

UNIVERSIDAD DE SEVILLA

Departamento de Matemática Aplicada II

**LOCALIZACIÓN DE
ESTRUCTURAS LINEALES Y
LINEALES A TROZOS**

José Miguel Díaz Báñez

Tesis Doctoral

T. 163

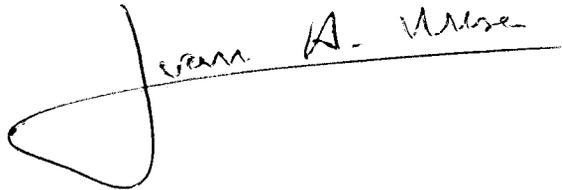
63

93
24 MAR. 1998

Alcaldía

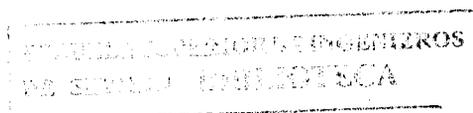
Memoria de Tesis presentada por **José Miguel Díaz Báñez**
para optar al grado de **Doctor en Matemáticas**
por la Universidad de Sevilla

Vº Bº
del Director de Tesis

A handwritten signature in black ink, reading "Juan A. Mesa". The signature is written in a cursive style and is positioned above a horizontal line that extends across the width of the signature.

Fdo. Dr. D. Juan A. Mesa López-Colmenar
Catedrático de Universidad
del Departamento de Matemática Aplicada II
de la Universidad de Sevilla

Sevilla, marzo de 1998.



Agradecimientos

Aunque muchas son las personas que han contribuido a la finalización de este trabajo, destacaré aquí algunas, que constituyen los tres pilares sobre los que se fundamenta.

Por una parte, deseo expresar mi agradecimiento al director, Juan A. Mesa, por sus consejos y ayuda incondicional.

Por otra, quisiera hacer constar que me siento orgulloso de haber recibido apoyo de un grupo de personas a los que denomino cariñosamente *gente de Geometría*: Paco Gómez, Ferrán Hurtado, Godfried Toussaint, Juan Carlos Dana y Pedro Díaz, sin cuya compañía no se hubiera realizado esta memoria.

Finalmente, en la parte más importante, la familiar, quisiera destacar la ayuda que siempre recibí de Inmaculada así como la confianza y ánimo recibidos de mi madre y hermanos.

Resumen

En esta memoria se estudia la localización óptima en el plano de estructuras lineales y lineales a trozos, como semirrectas, rectas y poligonales, para cuya resolución se usan fundamentalmente métodos de la Geometría Computacional. Los criterios que se contemplan son la minimización de la excentricidad, para el caso de rectas y semirrectas con origen en un punto prefijado y la minimización de la excentricidad y de la longitud, para el caso de poligonales con salida y entrada restringidas. Para los problemas minimax de poligonales se han considerado dos factores de restricción, el número de codos y la longitud del camino poligonal, que aparecen usualmente en problemas de diseño de trayectorias sobre un conjunto de puntos en el plano. Los procedimientos de resolución propuestos para el caso de poligonales se han desarrollado dependiendo de que estén compuestas por segmentos cualesquiera o bien sólo horizontales y verticales, denominadas respectivamente *poligonales genéricas* y *poligonales rectilíneas u ortogonales*. En la mayoría de los casos, el estudio de propiedades geométricas de una solución del problema ha sugerido el diseño de algoritmos eficientes que lo resuelven.

Índice

1	Introducción	9
1.1	El marco disciplinar	9
1.1.1	Análisis de Localizaciones y Geometría Computacional	9
1.1.2	El problema minimax	12
1.2	El trabajo realizado	14
2	El problema minimax para servicios lineales	17
2.1	Problemas con métrica euclídea	18
2.1.1	La recta centro	18
2.1.2	La semirrecta centro	24
2.1.3	El segmento centro	25
2.2	Problemas con distancia vertical	27
2.3	Problemas con métricas l_∞ y l_1	28
2.3.1	La semirrecta centro	30
2.4	Conclusiones y problemas abiertos	44

3	Camino rectilíneo más corto de rechazo y de cobertura.	47
3.1	Estudio del problema RMA.	49
3.2	Análisis del problema RMC.	59
3.2.1	Cobertura directa	60
3.2.2	Cobertura indirecta	65
3.3	Conclusiones y problemas abiertos	73
4	El problema minimax para trayectorias rectilíneas	75
4.1	Restricción en el número de codos	76
4.1.1	Formulación del problema	76
4.1.2	Propiedades de soluciones óptimas	78
4.1.3	Búsqueda de soluciones	81
4.1.4	El problema restringido	87
4.1.5	Un algoritmo eficiente para el caso no degenerado	90
4.2	Restricción en la longitud	95
4.2.1	Análisis de soluciones	98
4.2.2	Descripción del método	102
4.3	Conclusiones y problemas abiertos	106
5	El problema minimax para poligonales	109
5.1	Antecedentes	109
5.2	Definición de los problemas	113

5.3	Poligonales discretas	115
5.3.1	Poligonales de un solo codo	115
5.3.2	Poligonales de un codo apoyadas en convexos	119
5.3.3	Poligonales de k codos	122
5.3.4	Poligonales con longitud restringida	124
5.4	Poligonales libres	129
5.4.1	Poligonales libres de un codo	130
5.5	Conclusiones y problemas abiertos	134

Bibliografía

Lista de Figuras

2.1	Recta candidato del problema 1-recta centro.	20
2.2	Hipódromo contenedor mínimo.	26
2.3	Cuadrado mínimo englobador.	28
2.4	Entorno $E_\infty(s(\theta), R)$	31
2.5	Giro de un entorno de semirrecta óptimo.	32
2.6	Casos considerados.	32
2.7	Construcción geométrica del entorno.	33
2.8	Distintas posiciones del foco.	35
2.9	Distancia l_∞ entre un punto y una recta.	36
2.10	Regiones I_1, I_2, I_3	37
2.11	Distintos tipos de funciones distancia.	39
2.12	Pasos 5, 6 y 7 del algoritmo.	42
2.13	p_0 está en el interior de $CC(P)$	44
3.1	Trayectoria más corta a través de obstáculos rectangulares. . .	50
3.2	Obtención de la partición maximal.	51

3.3	Cálculo de vértices extremos.	53
3.4	Reducción no válida del problema con distancia d_1	54
3.5	Obtención de una poligonal genérica de anticobertura para d_1	55
3.6	Cadena poligonal rectilínea y distancia vertical.	56
3.7	Reducción del problema con distancia vertical.	57
3.8	Ilustrando una cota inferior.	59
3.9	El camino más corto no es x-monótono.	60
3.10	Construcción de un camino mínimo. (i) Posición no degenerada. (ii) Posición degenerada.	61
3.11	Obtención de los puntos z_j y \bar{z}_j	62
3.12	Caso $y(z) \geq y(\bar{t}_{j-1})$ en el lema 3.2.1.	64
3.13	Transformación del problema RMCI para el caso d_∞	66
3.14	Un algoritmo incremental sobre las abcisas no resuelve el problema.	67
3.15	Transformación del problema con distancia vertical.	68
3.16	Problema de decisión.	69
3.17	Obtención de sucesores.	70
4.1	Conjuntos de asignación.	79
4.2	Segmentos horizontales centro.	80
4.3	Situación de los segmentos verticales.	81
4.4	Asignación de los puntos en una vertical.	83

4.5	Cota impar del número de codos.	85
4.6	Considerando restricciones en los extremos.	88
4.7	Valores candidatos.	91
4.8	Interpretación geométrica del problema.	92
4.9	Los segmentos de trazo continuo (discontinuo) corresponden al barrido de derecha a izquierda (de izquierda a derecha).	93
4.10	Cálculo de valores candidatos: $e_{ik} = e_{ik-1} = d_v(a, b)$, $e_{ik+1} = d_v(p_{k+1}, b) > e_{ik}$	94
4.11	Adaptación al caso restringido.	95
4.12	Los segmentos horizontales no son segmentos-centro.	96
4.13	Puntos extremales de una cadena R	97
4.14	Situación de los segmentos horizontales.	101
5.1	Interpretación del error correspondiente a un segmento.	110
5.2	Distintos problemas de optimización.	115
5.3	El máximo de la distancia no se alcanza en el cierre convexo.	116
5.4	Descomposición del cierre convexo por la recta $x = x(c_1)$	117
5.5	Cálculo de la recta óptima que pasa por dos polígonos convexos.	120
5.6	Una de las dos subcadenas es óptima.	123
5.7	Construcción del polígono monótono $\pi(a, b)$	125
5.8	Obtención de e_{ij} en tiempo $O(\log(n))$	128
5.9	Interpretación geométrica de un problema	129

5.10 Resolución del problema en cada banda.	131
5.11 Obtención del codo c^*	133

Capítulo 1

Introducción

1.1 El marco disciplinar

1.1.1 Análisis de Localizaciones y Geometría Computacional

En esta memoria se tratan diversos problemas planteados en el marco general del Análisis de Localizaciones, aunque la resolución de los mismos ha resultado como consecuencia de abordarlos desde el punto de vista de la Geometría Computacional. De esta forma, en el desarrollo temporal de la investigación cuya consecuencia ha sido este trabajo, la Localización ha proporcionado el planteamiento de las cuestiones, siendo la Geometría la que ha aportado principalmente los procedimientos de resolución.

Aunque ambas disciplinas surgieron de forma independiente dentro de la Matemática Aplicada, la conexión entre ambas viene determinada por la propia naturaleza geométrica de múltiples problemas del Análisis de Localizaciones. Si quisiéramos expresar de forma concisa, aunque sin duda restrictiva, el objetivo general del Análisis de Localizaciones y la Geometría Computacional, diríamos que la primera trata sobre la *asignación o ubicación espacial de recursos en un espacio prefijado*, mientras que la segunda se refiere al *diseño*

y análisis de algoritmos para resolver problemas geométricos en un ordenador. Como consecuencia, no sería correcto decir que se trata de disciplinas con objetivos comunes puesto que la primera estudia un conjunto de problemas en un contexto determinado, mientras que la segunda se refiere más bien a una forma de actuación ante problemas geométricos en cualquier ámbito. Constituyen dos campos que han surgido (como tantos otros en la historia de las matemáticas) para cubrir distintas necesidades en momentos distintos y cuyo contacto produce (como en tantas otras ocasiones) resultados fructíferos, tanto desde el punto de vista interno de las matemáticas como desde la aplicación de ésta a necesidades reales. Pueden citarse [34], [54] y [23] como textos básicos del área de Localización y [74], [70] y [36] para la Geometría Computacional.

Las versiones clásicas de problemas de Análisis de Localizaciones (también denominada Localización de Servicios), consisten en la determinación de uno o varios puntos en un espacio prefijado, en el que se conoce la distribución de otros tantos denominados *puntos de demanda*. La elección de los puntos en los que se van a localizar los servicios, se hace de acuerdo con una o varias funciones objetivo y, posiblemente, una o varias restricciones, que miden la interrelación entre los puntos de demanda y los servicios. Las funciones y las restricciones contienen términos dependientes de las distancias.

Usualmente, los problemas de localización puntual se han clasificado en tres grupos de acuerdo con el espacio en el que se localiza, el cual puede ser un conjunto finito, un subconjunto propio o impropio de \mathbb{R}^n , $n \geq 1$, un grafo o una red. Sin embargo, cuando los servicios que se quieren localizar son lo suficientemente grandes, con respecto a sus actividades de interés y a su entorno como para ser representados como puntos, los modelos de localización puntual no pueden ser aplicados ya que el conjunto a localizar no consta sólo de puntos aislados sino que se ajusta a una determinada estructura sugerida por la forma del servicio.

Para obtener una formulación precisa y genérica de un problema de Análisis de Localizaciones, consideremos \mathcal{S} un espacio dado, \mathcal{C} una colección de subconjuntos de \mathcal{S} , $\mathcal{D} \subset \mathcal{S}$ un subconjunto especificado y $f_D : \mathcal{C} \rightarrow \mathbb{R}^p$, $p \geq 1$

una función vectorial. El problema general de localización consiste en :

$$\text{optimizar}_{\mathcal{A} \in \mathcal{C}} f_{\mathcal{D}}(\mathcal{A})$$

donde $f_{\mathcal{D}}$ es la función objetivo y, si hay restricciones, éstas están contenidas en la condición $\mathcal{A} \in \mathcal{C}$.

\mathcal{S} puede ser, por ejemplo, una región del espacio 2 ó 3-dimensional, una red o un conjunto discreto de puntos; \mathcal{C} una colección de superficies, caminos, conjunto de puntos aislados, etc.

Cuando a los conjuntos de \mathcal{C} se les puede asignar alguna medida espacial (longitud, área, etc.) diremos que el problema de localización es de *estructuras dimensionales*. Estos problemas pueden clasificarse en dos grupos: aquéllos en los que \mathcal{S} es un subconjunto (propio o impropio) del espacio \mathbb{R}^p , $p \geq 1$, y los que tienen como espacio \mathcal{S} a un grafo o una red. Para mayor claridad, se conviene denominarlos problemas de *localización de estructuras en espacios continuos* y *localización de estructuras sobre redes*, respectivamente. Precisamente, los problemas planteados en este trabajo pertenecen al primer grupo, cuyos avances se han orientado en gran medida hacia la localización de servicios lineales en el plano. Una de las metodologías más convenientes en este contexto es la de hacer uso de los aspectos geométricos que se derivan de la condición espacial de dichos problemas. De hecho, estas cuestiones se pueden describir tanto con un modelo analítico de optimización como con un modelo geométrico en los que la solución es una determinada figura o estructura geométrica.

El uso de técnicas propias de Geometría Computacional en la resolución de problemas de Análisis de Localizaciones aparece en la bibliografía reciente relativa a ambas disciplinas, tanto para el caso de localización de servicios puntuales como para el de estructuras. Justamente, en la tesis de Shamos [80], considerada como punto de arranque de la Geometría Computacional, se describen algoritmos geométricos que resuelven problemas de localización puntual como el conocido *minimum spanning circle*. A partir de ahí, múltiples han sido los resultados en el campo de Análisis de Localizaciones obtenidos mediante el uso de herramientas como la envolvente convexa de un conjunto

de puntos, los diagramas de Voronoi, la triangulación de Delaunay, algoritmos de divide y vencerás, etc. Citaremos, entre otros muchos, los trabajos de Lee y Wu [53], Toussaint [86], Bose y Toussaint [8], Robert y Toussaint [78], Okabe, Boots y Sugihara [67] y Okabe y Suzuki [68].

1.1.2 El problema minimax

Uno de los artículos más cortos aparecidos en revistas matemáticas fue el que, firmado por Sylvester y titulado *A Question in the Geometry of Situation*, se publicó en el *Quarterly Journal of Pure and Applied Mathematics* en 1857 [84]: “It is desired to find the least circle which shall contain a given system of points in a plane ...”. Dicho trabajo, puede considerarse como el punto de partida de un problema que muestra una fuerte conexión entre los campos de Análisis de Localizaciones y Geometría Computacional. La resolución del problema, también citado en textos clásicos como [75] y [15], posee una interesante historia que refleja dicha conexión y coincide además con el desarrollo de ambas disciplinas.

Es en 1860, cuando Sylvester proporciona un método de resolución geométrico que, sin embargo, es atribuido a Pierce. Desde entonces, fueron propuestos distintos procedimientos para resolver el problema del *menor círculo que contiene a un conjunto de puntos en \mathbb{R}^2* , también llamado problema del *círculo generador o englobador*. Esta formulación geométrica corresponde al problema denominado en Teoría de Localización del del 1-centro euclídeo no ponderado: dado un conjunto de puntos en el plano $P = \{p_i; i = 1, 2, \dots, n\}$, encontrar un punto p que minimice la máxima distancia euclídea sobre los puntos de P ,

$$\min_{p \in \mathbb{R}^2} \max_{1 \leq i \leq n} d(p_i, p),$$

y cuya aplicación resulta particularmente adecuada a la ubicación de servicios de urgencias, con objeto de minimizar el tiempo de respuesta en el peor caso. Puesto que la solución viene dada precisamente por el centro del menor círculo contenedor, el problema es intrínsecamente geométrico. Por otra parte, teniendo en cuenta que dicho círculo viene determinado por un par

de puntos diametrales o una terna de puntos, se puede resolver el problema considerando todos los posibles círculos determinados por los puntos de P , descartar los que no contengan a todos los puntos y elegir el menor de todos. Este procedimiento, que podemos denominar *de fuerza bruta* dará pie al uso de una técnica usual en problemas generalizados de éste que denominaremos *análisis de casos*. A dicho procedimiento inicial (de tiempo $O(n^4)$) siguieron otros de mayor eficiencia. El algoritmo geométrico de tiempo $O(n^2)$ propuesto por Elzinga y Hearn [30] en la década de los setenta, supone un importante avance en la resolución. Posteriormente, coincidiendo con la aparición y desarrollo de la Geometría Computacional, Shamos y Hoey [80] usan el Diagrama de Voronoi del punto más lejano y proponen un algoritmo $O(n \log(n))$. Sin embargo, el algoritmo publicado en [80] contiene un error a la hora de calcular el diámetro de un conjunto. Bhattacharya y Toussaint [7] proponen un contraejemplo y demuestran, a pesar de ello, que el algoritmo de [80] conduce siempre a una solución del problema. Finalmente, Megiddo [60] por un lado, y Dyer [28] por otro, encuentran algoritmos de tiempo $O(n)$ usando un método basado en programación lineal.

La evolución histórica que se observa en las distintas soluciones del problema 1-centro, sugiere precisamente la línea de investigación en la cual se enmarca esta memoria y constituye además un apartado de la Matemática Aplicada con evidente interés y actualidad. De hecho, el problema del *1-centro puntual*, que podemos considerar como básico para el criterio minimax, ha sido extendido y generalizado a diferentes contextos toda vez que se modifican los factores que componen el problema. Se han considerado distancias distintas de la euclídea [72],[55], calibradores [73]. Se ha considerado la versión en la que los puntos están ponderados de acuerdo con su importancia relativa [41], [58]. Ha sido estudiado el caso 3D euclídeo [31] y m-dimensional [24]. Con respecto a la multiplicidad de puntos a localizar se ha estudiado el problema denominado *p-centro*, mediante el uso de programación lineal para la norma l_1 [89] y programación no lineal para el caso l_p [54], [26]. Finalmente, en la actualidad, se está investigando sobre problemas puntuales de tipo minimax con restricciones [42].

1.2 El trabajo realizado

Otra posible generalización del problema básico 1-centro consiste en la consideración de servicios no puntuales, *localización minimax de estructuras dimensionales*. Morris y Norback [65] plantean el problema para rectas, resuelto en [53] en tiempo óptimo $\theta(n \log(n))$ para la métrica euclídea. Imai, Lee y Yang [47] proponen, para la misma métrica, la localización de un segmento de longitud fijada en el plano como problema intermedio entre el 1-punto y 1-recta centro.

Este marco de trabajo, sugiere la consideración de dos nuevas estructuras intermedias entre rectas y segmentos: semirrecta con origen en un punto prefijado y poligonal que conecte dos puntos situados fuera del rango de abscisas de los puntos de demanda. En esta memoria se plantea la localización de dichas estructuras para medidas de distancias alternativas a la euclídea, como las inducidas por las métricas l_∞ o l_1 y la distancia vertical. En la formulación de todos los problemas propuestos aparece, bien como objetivo o bien como restricción, la *excentricidad* de la estructura a localizar (máxima distancia de todos los puntos a la estructura), parámetro muy común a la hora de diseñar caminos en contextos donde se tiene en cuenta la interacción entre la estructura y los puntos entre los que ésta se instala.

Debido a la ausencia, en estos problemas, de propiedades como la diferenciabilidad o convexidad de la función a optimizar, así como a la propia naturaleza geométrica de los mismos, se ha creído conveniente el uso de técnicas de Geometría Computacional. En la mayoría de los casos, el estudio de propiedades geométricas de una solución nos ha permitido proponer algoritmos sencillos en su descripción que resuelven de forma eficiente las cuestiones planteadas.

Los resultados obtenidos en esta memoria se presentan agrupados en cuatro capítulos cuyo orden de aparición coincide con el orden temporal en que se fueron planteando como continuación de los problemas ya resueltos de localización de estructuras lineales. Así, en el capítulo 2 se recogen los principales resultados que se han obtenido para la localización óptima de rectas, semi-

rectas y segmentos cuando se toma el criterio minimax y se usa la métrica euclídea, haciendo hincapié en el uso de las técnicas de *calibres rotadores* y *análisis de casos* como herramientas comunes en la resolución de múltiples problemas. A partir de ahí, se discute el problema minimax con distancia vertical y se abordan los problemas no ponderados para rectas y semirrectas cuando se consideran métricas inducidas por las normas l_∞ y l_1 . Aunque para el caso de la recta, el problema se resuelve mediante programación lineal, el de la semirrecta encierra mayor complejidad y proponemos un algoritmo eficiente basado en la técnica de *poda y búsqueda*.

En el siguiente capítulo, se considera la localización de caminos poligonales de mínima longitud que conecten dos puntos a y b del plano prefijados de antemano y cuyas abcisas se encuentra a la izquierda y derecha respectivamente del rango de las abcisas de los puntos de demanda que conforman la entrada del problema. Se resuelven diversas situaciones, que dependen de la consideración de cotas inferiores o superiores en la excentricidad, para el caso de que los segmentos que componen el camino sean paralelos a los ejes de coordenadas (*caminos rectilíneos*). A la hora de buscar métodos de resolución para los problemas planteados, aparecen, como factores a tener en cuenta, el que la poligonal sea o no *x-monótona* (un camino se dice *x-monótono* si toda recta vertical lo intersecta en un punto o en un segmento vertical) y que el conjunto de puntos de entrada se encuentre o no en *posición degenerada*. De hecho, en la bibliografía de Geometría Computacional, una de las hipótesis que más frecuentemente se hacen sobre conjuntos de puntos en el plano es que no haya dos puntos que tengan la misma coordenada x , en cuyo caso se dice que la nube está en *posición general* o *no degenerada*. En algunas ocasiones, dicha hipótesis se realiza por claridad y simplicidad en la exposición. Sin embargo, para algunos algoritmos esa suposición es esencial pues no resuelven el problema para puntos con *degeneraciones*. Un ejemplo de tales algoritmos aparece en el trabajo de Mitchell, Snoeyink, Sundaram y Zhu [64] donde se muestra cómo generar uno de los posibles polígonos monótonos respecto a OX con vértices en un conjunto dado de puntos en \mathbb{R}^2 . Sobre la hipótesis de no degeneración en el conjunto de puntos existen trabajos recientes como los de Fortune [32], Gómez [37], Gómez, Ramaswami y Toussaint, [38]. Por

ello, tanto en este capítulo como en los posteriores se contemplan las distintas posiciones de los puntos de la entrada para los casos donde las degeneraciones son esenciales, así como consideración o no de caminos x -monótonos.

En el capítulo 4 se plantea la ubicación óptima de un camino rectilíneo para el criterio minimax y se usa la distancia vertical. Se propone un algoritmo eficiente cuando los puntos están en posición no degenerada y se considera una cota sobre el número de *codos* de la poligonal. Se propone asimismo en este capítulo un algoritmo de tiempo $O(n^2)$ para el caso en el que aparece el factor longitud en la restricción, situación muy común en el diseño de caminos.

En el último capítulo se aborda el problema minimax para el caso de poligonales generales y distancia vertical. A diferencia de los problemas desarrollados en los capítulos anteriores, esta cuestión aparece resuelta para casos particulares en un trabajo reciente de Hakimi y Schmeichel [40]. Nosotros proponemos aquí algoritmos que completan algunos casos no considerados en dicho trabajo, mejorando además la complejidad computacional de otros. Por otro lado, al igual que se hizo en el capítulo anterior, se introduce la longitud de la poligonal como factor de restricción, obteniendo un algoritmo de igual complejidad que los obtenidos por Hakimi y Schmeichel [40] para los problemas en cuya restricción aparece el número de codos de la poligonal.

Hemos escrito la memoria de manera que cada capítulo contenga un conjunto de problemas que provienen de la consideración de un factor u otro en la definición del problema de localización, de forma que sean autocontenidos. Sin embargo, las interrelaciones entre unos y otros son tantas que las referencias cruzadas han sido inevitables, de forma que se presentan las definiciones a medida que se van necesitando para la formulación y resolución de los problemas y se usan desde ese momento.

