial and Applied Mathematics.
- [6] P. Bose and G. Toussaint. Computing the constrained euclidean, geodesic and link centre of a simple polygon with applications. In *Computer Graphics International, 1996. Proceedings*, pages 102–111, jun 1996.
- [7] Timothy M. Chan. More planar two-center algorithms. *Comput. Geom. Theory Appl.*, 13:189–198, September 1999.
- [8] Bernard Chazelle and Jiří Matoušek. On linear-time deterministic algorithms for optimization problems in fixed dimension. In *Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms*,

- SODA '93, pages 281–290, Philadelphia, PA, USA, 1993. Society for Industrial and Applied Mathematics.
- [9] Kenneth L. Clarkson. Las vegas algorithms for linear and integer programming when the dimension is small. *J. ACM*, 42:488–499, March 1995.
- [10] G. Das and D. Joseph. The complexity of minimum convex nested polyhedra. In *In Proc. 2nd Canad. Conf. Comput. Geom., CCCG '90*, pages 296–301, 1990.
- [11] Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, third edition, 2008.
- [12] M E Dyer. On a multidimensional search technique and its application to the euclidean one centre problem. *SIAM J. Comput.*, 15:725–738, August 1986.
- [13] H. Edelsbrunner. Computing the extreme distances between two convex polygons. *Journal of Algorithms*, 6(2):213 – 224, 1985.
- [14] H. Edelsbrunner and F. P. Preparata. Minimum polygonal separation. *Information and Computation*, 77(3):218 – 232, 1988.
- [15] Jack Elzinga and Donald W. Hearn. Geometrical solutions for some minimax location problems. *Transportation Science*, 6(4):379–394, 1972.
- [16] David Eppstein. Faster construction of planar two-centers. In *Proceedings of the eighth annual ACM-SIAM symposium on Discrete algorithms*, SODA '97, pages 131–138, Philadelphia, PA, USA, 1997. Society for Industrial and Applied Mathematics.
- [17] V. Sacristan F. Hurtado and G. Toussaint. Some constrained minimax and maximim location problems. *Studies in Locational Analysis*, 15:17–35, 2000.
- [18] Tomás Feder and Daniel Greene. Optimal algorithms for approximate clustering. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, STOC '88, pages 434–444, New York, NY, USA, 1988. ACM.
- [19] Steve Fisk. Separating point sets by circles, and the recognition of digital disks. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(4):554 –556, july 1986.

- 
- [20] Teofilo F. Gonzalez. Covering a set of points in multidimensional space. *Information Processing Letters*, 40(4):181 – 188, 1991.
- [21] Arindam Karmakar, Sasanka Roy, and Sandip Das. Fast computation of smallest enclosing circle with center on a query line segment. *Information Processing Letters*, 108(6):343 – 346, 2008.
- [22] Chul E. Kim and Timothy A. Anderson. Digital disks and a digital compactness measure. In *STOC '84: Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 117–124, New York, NY, USA, 1984. ACM.
- [23] S.R. Lay. On separation by spherical surfaces. *Amer. Math. Monthly*, 78:1112–1113, 1971.
- [24] D.T. Lee. Farthest neighbor voronoi diagrams and applications. *Report 80-11-FC-04, Dept. Elect. Engrg. Comput. Sci.*, 1980.
- [25] Nimrod Megiddo. Linear-time algorithms for linear programming in  $\mathbb{R}^3$  and related problems. *SIAM Journal on Computing*, 12(4):759–776, 1983.
- [26] Nimrod Megiddo. Linear programming in linear time when the dimension is fixed. *J. ACM*, 31:114–127, January 1984.
- [27] Nimrod Megiddo and Kenneth J. Supowit. On the complexity of some common geometric location problems. *SIAM Journal on Computing*, 13(1):182–196, 1984.
- [28] Joseph O'Rourke, S. Rao Kosaraju, and Nimrod Megiddo. Computing circular separability. *Discrete and Computational Geometry*, 1:105–113, 1986.
- [29] F.P. Preparata. *Minimum spanning circle, in F.P. Preparata (Ed.), Steps in Computational Geometry*. University of Illinois, 1977.
- [30] Stefan Langerman Prosenjit Bose and Sasanka Roy. Smallest enclosing circle centered on a query line segment. In *CCCG'08*, pages –1–1, 2008.
- [31] M. S. Paterson R. J. Fowler and S. L. Tanimoto. Optimal packing and covering in the plane are np-complete. *Information Processing Letters*, 12(3):133–137, 1981.



- 
- [32] Sasanka Roy, Arindam Karmakar, Sandip Das, and Subhas C. Nandy. Constrained minimum enclosing circle with center on a query line segment. *Computational Geometry*, 42(6-7):632 – 638, 2009.
- [33] Michael Ian Shamos and Dan Hoey. Closest-point problems. In *Proceedings of the 16th Annual Symposium on Foundations of Computer Science*, SFCS '75, pages 151–162, Washington, DC, USA, 1975. IEEE Computer Society.
- [34] J. J. Sylvester. A Question in the Geometry of Situation. *Quarterly Journal of Pure and Applied Mathematics*, 1, 1857.
- [35] Emo Welzl. Smallest enclosing disks (balls and ellipsoids). In Hermann Maurer, editor, *New Results and New Trends in Computer Science*, volume 555 of *Lecture Notes in Computer Science*, pages 359–370. Springer Berlin / Heidelberg, 1991. 10.1007/BFb0038202.