Capítulo 2

El problema minimax para servicios lineales

El problema del *1-centro puntual*, como ya se ha indicado en el capítulo anterior, ha sido extendido y generalizado a diferentes contextos. De todas las posibles variantes del problema básico, destacaremos en este capítulo aquéllas que, considerando servicios no puntuales, permiten una dualidad en su formulación, esto es, que pueden formularse, bien analíticamente como problema de Localización de Servicios, bien geoméricamente como problema de Geometría Computacional. Podemos citar, entre otros, los problemas no ponderados que provienen de considerar servicios no puntuales como rectas, semirrectas, segmentos, circunferencias o poligonales.

El problema no ponderado de la recta centro usando la métrica euclídea es equivalente al conocido problema de la anchura de un conjunto en Geometría Computacional, que posee diversas aplicaciones como, por ejemplo, en problemas de movimiento sin colisión [43]. Por su parte, la localización de un segmento de longitud dada con el criterio minimax fue propuesta por Imai, Lee y Yang [47] como problema intermedio entre el 1-centro y 1-recta centro. El problema de la circunferencia centro o su equivalente, el problema del anillo mínimo que contiene a un conjunto de puntos, está directamente relacionado con el campo de la Metrología Computacional, concretamente con la medida

de circularidad de un conjunto (se dice que un conjunto de puntos tiene circularidad p si existe un anillo de radio p que contiene al conjunto) y ha sido extensamente estudiado en Geometría Computacional [3], [50]. Por último, la ubicación de poligonales está relacionada con uno de los problemas más conocidos en Matemáticas: la aproximación a un conjunto de puntos, usando en este caso funciones lineales a trozos.

En este capítulo consideramos la localización de servicios que denominamos *lineales* (rectas, semirrectas y segmentos). Los problemas resultan esencialmente distintos cuando se consideran medidas de distancias diferentes, por lo que dividimos el capítulo en tres secciones, según la distancia usada sea la euclídea, la vertical o la inducida por las normas l_1 y l_∞ . En la primera sección recogeremos los principales resultados que se han obtenido para los correspondientes problemas de rectas, semirrectas y segmentos cuando se usa la métrica euclídea, haciendo hincapié en el uso de las técnicas de *calibres rotadores* y *análisis de casos* como herramientas comunes en la resolución de múltiples problemas. En la segunda sección se discute el problema minimax con distancia vertical y en la siguiente, se abordan los problemas no ponderados para rectas y semirrectas cuando se consideran métricas inducidas por las normas l_∞ y l_1 . Aunque para el caso de una recta, el problema se resuelve via programación lineal, el problema para una semirrecta encierra mayor complejidad y proponemos un algoritmo basado en la técnica *poda y búsqueda*.

2.1 Problemas con métrica euclídea

2.1.1 La recta centro

La localización de servicios lineales para el criterio minimax con métrica euclídea ha sido un problema que ha despertado interés en un gran número de investigadores correspondientes a distintos campos. El artículo de Morris y Norback (1980) [65] puede considerarse como el punto de partida. Ahí se presenta el problema de la 1-recta centro y se hacen caracterizaciones básicas que han supuesto la referencia a trabajos posteriores.

La formulación general del problema 1-recta centro es la siguiente. Dado un conjunto P de puntos de demanda $p_j = (x_j, y_j)$, $j = 1, \dots, n$ en el plano, y unos pesos asociados $w_j > 0$, $j = 1, \dots, n$, (para el caso no ponderado todos los pesos son iguales a 1), se trata de encontrar una recta $r \equiv ax + by + c = 0$ que resuelva el problema

$$\min_r (\max_j w_j d(p_j, r))$$

Si $d(p_j, r)$ representa la distancia euclídea de punto a recta, el problema puede formularse como

$$\min_{a,b,c} \{ \max_j w_j |ax_j + by_j + c|, a^2 + b^2 = 1 \}$$

El resultado básico que prueban Morris y Norback es

Teorema 2.1.1 ([65]) *Toda solución al problema 1-recta centro se encuentra a máxima distancia ponderada de al menos tres puntos de demanda.*

Esta caracterización de una solución es fundamental pues ofrece una “discretización” del problema en el sentido de que va a permitir buscar entre un conjunto finito de posibilidades. Esto sugiere usar un procedimiento general conocido como “análisis de casos” o “por casos candidatos” que usaremos frecuentemente como primer método de resolución en problemas análogos.

En efecto, la propiedad anterior permite resolver el problema considerando todos los conjuntos de tres puntos. Cada uno de éstos, determina tres rectas, una por cada dos puntos, los cuales fijan la pendiente, en tanto que el tercero determina la ordenada de la recta que está a la misma distancia ponderada de los tres puntos. La descripción del método *análisis de casos* será: (a) considerar todas las rectas determinadas por tres puntos equidistantes; (b) descartar las rectas para las que la distancia máxima a todos los puntos no viene dada por la distancia a los puntos equidistantes que la determinan; (c) de entre las restantes rectas nos quedamos con la de menor distancia máxima. Este procedimiento (digamos *ingenuo* o de *fuerza bruta*) es de tiempo $O(n^4)$, por lo que se debe hacer uso de métodos más eficientes. En particular, para el caso no ponderado, al que dedicaremos especial atención por poseer una clara

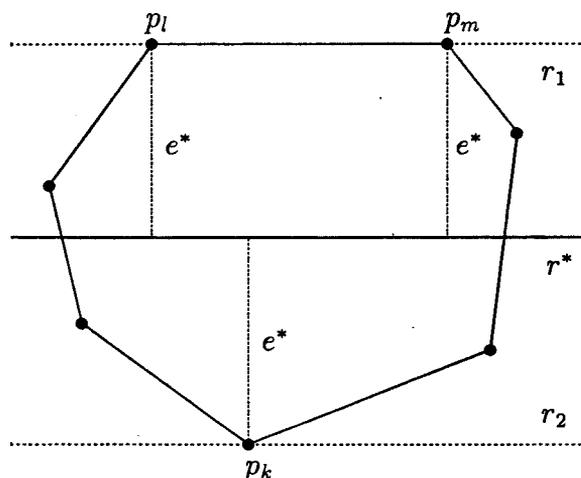


Figura 2.1: Recta candidato del problema 1-recta centro.

interpretación geométrica, el conjunto de puntos de demanda que determinan la solución deben encontrarse obviamente en el cierre convexo de los puntos. Además, en este caso, se cumplen las siguientes propiedades para una recta solución r^* de distancia máxima e^* :

1. Existen dos líneas paralelas de soporte sobre el cierre convexo de los puntos, r_1 y r_2 cuya distancia euclídea es $2e^*$; una de las líneas, digamos r_1 , pasa por dos puntos consecutivos del cierre convexo p_l , p_m , y la otra recta, r_2 pasa por otro punto del cierre p_k cumpliéndose que la base de la perpendicular desde p_k a r^* se encuentra comprendida entre las correspondientes a los puntos p_l y p_m ; y
2. La solución r^* está situada justo a la mitad de la banda determinada por las rectas r_1 y r_2 (ver la Figura 2.1).

Precisamente la versión no ponderada del problema de la 1-recta centro es equivalente, a un problema fundamental en Geometría Computacional como es el de la *anchura* de un conjunto. “La anchura de un conjunto de puntos P (o un polígono simple P) en dos dimensiones, es la menor distancia posible entre dos rectas paralelas que soportan a P ”. Houle y Toussaint [43] resuelven este problema usando el procedimiento denominado por Toussaint [85], técnica de

los *calibres* (*rotating calipers*). Básicamente el método se apoya en la propiedad geométrica de una solución descrita anteriormente y se puede resumir con la siguiente descripción: según se ha indicado, dos rectas de soporte r_1 y r_2 que corresponden a una solución, contienen a un vértice y a un lado del cierre convexo de los puntos, $CC(P)$, respectivamente. Llamando *par antipodal* a cada par (recta, vértice) correspondientes a dos rectas de soporte en las condiciones anteriores, se trata de obtener todos los posibles pares antipodales y tomar el que determine la banda de menor anchura. Una vez calculado el cierre convexo $CC(P)$, se enumeran todos los pares antipodales en tiempo $O(n)$, empezando con una banda inicial y rotando ésta alrededor de $CC(P)$. Como consecuencia, se obtiene un algoritmo de tiempo $O(n \log(n))$, donde domina el cálculo del cierre convexo.

Además, Lee y Wu [53] prueban, reduciendo el problema del “Maximun Gap” sobre un círculo al problema de la 1-recta centro, que $\Omega(n \log(n))$ es cota inferior. Como consecuencia de lo anterior, se tiene el siguiente teorema:

Teorema 2.1.2 ([53]) *El problema 1-recta centro no ponderado en el plano se resuelve en tiempo óptimo $\theta(n \log(n))$*

El tratamiento de la resolución de este problema desde distintos puntos de vista, como también se hizo en el problema del 1-centro puntual, da pie a tratar problemas propios del Localización de Servicios desde el punto de vista de la Geometría Computacional. En este contexto trabajaremos a lo largo de toda la tesis.

A continuación veremos que el caso ponderado puede reducirse al no ponderado. En efecto, el problema puede formularse como

$$\begin{array}{ll} \min & Z \\ Z, a, b, c & \text{s.a. } w_j |ax_j + by_j + c| \leq Z, j = 1, \dots, n, \\ a^2 + b^2 + c^2 = 1 & \end{array}$$

que puede transformarse (véase [65]) a

$$\begin{array}{ll} \min & Z \\ Z, a, b & \text{s.a. } \frac{w_j w_k}{w_j + w_k} |a(x_k - x_j) + b(y_k - y_j)| \leq Z, j \neq k \in [1, n]. \\ a^2 + b^2 = 1 & \end{array}$$

Nótese que $[w_j w_k / (w_j + w_k)] |a(x_k - x_j) + b(y_k - y_j)|$ es precisamente la distancia del punto “compuesto ponderado”

$$p_{jk} = \left(\frac{w_j w_k}{w_j + w_k} (x_k - x_j), \frac{w_j w_k}{w_j + w_k} (y_k - y_j) \right)$$

a la recta $ax + by = 0$ con $a^2 + b^2 = 1$ (que pasa por el origen).

Puede verse con facilidad que el valor óptimo Z_p está determinado por al menos dos puntos “compuestos ponderados”, que tienen la misma distancia (no ponderada) a la recta $ax + by = 0$. Si denominamos $S_c = \{p_{jk} : j, k = 1, \dots, n\}$ y D_c al conjunto de puntos “compuestos ponderados” que determinan la solución del problema, puede probarse el siguiente lema,

Lema 2.1.1 ([53]) *D_c es un subconjunto del cierre convexo de S_c y los puntos de D_c determinan al menos un par de ejes simétricos de la envolvente de S_c .*

Este resultado permite resolver el problema usando la técnica de los calibres donde domina el cálculo del cierre convexo del conjunto S_c que es, en este caso, de complejidad $O(n^2 \log(n))$. Como consecuencia, el problema 1-recta centro ponderado con la métrica euclídea en el plano se puede resolver en tiempo $O(n^2 \log(n))$.

Sin embargo, posteriormente, Houle et al. [44] demostraron que este problema es resoluble en tiempo óptimo $\theta(n \log(n))$, luego podemos establecer el siguiente resultado:

Teorema 2.1.3 ([44]) *El problema 1-recta centro ponderado con la métrica euclídea en el plano se puede resolver en tiempo $\theta(n \log(n))$.*

Una variante del problema 1-recta centro consiste en exigir a la recta óptima que pase por un punto prefijado. Si tomamos ese punto el origen de coordenadas, el problema 1-recta centro restringido puede expresarse como:

$$\begin{array}{ll} \min & Z \\ Z, a, b & \text{s.a. } w_j |ax_j + by_j| \leq Z, j = 1, \dots, n, \\ a^2 + b^2 = 1 & \end{array}$$

Si consideramos el conjunto de “puntos ponderados” $S_w = \{(w_j x_j, w_j y_j)\}$ obtenidos del conjunto de puntos de demanda original, aparece un problema no ponderado cumpliéndose que existen al menos dos puntos en S_w que están a distancia máxima de la recta óptima. Esto permite un algoritmo de búsqueda geométrica usando el concepto de pares antipodales [43], que encuentra la solución en tiempo $O(n \log(n))$. De forma análoga al caso 1-recta centro no ponderado se puede obtener la cota inferior $\Omega(n \log(n))$, lo que garantiza el óptimo en la complejidad,

Teorema 2.1.4 ([53]) *El problema 1-recta centro restringido se resuelve en tiempo óptimo $\theta(n \log(n))$.*

El problema euclídeo de la recta centro admite diversas *generalizaciones*, según se modifique el espacio ambiente (\mathbb{R}^2), el servicio (recta) o la demanda (conjunto de puntos).

Por una parte, manteniendo como servicio una recta, en 3-D aparece el problema del *mínimo cilindro generador* que posee diversas aplicaciones como por ejemplo en *tolerancia geométrica*. Usando la técnica de *búsqueda paramétrica* propuesta por Megiddo [59], Shömer et al. [83] resuelven el problema en tiempo $O(n^5 \log^{O(1)}(n))$. Por otro lado, si lo que aumentamos es la dimensión del servicio, aparece el problema de *aproximación de hiperplanos por un conjunto de puntos*, concretamente para el criterio minimax se habla de *aproximación l_∞* [44] (véase el trabajo de Korneenko y Martini [56]). Finalmente, Robert y Toussaint [78] consideran el problema de la recta centro con demanda no puntual (concretamente la demanda viene constituida por un conjunto de polígonos convexos), que resuelven en tiempo $O(n^2 \log(n))$.

Como *problema múltiple*, podemos destacar el problema 2-recta centro: dado un conjunto de puntos en el plano, D , encontrar dos bandas cuya unión contenga al conjunto de puntos de entrada P de forma que, la mayor de las anchuras de dichas bandas sea lo menor posible. Usando la técnica de *búsqueda paramétrica*, Agarwal y Sharir [2] proponen un algoritmo $O(n^2 \log^5(n))$ que resuelve el problema. Por último, el problema p-recta centro es NP-duro. La prueba es, en efecto, un caso particular de la que se muestra en el artículo de

Megiddo y Tamir [61].

2.1.2 La semirrecta centro

En el problema de la semirrecta centro se trata de encontrar una semirrecta que parte de un punto fijado del plano de forma que se minimiza la máxima distancia ponderada de los puntos de demanda a la semirrecta. Podemos suponer, sin pérdida de generalidad, que la semirrecta parte del origen y por tanto, la denotamos $s(\theta)$ donde θ es el ángulo, medido en sentido positivo, que forma la semirrecta con el eje x . Entonces, el problema podemos expresarlo como

$$\min_{Z, \theta} Z \quad \text{s.a. } w_j d(p_j, s(\theta)) \leq Z, \quad j = 1, \dots, n,$$

donde $d(p_j, s(\theta))$ denota la distancia euclídea entre el punto p_j y la semirrecta $s(\theta)$. Esta distancia tiene distinta interpretación según se tome sobre el punto inicial (origen) o sobre la recta $l(\theta)$ ($l(\theta)$ es la recta que contiene a $s(\theta)$). Por tanto, dada una semirrecta $s(\theta)$ y el conjunto de puntos de demanda, podemos distinguir dos tipos de puntos:

1. Los puntos p_j cuyo ángulo polar $\theta_j \in [\theta - \frac{\pi}{2}, \theta + \frac{\pi}{2}]$, para los que se cumple $d(p_j, s(\theta)) = d(p_j, l(\theta))$.
2. Los puntos de la región restante para los que se tiene $d(p_j, s(\theta)) = d(p_j, O)$.

Al igual que se hizo para el problema restringido 1-recta centro, podemos transformar el problema en uno no ponderado, sin más que considerar el conjunto $S_w = \{(w_j x_j, w_j y_j)\}$. Aunque el problema es más complejo pues debemos considerar dos tipos de distancia, podemos reformularlo como

$$\min_{\theta} Z(\theta)$$

donde

$$Z(\theta) = \max(\{X(\theta), Y(\theta)\}),$$

$$X(\theta) = \text{máx}\{d(p_j, l(\theta)), \theta_j \in [\theta - \frac{\pi}{2}, \theta + \frac{\pi}{2}]\}.$$

$$Y(\theta) = \text{máx}\{d(p_j, O), \theta_j \in [\theta + \frac{\pi}{2}, \theta + \frac{3\pi}{2}]\}.$$

Lee y Wu [53] muestran que, para buscar la solución del problema, es suficiente considerar los puntos de la envolvente de S_w y presentan un algoritmo de tiempo $O(n \log(n))$ que hace uso del método de “rotating calipers”. Sin embargo, en dicho trabajo dejan abierto el problema de encontrar una cota inferior. Así pues, se puede establecer el siguiente resultado,

Teorema 2.1.5 ([53]) *El problema 1-semirrecta centro euclídeo puede resolverse en tiempo $O(n \log(n))$.*

2.1.3 El segmento centro

En el trabajo de Imai, Lee y Yang [47] aparece el problema del 1-segmento centro: dado un conjunto de puntos en \mathbb{R}^2 y un segmento s de longitud dada l , encontrar el emplazamiento de s de forma que se minimice la máxima distancia (euclídea) de los puntos al segmento.

Cuando $l = 0$, el problema se reduce al problema del centro puntual, mientras que si l es suficientemente grande, tenemos el problema de la recta centro. Por tanto, este problema puede ser considerado intermedio entre los dos referenciados. Como sabemos, la complejidad de los problemas 1-punto centro y 1-recta centro es esencialmente diferente, lo que proporciona mayor interés precisamente al problema intermedio.

El segmento centro puede ser representado por (x, y, θ) , donde (x, y) son las coordenadas del punto inicial y θ es la orientación del segmento con respecto al eje x positivo.

Dado un segmento s de longitud l , representado por (x, y, θ) , el lugar geométrico de los puntos equidistantes de s a una distancia r se conoce como *hipódromo* centrado en (x, y, θ) y radio r . Obviamente el problema es equiva-

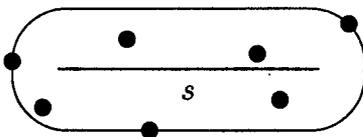


Figura 2.2: Hipódromo contenedor mínimo.

lente a encontrar el hipódromo de menor radio que contenga a todos los puntos de entrada (Figura 2.2).

Mientras tres puntos no colineales determinan un único círculo, Imai et al. [47] demuestran que para fijar un hipódromo son necesarios cuatro puntos. El algoritmo de Imai, Lee y Yang pasa por hacer un análisis exhaustivo de todos los posibles conjuntos de cuatro puntos y determinar las posibles situaciones de los puntos en las diferentes partes de la frontera del entorno o hipódromo del segmento solución. Este estudio permite resolver el problema en tiempo $O(n^5)$. Sin embargo, usando como preprocesado la envolvente convexa de los puntos y el diagrama de Voronoi del punto más lejano, puede reducirse sensiblemente la complejidad, como indica este teorema

Teorema 2.1.6 ([47]) *El problema 1-segmento centro euclídeo puede resolverse en tiempo $O(n^4 \log(n))$.*

Desde la publicación del artículo citado, han ido apareciendo otros cuyo objetivo ha sido reducir la complejidad. El trabajo de Agarwal, Efrat, Sharir y Toledo [3], reducen la complejidad del problema haciendo uso de un sofisticado algoritmo de programación basado en la técnica paramétrica de Megiddo. Posteriormente, Efrat y Sharir consiguen un algoritmo de complejidad cercana a la lineal [29].

Finalmente, podemos hacer referencia a algunos casos particulares como son el problema 1-segmento centro orientado o el problema 1-segmento centro con pivote, planteados por Imai et al. [47]. En el primero se considera fijada la orientación θ y puede resolverse en tiempo lineal usando la técnica de *poda y búsqueda*. En el segundo, se da como dato un punto p (llamado pivote) y

un valor $d > 0$ y se exige que el segmento solución esté en una recta que pase por p de forma que la distancia del pivote al segmento sea d . Este problema puede resolverse en tiempo $O(n)$ o $O(n^2)$ según el pivote p se encuentre en el interior o no del cierre convexo de los puntos dados inicialmente.

2.2 Problemas con distancia vertical

El problema 1-recta centro para la distancia vertical equivale a un conocido problema en teoría de regresión: el problema de aproximación de Chebyshev [77], cuya formulación es la siguiente: dados p_1, p_2, \dots, p_n en \mathbb{R}^2 , encontrar la recta r solución del problema

$$\min_r \max_i d_v(p_i, r).$$

Si r tiene por ecuación $y = b_0 + b_1x$, se define $d_v(p_i, r) = |y(p_i) - (b_0 + b_1x(p_i))|$ donde $x(p_i)$ e $y(p_i)$ representan las coordenadas cartesianas de p_i .

Para el caso no ponderado, Shamos [79] resolvió el problema en tiempo $O(n \log(n))$ haciendo uso de la técnica de los calibres comentada anteriormente, y demostró que $\Omega(n)$ es cota inferior. Sin embargo, el problema puede ser planteado como un problema de programación lineal en dimensión 3, que se resuelve en tiempo lineal (véase [60]). De hecho, el problema ponderado para una dimensión fijada d , cuya formulación es

$$\min_{b_0, \dots, b_d} \max_i w_i \left| y(p_i) - \sum_{j=2}^d b_j x(p_i) \right|$$

puede reformularse como un problema de programación lineal de dimensión $d+1$ resoluble igualmente en tiempo lineal (cuando la dimensión d está fijada) [60]. Así pues, el problema está cerrado y podemos enunciar el siguiente resultado:

Teorema 2.2.1 *El problema 1-recta centro con distancia vertical se resuelve en tiempo óptimo $\theta(n)$.*

Por otra parte, si consideramos como objetivo una semirrecta que parte de un punto fijado p_0 (sup. sin pérdida de generalidad que p_0 es el origen

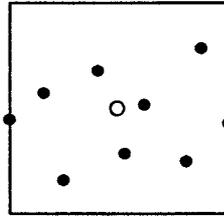


Figura 2.3: Cuadrado mínimo englobador.

de coordenadas), debemos imponer que la abcisa de p_0 se encuentre fuera del rango de abcisas de los puntos del *input*. En ese caso, el problema es equivalente al caso de la 1-recta centro restringido formulado como

$$\min_b \max_i |y(p_i) - bx(p_i)|$$

que puede ser reducido nuevamente a un problema de programación lineal y, como consecuencia, resuelto en tiempo lineal.

2.3 Problemas con métricas l_∞ y l_1

Las métricas l_∞ y l_1 se han usado con frecuencia en problemas de localización. Es sabido, por ejemplo que la distancia rectangular o l_1 se adapta mejor que la euclídea a ciertas situaciones como el la localización en un mallado determinado por las calles de una ciudad.

Centrándonos en el criterio minimax para problemas no ponderados, el problema del 1-centro para la métrica l_∞ es equivalente al problema del cuadrado mínimo que contiene a P y puede resolverse fácilmente. Dada la nube $P = \{p_i = (x_i, y_i); i = 1, 2, \dots, n\}$, el lado del cuadrado óptimo viene dado por

$$\max_{1 \leq i, j \leq n} \{|x_i - x_j|, |y_i - y_j|\},$$

por tanto, obtenemos la solución en tiempo $O(n)$ (Figura 2.3).

Para la métrica l_1 podemos usar el mismo mecanismo rotando previamente la nube de puntos un ángulo $\frac{\pi}{4}$, obteniendo el cuadrado generador para los puntos transformados que, tras la rotación correspondiente, se convertirá en el “rombo generador”. Por consiguiente, se resuelve en tiempo lineal. La razón de fondo de esta similitud entre los procedimientos usados para las métricas l_∞ y l_1 es que se puede establecer una isometría entre los espacios métricos correspondientes. Para ver la equivalencia entre las métricas inducidas por las normas l_1 y l_∞ denotemos por \mathbb{R}_∞^2 y \mathbb{R}_1^2 el espacio métrico \mathbb{R}^2 y d_1, d_2 las distancias inducidas cuando se usan las métricas l_∞ y l_1 respectivamente. La equivalencia está garantizada por el siguiente resultado:

Teorema 2.3.1 *Los espacios métricos \mathbb{R}_∞^2 y \mathbb{R}_1^2 son isométricos.*

Demostración: Sea la función $f : \mathbb{R}_\infty^2 \longrightarrow \mathbb{R}_1^2$ $f(x, y) = \left(\frac{y+x}{2}, \frac{y-x}{2}\right)$

Dados $p_i = (x_i, y_i)$ y $p_j = (x_j, y_j)$, supongamos $x_j > x_i, y_j > y_i$ y $x_j - x_i > y_j - y_i$. Se verifica que:

$$\begin{aligned} d_1(f(p_i), f(p_j)) &= \left| \frac{y_j + x_j}{2} - \frac{y_i + x_i}{2} \right| + \left| \frac{y_j - x_j}{2} - \frac{y_i - x_i}{2} \right| \\ &= \frac{((y_j - y_i) + (x_j - x_i)) + ((x_j - x_i) - (y_j - y_i))}{2} \\ &= \max\{(y_j - y_i), (x_j - x_i)\} = d_\infty(p_i, p_j). \end{aligned}$$

La prueba para los otros casos es análoga, con lo que se tiene que f conserva las distancias. Además, como puede comprobarse, f es biyectiva y f^{-1} conserva también las distancias. Por tanto, f es una isometría. \square

Por otro lado, f es la composición de un giro de centro el origen y ángulo $-\frac{\pi}{4}$ con una homotecia de centro el origen y razón $\frac{1}{\sqrt{2}}$, con lo que las rectas se transforman en rectas y las semirrectas con origen en el punto $(0, 0)$ en semirrectas con el mismo origen. Por tanto, basta resolver los problemas recta centro y semirrecta centro con una de las dos métricas. Megiddo [61]

probó, usando programación lineal, que el problema de la 1-recta centro es lineal para la métrica l_1 y por tanto, también para l_∞ . Así pues, obtenemos el siguiente resultado

Teorema 2.3.2 *El problema 1-recta centro con métrica l_∞ (l_1) se resuelve en tiempo óptimo $\theta(n)$.*

Centrémonos entonces en el problema de la 1-semirrecta centro. Cuando tratamos de localizar una semirrecta aparece la dificultad ya comentada en el caso euclídeo de que existen dos tipos de distancias a considerar: la distancia al punto inicial (que llamaremos *foco*), y la distancia sobre la recta que contiene a la semirrecta. Abordamos este problema considerando, por comodidad, la métrica l_∞ , en la siguiente sección.

2.3.1 La semirrecta centro

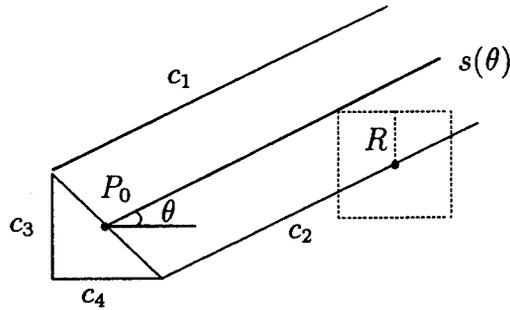
Dados un conjunto de puntos en el plano $P = \{p_i; i = 1, \dots, n\}$ y un punto p_0 (que denominaremos *foco*), el problema de la *semirrecta centro* consiste en encontrar una semirrecta $s(\theta)$, $\theta \in [0, 2\pi)$ que emane del punto p_0 que minimice la máxima distancia de los puntos a la semirrecta:

$$\min_{\theta} (\max_i d_\infty(p_i, s(\theta)))$$

Comenzaremos la resolución de este problema usando un procedimiento general conocido como *análisis de casos* y que ha sido usado en algunas ocasiones como en el problema del 1-segmento centro euclídeo.

Definición 2.3.1 Dado una semirrecta $s(\theta)$ que parte del punto p_0 , el entorno con centro $s(\theta)$ y radio R , $E_\infty(s(\theta), R)$, es el lugar geométrico de los puntos cuya distancia de Chebyshev a $s(\theta)$ es menor o igual que R .

La frontera de un entorno consta de dos tipos distintos de caras, las que corresponden a los puntos que toman su distancia sobre la recta que contiene

Figura 2.4: Entorno $E_\infty(s(\theta), R)$.

a la semirrecta y las que corresponden a puntos que la toman sobre p_0 . Las denotaremos c_1, c_2 y c_3, c_4 respectivamente (Figura 2.4).

Según la definición anterior, el problema puede formularse de la siguiente forma:

$$\begin{aligned} & \text{mín } R \\ & \text{s.a. : } P \subset E_\infty(s(\theta), R) \end{aligned}$$

Por tanto, encontrar una semirrecta óptima es equivalente a encontrar un entorno de radio mínimo que contenga a todos los puntos de P .

Lema 2.3.1 *Existe una solución $s(\theta^*)$ con al menos dos puntos de P a máxima distancia.*

Demostración: Es evidente que toda solución del problema contiene, al menos un punto de P en su frontera. Supongamos que sólo hay un punto en la frontera del entorno de una solución $s(\theta)$ y está en la cara c_1 o c_2 ; podría hacerse, entonces, con centro en p_0 , un pequeño giro de la semirrecta y obtener un entorno del mismo radio sin ningún punto en su frontera (Figura 2.5 (a)). Así, podría encontrarse un entorno de menor radio y $s(\theta)$ no sería solución.

Si el único punto situado en la frontera de un entorno óptimo estuviese en la cara c_3 o c_4 , girando con centro p_0 se obtendría otro entorno con el mismo radio y con algún otro punto más en c_1 o en c_2 (Figura 2.5 (b)). \square

De este modo, la consideración de todas las posiciones relativas de

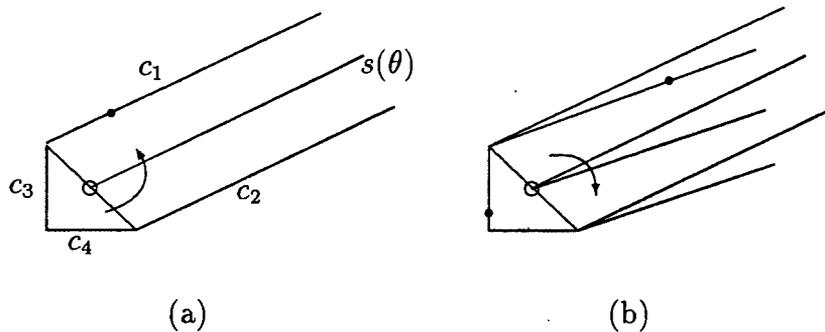


Figura 2.5: Giro de un entorno de semirrecta óptimo.

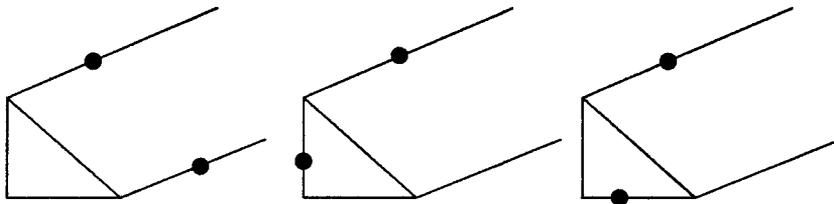


Figura 2.6: Casos considerados.

dos puntos en las caras del entorno resolverá el problema usando el método *análisis de casos*. Obsérvese que cuando los dos puntos estén sobre c_3 y c_4 , podrá encontrarse otro punto en c_1 o c_2 sin más que girar convenientemente. Por ello, no consideraremos este caso. Razonando de forma análoga tampoco contemplaremos los casos $\{c_1, c_4\}$ y $\{c_2, c_4\}$. Por tanto, las posibilidades de que dos puntos definan un posible entorno solución con un punto en cada una de sus caras son: $\{c_1, c_2\}$, $\{c_1, c_3\}$ y $\{c_2, c_3\}$ (Figura 2.6).

En el siguiente punto daremos un método geométrico que permite obtener el entorno para cada uno de los casos que se hace referencia más arriba, y por tanto, la determinación de la semirrecta que determina la situación de dos puntos a máxima distancia.

Lema 2.3.2 *Es posible construir geoméricamente un único entorno para cada uno de los casos dados por las posiciones relativas de dos puntos en la frontera.*

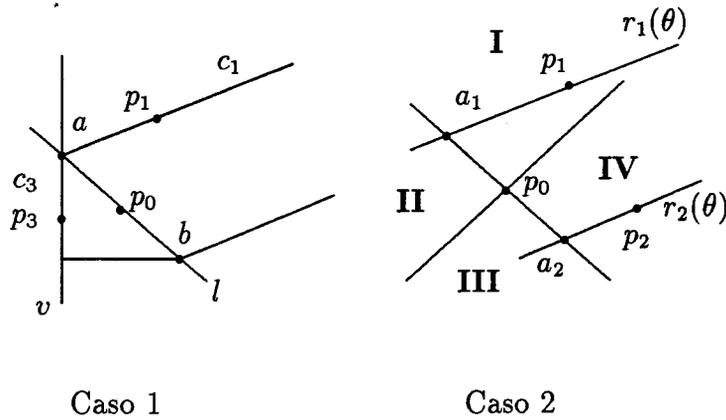


Figura 2.7: Construcción geométrica del entorno.

Demostración: Por simetría, podemos reducirnos a considerar dos casos: un punto en c_1 o c_2 y el otro en c_3 o c_4 y un punto en c_1 y el otro en c_2 .

Caso 1. Sin pérdida de generalidad consideraremos el caso $\{c_1, c_3\}$. Sean $p_0 = (x_0, y_0)$, $p_1 = (x_1, y_1) \in c_1$, $p_3 = (x_3, y_3) \in c_3$ y supongamos que $x_0 > x_3$, $x_0 - x_3 \leq y_1 - y_0$. Sea v la recta vertical pasando por p_3 y l la recta con pendiente -1 que pasa por p_0 (Figura 2.7. Caso 1). Denotemos por a el corte de l y v , con lo que el segmento ap_1 determina la cara c_1 . Considerar sobre l el punto b tal que p_0 sea el punto medio del segmento ab . El eje c_2 está determinado por la recta paralela a ap_1 que pasa por b , con lo que se concluye la construcción del entorno.

Cuando $x_0 > x_3$ y $x_0 - x_3 > y_1 - y_0$ no existe ningún entorno; y cuando $x_0 < x_3$ y $x_3 - x_0 \leq y_1 - y_0$ el procedimiento es similar tomando la recta L con pendiente 1 . Por último, el caso en que $x_0 < x_3$ y $x_3 - x_0 > y_1 - y_0$ tampoco determina ningún entorno.

Caso 2. Cuando los puntos están en c_1 y en c_2 , denotemos por I, II, III y IV las regiones del plano determinadas por el punto p_0 y las rectas que con pendiente 1 y -1 pasan por él (Figura 2.7. Caso 2). Si los puntos están en regiones opuestas, no existe ningún entorno que los contenga en esa posición. Por tanto vamos a suponer que p_1 y p_2 están en regiones adyacentes. Supondremos, sin pérdida de generalidad, que $p_1 \in I$ y $p_2 \in IV$.

Sea l la recta pasando por p_0 con pendiente -1 y $r_1(\theta), r_2(\theta)$ un par de rectas paralelas arbitrarias pasando por p_1 y p_2 respectivamente (ver Figura 2.7 Caso 2). Entonces, la solución está determinada por el ángulo θ para el cual p_0 es el punto medio del segmento de extremos $a_1(\theta)$ y $a_2(\theta)$, siendo $a_1(\theta)$ y $a_2(\theta)$ los puntos de corte de l con $r_1(\theta)$ y $r_2(\theta)$ respectivamente.

Finalmente, esta misma construcción es válida si los puntos están situados en la misma región, por lo que, en cualquiera de los casos, se puede determinar geoméricamente la semirrecta correspondiente. es idéntica \square

Basado en las propiedades anteriores proponemos el algoritmo siguiente:

ALGORITMO SRC1

Entrada: El conjunto de puntos dados P y p_0 .

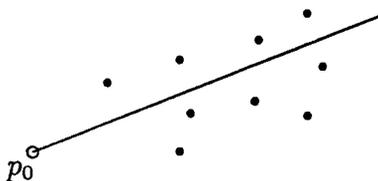
Salida: El ángulo θ^* de una semirrecta óptima y el radio R^* .

1. Para cada par de puntos de P , considerar los casos $\{c_1, c_2\}, \{c_1, c_3\}, \{c_1, c_4\}, \{c_2, c_3\}$ y $\{c_2, c_4\}$ y construir los correspondientes entornos (Lema 2.3.2).
2. Eliminar aquellos entornos que no contengan a todos los puntos.
3. Tomar el entorno de radio mínimo.

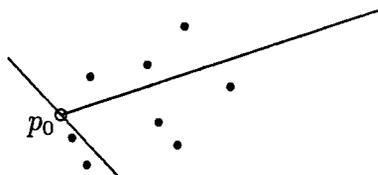
Teorema 2.3.3 *El problema de la semirrecta centro con métrica l_∞ puede resolverse en tiempo $O(n^3)$.*

Demostración: Puesto que elegir todos los pares requiere tiempo $O(n^2)$ y la construcción del entorno para un par dado es constante, el paso 1 se puede realizar en tiempo $O(n^2)$. El paso 2 requiere tiempo $O(n^3)$ (decidir si un entorno contiene a todos los puntos se hace en tiempo $O(n)$) que es dominante puesto que el paso 3 se realiza en tiempo $O(n^2)$. \square

A continuación y, con objeto de reducir la complejidad del problema, consideraremos una restricción: supondremos que el foco p_0 se encuentra en el exterior de $CC(P)$ (cierre convexo de P), como en la Figura 2.8.



Los puntos toman la distancia sobre la recta



Hay puntos que toman la distancia sobre P_0

Figura 2.8: Distintas posiciones del foco.

Obsérvese que cuando los puntos de P están suficientemente alejados de p_0 , de forma que la distancia a la semirrecta se tome sobre puntos distintos de p_0 (Figura 2.8), el problema coincide con el de la recta centro restringida a que pase por un punto dado, para el que puede usarse un método similar al propuesto por Schöbel [82] para rectas y que es lineal para cualquier norma bloque, en particular la métrica l_∞ . Sin embargo, el método de Schöbel no puede aplicarse si hay puntos que tomen la distancia sobre p_0 . Desarrollaremos aquí un procedimiento que resuelve el problema en cualquiera de los casos en tiempo lineal. Para describir el método, analizaremos con detalle algunas propiedades de la función distancia de los puntos a la semirrecta, que denominaremos $d_i := d_i(\theta) := d_\infty(p_i, s(\theta))$.

Con objeto de simplificar el análisis de la función d_i supondremos que $p_0 = (0, 0)$ y que todos los puntos de P tienen ordenada positiva, (si p_0 es exterior a $CC(P)$, los puntos están en un semiplano definido por una línea que pasa por p_0). La primera propiedad es un resultado técnico que resulta fundamental en la eficiencia del algoritmo que se va a proponer.

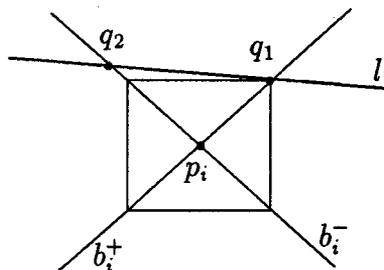


Figura 2.9: Distancia l_∞ entre un punto y una recta.

Lema 2.3.3 Para todo par $i \neq j$ las funciones d_i y d_j tienen un número finito de puntos de corte diferentes.

Demostración: La distancia d_∞ de un punto p a la recta l se define como:

$$d_\infty(p, l) = \min\{|\lambda| : (p + \lambda B) \cap l \neq \emptyset\} = \min\{|\lambda| : \exists q \in l, q \in (p + \lambda B)\}$$

donde B es la bola unidad de la norma l_∞ (Figura 2.9).

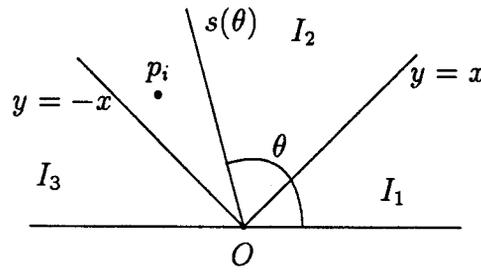
Sin embargo, para obtener la expresión de la función $d_i = d_\infty(p_i, s(\theta))$ consideraremos tres casos, según se encuentre el punto $p_i = \rho_i(\cos \theta_i, \sin \theta_i) = (x_i, y_i)$ en una de las regiones I_1, I_2 o I_3 determinadas por las rectas $y = x, y = -x$ y el eje OX (Figura 2.10):

$$d_i = \begin{cases} d_i^1, & \text{si } \theta_i \in [0, \frac{\pi}{4}] & (p_i \in I_1) \\ d_i^2, & \text{si } \theta_i \in [\frac{\pi}{4}, \frac{3\pi}{4}] & (p_i \in I_2) \\ d_i^3, & \text{si } \theta_i \in [\frac{3\pi}{4}, \pi] & (p_i \in I_3) \end{cases}$$

Caso 1. $p_i \in I_1$

Distiguiremos dos subcasos:

1.a) $\theta \in [0, \frac{\pi}{2})$. En primer lugar, calculamos el punto q_2 en el que se cortan la recta $r(\theta)$ con la paralela a $y = -x$ que pasa por p_i . Entonces,

Figura 2.10: Regiones I_1, I_2, I_3 .

si $b_i^- := y - y_i = -(x - x_i)$ y $r(\theta) := y = (\operatorname{tg} \theta)x$ obtenemos que $q_2 = (\frac{x_i + y_i}{1 + \operatorname{tg} \theta}, \frac{\operatorname{tg} \theta}{1 + \operatorname{tg} \theta}(x_i + y_i))$ (Figura 2.9). Por tanto:

$$d_i^1 = d_\infty(p_i, s(\theta)) = d_\infty(p_i, q_2) = \left| \frac{x_i + y_i}{1 + \operatorname{tg} \theta} - x_i \right|$$

1.b) $\theta \in [\frac{\pi}{2}, \pi]$. El cuadrado más pequeño centrado en $p_i \in I_1$ corta por primera vez a la semirrecta $s(\theta)$ en el punto $p_0 = (0, 0)$ con lo que $d_i^1 = d_\infty(p_i, s(\theta)) = d_\infty(p_i, p_0) = x_i$. Luego,

$$d_i^1(\theta) = \begin{cases} \left| \frac{x_i + y_i}{\operatorname{tg} \theta + 1} - x_i \right|, & \text{si } \theta \in [0, \frac{\pi}{2}) \\ x_i, & \text{si } \theta \in [\frac{\pi}{2}, \pi] \end{cases}$$

Caso 2. $p_i \in I_2$. Notemos b_i^+ y b_i^- a las rectas que pasan por p_i y son paralelas a $y = x$ y $y = -x$ respectivamente. La distancia del punto p_i a la semirrecta $s(\theta)$ es $d_\infty(p_i, s(\theta)) = \min\{d_\infty(p_i, q_1), d_\infty(p_i, q_2)\}$ donde $q_1 = s(\theta) \cap b_i^+$, y $q_2 = s(\theta) \cap b_i^-$ (Figura 2.9). Por tanto,

$$d_i^2(\theta) = \begin{cases} d_\infty(p_i, q_2), & \text{si } \theta \in [0, \frac{\pi}{2}] \\ d_\infty(p_i, q_1), & \text{si } \theta \in [\frac{\pi}{2}, \pi] \end{cases}$$

Puesto que $q_1 = (\frac{y_i - x_i}{\operatorname{tg} \theta + 1}, \frac{\operatorname{tg} \theta(y_i - x_i)}{\operatorname{tg} \theta - 1})$ y $q_2 = (\frac{x_i + y_i}{\operatorname{tg} \theta + 1}, \frac{\operatorname{tg} \theta(x_i + y_i)}{\operatorname{tg} \theta + 1})$

entonces

$$d_i^2(\theta) = \begin{cases} \left| \frac{x_i + y_i}{\operatorname{tg} \theta + 1} - x_i \right|, & \text{si } \theta \in [0, \frac{\pi}{2}) \\ \left| \frac{y_i - x_i}{\operatorname{tg} \theta - 1} - x_i \right|, & \text{si } \theta \in (\frac{\pi}{2}, \pi) \\ |x_i|, & \text{si } \theta = \frac{\pi}{2} \end{cases}$$

Caso 3. $p_i \in I_3$

Un argumento similar al del caso 1 nos lleva a la siguiente expresión:

$$d_i^3(\theta) = \begin{cases} |x_i|, & \text{si } \theta \in [0, \frac{\pi}{2}] \\ \left| \frac{y_i - x_i}{\operatorname{tg} \theta - 1} - x_i \right|, & \text{si } \theta \in (\frac{\pi}{2}, \pi) \end{cases}$$

Por tanto, debemos considerar tres tipos de funciones (ver Figura 2.11). Introduciendo el cambio de variable $x = \cos \theta$ obtenemos:

$$\begin{aligned} 1. \quad & \left| \frac{y_i - x_i}{\operatorname{tg} \theta - 1} - x_i \right| = \left| \frac{(y_i - x_i) \cos \theta}{-\cos \theta + \operatorname{sen} \theta} - x_i \right| = \left| \frac{(y_i - x_i)x}{-x + \sqrt{1 - x^2}} - x_i \right| = \\ & = \left| \frac{K_1 x}{\sqrt{1 - x^2} - x} - K_2 \right| \\ 2. \quad & \left| \frac{x_i + y_i}{\operatorname{tg} \theta + 1} - x_i \right| = \left| \frac{(x_i + y_i) \cos \theta}{\operatorname{sen} \theta + \cos \theta} - x_i \right| = \left| \frac{(x_i + y_i)x}{x + \sqrt{1 - x^2}} - x_i \right| = \\ & = \left| \frac{K_3 x}{x + \sqrt{1 - x^2}} - K_4 \right| \end{aligned}$$

LLamando $D_i(x)$ y $\widetilde{D}_i(x)$ a las funciones de 1 y 2 respectivamente, se tiene que,

$$d_i^1(x) = \begin{cases} \widetilde{D}_i(x), & \text{si } x \in [0, 1] \\ C_i, & \text{si } x \in [-1, 0] \end{cases}$$

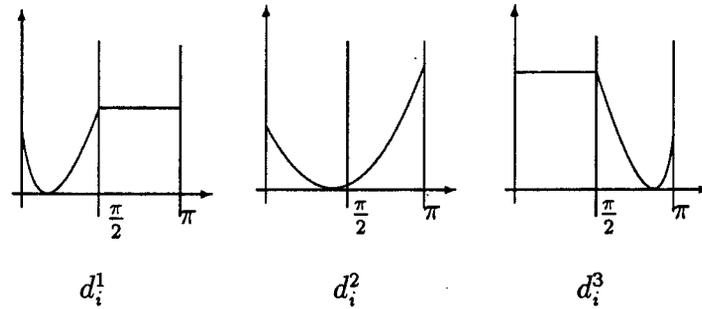


Figura 2.11: Distintos tipos de funciones distancia.

$$d_i^2(x) = \begin{cases} \widetilde{D}_i(x), & \text{si } x \in [0, 1] \\ D_i, & \text{si } x \in [-1, 0] \end{cases}$$

$$d_i^3(x) = \begin{cases} C_i, & \text{si } x \in [0, 1] \\ D_i, & \text{si } x \in [-1, 0] \end{cases}$$

Por tanto, para calcular los puntos de corte de d_i y d_j tendremos que considerar tres tipos distintos de ecuaciones:

- (i) $D_i(x) = D_j(x)$, $\widetilde{D}_i(x) = D_j(x)$ o $\widetilde{D}_i(x) = \widetilde{D}_j(x)$ en cuyo caso, puesto que el grado de la ecuación que resulta es 4, existen a lo sumo 16 puntos de corte.
- (ii) $D_i(x) = C_j$, $\widetilde{D}_i(x) = K_j$ que tienen 4 soluciones a lo sumo.
- (iii) $C_i = C_j$ en cuyo caso sólo consideramos como puntos de corte los extremos del intervalo, es decir, 0 y $\frac{\pi}{2}$ o $\frac{\pi}{2}$ y π . \square

A continuación se exponen dos resultados cruciales para nuestro algoritmo, basados en el hecho de que la función distancia de un punto a una semirrecta tiene un único mínimo en el intervalo que estamos considerando.

Lema 2.3.4 *Dada una semirrecta con ángulo $\theta' \in [0, \pi]$ se puede decidir en tiempo $O(n)$ si existe una solución $s(\theta^*)$ con ángulo mayor, menor o igual que*

θ' .

Demostración: Nótese que la solución del problema considerado consiste en encontrar el mínimo de la función $f(\theta) = \max_{1 \leq i \leq n} d_i(\theta)$ donde las funciones $d_i(\theta)$ tienen un único mínimo local (que a su vez es global) que denotaremos por θ_i^* . Este mínimo se toma en la única semirrecta que emana del origen y pasa por el punto p_i . Por tanto, $d_i(\theta_i^*) = 0$ y se requiere un tiempo de $O(n)$ para calcular θ_i^* , $i = 1, \dots, n$ ya que es constante para cada punto.

Dado $\theta' \in [0, \pi]$ podemos determinar en tiempo $O(n)$ aquellos puntos p_j , $j = 1, \dots, p$ para los cuales $d_j(\theta') = \max_{1 \leq i \leq n} d_i(\theta')$. Entonces tenemos las siguientes posibilidades:

1. $\theta' < \theta_j^* \quad \forall j = 1, \dots, p.$

En este caso, debido a la naturaleza de las funciones, existe un entorno a la derecha de θ' donde las funciones d_j son decrecientes y como consecuencia $d_j(\theta') \geq d_j(\theta)$, $j = 1, \dots, p$, $\forall \theta \in [\theta', \theta' + \varepsilon]$. Así pues, $f(\theta') \geq f(\theta)$ $\forall \theta \in [\theta', \theta' + \varepsilon]$ con lo que existe θ^* con $\theta' \leq \theta^*$.

2. $\theta' > \theta_j^* \quad \forall j = 1, \dots, p.$

Ahora, las funciones d_j son no decrecientes en un entorno a la izquierda de θ' . Así, $f(\theta') \geq f(\theta) \quad \forall \theta \in [\theta' - \varepsilon, \theta']$ y existe θ^* con $\theta^* \leq \theta'$.

3. Para algunos $j_1 \in \{1, 2, \dots, p\}$ es $\theta' > \theta_{j_1}^*$ y para algunos $j_2 \in \{1, 2, \dots, p\}$ $\theta' < \theta_{j_2}^*$.

En este caso θ' es un punto de no decrecimiento para las funciones d_{j_1} cuyos $j_1 \in \{1, 2, \dots, p\}$ para los cuales $\theta' > \theta_{j_1}^*$. Mediante un argumento similar al del caso 2, podemos probar que $f(\theta') \leq f(\theta) \quad \forall \theta \in [\theta', \theta' + \varepsilon]$. Análogamente, θ' es un punto de no crecimiento para las funciones d_{j_2} con $j_2 \in \{1, 2, \dots, p\}$ siendo $\theta' < \theta_{j_2}^*$. Entonces de forma similar al caso 1 obtenemos que $f(\theta') \leq f(\theta)$, $\forall \theta \in [\theta' - \varepsilon, \theta']$.

En definitiva

$$f(\theta') \leq f(\theta) \quad \forall \theta \in [\theta', \theta' + \varepsilon]$$

$$f(\theta') \leq f(\theta) \quad \forall \theta \in [\theta' - \varepsilon, \theta']$$

y entonces $\theta^* = \theta'$.

□

Lema 2.3.5 *Si θ_1 y θ_2 son dos soluciones del problema de la semirrecta centro con distancia l_∞ , entonces todo $[\theta_1, \theta_2]$ es un intervalo de soluciones.*

Demostración: Puesto que todas las funciones $d_i(\theta)$ son explícitamente cuasiconvexas [57] también lo es la función $f(\theta) = \max_{1 \leq i \leq n} f_i(\theta)$. Como consecuencia inmediata, $f(\theta)$ constante en el intervalo $[\theta_1, \theta_2]$.

□

ALGORITMO SMC2

Entrada: El conjunto de puntos dados $P = \{p_i, i = 1, \dots, n\}$.

Salida: El ángulo θ^* de una semirrecta óptima.

1. Hacer $P = \{p_i, i = 1, \dots, n\}$ y $K = 1$.
2. Distribuir aleatoriamente en pares $\{p_i, p_j\}$, $i \neq j$, los puntos de P .
3. Para cada par $\{p_i, p_j\}$ determinar los ángulos que corresponden a los puntos de corte de sus respectivas funciones distancia, como se indica en el lema 2.3.3. Sea $R = \{\theta_1, \dots, \theta_q\}$ el conjunto formado por los ángulos asociados a todos los pares.
4. Tomar $C > \log q$ un entero positivo. (Teorema 2.3.4)
5. Determinar la media θ_m de los ángulos de R , y aplicando el lema 2.3.4 decidir si existe un ángulo óptimo $\bar{\theta}$ mayor, menor o igual que θ_m .
 - (a) Si $\theta_m = \bar{\theta}$, θ_m es una solución. Salir con $\theta^* = \theta_m$.
 - (b) Si $\theta_m > \bar{\theta}$ eliminar de R aquellos ángulos θ tales que $\theta > \theta_m$.

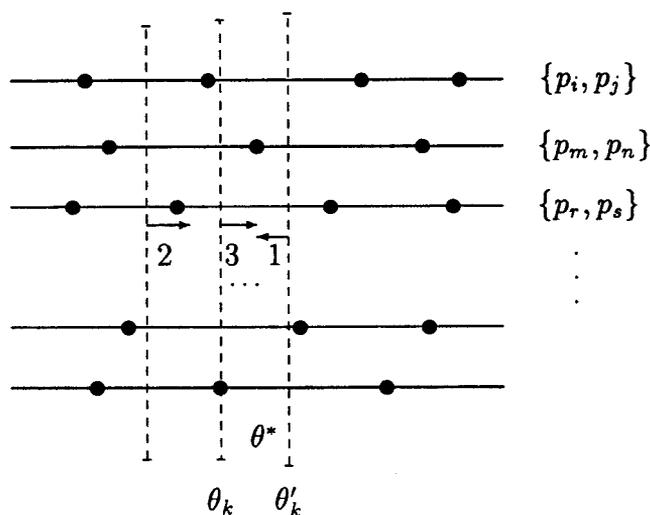


Figura 2.12: Pasos 5, 6 y 7 del algoritmo.

- (c) Si $\theta_m < \bar{\theta}$ eliminar de R aquellos ángulos θ tales que $\theta < \theta_m$.
6. Hacer $C = C - 1$. Si $C > 0$ ir al paso 5 (ver Figura 2.12).
7. Sea $[\theta_k, \theta'_k]$ el intervalo obtenido mediante el método de bisección descrito en los pasos 5 y 6. (Del lema 2.3.4 se deduce la existencia de al menos una solución θ^* en este intervalo).
- Para todos los pares de puntos $\{p_i, p_j\}$ sin ángulos de intersección en R , decidir el intervalo de ángulos asociados a cada par donde se encuentra $[\theta_k, \theta'_k]$ (que incluye a una solución).
- Para cada uno de esos pares, obtener el punto cuya función distancia es más pequeña en este intervalo y eliminarlo de P .
8. Si $|P| \leq 4$, construir geoméricamente la solución. Sea θ^* el ángulo asociado a esa solución y tomar $[\theta, \theta'] = \bigcap_{i=1}^k [\theta_i, \theta'_i]$, entonces tenemos,
- Si $\theta^* \in [\theta, \theta']$ entonces θ^* es una solución.
 - Si $\theta^* \notin [\theta, \theta']$ entonces θ o θ' es una solución (lema 2.3.5).
- En otro caso ($|P| > 4$), hacer $K = K + 1$ e ir al paso 2.

Teorema 2.3.4 *El problema de la semirrecta centro con distancia l_∞ puede resolverse en tiempo $O(n)$ si p_0 está fuera de $CC(P)$.*

Demostración: Analicemos la complejidad del Algoritmo 2. Para n puntos dados, el conjunto R contiene al menos $\frac{n}{2}q$ ángulos, donde q es el número máximo de ángulos de corte (constante según el lema 1.3.3). Ya que en cada iteración se elimina de R la mitad de los ángulos, el número total de ángulos en R cuando se completan las C iteraciones es $\frac{n}{2^C}q$. Estos ángulos corresponden a lo sumo a $\frac{n}{2^C}q$ pares de puntos. Por tanto, quedan $\frac{n}{2} - \frac{n}{2^C}q$ pares de puntos cuyos ángulos asociados no están en R . Aplicando el paso 7, de cada uno de estos pares se elimina un punto quedando así $\frac{n}{2} - \frac{n}{2^C}q = \alpha n$ puntos. Luego, tomando $C > \log q$ conseguimos que sea $0 < \alpha < 1$ y obtenemos la siguiente ecuación recursiva,

$$T(n) = T((1 - \alpha)n) + O(n), \quad 1 - \alpha < 1$$

de donde se sigue que $T(n) = O(n)$ □

El hecho fundamental que ha permitido el funcionamiento del algoritmo anterior es la unicidad de mínimos locales de una función distancia $d_i(p_i, s(\theta))$. Esta propiedad está asegurada como hemos visto en cierto intervalo de amplitud π . Esto nos ha sugerido resolver el problema general (el punto p_0 puede ser interior al cierre convexo) como indica el siguiente resultado,

Teorema 2.3.5 *El caso general del problema puede resolverse en tiempo $O(n^2)$.*

Demostración: Si p_0 se encuentra en el interior del cierre convexo del conjunto P , las funciones d_i que miden la distancia de un punto p_i a una semirrecta anclada en p_0 no son cuasiconvexas, esto es, no tenemos asegurada la unicidad de mínimos locales. Sin embargo, denotando por θ_i el ángulo correspondiente el segmento p_0p_i , resulta fácil observar que, en cada intervalo $(\theta_i + \pi, \theta_{i+1} + \pi)$, la función d_i sí posee un único valor mínimo local para $i = 1, 2, \dots, n$ (ver Figura 2.13). De esta manera, las propiedades requeridas para ejecutar el Algoritmo 2 se tienen en cada intervalo. Así pues, es suficiente resolver n problemas en

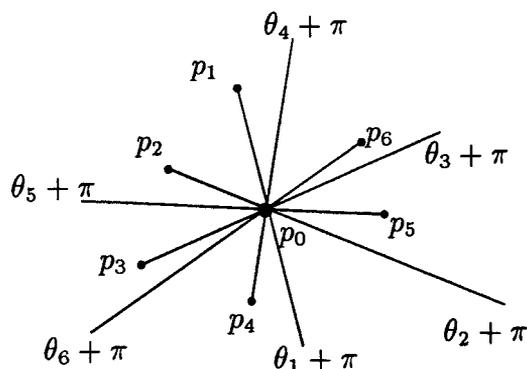


Figura 2.13: p_0 está en el interior de $CC(P)$.

tiempo lineal (obtener una semirrecta óptima para cada intervalo) y escoger la mejor solución. Como consecuencia, resulta un algoritmo de tiempo $O(n^2)$.

□

2.4 Conclusiones y problemas abiertos

A lo largo de este capítulo ha quedado patente la doble visión que permiten los problemas no ponderados de criterio minimax para servicios puntuales y lineales. Así, desde el punto de vista de la Geometría Computacional, en el problema general, se trata de encontrar el menor “entorno” que contiene a una nube de puntos. Esto permite resolver el problema, en primera instancia, encontrando una caracterización de la (una) solución por puntos de la frontera del entorno y hacer uso del método que hemos denominado *análisis de casos*.

Generalmente, este procedimiento no produce algoritmos de aceptable complejidad por lo que se requieren herramientas más potentes, usuales en Geometría Computacional, como el cierre convexo de un conjunto de puntos, los diagramas de Voronoi, la técnica de los calibres o algoritmos de poda y búsqueda.

Existen algoritmos eficientes que resuelven los problemas euclídeos de rectas, semirrectas y segmentos. Aquí hemos ampliado el estudio para otro tipo de métricas muy usuales en problemas de Localización de Servicios como son la métrica de Manhattan o l_1 y la métrica de Chebyshev o l_∞ . Se ha mostrado que es suficiente resolver los problemas de puntos, rectas y semirrectas para una sola métrica (por ejemplo, l_∞). Si bien, para el problema de la recta centro podemos hacer uso del método propuesto por Megiddo [61], para el problema de la semirrecta que parte de un punto fijado p_0 , ha sido necesario un minucioso estudio de la función distancia de punto a semirrecta. Esto nos ha conducido a un algoritmo de tiempo $O(n^2)$ en el que hacemos uso de la técnica de poda y búsqueda y de la construcción geométrica de la semirrecta determinada por dos puntos a máxima distancia. Este resultado ha sido presentado en un congreso de ámbito español [17] y constituye la parte principal del trabajo [18].

Servicios lineales	minmax d_2	minmax d_∞, d_1	minmax d_v
Recta	$\theta(n \log(n))^*$	$O(n)^+$	$\theta(n)^{++}$
Semirrecta	$O(n \log(n))^*$	$O(n^2)$	$O(n)$
Segmento	$O(n^{1+\epsilon})^{**}$	ABTO	N. A.

Tabla 2.4. Localización minimax de servicios lineales en el plano.

* Lee y Wu [53]. ** Efrat y Sharir [29]. + Schöbel [82]. ++ Megiddo [60]. ABTO: Problema abierto. N. A.: No aplicable.

Para el caso del segmento centro puede hacerse uso de la técnica general “análisis de casos” y encontrar un algoritmo similar al que Imai et al. [47] propusieron para el caso euclídeo. Sin embargo, el método es laborioso y la complejidad obtenida es de tipo $O(n^5)$. Este hecho, unido a la posibilidad de que este tipo de métricas permita el uso de técnicas más adecuadas, nos ha llevado a plantear la cuestión en esta tesis como problema abierto. En la Tabla 2.4 aparecen los resultados actuales, de menor complejidad computacional, que han sido propuestos para problemas de localización minimax de servicios lineales.

Desde luego, en el amplio campo de la localización de servicios lineales quedan abiertos una gran cantidad de problemas como el de encontrar cotas

inferiores para el problema de la semirrecta tanto con métrica euclídea como con l_1 o l_∞ , los problemas que provienen de considerar otras normas como las denominadas “bloque” o los que se pueden plantear cuando aumentamos la dimensión del espacio ambiente a 3-D.

Capítulo 3

Caminos rectilíneos más cortos de rechazo y de cobertura.

Una de las posibles extensiones de problemas de localización de servicios lineales es la consideración de servicios lineales a trozos, esto es, poligonales. Se trata, por tanto, de encontrar una poligonal que optimice cierta función objetivo. Además, se pueden considerar dos tipos diferentes de poligonales: las *poligonales genéricas* y las *poligonales rectilíneas u ortogonales*, según que los segmentos que la compongan sean generales o bien sólo horizontales y verticales.

En campos tales como robótica (*planning robot motion*) o diseño de circuitos integrados (*VLSI layout design*), aparecen problemas de optimización cuyo objetivo es encontrar trayectorias o caminos ortogonales, también llamados L_1 -caminos. De hecho, un tema extensamente tratado en Geometría Computacional es el problema de buscar la trayectoria rectilínea más corta entre dos puntos evitando una serie de obstáculos, para el que podemos citar trabajos básicos como el de Mitchell y Papadimitriou [63], o más específicos como el de Yang [91]. Por otro lado, el estudio de este problema viene motivado en el contexto de Localización de Servicios por aplicaciones en la construcción de rutas urbanas [88]. Así, nosotros trabajaremos en este capítulo con poligonales que llamaremos rectilíneas (por afinidad con el inglés) cuya interpretación

en el contexto de Localización de Servicios va a corresponderse con caminos o trayectorias que conectan dos puntos a y b del plano y atraviesan una nube de puntos de demanda $P = \{p_1, p_2, \dots, p_n\}$ comprendida entre a y b (esto es, $x(a) < x(p_i) < x(b)$, $i = 1, 2, \dots, n$).

Definición 3.0.1 Definiremos *trayectoria* o *camino rectilíneo* R , uniendo a y b , como un conjunto de segmentos (horizontales o verticales) consecutivos conectando a y b ,

$$R = \{\overline{c_0c_1}, \overline{c_1c_2}, \dots, \overline{c_kc_{k+1}}\}, \quad c_0 = a, \quad c_{k+1} = b.$$

Así pues, cada segmento $\overline{c_i c_{i+1}}$ es horizontal o vertical y diremos que $q \in R$ si $q \in \overline{c_i c_{i+1}}$ para algún $i = 1, \dots, k$.

Definición 3.0.2 Diremos que una trayectoria o camino R que conecta a y b es *x-monótono*(*y-monótono*) si es monótono en la dirección de OX(OY), esto es, cumple que la intersección de la trayectoria con cualquier recta vertical(horizontal) es un conjunto conexo. Un camino se dice *monótono* si es x-monótono o y-monótono.

La monotonía del camino buscado es de gran utilidad en la resolución de algunos de los problemas que se van a plantear pues permitirá, en su caso, usar técnicas de barrido. Así, para caminos rectilíneos que atraviesan una nube de puntos, la monotonía en la dirección de OX será una propiedad crucial.

En el presente capítulo, de la misma forma que aparece en los citados, el objetivo va a ser minimizar la longitud de la trayectoria. Desde luego, si queremos abordar un problema de localización de servicios, debemos imponer alguna condición a la trayectoria con respecto a los puntos de la nube P . Así, podemos plantear dos problemas fundamentales que pueden ser considerados de localización-rutas: el primero, se obtiene de imponer al camino R una medida de separación de los puntos y el segundo, de cercanía. Concretamente, dada una constante no negativa k , se tienen los problemas siguientes:

$$\text{mín } l(R) \quad \text{s.a. : } d(p_i, R) \geq k \quad (RMA)$$

$$\text{mín } l(R) \quad \text{s.a. : } d(p_i, R) \leq k \quad (\text{RMC})$$

donde $d(p_i, R) = \text{mín}_{q \in R} d(p_i, q)$ es la distancia del punto p_i al camino R .

El primer problema (*Trayectoria Rectilínea Mínima de Anticobertura o rechazo*), aparece cuando p_i representa un punto nocivo o peligroso, mientras que en el segundo (*Trayectoria Rectilínea Mínima de Cobertura*) los puntos se identifican con clientes que deben ser abastecidos durante el recorrido R .

3.1 Estudio del problema RMA.

En esta sección haremos hincapié en la formulación geométrica del problema RMA según sea la distancia a considerar. En primer lugar, haremos la traducción del problema para los casos de distancias inducidas por las métricas l_2 , l_1 y l_∞ . Resulta bien conocido que toda norma puede ser caracterizada por su bola unidad. Así pues, considerando los conjuntos $B_p(p_i, \lambda) = \{p \in \mathbb{R}^2 : d_p(p, p_i) \leq \lambda\}$, para cada $i = 1, \dots, n$ y cada $p = 1, 2, \infty$, podemos establecer el siguiente resultado:

Lema 3.1.1 *Dado un punto $p_i \in P$ y un camino rectilíneo R , se tiene que $d_p(p_i, R) = \text{mín}\{|\lambda| : B_p(p_i, \lambda) \cap R \neq \emptyset\}$.*

$$\begin{aligned} \text{Demostración: } d_p(p_i, R) &= \text{mín}_{q \in R} d_p(p_i, q) = \\ &= \text{mín}_{q \in R} (\text{mín}\{|\lambda| : q \in B_p(p_i, \lambda)\}) = \\ &= \text{mín}(\text{mín}_{q \in R}\{|\lambda| : q \in B_p(p_i, \lambda)\}) = \\ &= \text{mín}\{|\lambda| : \exists q \in R, q \in B_p(p_i, \lambda)\} = \\ &= \text{mín}\{|\lambda| : B_p(p_i, \lambda) \cap R \neq \emptyset\}. \quad \square \end{aligned}$$

Como consecuencia directa, si denotamos $C_i^p(\lambda) = \{p \in \mathbb{R}^2 : d_p(p, p_i) < \lambda\}$, se deduce que $d_p(p_i, R) \geq k$ equivale a que $C_i^p(k) \cap R = \emptyset$. Esto nos conduce

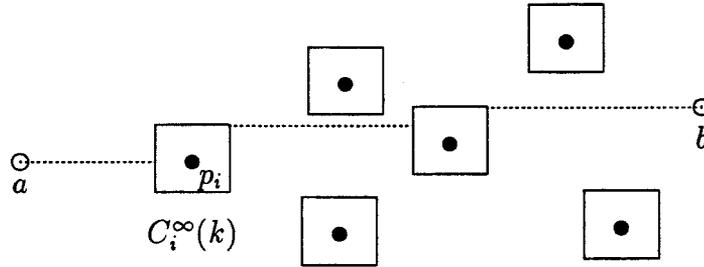


Figura 3.1: Trayectoria más corta a través de obstáculos rectangulares.

a plantear la transformación del problema a una cuestión de trayectorias más cortas a través de obstáculos. De esta forma, si el problema geométrico de obstáculos está resuelto, la resolución se compone de dos pasos: la transformación o reducción del problema y el algoritmo de cálculo que resuelve el problema geométrico. Desafortunadamente, la reducción a un problema de camino más corto que evita una serie de obstáculos, es a veces costosa y no siempre el correspondiente problema geométrico está resuelto. Sin embargo, veremos cómo para el caso de la norma l_∞ pueden obtenerse interesantes resultados.

Cuando se considera la distancia inducida por la métrica l_∞ , el problema RMA se formula como

$$\text{mín } l(R) \quad \text{s.a. : } d_\infty(p_i, R) \geq k,$$

donde $d_\infty(p_i, R) = \text{mín}_{q \in R} d_\infty(p_i, q)$. Estudiamos la posible transformación a un problema de obstáculos dependiendo de la posición relativa de los puntos de P . Si ocurre que los conjuntos C_i^∞ definidos para cada p_i no intersecan, $C_i^\infty(k) \cap C_j^\infty(k) = \emptyset$, $i \neq j$, $i, j = 1, \dots, n$, entonces el problema se reduce a encontrar la trayectoria rectilínea más corta entre a y b que evita un conjunto de obstáculos *rectangulares* disjuntos (cuadrados en nuestro caso) como se indica en la Figura 3.1. En la bibliografía se encuentran un gran cantidad de trabajos que se refieren a trayectorias que evitan un conjunto de obstáculos disjuntos. Para este caso (obstáculos rectangulares), puede aplicarse el algoritmo de DeRezende et al. [16] que resuelve dicho problema en tiempo óptimo $\theta(n \log(n))$. Puesto que en este caso, se obtienen los obstáculos en tiempo

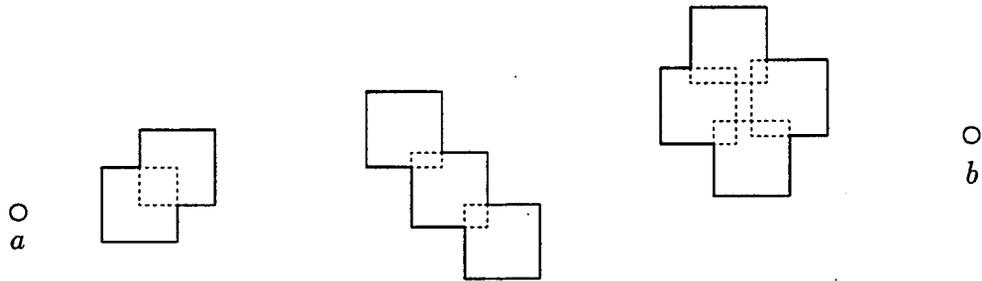


Figura 3.2: Obtención de la partición maximal.

$O(n)$, se resuelve el problema original en tiempo $O(n \log(n))$.

Si, por el contrario, existen $i, j \in \{1, 2, \dots, n\}$ tal que los conjuntos $C_i^\infty(k)$, $C_j^\infty(k)$, $i \neq j$ tienen intersección no vacía, esto es, $|x(p_i) - x(p_j)| < k$ y $|y(p_i) - y(p_j)| < k$, se puede transformar el problema en uno de trayectoria más corta a través de polígonos *rectilíneos u ortogonales* (de lados son paralelos a los ejes de coordenadas) que son disjuntos entre sí. Para ello, debemos obtener los polígonos que provienen de hacer *uniones máximas* de cuadrados con intersección no vacía (Figura 3.2).

Definición 3.1.1 Dado un conjunto de cuadrados $C = \{C_1, C_2, \dots, C_n\}$ en \mathbb{R}^2 ordenados lexicográficamente (a partir de la ordenación de sus centros), llamaremos *partición maximal* de C , a la partición disjunta que se puede establecer en C , de forma que para cada conjunto de la partición se cumple que todo cuadrado tiene intersección no vacía con algún cuadrado precedente del conjunto, esto es, diremos que $\{R_1, \dots, R_k\}$ es la partición maximal de C si

1. $C = R_1 \cup \dots \cup R_k$.
2. Dados $C_i \in R_i$ y $C_j \in R_j$, $i \neq j$, $C_i \cap C_j = \emptyset$
3. Para cada $i \in \{1, \dots, k\}$ y $C \in R_i$ existe $C' \in R_i$ cuyo centro está a la izquierda del centro de C , cumpliéndose $C \cap C' \neq \emptyset$.

Lema 3.1.2 *Dados n cuadrados en el plano, podemos encontrar la partición maximal del conjunto de cuadrados en tiempo $O(n^3)$.*

Demostración: Se trata de hallar conjuntos maximales de cuadrados, de forma que cada cuadrado tenga intersección no vacía con algún otro cuadrado del conjunto situado a la izquierda de aquél. Dos cuadrados C_i, C_j correspondientes a puntos p_i, p_j , tienen intersección no vacía si $d_\infty(p_i, p_j) < k$. Para cada punto p_i , podemos calcular el conjunto de los puntos a distancia menor que k en tiempo $O(n)$. Asimismo, para cada uno de estos puntos debemos obtener nuevamente los puntos del conjunto restante que se encuentran a distancia menor que k . De esta forma, y actuando de izquierda a derecha, para calcular el primer elemento de la partición maximal, se requiere tiempo $O(n^2)$. En el peor caso, tendremos que hacer esto para los n puntos, por lo que calcular todos los conjuntos maximales de cuadrados nos llevará tiempo $O(n^3)$. \square

Una vez obtenida la partición maximal del conjunto de cuadrados, nos interesa obtener los vértices de los *obstáculos* para aplicar alguno de los algoritmos que aparecen en la bibliografía que resuelven el problema de encontrar la trayectoria rectilínea más corta entre dos puntos evitando un conjunto de obstáculos rectilíneos. La mayoría de los algoritmos que resuelven esta cuestión se basan en la construcción de un grafo que contenga un camino de mínima longitud (*shortest path preserving graph*). Podemos citar el de Clarkson et al. [13], que resuelve el problema en tiempo $O(e \log^{\frac{3}{2}}(e))$ siendo e el número de lados del conjunto de obstáculos. Sin embargo, como se muestra en los trabajos de Wu, et al. [90] y de Yang [91], para obtener el grafo que contiene a una solución, es suficiente considerar los lados y vértices *extremos* de los obstáculos.

Definición 3.1.2 Se dice que un lado de un polígono rectilíneo (ortogonal) es *extremo* si los dos lados adyacentes se encuentran en el mismo semiplano de los determinados por la recta que pasa por él. Los vértices de un lado extremo se denominan vértices extremos.

De esta manera, dados n puntos de la entrada y una constante positiva k (dados n cuadrados centrados en p_i y de lado $2k$), podemos obtener los vértices extremos de la partición maximal de cuadrados definida anteriormente a la vez que se construye ésta. Para ello, cuando encontramos un cuadrado con

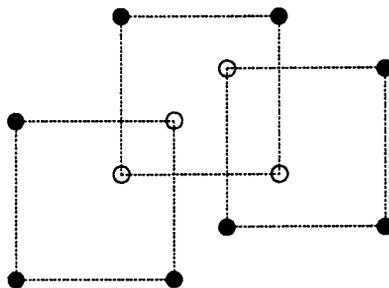


Figura 3.3: Cálculo de vértices extremos.

intersección no vacía con el anterior, podemos eliminar los vértices interiores (se encuentran en el interior del otro cuadrado) y actualizar así el conjunto de vértices extremos (ver Figura 3.3). Obsérvese que los vértices extremos de un polígono ortogonal resultante de unir varios cuadrados provienen sólo de los vértices de los cuadrados (y no de la intersección de sus lados).

LLamando e y v al número de lados y vértices extremo respectivamente, podemos usar el algoritmo de Yang [91] de tiempo $O(e \log(v)) + v \log^{\frac{3}{2}}(v)$. En nuestro caso, se cumple que para n cuadrados $l, v \leq 4n$ y, por tanto, podemos establecer el siguiente resultado:

Teorema 3.1.1 *Podemos resolver el problema RMA para la distancia inducida por la métrica l_∞ en tiempo $O(n \log^{\frac{3}{2}}(n))$ con preprocesamiento $O(n^3)$.*

Una vez resuelto el problema para la distancia inducida por l_∞ , podemos plantear el uso del mismo método para el caso de las distancias inducidas por l_1 y por l_2 . Si consideramos, por ejemplo, el problema RMA con la distancia inducida por la métrica l_1 ,

$$\text{mín } l(C) \quad \text{s.a. : } d_1(p_i, C) \geq k,$$

donde $d_1(p_i, C) = \text{mín}_{p \in C} d_1(p_i, p)$, podemos hacer una interpretación análoga a la anterior sin más que usar las bolas de radio k y centro p_i para cada $i = 1, \dots, n$, esto es, los conjuntos $B_1(p_i, k) = \{p \in \mathbb{R}^2 : d_1(p, p_i) \leq k\}$ ($d_1(p, p_i) = |x(p) - x(p_i)| + |y(p) - y(p_i)|$). Sin embargo, no aparecen en la

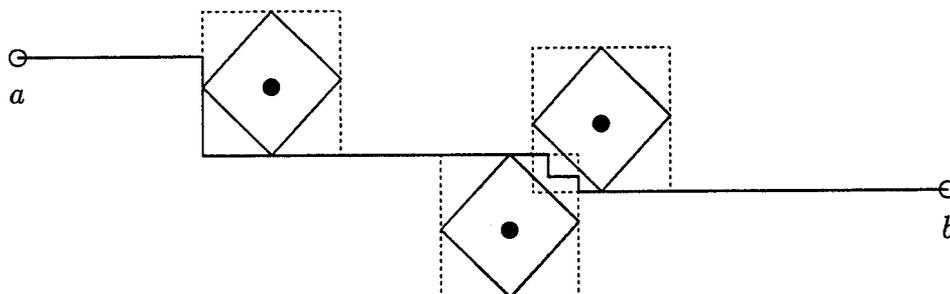


Figura 3.4: Reducción no válida del problema con distancia d_1 .

bibliografía algoritmos que obtengan caminos rectilíneos de mínima longitud entre obstáculos que no sean rectilíneos u ortogonales. Además, si para cada $i \in I$, se obtiene el menor cuadrado $C(p_i)$ que contiene a $B_1(p_i, k)$, no se cumple que una trayectoria rectilínea mínima que una dos puntos a través del conjunto de obstáculos $C(p_i)$, es una trayectoria óptima para el problema RMA cuando se usa la distancia d_1 (véase la Figura 3.4). Un ejemplo análogo puede darse para el caso euclídeo.

Desde luego, actuando de forma análoga a como se ha hecho en el caso de la norma l_∞ , la resolución de los correspondientes problemas geométricos (con *rombos* y *círculos* como obstáculos) para las métricas l_1 y l_2 , garantiza la solución. Actualmente, en la bibliografía no aparece el problema resuelto para caminos rectilíneos y obstáculos generales, por lo que proponemos aquí esta cuestión como problema abierto. Sin embargo, cabe señalar que si se considera una *poligonal genérica* (no se impone restricción sobre los segmentos que la componen), puede aplicarse el método de resolución propuesto, actuando de forma análoga para los problemas con métricas l_∞ y l_1 . En efecto, el método básico que se utiliza para obtener caminos de mínima longitud entre dos puntos a través de un conjunto de obstáculos es el siguiente: se construye un grafo que contiene un vértice por cada vértice extremo de cada polígono-obstáculo y una arista por cada dos vértices mutuamente visibles. Una vez probado que dicho grafo contiene un camino óptimo (*shortest path preserving graph*), basta aplicar el algoritmo de Dijkstra sobre el grafo para calcular el camino de menor longitud que evita el conjunto de polígonos. Puesto que el grafo

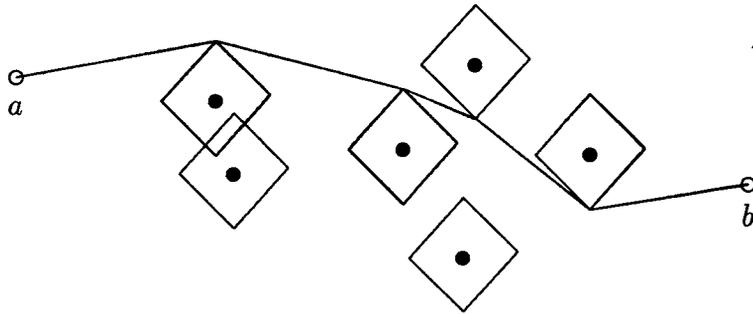


Figura 3.5: Obtención de una poligonal genérica de anticobertura para d_1 .

mencionado contiene $O(n^2)$ aristas, puede construirse en tiempo $O(n^2 \log(n))$ y espacio $O(n^2)$. Este método ha sido utilizado por diversos autores para obtener la poligonal de mínima longitud que conecta dos puntos y evita un conjunto de polígonos disjuntos. Citaremos los trabajos iniciales en este campo como el de Sharir y Schorr [81] que propone un algoritmo de tiempo $O(n^2 \log(n))$ o el de Asano et al. [4] de tiempo $O(n^2)$.

En la Figura 3.4 se observa que un grafo cuyos vértices son los vértices de los obstáculos no contiene a un camino rectilíneo óptimo por lo que el método denominado *shortest path preserving graph* no es aplicable. Para el caso de una poligonal genérica, sin embargo, podemos obtener el grafo mencionado independientemente de la forma de los polígonos que se consideran obstáculos (véase la Figura 3.5). Por tanto, el método puede ser aplicado tanto para el caso de la norma l_∞ como de l_1 , para lo que se requiere una transformación previa del problema a un problema de obstáculos, que como se ha visto puede llevarse a cabo en tiempo $O(n^3)$ (la obtención del conjunto de obstáculos es análoga en ambos casos).

A continuación, vamos a considerar una medida de distancia que aparece fundamentalmente en problemas de Investigación Operativa, Estadística y otros campos, como es la inducida por la distancia vertical entre dos puntos. El uso de la distancia vertical de un punto al servicio se centra principalmente en problemas de aproximación como los problemas de regresión. En nuestro contexto, el uso de la distancia vertical viene justificado cuando los puntos p_i

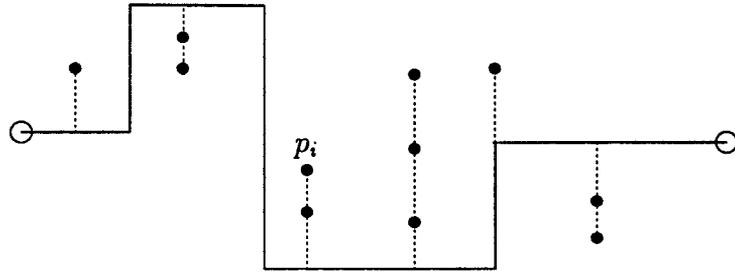


Figura 3.6: Cadena poligonal rectilínea y distancia vertical.

se encuentran distribuidos en pasillos verticales, de forma que el acceso a la ruta R se realiza en las direcciones $(0, 1)$ ó $(0, -1)$ (Figura 3.6).

Definición 3.1.3 Dado un punto $p_i \in P$ y una cadena poligonal rectilínea R , que une a y b , se define la *distancia vertical* de p_i a R como la distancia euclídea de p_i a su proyección vertical sobre R .

$$d_v(p_i, R) = \min\{d(p_i, q), q \in R, x(q) = x(p_i)\}.$$

Haciendo abuso de lenguaje, en lo sucesivo llamaremos distancia a la distancia vertical y la denotaremos por $d(p_i, R)$.

En este caso, el problema RMA puede interpretarse geoméricamente como sigue: dados dos puntos a y b en \mathbb{R}^2 y un conjunto de segmentos verticales de longitud $2k$ situados entre a y b , encontrar una trayectoria rectilínea de mínima longitud que conecte a y b y que no intersekte a ningún segmento. En efecto, si para cada punto p_i consideramos el segmento vertical $I_i = (y(p_i) - k, y(p_i) + k)$, resulta evidente probar el siguiente resultado

Lema 3.1.3 Dada una cadena rectilínea R que conecta a y b , se cumple que $d_v(p_i, R) \geq k$ si y sólo si $I_i \cap R = \emptyset$.

Como consecuencia, podemos encontrar una trayectoria óptima sin más que buscar el camino rectilíneo más corto que evita el conjunto de segmentos I_i , $i = 1, \dots, n$. (Figura 3.7).

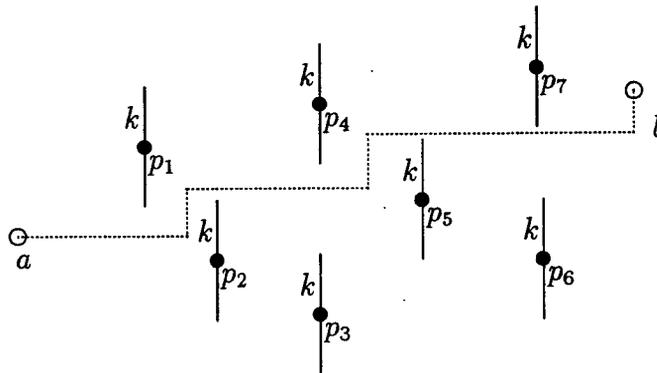


Figura 3.7: Reducción del problema con distancia vertical.

Pudiera ocurrir que dos segmentos con la misma abcisa tuviesen intersección no vacía, esto es, que existan puntos p_i, p_j con $x(p_i) = x(p_j)$ y $d_v(p_i, p_j) \leq k$. Por tanto, para obtener el conjunto de obstáculos disjuntos necesarios para la aplicación de algoritmos geométricos, como se hizo para el caso de l_∞ , basta considerar los segmentos resultantes de unir todos los segmentos correspondientes a puntos que verifiquen dicha condición, a los cuales llamaremos *segmentos maximales*.

Lema 3.1.4 *Dados n segmentos verticales en el plano, podemos encontrar el conjunto de segmentos maximales disjuntos, obtenidos a partir de los segmentos iniciales, en tiempo $O(n)$.*

Demostración: Dos segmentos verticales I_i, I_j con intersección no vacía (esto es, $d_v(p_i, p_j) < k$) deben tener la misma abcisa. Por tanto, dados los segmentos correspondientes a los puntos ordenados lexicográficamente, debemos obtener la unión de segmentos verticales trabajando en cada posición de abcisa.

Para ello, basta poner dos punteros, uno indicando el extremo inferior del segmento, que iremos cambiando cuando se terminen de hacer las uniones, y otro referido al extremo superior, que cambiamos cada vez que haya que unir un segmento (uniremos los segmentos I_i, I_j cuando $I_i \cap I_j \neq \emptyset$, esto es, $d_v(p_i, p_j) < k$). Como consecuencia, con un proceso incremental a lo largo

de la ordenación de los puntos podemos generar el conjunto de segmentos maximales disjuntos en tiempo $O(n)$. \square

Por consiguiente, nos centraremos en resolver el problema para un conjunto de segmentos verticales disjuntos.

Teorema 3.1.2 *Dado un conjunto de segmentos verticales disjuntos cualesquiera, podemos encontrar una trayectoria de mínima longitud entre dos puntos a y b que no corta a los segmentos en tiempo $\theta(n \log(n))$.*

Demostración: Haciendo uso del algoritmo de DeRezende et al. [16] para este caso particular (los segmentos se interpretan como obstáculos rectangulares degenerados), podemos resolver el problema en tiempo $O(n \log(n))$.

Para obtener la cota inferior $\Omega(n \log(n))$, mostraremos que la ordenación de un conjunto de números enteros puede reducirse a una instancia del problema RMA para la distancia vertical.

Dado un conjunto de enteros $\{a_1, a_2, \dots, a_n\}$ a ordenar, considerar dos puntos $(a, 0)$ y $(b, 0)$ de forma que $a < a_1, a_2, \dots, a_n < b$. (a y b se calculan en tiempo lineal). Para cada $i = 1, 2, \dots, n$ se considera el segmento vertical de extremo inferior $(x_i, -\frac{1}{n})$ y longitud 5 y el segmento vertical de extremo superior $(x_i + \frac{1}{2}, 0)$ e igual longitud ilustrados en la Figura 3.8.

Al calcular el camino rectilíneo más corto desde a hasta b que no intersecte a los segmentos, obtenemos una trayectoria que ordena el conjunto de enteros puesto que, al ser x -monótona, visita los extremos $(x_i, -\frac{1}{n})$ de los segmentos en orden creciente. Por tanto, tenemos reducido el problema de la ordenación entera a un caso particular de nuestro problema y se tiene la cota inferior. \square

Como consecuencia de los resultados anteriores obtenemos el siguiente teorema:

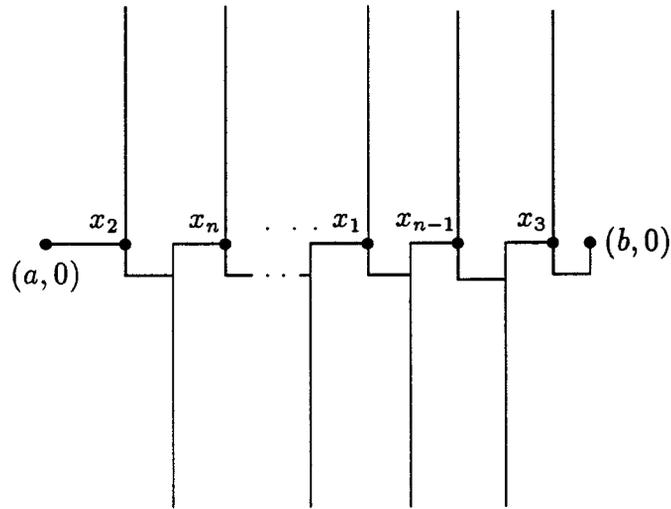


Figura 3.8: Ilustrando una cota inferior.

Teorema 3.1.3 Podemos resolver el problema RMA con distancia vertical en tiempo $\theta(n \log(n))$.

3.2 Análisis del problema RMC.

Dada una constante $k \geq 0$, se trata de encontrar un camino poligonal rectilíneo R uniendo a y b de mínima longitud cuya distancia a los puntos de P esté acotada por k .

$$\text{mín } l(R) \quad \text{s.a. : } d(p_i, R) \leq k \quad (RMC)$$

Una solución R^* de TMC se denomina trayectoria rectilínea mínima con cobertura sobre P . Si $k = 0$, diremos que el cobertura es *directa* y para los valores $k > 0$ hablaremos de cobertura *indirecta*.

Aunque se puede abordar el problema general, trataremos aquí, por simplicidad en su desarrollo, los dos tipos de cobertura por separado. Así pues consideramos los problemas

$$\text{mín } l(R) \quad \text{s.a. : } d(p_i, R) = 0 \quad (RMCD)$$

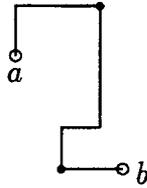


Figura 3.9: El camino más corto no es x-monótono.

$$\text{mín } l(R) \quad \text{s.a. : } d(p_i, R) \leq k, k > 0 \quad (\text{RMCI})$$

3.2.1 Cobertura directa

Dados dos puntos a y b en \mathbb{R}^2 y un conjunto P de puntos comprendidos entre a y b ($x(a) < x(p_i) < x(b) \quad \forall p_i \in P$), el camino rectilíneo más corto que pasa por todos los puntos no posee la propiedad de x-monotonía (Figura 3.9). En consecuencia, el problema para trayectorias rectilíneas generales no es resoluble usando técnicas de barrido. De hecho, el problema del camino de menor longitud que pasa por todos los puntos de un conjunto de \mathbb{R}^2 , (*TSP geométrico*), es un conocido problema NP duro. Así pues, proponemos una resolución del problema para el caso restringido de caminos rectilíneos x-monótonos (*Trayectoria Rectilínea Monótona Mínima de Cobertura Directa*):

$$\text{mín } l(R) \quad \text{s.a. : } p_i \in R, R \text{ x-monótono} \quad (\text{RMMCD})$$

Si todos los puntos de P tienen abcisas diferentes, la trayectoria x-monótona más corta desde a hasta b que pasa por todos los puntos viene dada justamente por la ordenación de la nube, de forma que longitud (de una trayectoria mínima) entre cada dos puntos consecutivos viene dada precisamente por la distancia de Manhattan o d_1 entre dichos puntos (Figura 3.10 (i)).

Contemplamos entonces la posibilidad de que los puntos pueden compartir abcisa. Para los algoritmos que desarrollamos en esta sección, la situación de los puntos en cada vertical es esencial, por lo que distinguiremos los casos en función de esa situación. A continuación veremos que para buscar

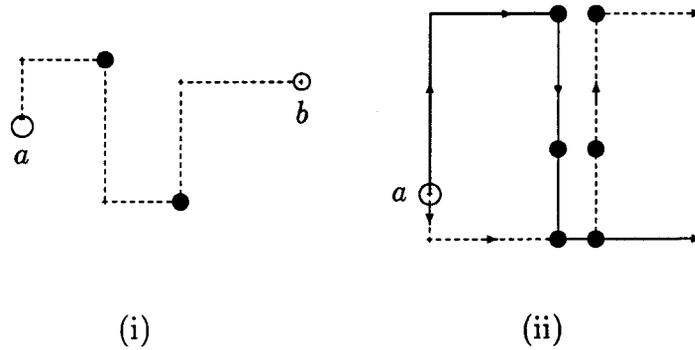


Figura 3.10: Construcción de un camino mínimo. (i) Posición no degenerada. (ii) Posición degenerada.

un camino rectilíneo de mínima longitud basta considerar un número finito, aunque no polinomial, de trayectorias.

Por una parte, dada una poligonal rectilínea x -monótona conectando a con b , $R = \{\overline{c_0c_1}, \overline{c_1c_2}, \dots, \overline{c_kc_{k+1}}\}$, la longitud viene dada por la suma de las longitudes de los segmentos que la componen $s_1 = \overline{c_0c_1}, s_2 = \overline{c_1c_2}, \dots, s_{k+1} = \overline{c_kc_{k+1}}$ y por tanto se obtiene que

$$l(C) = \sum_{j=0}^{k+1} l(s_j) = \sum_{s_j \text{ horiz.}} l(s_j) + \sum_{s_j \text{ vert.}} l(s_j) = (x(b) - x(a)) + \sum_{s_j \text{ vert.}} l(s_j)$$

de tal suerte que el objetivo es minimizar la suma de las longitudes correspondientes a los segmentos verticales de la cadena que denominaremos *longitud vertical*, $l_v(R)$.

Por otra, sean $a = p_0, p_1, \dots, p_n, p_{n+1} = b$ los puntos ordenados lexicográficamente (si fuera necesario los renombramos para que así sea) y sean $a_0 = x_0, a_1, \dots, a_m, a_{m+1} = x_{m+1}$ las abscisas distintas de los puntos. Se observa fácilmente que basta buscar un camino mínimo entre los que cumplen la siguiente condición: *la cobertura para los puntos de cada vertical $x = a_j$ se realiza entrando por el punto de menor ordenada y saliendo por el punto de mayor ordenada o al contrario* (véase la Figura 3.10 (ii)). El número de caminos x -monótonos posibles que verifican esa condición es 2^m , por lo que

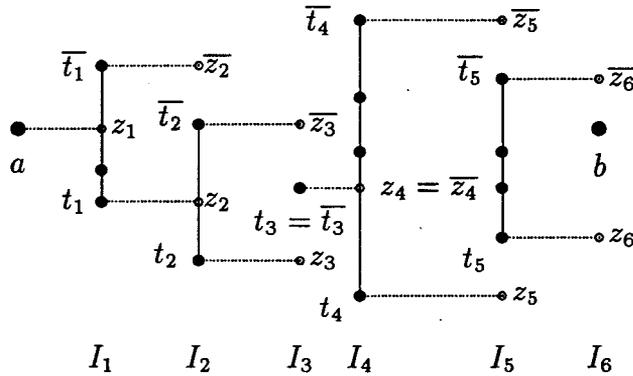


Figura 3.11: Obtención de los puntos z_j y \bar{z}_j .

el método combinatorio no puede usarse. Haremos uso, sin embargo, de un método basado en la técnica de *programación dinámica* introducida por Bellman [5] (véase [14] para una exposición reciente).

Dados dos puntos $p_i, p_j, i < j$, sabemos que el camino de longitud mínima desde p_i hasta p_j posee longitud vertical $l_v(R) = |y(p_i) - y(p_j)|$. Puesto que debemos minimizar la suma total de longitudes verticales, es suficiente hacer énfasis en la posición de ordenada (altura) del camino para cada posición de $j = 1, \dots, m$. De esta forma, dado $j = 1, 2, \dots, m$ se definen los siguientes parámetros (ver Figura 3.11):

$$I_j = \{p_i \in P, x(p_i) = a_j\}$$

$$t_0 = \bar{t}_0 = a$$

$$t_j = (a_j, \min \{y(p_i), p_i \in I_j\}), \bar{t}_j = (a_j, \max \{y(p_i), p_i \in I_j\})$$

$$z_j = (a_j, y(t_{j-1})), \bar{z}_j = (a_j, y(\bar{t}_{j-1}))$$

Nuestro objetivo es obtener una fórmula de recurrencia que calcule la longitud de camino más corto desde una posición z hasta el final b , visitando todos los puntos a la derecha de z . Puesto que, según se ha observado, basta considerar caminos que entren y salgan por los puntos de mayor y menor ordenada de cada vertical, para calcular la longitud vertical mínima de un camino rectilíneo que pasa por todos los puntos basta tener en consideración

los parámetros anteriores. Por simplicidad en las expresiones, sólo consideraremos la suma de las longitudes invertidas en movimientos verticales, pues, como sabemos, la suma de las longitudes de los segmentos horizontales está determinada. Por tanto, continuar a partir de la posición $t_j(\bar{t}_j)$ es equivalente a continuar a partir del punto $z_{j+1}(\bar{z}_{j+1})$ (Figura 3.11).

Para cada $j = 1, 2, \dots, m$ consideramos la función $f_j^* : \{z_j, \bar{z}_j\} \rightarrow \mathbb{R}^+$ donde $f_j^*(z)$ es la longitud vertical de un camino rectilíneo mínimo desde el punto z hasta b visitando todos los puntos de I_j, I_{j+1}, \dots, I_m . Se cumple la siguiente propiedad:

Lema 3.2.1 Para $z \in \{z_{j-1}, \bar{z}_{j-1}\}$ se tiene que

1. Si $y(z) \geq y(\bar{t}_{j-1})$, $f_{j-1}^*(z) = (y(z) - y(\bar{t}_{j-1})) + (y(\bar{t}_{j-1}) - y(t_{j-1})) + f_j^*(z_j)$
2. Si $y(z) \leq y(t_{j-1})$, $f_{j-1}^*(z) = (y(t_{j-1}) - y(z)) + (y(\bar{t}_{j-1}) - y(t_{j-1})) + f_j^*(\bar{z}_j)$
3. Si $y(t_{j-1}) < y(z) < y(\bar{t}_{j-1})$, $f_{j-1}^*(z) = \min\{(y(\bar{t}_{j-1}) - y(z)) + (y(\bar{t}_{j-1}) - y(t_{j-1})) + f_j^*(z_j), (y(z) - y(t_{j-1})) + (y(\bar{t}_{j-1}) - y(t_{j-1})) + f_j^*(\bar{z}_j)\}$

Demostración: El caso 1 y 2 son análogos y el caso 3 es evidente (ver Figura 3.11), por lo que basta demostrar la fórmula de 1.

Supongamos $y(z) \geq y(\bar{t}_{j-1})$ y estamos en la situación de la Figura 3.12. Entonces se cumple $f_{j-1}^*(z) = \min\{\alpha, \beta\}$, donde $\alpha = (y(z) - y(\bar{t}_{j-1})) + (y(\bar{t}_{j-1}) - y(t_{j-1})) + f_j^*(z_j)$ y $\beta = (y(z) - y(t_{j-1})) + (y(\bar{t}_{j-1}) - y(t_{j-1})) + f_j^*(\bar{z}_j)$.

Por otra parte, se tiene que $f_j^*(z_j) \leq y(\bar{z}_j) - y(z_j) + f_j^*(\bar{z}_j)$ y por tanto, $\alpha \leq (y(z) - y(\bar{t}_{j-1})) + (y(\bar{t}_{j-1}) - y(t_{j-1})) + y(\bar{z}_j) - y(z_j) + f_j^*(\bar{z}_j) = (y(z) - y(t_{j-1})) + y(\bar{z}_j) - y(z_j) + f_j^*(\bar{z}_j) = (y(z) - y(t_{j-1})) + (y(\bar{t}_{j-1}) - y(t_{j-1})) + f_j^*(\bar{z}_j) = \beta$ de donde se tiene el resultado. \square

Como consecuencia del lema anterior, la longitud mínima de un camino rectilíneo de cobertura directa desde una posición en I_{j-1} involucra la longitud del camino a partir de una posición en el siguiente conjunto I_j y, por tanto,

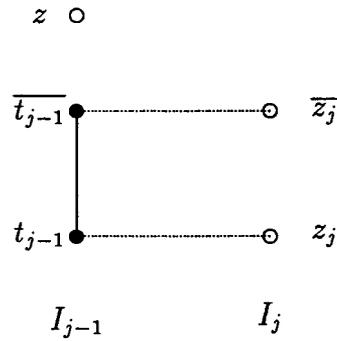


Figura 3.12: Caso $y(z) \geq y(\overline{t_{j-1}})$ en el lema 3.2.1.

podemos usar programación dinámica. A saber, definiendo $f_{m+1}^*(z) = |y(z) - y(b)|$ para $z = z_{m+1}$ y $\overline{z_{m+1}}$, podemos calcular recursivamente los valores de $f_m^*, f_{m-1}^*, \dots, f_2^*, f_1^*$ y obtenemos el siguiente resultado:

Lema 3.2.2 *La longitud de un camino rectilíneo x -monótono mínimo desde a hasta b que visita todos los puntos de la nube P , viene dado por $l^*(a, b) = f_1^*(z_1) + (x(b) - x(a))$.*

Lema 3.2.3 *$l^*(a, b)$ se obtiene en tiempo y espacio lineal cuando los puntos de la nube P están ordenados.*

Demostración: Si llamamos $T(f_j^*)$ al tiempo invertido en calcular $f_j^*(z_j)$ y $f_j^*(\overline{z_j})$, se tiene la ecuación de recurrencia $T(f_{j-1}^*) = T(f_j^*) + O(1)$ mediante uso de los lemas 3.2.1 y 3.2.2, de donde deducimos que $T(f_1^*) = O(m)$.

Por su parte, en cada paso, necesitamos almacenar los valores $t_j, \overline{t_j}, z_j, \overline{z_j}$ en cada vertical I_j así como las cantidades f_j^* en el recorrido inverso. Por tanto, se requiere espacio lineal. \square

Teorema 3.2.1 *El problema (RMMCD) se resuelve en tiempo $O(n \log(n))$ y espacio $O(n)$.*

Demostración: Se obtiene como consecuencia directa de los resultados anteriores ejecutando el siguiente algoritmo:

ALGORITMO RMMCD

Entrada: $P = \{p_1, p_2, \dots, p_n\}$, a , b .

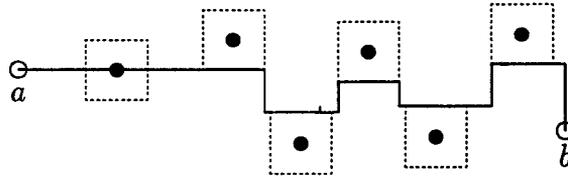
Salida: $l^*(a, b)$ y la lista de orden de visita desde a hasta b .

1. Ordenar P , llamar $a_0 = x(a), a_1, \dots, a_m, a_{m+1} = x(b)$ a las diferentes abscisas, $t_0 = \bar{t}_0 = a$ y $t_{m+1} = \bar{t}_{m+1} = b$.
2. Para $j = 1, 2, \dots, m + 1$ calcular $I_j, t_j, \bar{t}_j, z_j, \bar{z}_j$.
3. Hacer $f_{m+1}^*(z_{m+1}) = |y(z_{m+1}) - y(t_{m+1})|$ y $f_{m+1}^*(\bar{z}_{m+1}) = |y(\bar{z}_{m+1}) - y(t_{m+1})|$
4. Para $j = m, m - 1, \dots, 2, 1$
 - (a) Obtener $f_j^*(z_j)$ usando el lema 3.2.1 y guardar la lista de orden de visita de z_j a b .
 - (b) Obtener $f_j^*(\bar{z}_j)$ usando el lema 3.2.1 y guardar la lista de orden de visita de \bar{z}_j a b .
5. Salir con $l^*(a, b) = f_1^*(z_1) + (a_{m+1} - a_0)$ y la lista de orden de visita de z_1 a b .

□

3.2.2 Cobertura indirecta

Al igual que en la sección anterior, el objetivo es minimizar la suma de las longitudes de los segmentos de la cadena, pero ahora impondremos una restricción de *cobertura*.

Figura 3.13: Transformación del problema RMCI para el caso d_∞ .

Definición 3.2.1 Diremos que la nube P está k -cubierta por R o que R tiene *cobertura* de orden k sobre P , si la distancia de los puntos de P a R está acotada por la cantidad positiva fijada k .

El problema que se plantea (Trayectoria Rectilínea Mínima con Cobertura Indirecta) se formula como sigue:

$$\text{mín } l(R) \quad \text{s.a. : } d(p_i, R) \leq k, \quad k > 0 \quad (RMCI)$$

Puede mostrarse con un ejemplo análogo al caso de cobertura directa que, la trayectoria rectilínea mínima con cobertura indirecta no es, en general, x -monótona. Nosotros plantearemos la cuestión para el caso de caminos x -monótonos y, como consecuencia, el objetivo será minimizar la *longitud vertical* de la cadena R , esto es, la suma de las longitudes de los segmentos verticales.

$$\text{mín } l(R) \quad \text{s.a. : } d(p_i, R) \leq k, \quad R \text{ } x\text{-monótono} \quad (RMMCI)$$

Si queremos resolver el problema RMMCI con una distancia dada d , definida entre punto y camino, bueno será considerar la transformación del problema haciendo uso de la bola unidad (para cada punto) determinada por la correspondiente distancia. En este caso, aparecerá un problema de camino monótono que debe intersecar a un conjunto de obstáculos situados entre dos puntos. En el caso de la distancia d_∞ definida anteriormente, aparecen obstáculos de tipo cuadrado (Figura 3.13). Si suponemos que los puntos tienen distinta coordenada x y además, se cumple $|x(p_i) - x(p_{i+1})| > 2k, \forall p_i \in P$, podemos resolver fácilmente el problema en tiempo lineal sin más que aplicar

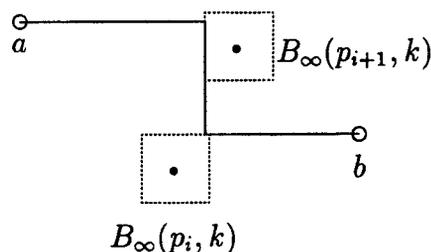


Figura 3.14: Un algoritmo incremental sobre las abcisas no resuelve el problema.

un algoritmo incremental sobre la ordenación de las abcisas (desde cada posición se accede al cuadrado siguiente con la menor longitud vertical posible) (obsérvese la Figura 3.13). Sin embargo, en general, esta técnica no resuelve el problema pues como se observa en la Figura 3.14, el *orden de cobertura* de un camino óptimo puede no coincidir con la ordenación lexicográfica de la nube.

Asimismo, si se consideran otras distancias aparecen diversos problemas geométricos con un mismo enunciado: *encontrar el camino mínimo que toca a un conjunto de objetos en el plano*, que no están resueltos.

Aquí resolveremos el problema para el caso de la distancia vertical d_v definida anteriormente. La interpretación geométrica del problema es quizás más interesante que la de camino de cobertura: dado un conjunto de segmentos verticales en el plano (*ventanas*), se trata de encontrar un camino monótono de longitud mínima que conectando un origen y un destino pase por todas las ventanas (Figura 3.15).

En este caso, veremos que una trayectoria óptima posee propiedades que permiten resolver el problema usando métodos clásicos como barrido y programación dinámica. En efecto, probaremos que la longitud de un camino óptimo desde una posición puede expresarse en función del camino óptimo desde la posición siguiente, por lo que la programación dinámica es aplicable.

Supongamos en primer lugar que los puntos están en “posición general”, esto es, todos tienen distinta coordenada x . En este caso, denotando por

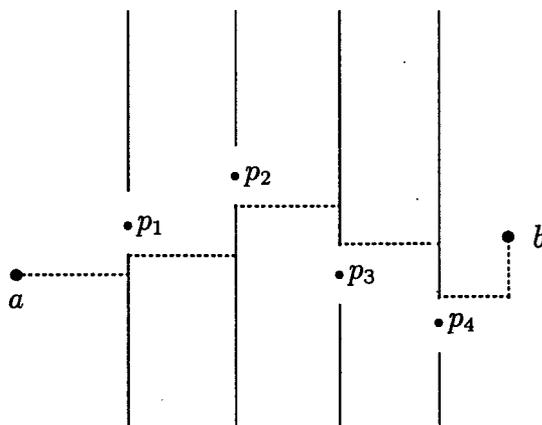


Figura 3.15: Transformación del problema con distancia vertical.

$y(p_1), y(p_2), \dots, y(p_n)$ a las ordenadas de los puntos, podemos transformar el problema en una cuestión de trayectoria más corta a través de obstáculos en un sentido contrario a como se hizo para los problemas de anticobertura. Esto es, considerando como obstáculos entre a y b las semirrectas $(y(p_i) + k, \infty)$ y $(-\infty, y(p_i) - k)$ para cada $i = 1, 2, \dots, n$ (Figura 3.15). Justamente obtenemos un caso particular del problema resuelto por deRezende et al. [16], por lo que podemos aplicar su algoritmo para resolver el problema en tiempo $O(n \log(n))$.

Para el caso degenerado, esto es, si admitimos la existencia de puntos con la misma coordenada x , consideraremos los siguientes casos: en primer lugar, supongamos que los puntos sobre una misma vertical no están alejados entre sí más de $2k$. Dada la nube ordenada lexicográficamente, consideremos los conjuntos I_j y los puntos t_j, \bar{t}_j , $j = 1, 2, \dots, m$ definidos en la sección anterior, y supongamos que se cumple $y(\bar{t}_j) - y(t_j) \leq 2k$ para todos los valores de j . En esta situación podemos igualmente resolver el problema tomando para cada j , las semirrectas $(y(t_j) + k, \infty)$ y $(-\infty, y(\bar{t}_j) - k)$ como barreras y aplicar el algoritmo de deRezende et al. [16], ya que una solución para el problema de las barreras es solución a nuestro problema.

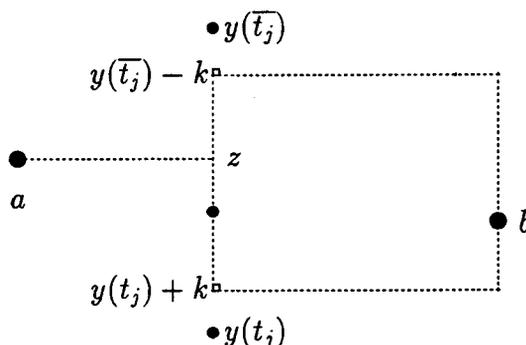


Figura 3.16: Problema de decisión.

Finalmente, si consideramos una posición arbitraria de los puntos, podemos observar que en el caso de puntos en una vertical que estén “alejados” más de $2k$ aparece un problema de decisión para el que el método anterior no es válido. Consideremos el caso de la Figura 3.16; para trazar un camino de cobertura desde z hasta b de mínima longitud tenemos dos opciones: acudir primero al punto $y(\bar{t}_j) - k$, después a $y(t_j) + k$ y seguir adelante o, tomar el camino inverso. Por tanto, tenemos una situación análoga al problema (*RMMCD*) y actuaremos de forma parecida.

Sean, para $j = 1, 2, \dots, m$, los conjuntos I_j y los puntos t_j, \bar{t}_j definidos en la sección anterior. Considerar además, los puntos $(a_j, y(t_j) + k)$ y $(a_j, y(\bar{t}_j) - k)$.

Definición 3.2.2 Diremos que un punto z con abscisa a_j es una *posición que cubre* a I_j si se cumple $d_v(z, p_j) \leq k, \forall p_j \in I_j$ (o sea, si $|y(z) - y(t_j)| \leq k$ y $|y(z) - y(\bar{t}_j)| \leq k$).

Definición 3.2.3 Dada una posición z con $x(z) = a_j$, definimos *sucesores* de z (Figura 3.17) a las posiciones en la recta vertical $x = a_j$ siguientes:

1. Si $|y(\bar{t}_j) - y(t_j)| > 2k$, posiciones $(a_{j+1}, y(t_j) + k)$ y $(a_{j+1}, y(\bar{t}_j) - k)$.
2. Si $|y(\bar{t}_j) - y(t_j)| \leq 2k$

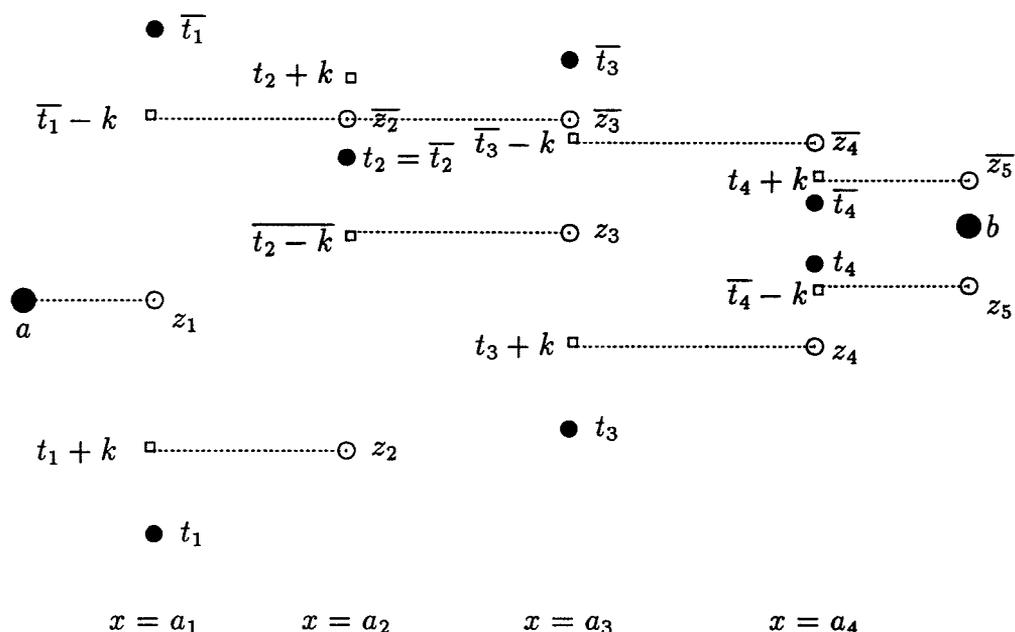


Figura 3.17: Obtención de sucesores.

- (a) Si z cubre a I_j , posición $(a_{j+1}, y(z))$.
- (b) Si z no cubre a I_j
 - i. Si $|y(z) - (y(\bar{t}_j) - k)| < |y(z) - (y(t_j) + k)|$, posición $(a_{j+1}, y(\bar{t}_j) - k)$.
 - ii. En caso contrario, posición $(a_{j+1}, y(t_j) + k)$.

Obsérvese que cada posición z en una vertical $x = a_j$ posee a lo más dos sucesores en la recta vertical siguiente (véase Figura 3.17); además, dadas dos posiciones distintas en la recta $x = a_j$, el conjunto compuesto por los sucesores de ambos tiene a lo más dos elementos que denominaremos z_{j+1} y \bar{z}_{j+1} ($y(z_{j+1}) \leq y(\bar{z}_{j+1})$).

Definición 3.2.4 Dado $z_1 = \bar{z}_1 = (x_1, y(a))$, definimos el conjunto de sucesores de z_1 como $\mathcal{S} = \{z_2, \bar{z}_2, z_3, \bar{z}_3, \dots, z_m, \bar{z}_m, z_{m+1}, \bar{z}_{m+1}\}$, donde $\{z_j, \bar{z}_j\}$ es el conjunto de sucesores de $\{z_{j-1}, \bar{z}_{j-1}\}$, $j = 2, 3, \dots, m + 1$.

Lema 3.2.4 *Dados a, p_1, \dots, p_n, b ordenados lexicográficamente, y z_1 la posición definida anteriormente, se puede determinar el conjunto S en tiempo y espacio lineal.*

Demostración: Se obtiene directamente de las definiciones anteriores mediante un proceso incremental sobre las abcisas $a_j, j = 1, \dots, m$. \square

Vamos a mostrar a continuación cómo la longitud de un camino mínimo desde una posición z en $x = a_j$ involucra la longitud de un camino mínimo desde una posición en $x = a_{j+1}$, lo que nos permitirá usar un procedimiento de programación dinámica. La situación es similar a la planteada para el problema (RMMCD) y la actuación será análoga.

Definición 3.2.5 Para $j = 1, 2, \dots, m$ definimos las funciones $f_j^* : \{z_j, \bar{z}_j\} \rightarrow \mathbb{R}^+$ donde $f_j^*(z)$ es la longitud mínima de un camino rectilíneo que conecta la posición z con b que cubre todos los puntos de I_j, I_{j+1}, \dots, I_m y $g_j^* : \{z_j, \bar{z}_j\} \rightarrow \mathbb{R}^+$ donde $g_j^*(z)$ es el recorrido mínimo sobre la recta vertical $x = a_j$ que partiendo de z cubre los puntos de I_j .

Lema 3.2.5 *Dado $z \in \{z_j, \bar{z}_j\}$, y $j = m, m-1, \dots, 2, 1$ se tiene que:*

1. Si $(y(\bar{t}_j) - y(t_j)) \leq 2k$ (z tiene un solo sucesor z'), entonces $f_j^*(z) = g_j^*(z) + f_{j+1}^*(z')$.
2. Si $(y(\bar{t}_j) - y(t_j)) > 2k$ (z tiene dos sucesores $z_{j+1} < \bar{z}_{j+1}$), entonces $f_j^*(z) = \min\{|y(z) - (y(\bar{t}_j) - k)| + M_j + f_j^*(z_{j+1}), |y(z) - (y(t_j) + k)| + M_j + f_{j+1}^*(\bar{z}_{j+1})\}$ donde $M_j = (y(\bar{t}_j) - k) - (y(t_j) + k)$

Demostración: Análoga al Lema 3.2.1. \square

Como consecuencia del lema y definiendo los valores $f_{m+1}^*(z_{m+1}) = |y(z_{m+1}) - y(b)|$ y $f_{m+1}^*(\bar{z}_{m+1}) = |y(\bar{z}_{m+1}) - y(b)|$, podemos calcular recursivamente los valores de $f_m^*, f_{m-1}^*, \dots, f_2^*, f_1^*$ y tenemos, al igual que en la sección anterior los resultados siguientes:

Lema 3.2.6 *La longitud mínima de un camino rectilíneo de cobertura que conecta a con b viene dada por $l^*(a, b) = f_1^*(z_1) + (x(b) - x(a))$.*

Lema 3.2.7 *$l^*(a, b)$ se puede obtener en tiempo y espacio lineal cuando los puntos de la nube \mathcal{P} están ordenados.*

Demostración: Si denominamos $T(f_j^*)$ al tiempo de cálculo necesario para obtener $f_j^*(z_j)$ y $f_j^*(\bar{z}_j)$, podemos establecer la ecuación de recurrencia $T(f_j^*) = T(f_{j+1}^*) + O(1)$ por lo que $T(f_1^*) = O(m)$. Por su parte, debemos almacenar desde $j = 1$ hasta $j = m + 1$ las posiciones $y(t_j)$, $y(\bar{t}_j)$, $y(t_j) + k$, $y(\bar{t}_j) - k$, z_j , \bar{z}_j así como los valores de las funciones f_j^* en el recorrido inverso. Por tanto, obtenemos espacio lineal. \square

Como consecuencia de los resultados anteriores, tenemos el teorema siguiente:

Teorema 3.2.2 *El problema (RMMCI) se resuelve en tiempo $O(n \log(n))$ y espacio $O(n)$.*

Demostración: Basta aplicar el algoritmo descrito a continuación:

ALGORITMO RMMCI

Entrada: $P = \{p_1, p_2, \dots, p_n\}$, a , b .

Salida: $l^*(a, b)$ y la lista de orden de visita desde a hasta b .

1. Ordenar P , llamar $a_0 = x(a)$, a_1, \dots, a_m , $a_{m+1} = x(b)$ a las diferentes abcisas y hacer $z_1 = \bar{z}_1 = (a_1, y(a))$.
2. Para $j = 1, 2, \dots, m + 1$ calcular I_j , t_j , \bar{t}_j , z_j , \bar{z}_j , $t_{m+1} = \bar{t}_{m+1} = b$.
3. Hacer $f_{m+1}^*(z_{m+1}) = |y(z_{m+1}) - y(b)|$ y $f_{m+1}^*(\bar{z}_{m+1}) = |y(\bar{z}_{m+1}) - y(b)|$.

4. Para $j = m, m - 1, \dots, 2, 1$
 - (a) Obtener $f_j^*(z_j)$ usando el lema 3.2.5 y guardar la lista de orden de visita de z_j a b .
 - (b) Obtener $f_j^*(\bar{z}_j)$ usando el lema 3.2.5 y guardar la lista de orden de visita de \bar{z}_j a b .
5. Salir con $l^*(a, b) = f_1^*(z_1) + (a_{m+1} - a_0)$ y la lista de orden de visita desde z_1 a b .

□

3.3 Conclusiones y problemas abiertos

En este capítulo se han resuelto problemas de localización óptima de caminos rectilíneos cuya función objetivo ha sido la longitud y la restricción involucra las distancias entre la trayectoria y los puntos de la nube que ésta atraviesa. Imponiendo en la restricción cotas inferiores o superiores sobre la distancia, aparecen dos problemas que, aunque cercanos en su formulación, son esencialmente distintos. De hecho, se ha visto cómo en el primer caso, una técnica adecuada para resolver el problema es la transformación a un problema de caminos rectilíneos que evitan un conjunto de determinados obstáculos. La transformación se basa en la consideración de la bola unidad (correspondiente a la distancia de trabajo) centrada en cada punto p_i cuya interacción con el camino R quiere contemplarse.

El segundo problema planteado (camino rectilíneo de cobertura) posee mayor complejidad. De hecho, se ha mostrado cómo el camino óptimo no posee la propiedad de x -monotonía en general y, por tanto, no podemos aplicar las técnicas propias que aprovechan dicha propiedad. De hecho, el problema de encontrar un camino rectilíneo de mínima longitud y que pase por todos los puntos (cobertura directa) es NP-duro [35]. Sin embargo, aquí se ha resuelto el problema para un caso particular: se ha supuesto el camino x -monótono y se ha considerado la distancia vertical entre punto y trayectoria. Cabe destacar que

en los algoritmos propuestos para estos problemas restringidos, la complejidad es lineal si consideramos la ordenación de los puntos como preprocesado. Este es debido a que se basan en el uso de barrido y programación dinámica.

En el desarrollo del capítulo, se han expuesto los distintos problemas conforme han ido apareciendo de forma natural al imponer una u otra restricción o al considerar una u otra distancia. La idea central que subyace en la resolución de los problemas aquí tratados es la de interpretarlos como cuestiones geométricas. En este sentido, cabe observar que el método usado no exige que los *entornos* centrados en los puntos p_i (cuadrados, segmentos) sean del mismo tamaño, por lo que el caso ponderado queda igualmente resuelto. Por otra parte, el procedimiento desarrollado para el problema RMA es aplicable igualmente si se consideran poligonales genéricas, puesto que en la bibliografía existen múltiples algoritmos que obtienen, de forma eficiente, caminos poligonales de mínima longitud evitando un conjunto de obstáculos (véanse los trabajos de Lee y Preparata [51] y Mitchell y Papadimitriou [63]). Sin embargo, los problemas que provienen de considerar la distancia l_1 o euclídea no han sido resueltos con el mismo procedimiento puesto que los correspondientes problemas geométricos no lo están. Estas cuestiones quedarán aquí como problemas abiertos.

Consideramos finalmente, que lo que aparece en este capítulo no es más que una muestra de un estudio de problemas de caminos rectilíneos de mínima longitud en el plano planteados desde la óptima de la Localización de Servicios y que pueden resolverse de forma eficiente con técnicas usuales en la disciplina de Geometría Computacional.

Capítulo 4

El problema minimax para trayectorias rectilíneas

En este capítulo se plantea la localización óptima de caminos rectilíneos (ortogonales) usando otro criterio distinto al de *mínima longitud*. Uno de los criterios más usados en teoría de localización es el *minimax*. Plantearemos aquí el problema minimax correspondiente.

Así pues, se trata de encontrar una trayectoria rectilínea que atraviese una nube de puntos de forma que, la máxima distancia que tome sobre los puntos sea minimizada. Usualmente, como se pudo apreciar en el capítulo 2, al servicio óptimo que proviene de considerar el criterio minimax, se le denomina “servicio centro”. Por extensión, llamaremos *trayectoria centro* a una solución del problema general que va a plantearse. La formulación de dicho problema es la siguiente: sea $P = \{p_i, i = 1, 2, \dots, n\}$ un conjunto de puntos en \mathbb{R}^2 y $X(P)$ el rango de abscisas de P . Denotaremos por \mathcal{R}_P al conjunto de trayectorias rectilíneas que *atraviesan* la nube P , esto es, $R \in \mathcal{R}_P$ si para todo $p \in P$ existe un punto $q \in R$ que cumple $x(q) = x(p)$. El problema se expresa entonces

$$\min_{R \in \mathcal{R}_P} \max_{p_i \in P} d(p_i, R).$$

Definición 4.0.1 La *excentricidad* de una trayectoria $R \in \mathcal{R}_P$, es la máxima

de las distancias de los puntos a la trayectoria:

$$e(R) = \max_{p_i \in P} d(p_i, R)$$

Haciendo uso de la defición anterior, el problema quedaría

$$\min_{R \in \mathcal{R}_P} e(R).$$

Si no imponemos restricciones sobre la trayectoria, la solución está dada por una poligonal que pase por todos los puntos de la nube. Por tanto, consideraremos rectricciones sobre dos factores que son generalmente utilizados para la construcción de caminos: el número de codos y la longitud. Como distancia de punto a trayectoria se usará la distancia vertical definida en el capítulo anterior, por lo que los problemas que vamos a plantear pueden incluirse en el contexto de *ajuste de una poligonal R a una nube de puntos $P = \{p_1, \dots, p_n\}$* en el sentido de Chebichev, esto es, el error considerado en la aproximación es $E = \max_{1 \leq i \leq n} d_v(p_i, R)$. Puesto que, tanto en este capítulo como en el siguiente, trabajaremos con la distancia vertical de punto a poligonal, haremos en lo que sigue abuso de lenguaje y llamaremos distancia a distancia vertical denotando d por d_v .

Existen interesantes trabajos que abordan problemas de tipo minimax en el contexto de *aproximación de funciones o aproximación de poligonales complejas por otras más simples*, pero siempre la poligonal buscada es no rectilínea (ver [45] y [40] entre otros). Trataremos este tipo de poligonales más generales en el siguiente capítulo y consideraremos en éste los problemas para trayectorias rectilíneas usando los factores de restricción ya mencionados.

4.1 Restricción en el número de codos

4.1.1 Formulación del problema

Dada una trayectoria rectilínea $R \in \mathcal{R}_P$, sea $|R|$ el número de ángulos rectos o *codos* de R y c' la cota para el número de codos. El problema (*trayectoria*

rectilínea centro con restricción en el número de codos) consiste en minimizar

$$\max_{p_i \in P} d(p_i, R) \quad s.a. : |R| \leq c' \quad (RCC)$$

o bien, usando el concepto de excentricidad definido

$$\min_{R \in \mathcal{R}_P} e(R) \quad s.a. : |R| \leq c'.$$

Dado que un segmento no extremo de la trayectoria posee dos codos, uno a cada lado, y teniendo en cuenta que cada trayectoria puede empezar y terminar con segmentos horizontales o verticales, se cumple que:

$$|R| \leq 2h(R) \leq |R| + 2$$

siendo $h(R)$ el número de segmentos horizontales de R .

Primero, consideramos el caso de trayectorias que empiecen y terminen con segmentos horizontales, en cuyo caso $|R| = 2h(R) - 2$ y el problema será formulado entonces

$$\min_{t \in \mathcal{R}_P} e(R) \quad s.a. : h(R) \leq \frac{c' + 2}{2} = k.$$

A continuación, extenderemos el problema permitiendo un número impar de codos. Además, si la trayectoria se restringe en sus extremos por segmentos verticales o puntos, aparecen variantes del problema libre. Para el primer caso, denominamos $I_e = [y(a), y(a')]$ y $I_s = [y(b), y(b')]$ los intervalos de entrada y salida, situados fuera del rango $X(P)$, esto es, si $x(a)$ y $x(b)$ son las abcisas de los segmentos I_e y I_s , entonces, $x(a) < x(p_i) < x(b)$ para $i = 1, \dots, n$. Si denotamos por $\mathcal{R}_P(I_e, I_s)$ el subconjunto de trayectorias de \mathcal{R}_P que empiezan y acaban en puntos de los segmentos de la restricción, respectivamente, el problema restringido a intervalos puede formularse como

$$\min_{R \in \mathcal{R}_P(I_e, I_s)} e(R) \quad s.a. : |R| \leq c'.$$

En segundo lugar, los intervalos pueden ser degenerados a los puntos O y D (origen y destino respectivamente) y entonces, el problema restringido a puntos consiste en

$$\min_{R \in \mathcal{R}_P(O, D)} e(R) \quad s.a. : |R| \leq c' \quad (RCCR).$$

Nótese que la formulación original resulta como “límite” de la formulación del problema restringido a intervalos cuando $y(a), y(b) \rightarrow +\infty, y(a'), y(b') \rightarrow -\infty$. Sin embargo, resolveremos los problemas restringidos usando el método propuesto para el correspondiente libre.

4.1.2 Propiedades de soluciones óptimas

Para el caso $|R|$ par, una solución es una colección $s_1 s'_1 s_2 s'_2 \dots s'_{k-1} s_k$ de segmentos horizontales s_j y verticales s'_j . Puesto que es suficiente determinar los segmentos horizontales $s_j; j = 1, 2, \dots, k$, identificaremos una trayectoria rectilínea por sus segmentos horizontales.

Cada segmento horizontal s_j está determinado por un par (q_j, l_j) siendo q_j el punto del segmento con menor abcisa y l_j su longitud. Así, tendremos $x(q_{j+1}) = x(q_j) + l_j; j = 1, 2, \dots, k - 1$.

Definición 4.1.1 Dada una trayectoria rectilínea $R \in \mathcal{R}_P$ determinada por los segmentos horizontales $s_1 s_2 \dots s_k$, definimos el *conjunto de asignación del segmento* s_j , $\mathcal{A}(s_j)$, como el conjunto de puntos de P que toman la distancia a R sobre el segmento s_j . Formalmente, diremos que $p \in P$ es un punto de $\mathcal{A}(s_j)$ si, bien $x(q_j) < x(p) < x(q_{j+1})$, bien $x(p) = x(q_j)$, $y(p) \notin I(y(q_{j-1}), y(q_j))$ y $d(p, s_j) < d(p, s_{j-1})$, o bien $x(p) = x(q_{j+1})$, $y(p) \notin I(y(q_j), y(q_{j+1}))$ y $d(p, s_j) < d(p, s_{j+1})$ donde $j = 2, \dots, k - 1$ (Figura 4.1). Para los segmentos extremos, s_1 y s_k la formalización del concepto es análoga .

Obsérvese que para los puntos $p \in P$ tal que $p \notin \bigcup_{j=1}^k \mathcal{A}(s_j)$ cumplen $d(p, s'_i) = 0$ para algún segmento vertical s'_i de R (esto es, están en algún segmento vertical). Así pues, un punto p está en un segmento vertical de la trayectoria y no pertenece a ningún conjunto de asignación, o está contenido en $\mathcal{A}(s_j)$ donde s_j es el segmento horizontal más cercano cumpliendo $x(q_j) \leq x(p) \leq x(q_{j+1})$.

Definición 4.1.2 Dada $R \in \mathcal{R}_P$, decimos que un segmento horizontal de R ,

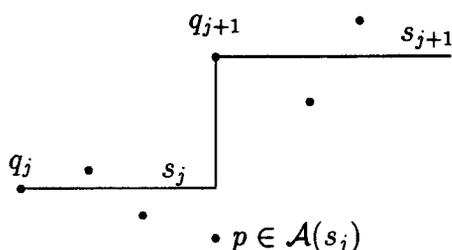


Figura 4.1: Conjuntos de asignación.

s es *segmento-centro* respecto a un conjunto A si se cumple que

$$\max_{p,q \in A} d(p, q) = 2 \max_{p \in A} d(p, s)$$

donde $d(p, s)$ es la distancia vertical de punto a segmento.

Proposición 4.1.1 *Existe una trayectoria solución cuyos segmentos horizontales son segmentos-centro respecto de sus puntos asociados.*

Demostración: Sea R^* una solución de segmentos horizontales $s_1^* s_2^* \dots s_{k-1}^* s_k^*$, o sea:

$$e(R^*) = \min_{R \in \mathcal{R}_P} e(R), \quad c(R^*) \leq c'$$

Si tomamos $s^* \in \{s_1^*, s_2^*, \dots, s_k^*\}$ de forma que $e(s^*) = e(R^*)$, se tiene que $e(R^*) = \max_{p_i \in \mathcal{A}(s^*)} d(p_i, s^*)$. Si s^* no es el segmento centro de $\mathcal{A}(s^*)$, llamando $d = \max_{p_i, p_j \in \mathcal{A}(s^*)} d(p_i, p_j)$, se cumple que $e(s^*) > \frac{d}{2}$ y, por tanto, $e(R^*) > \frac{d}{2}$.

Sea s_c^* el segmento centro de $\mathcal{A}(s^*)$ (se tiene $e(s_c^*) = \frac{d}{2} < e(s^*) = e(R^*)$) y R_c^* la trayectoria resultante de sustituir en R^* , s^* por s_c^* , para todo $s^* \in R^*$ que verifique $e(s^*) = e(R^*)$.

Es fácil deducir que $e(R_c^*) < e(R^*)$, dado que, si existe, en el peor caso, un punto p en una vertical adyacente a algún segmento s^* ($e(s^*) = e(R^*)$) que cumple $d(p, R^*) = 0$ y $d(p, R_c^*) > 0$, se tiene que $d(p, R_c^*) \leq \frac{d}{2}$ al estar p en el conjunto $\mathcal{A}(s_c^*)$ y, por tanto, no se aumenta el valor de $e(R^*) = \frac{d}{2}$ (véase la Figura 4.2).

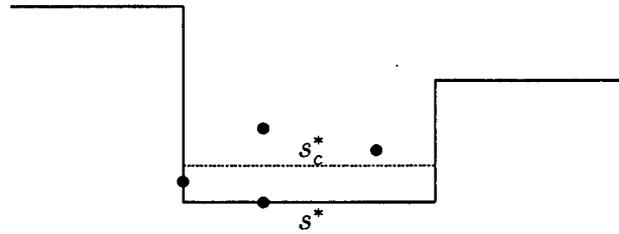


Figura 4.2: Segmentos horizontales centro.

Como consecuencia, podemos encontrar una trayectoria con menor excentricidad que R^* y eso contradice que R^* sea solución. Se concluye, que s^* debe ser el segmento centro. Finalmente, sustituyendo los restantes segmentos horizontales s de R_c^* verificando $e(s) < e(R^*)$ por los segmentos-centro respectivos, la excentricidad total de la trayectoria no se altera. Por tanto, siempre podemos considerar una solución $R_c = s_1 s_2 \dots s_k$, donde s_i es el segmento centro de $\mathcal{A}(s_i)$. \square

Proposición 4.1.2 *Siempre existe una solución R^* que tiene situado cada segmento vertical sobre la abscisa de algún punto de la nube.*

Demostración: Sea R la trayectoria considerada en la proposición 4.1.1. Supongamos que existe un segmento vertical v de la trayectoria cuya abscisa $x(v) \neq x(p_i)$ para todo $p_i \in P$ según ilustra la Figura 4.3. Sean h, h' los segmentos horizontales adyacentes a v . Se tiene entonces que, para todo $p \in \mathcal{A}(h)$ se tiene $x(p) < x(v)$ y, para todo $q \in \mathcal{A}(h')$ es $x(q) > x(v)$.

Sea $q_1 \in \mathcal{A}(h')$ tal que $x(q_1) \leq x(q') \forall q' \in \mathcal{A}(h')$ y v^* el segmento vertical con abscisa $x(v^*) = x(q_1)$ (ver Figura 4.3).

La nueva trayectoria R^* que se obtiene desplazando horizontalmente v hasta v^* cumple $e(R^*) = e(R)$ y $c(R^*) = |R| \leq c'$ y, por tanto, también es solución. \square

Como consecuencia directa de las proposiciones 4.1.1 y 4.1.2 se tiene el teorema siguiente:

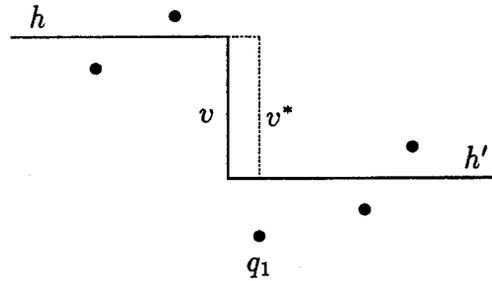


Figura 4.3: Situación de los segmentos verticales.

Teorema 4.1.1 *Existe una solución al problema de la trayectoria rectilínea centro con restricción el número de codos (RCC) que consta de segmentos horizontales centro y segmentos verticales situados sobre la abcisa de algún punto de la nube.*

4.1.3 Búsqueda de soluciones

Los resultados obtenidos en la sección anterior indican que podemos centrar la búsqueda de soluciones sobre trayectorias que verifiquen las condiciones del teorema 4.1.1. En esta sección cuantificaremos el número de trayectorias que se encuentran en esa situación. Para ello, comenzaremos denotando por $\bar{\mathcal{R}}_P$ el subconjunto de trayectorias de \mathcal{R}_P para las que las abcisas de los segmentos verticales coincide con la abcisa de algún punto de P . Sea además, $\bar{\mathcal{R}}_P(k)$ el subconjunto de $\bar{\mathcal{R}}_P$ con k segmentos horizontales.

Si tenemos n puntos de demanda, existen a lo más $\binom{n}{k-1}$ posibilidades para la elección de $k-1$ segmentos verticales. Para cada una de esas configuraciones, los segmentos horizontales(centro) pueden ser calculados después de determinar los correspondientes conjuntos de asignación. Por tanto, actuando con fuerza bruta, la consideración de todas las posibles situaciones de k segmentos verticales y los correspondientes conjuntos de asignación entre cada dos verticales nos llevará a encontrar una solución a nuestro problema. Nos centramos entonces en la obtención de conjuntos de asignación para trayectorias del conjunto $\bar{\mathcal{R}}_P$. Mientras que los puntos cuyas abcisas están situadas

en el interior de algún intervalo determinado por dos segmentos verticales adyacentes serán automáticamente asignados a dicho intervalo, los puntos que pertenezcan a las líneas verticales pueden ser asignados bien a la izquierda o bien a la derecha, o incluso pertenecer al correspondiente segmento vertical.

A continuación, analizamos las distintas asignaciones que pueden ser consideradas para una configuración de abcisas dada $\mathcal{C} = \{x'_1, \dots, x'_{k-1}\}$. Para cada banda vertical determinada por $\{x'_{j-1}, x'_j\}$, consideremos los conjuntos de asignación correspondientes al interior de la banda, $\overline{\mathcal{A}}_1 = \{p \in P : x(p) < x'_1\}$, $\overline{\mathcal{A}}_j = \{p \in P : x'_{j-1} < x(p) < x'_j\}$ y $\overline{\mathcal{A}}_k = \{p \in P : x'_{k-1} < x(p)\}$.

A la hora de añadir puntos situados en las verticales adyacentes a $\overline{\mathcal{A}}_j$ haremos algunas consideraciones. Primero, que los puntos situados en la vertical que delimitan los conjuntos $\overline{\mathcal{A}}_j$ y $\overline{\mathcal{A}}_{j+1}$ que tienen ordenadas en el rango de las ordenadas de los puntos de $\overline{\mathcal{A}}_j \cup \overline{\mathcal{A}}_{j+1}$, serán asignados al conjunto $\overline{\mathcal{A}}_j$ o $\overline{\mathcal{A}}_{j+1}$ con segmento centro más cercano. Por otra parte, los puntos que no cumplan la condición anterior serán agrupados como sigue. Para una configuración dada $\{x'_1, \dots, x'_{k-1}\}$ (Figura 4.4) se consideran los conjuntos

$$\begin{aligned} W_j^- &= \{p \in P : x(p) = x'_j \text{ e } y(p) < y(\overline{p}) \quad \forall \overline{p} \in \overline{\mathcal{A}}_j \cup \overline{\mathcal{A}}_{j+1}\} \\ W_j^+ &= \{p \in P : x(p) = x'_j \text{ e } y(p) > y(\overline{p}) \quad \forall \overline{p} \in \overline{\mathcal{A}}_j \cup \overline{\mathcal{A}}_{j+1}\} \end{aligned}$$

Cabe observar que la asignación de los puntos de cada uno de estos conjuntos puede hacerse a la misma banda, debido a que el punto más lejano a $\overline{\mathcal{A}}_j \cup \overline{\mathcal{A}}_{j+1}$ en cada conjunto determina la excentricidad global. Además, no es necesario considerar todas las posibilidades pues cada conjunto W_j debe ser asignado a la banda de la izquierda o la banda de la derecha, pero lógicamente no a las dos.

Definición 4.1.3 Dada una configuración de abcisas \mathcal{C} , llamamos *asignaciones globales compatibles* a aquellas para las que los conjuntos W_j^+ y W_j^- han sido asignados a una y sólo a una banda adyacente.

Proposición 4.1.3 *El número de asignaciones globales compatibles para una configuración dada es a lo más 2^{k-1} .*

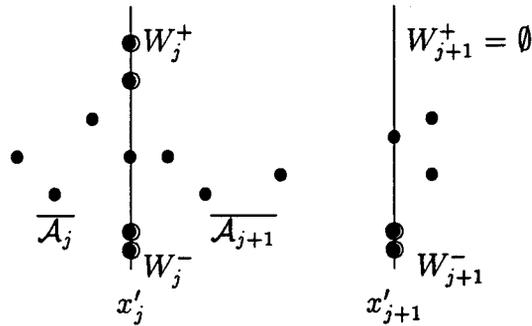


Figura 4.4: Asignación de los puntos en una vertical.

Demostración: Para la primera banda $\{x_1, x_2\}$ hay cuatro posibilidades dependiendo de que cada uno o ambos conjuntos W_1^+ , W_1^- se asignen a \bar{A}_1 . Para la segunda banda, la decisión tomada en la primera va a determinar los conjuntos W pertenecientes a la primera línea vertical que se añadan a \bar{A}_2 , pero hay también cuatro posibilidades para asignar los conjuntos de la segunda vertical. Así pues, para cada banda tenemos 2^2 posibilidades que corresponden a la asignación de la vertical a su derecha y por tanto, el número total es $2^{2(k-1)}$. Sin embargo, en el cómputo se han repetido algunas asignaciones. De hecho, cuando ambos conjuntos W_j^+ y W_j^- van al mismo lado (pongamos a la izquierda), existe otra configuración de abcisas para la que el conjunto de puntos interiores a la banda correspondiente es $\bar{A}_j \cup W_j^+ \cup W_j^-$. Esta configuración tiene los mismos elementos excepto que x'_j ha sido sustituido por la siguiente abcisa. Por tanto, basta considerar 2^{k-1} asignaciones globales compatibles. \square

Basándonos en los resultados obtenidos anteriormente, proponemos un algoritmo que construye una solución del problema RCC para el caso par en la cota, que pasa por considerar todas las posibles configuraciones y sus correspondientes conjuntos de asignación.

Es necesario observar que puede haber infinitas soluciones ya que, los segmentos horizontales que no proporcionan la excentricidad de la trayectoria pueden desplazarse infinitésimamente en dirección vertical de forma que posee un rango de posibles valores. Asimismo, una pequeña traslación de los

segmentos verticales no afecta a la excentricidad total de la trayectoria. Aquí se trata la localización de una solución óptima. Cualquier otra solución en las condiciones del teorema 4.1.1 puede encontrarse a partir de ésta y de las consideraciones anteriores.

ALGORITMO RCC

Entrada: $P = \{p_i, i = 1, \dots, n\}$ y la cota par del número de codos $|R| = c'$.

Salida: La excentricidad de una trayectoria solución.

1. Sea $k = \frac{c'+2}{2}$ y ordenar lexicográficamente la nube P .
2. Para cada $u = 1, 2, \dots, \binom{n}{k-1}$, obtener todas las configuraciones $C_u = \{x'_{1u}, \dots, x'_{k-1u}\}$, que contienen $k-1$ elementos tomados de las abcisas de los puntos de P .
3. Para cada configuración C_u obtener los conjuntos de asignación correspondientes a los puntos del interior de cada banda, $\{\bar{A}_{1u}, \bar{A}_{2u}, \dots, \bar{A}_{ku}\}$.
4. Para cada configuración C_u determinar todas las asignaciones globales compatibles añadiendo los puntos sobre verticales

$$\{\mathcal{A}_{1u}^r, \mathcal{A}_{2u}^r, \dots, \mathcal{A}_{ku}^r\}; \quad r = 1, 2, \dots, 2^{k-1}.$$

5. Para cada configuración $C_u = \{x'_{1u}, \dots, x'_{k-1u}\}$ y cada $r = 1, 2, \dots, 2^{k-1}$ construir la poligonal R_u^r que está formada por segmentos horizontales s_{ju}^r que son segmentos centro correspondientes a los conjuntos \mathcal{A}_{ju}^r , $j = 1, 2, \dots, k$, y que tienen puntos extremos con abcisas

$$\min_{p \in P} \{x(p)\} \text{ y } x'_{1u}, \text{ para el extremo izquierdo;}$$

$$x'_{j-1u} \text{ y } x'_{ju}, \text{ para } j = 2, \dots, k-1;$$

$$x'_{k-1u} \text{ y } \max_{p \in P} \{x(p)\} \text{ para el extremo derecho.}$$

6. Calcular la excentricidad de cada trayectoria R_u^r :

$$e(R_u^r) = \max_{p \in P} d(p, R_u^r) = \max_{1 \leq j \leq k} e(s_j^r u).$$

7. Salir con el valor $\min_{u,r} e(R_u^r)$.

En todo el desarrollo anterior se ha tomado c' par. El caso impar en el número de codos de la restricción puede considerarse como una variante del anterior. En este caso, debemos considerar trayectorias que empiezan o terminan con segmentos verticales, puesto que pueden mejorar la excentricidad de una trayectoria cuyos segmentos extremos sean obligatoriamente horizontales. Esto se debe a que los segmentos verticales pueden contener a algunos puntos. Es fácil probar que siempre existe una solución al problema en el conjunto de trayectorias que empiecen o terminen con un segmento vertical, por lo que nos restringiremos a este conjunto.

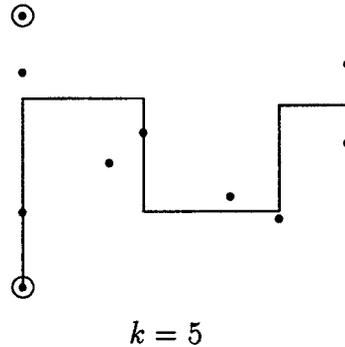


Figura 4.5: Cota impar del número de codos.

Para determinar una poligonal óptima en estas condiciones, consideraremos x_1 la primera abcisa de la nube P y los conjuntos de puntos P_1^+ and P_1^- obtenidos de $P \setminus \{p \in P : x(p) = x_1\}$ añadiendo el punto $p \in P$ tal que $x(p) = x_1$ con la mayor o menor ordenada, respectivamente como se indica en la Figura 4.5. (En el caso de tener el conjunto $\{p \in P : x(p) = x_1\}$ un único elemento, sea $P^+ = P^-$). De igual forma, sea x_m la mayor abcisa de P y P_m^+ ,

P_m^- los conjuntos de puntos de abcisa distinta a x_m añadiéndoles el punto de la vertical $x = x_m$ con mayor y menor ordenada respectivamente.

Para resolver el caso impar, ejecutaremos el algoritmo RCC cuatro veces con $k = E(\frac{c'+2}{2})$ (parte entera de $\frac{c'+2}{2}$), una vez para cada conjunto de puntos P_1^+ , P_1^- , P_m^+ , P_m^- . Para cada caso, la trayectoria se construye añadiendo, al comienzo (o al final) un segmento vertical en el lado opuesto al punto añadido a $P \setminus \{p \in P : x(p) = x_1\}$ ($P \setminus \{p \in P : x(p) = x_m\}$) conectando con el correspondiente segmento horizontal. La excentricidad óptima se obtendrá comparando las cuatro trayectorias consideradas.

Teorema 4.1.2 *Cuando n es el número de puntos de demanda y la constante c' es la cota del número de codos, puede encontrarse una solución al problema RCC en tiempo $O(n^k)$ siendo $k = E(\frac{c'+2}{2})$.*

Demostración: Como es sabido la ordenación lexicográfica se hace en tiempo $O(n \log n)$. Puesto que el número total de configuraciones es $\binom{n}{k-1}$, el paso 1 requiere $O(n^{k-1})$. Por otra parte, la asignación de todos los puntos de cada conjunto \bar{A}_{j_u} para una configuración dada lleva $(k-1)n$ operaciones, esto es, $O(n)$. En el paso 3 debemos obtener el máximo y el mínimo de las ordenadas correspondientes a los puntos de $\bar{A}_{j_u} \cup \bar{A}_{j+1_u}$, y así quedan determinados los conjuntos W_j^+ , W_j^- . Esto requiere tiempo $O(n)$. Dado que por cada configuración existen 2^{k-1} asignaciones compatibles, el paso 3 toma tiempo $O(n)$.

La construcción de todas las trayectorias que resultan de las correspondientes asignaciones compatibles se hace en tiempo $O(n^k)$, teniendo en cuenta que la obtención de un segmento centro es $O(n)$ [61] y que existen $O(n^{k-1})$ configuraciones, para cada una de las cuales tenemos $O(2^{k-1})$ asignaciones compatibles. Finalmente, los pasos 5 y 6 requieren tiempo $O(n^{k-1})$ cada uno y como consecuencia el cómputo total es $O(n^k)$. \square

4.1.4 El problema restringido

Hasta ahora habíamos permitido libertad para los extremos de la trayectoria y reducido el problema a considerar segmentos horizontales tanto en la salida como en la entrada de la poligonal. Sin embargo, si imponemos restricciones en los extremo (el camino comience y termine en dos puntos o dos segmentos verticales prefijados), ciertas modificaciones del algoritmo RCC son necesarias para resolver el problema.

El método usado para el problema no restringido ha consistido en la reducción de candidatos de un conjunto continuo a otro finito considerando trayectorias cumpliendo que todo segmentos horizontal sea segmento-centro de su correspondiente conjunto de asignación. Para obtener un número finito de candidatos para el problema restringido podemos considerar modificaciones convenientes en el conjunto de candidatos que se disponía para el caso libre. La modificación consiste en cambiar los segmentos extremos de la trayectoria, añadiendo un nuevo segmento vertical, si lo permite la cota en los codos, o moviendo los segmentos extremos para conseguir la restricción impuesta.

Hay que tener en cuenta en este apartado que el desplazamiento de los segmentos horizontales extremos puede modificar la excentricidad global. O sea, las trayectorias con menos segmentos horizontales podrían tener una excentricidad menor. Más precisamente, si la cota para el número de codos c' es par, entonces el máximo número de segmentos horizontales es $k = \frac{c'+2}{2}$. Sea \mathcal{R}_k el conjunto de trayectorias que considera el algoritmo RCC. A continuación, consideramos los conjuntos de trayectorias que se derivan del anterior cuando modificamos los segmentos extremos. En primer lugar, sea el conjunto $\mathcal{R}_k^{(1)}$ de las trayectorias obtenidas del conjunto \mathcal{R}_k moviendo, si fuera necesario, uno o ambos segmentos horizontales extremos hasta conseguir el objetivo de la restricción (Figura 4.6 (a)).

Por otra parte, sea $\mathcal{R}_{k-1}^{(1)}$ el conjunto de las trayectorias obtenidas de \mathcal{R}_{k-1} añadiendo dos codos (uno a cada lado) y los correspondientes segmentos verticales que nos garantice el cumplimiento de la restricción (Figura 4.6 (b)).

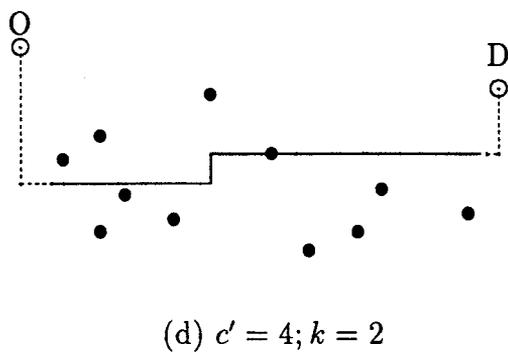
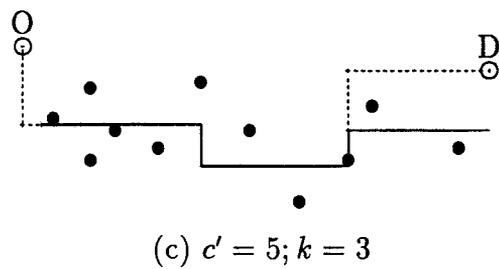
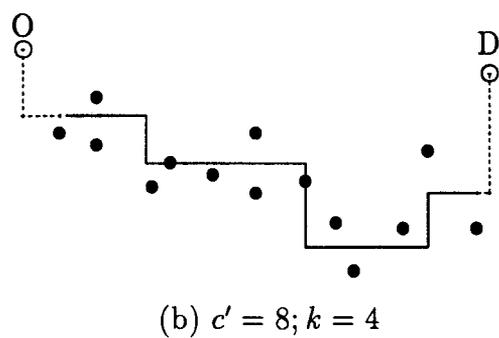
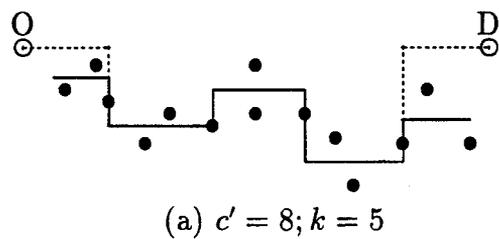


Figura 4.6: Considerando restricciones en los extremos.

Si el máximo número de codos permitido es impar, sea $\mathcal{R}_k^{(2)}$ el conjunto de trayectorias que se obtienen de \mathcal{R}_k modificando (moviendo verticalmente), si es necesario, uno de los segmentos de los extremos y añadiendo un codo y un segmento vertical al otro lado (Figura 4.6 (c)). En este caso, debemos considerar además, el conjunto $\mathcal{R}_{k-1}^{(2)}$ obtenido de \mathcal{R}_{k-1} añadiendo dos codos y dos segmentos verticales, uno a cada lado (Figura 4.6 (d)).

La idea fundamental es que en $\mathcal{R}_k^{(\nu)} \cup \mathcal{R}_{k-1}^{(\nu)}$, $\nu = 1, 2$ existe al menos una solución del problema pues si consideramos cualquier trayectoria con menos segmentos horizontales, existirá otra en $\mathcal{R}_k^{(\nu)} \cup \mathcal{R}_{k-1}^{(\nu)}$ derivada de ella con menor o igual excentricidad.

El siguiente algoritmo encuentra una solución al problema restringido:

Algoritmo RCCR

Entrada: $P = \{p_i, i = 1, \dots, n\}$, la cota par del número de codos $|R| = c'$ y dos intervalos I_e, I_s o dos puntos O, D .

Salida: Una trayectoria de mínima excentricidad.

1. Obtener el conjunto \mathcal{R}_k para $k = E(\frac{c'+2}{2})$ ejecutando los pasos 1,2,3,4 del algoritmo RCC.
Si c' es par ir al paso 2.
Si c' es impar ir al paso 3.
2. Obtener $\mathcal{R}_k^{(1)}$ y $\mathcal{R}_{k-1}^{(1)}$ e ir al paso 4.
3. Obtener $\mathcal{R}_k^{(2)}$ y $\mathcal{R}_{k-1}^{(2)}$ e ir al paso 4.
4. Extraer del conjunto $\mathcal{R}_k^{(\nu)} \cup \mathcal{R}_{k-1}^{(\nu)}$, $\nu = 1, 2$ la trayectoria de mínima excentricidad.

Puesto que la modificación realizada con respecto al algoritmo RCC requiere analizar un número de trayectorias del mismo orden, se obtiene la misma complejidad y se puede establecer el siguiente resultado:

Teorema 4.1.3 *El problema restringido RCCR puede resolverse en tiempo $O(n^k)$ donde k es la cota del número de codos.*

4.1.5 Un algoritmo eficiente para el caso no degenerado

En los algoritmos anteriores, el índice k de la cota del número de codos, aparece en el orden de complejidad. Por ello, éstos son aplicables para valores pequeños de k . Sin embargo, cuando los puntos del conjunto P están en posición no degenerada, esto es, no hay dos puntos sobre la misma vertical, podemos desarrollar un algoritmo eficiente válido para cualquier valor de k .

Supongamos entonces, que los puntos de P (ya ordenados) cumplen $x(p_1) < x(p_2) < \dots < x(p_n)$. Resolveremos el problema sin restricciones en los extremos, resultando el caso restringido una derivación de éste. Así pues, en este caso podemos suponer, sin pérdida de generalidad, que la poligonal buscada empieza y termina con segmentos horizontales (posiblemente degenerados). Es fácil observar que dado $k > 0$, el problema es equivalente a

$$\text{mín máx } d_v(p_i, R) \quad \text{s.a. : } h(R) \leq h$$

donde $h = \frac{k+2}{2}$ y $h(R)$ el número de segmentos horizontales de R .

El método que proponemos se basa en la generación de un conjunto finito de valores candidatos para el valor $e(R^*)$ de cualquier solución. A continuación hacemos la descripción del método: según vimos en una sección anterior (4.1.1), basta buscar una solución entre las trayectorias cuyos segmentos horizontales son segmentos-centro respecto de sus correspondientes conjuntos de asignación. Por tanto, basta considerar los conjuntos de puntos consecutivos

$$S_{12}, S_{13}, S_{14}, \dots, S_{1n}, S_{23}, S_{24}, \dots, S_{2n}, \dots, S_{n-1n}$$

donde $S_{ij} = \{p_i, p_{i+1}, \dots, p_j\}$, $j > i$ y obtener para cada uno de ellos el valor $e_{ij} = \frac{1}{2} \text{máx}_{p,q \in S_{ij}} d_v(p, q)$. Puesto que $e(R^*) \in \{e_{ij}\}_{i < j}$, obtenemos un

conjunto de $\frac{n(n-1)}{2}$ valores candidatos que denominaremos:

$$\mathcal{C} = \{e_{12}, e_{13}, \dots, e_{1n}, e_{23}, e_{24}, \dots, e_{2n}, \dots, e_{n-1n}\}.$$

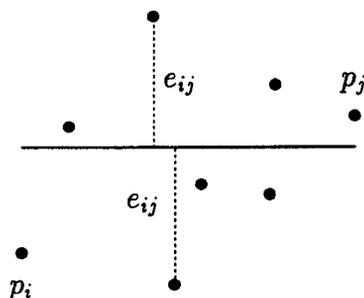


Figura 4.7: Valores candidatos.

Para cada valor del conjunto \mathcal{C} , e_{ij} , resolveremos el problema que llamaremos *inverso* y que proviene de alternar los factores *objetivo* y *restricción*,

$$\text{mín } h(R) \quad \text{s.a. : } d_v(p_i, R) \leq e_{ij}.$$

Denominaremos h_{ij} la solución (veremos que se puede obtener en tiempo lineal). Si para cada valor de e_{ij} resolvemos el correspondiente problema inverso, generamos una lista de pares $(e_1, h_1), (e_2, h_2), \dots, (e_r, h_r)$ de forma que $e_1 < e_2 < \dots < e_r$ y $h_1 \geq h_2 \geq \dots \geq h_r$. Finalmente, realizando búsqueda binaria sobre la lista ordenada de pares, la solución vendrá dada por el primer valor e_j , $j = 1, \dots, r$ que cumpla la restricción, $h_j \leq h$.

Teorema 4.1.4 *Dado $e > 0$, el problema de encontrar una trayectoria rectilínea y x -monótona R con el menor número de segmentos horizontales cuya distancia a cada punto de P es no superior a e , puede resolverse en tiempo $O(n)$.*

Demostración: El problema tiene la siguiente interpretación geométrica: dado un conjunto de segmentos verticales de longitud $2e$ con distinta abscisa, se

trata de encontrar un camino rectilíneo con el mínimo número de segmentos horizontales que atravesase a todos los segmentos, o bien, encontrar el mínimo número de segmentos horizontales consecutivos que “insertan” a los segmentos (Figura 4.8).

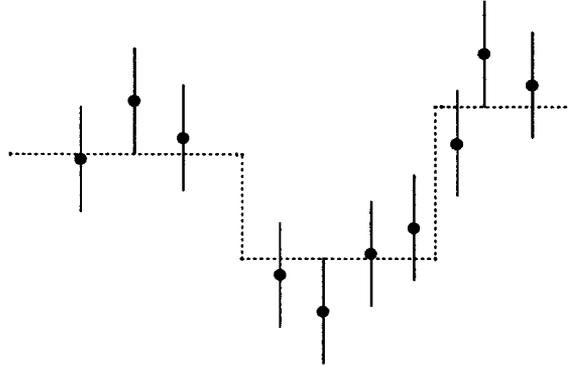


Figura 4.8: Interpretación geométrica del problema.

Puesto que buscamos el menor número de segmentos horizontales consecutivos, podemos resolver el problema obteniendo, mediante barrido de izquierda a derecha, una partición disjunta compuesta por conjuntos maximales de segmentos horizontales consecutivos de forma que cada conjunto pueda ser atravesado por un solo segmento horizontal. En efecto, esto puede realizarse con barrido de izquierda a derecha sobre los puntos $p_i \in P$ usando una recta vertical. Para obtener un proceso lineal, tomaremos dos punteros sobre la recta vertical que indican (en cada posición) la ordenada mayor y menor de los puntos del conjunto que se está construyendo. De esta forma, la actualización de los punteros y la inclusión o no del punto siguiente al conjunto (según la distancia vertical con los anteriores sea menor o igual que $2e$ o no) puede realizarse en tiempo $O(1)$. Como consecuencia, obtenemos una partición del conjunto P , digamos $P = Q_1 \cup Q_2 \cup \dots \cup Q_{h(e)}$ que es maximal en el sentido siguiente: cumpliéndose para cada conjunto $Q_j = \{p_i, p_{i+1}, \dots, p_{i+s}\}$, $j = 1, \dots, h(e)$, se verifica que $\max_{p, q \in Q_j} d_v(p, q) \leq 2e$, y además existen puntos p y q de Q_j de forma que $d_v(p, p_{i+s+1}) > 2e$ y $d_v(p_{i-1}, q) > 2e$. De esta manera, el número mínimo de segmentos lo indica el cardinal de la partición. Hay que observar que si el barrido se realiza de derecha a izquierda, se obtiene en general otra

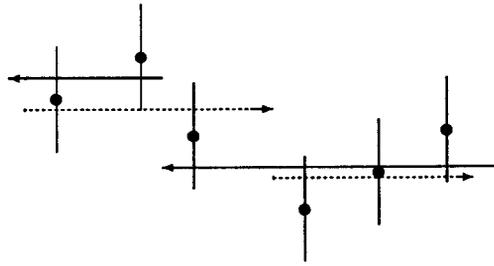


Figura 4.9: Los segmentos de trazo continuo (discontinuo) corresponden al barrido de derecha a izquierda (de izquierda a derecha).

partición, aunque el cardinal de la partición no varía respecto al obtenido en el barrido de izquierda a derecha. Esto es debido a que la consideración de un nuevo conjunto en el barrido equivale a la aparición de un nuevo punto a distancia superior a $2e$ de alguno anterior, por lo que el número de conjuntos de la partición no se altera (ver Figura 4.9). Por tanto, se puede obtener la solución al problema, $h(e)$, en tiempo lineal.

□

Para establecer la complejidad del método propuesto analizaremos el gasto computacional necesario para obtener el conjunto de valores candidatos \mathcal{C} . Para cada valor de i , podemos calcular $e_{i(i+1)}, e_{i(i+2)}, \dots, e_{in}$ en tiempo $O(n)$. En efecto, usando *actualización de datos*, podemos obtener e_{ik} a partir de $e_{i(k-1)}$ sin más que comparar la ordenada del punto p_k con la mayor y menor ordenada del conjunto anterior $S_{i(k-1)}$ (Figura 4.10). Para ello, haremos uso nuevamente del barrido plano de izquierda a derecha usando una recta vertical, en la que se destacarán dos punteros indicando los puntos de mayor y menor ordenada. De esta forma, cada paso del barrido se realiza en tiempo constante y además obtenemos el siguiente valor e_{ik} en tiempo $O(1)$. Por tanto, puesto que la operación anterior se realiza n veces, la actualización de punteros en el método de barrido plano permite calcular eficientemente los valores candidatos de \mathcal{C} en tiempo $O(n^2)$.

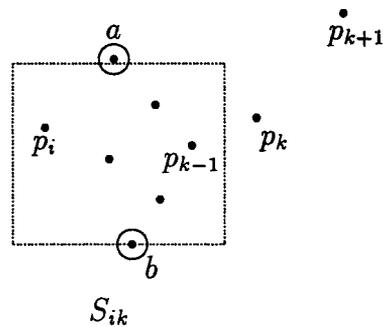


Figura 4.10: Cálculo de valores candidatos: $e_{ik} = e_{ik-1} = d_v(a, b)$, $e_{ik+1} = d_v(p_{k+1}, b) > e_{ik}$.

Una vez calculado el conjunto de valores candidatos, la búsqueda binaria sobre la lista de candidatos ordenada visita $O(\log(n))$ elementos. Puesto que en cada paso de la búsqueda, se obtiene el valor correspondiente h_j en tiempo $O(n)$ (Teorema 4.1.4), encontraremos la solución en tiempo $O(n \log(n))$. Como consecuencia, podemos establecer el siguiente teorema,

Teorema 4.1.5 *El problema (RCC) para el caso de un conjunto P no degenerado, puede resolverse en tiempo $O(n^2)$.*

Finalmente, al igual que se hizo para el caso de posición degenerada de los puntos de P , posee gran interés la consideración del caso restringido, esto es, se tiene una fuente y un sumidero o bien intervalos de entrada y salida a los cuales debe restringirse el camino. A continuación veremos que puede resolverse el problema en tiempo $O(n^2)$ debido a que es suficiente realizar una variante sobre el algoritmo propuesto para el caso no restringido. En efecto, imponemos como restricción sobre la trayectoria el que comience desde una *ventana* o segmento vertical I_e y termine en otro segmento prefijado I_s (Figura 4.11) (obsérvese que la consideración de fuente y sumidero es un caso particular de éste). Para resolver el problema modificamos el algoritmo anterior en los siguientes términos: puesto que un camino solución verifica nuevamente la condición de segmento-centro para sus segmentos horizontales no extremos, es válido el procedimiento anterior pero adaptando la búsqueda binaria sobre la

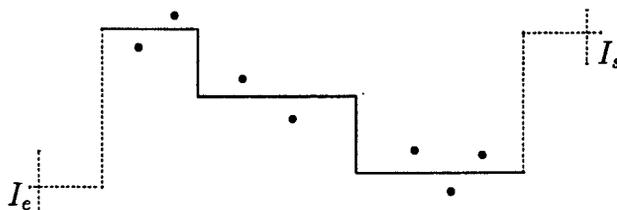


Figura 4.11: Adaptación al caso restringido.

lista ordenada de valores candidatos $e_1 < e_2 < \dots < e_r$ de la forma que sigue:

1. Se obtiene el primer valor e_j de forma que $h_j \leq h$. Llamar e^* a e_j y continuar.
2. (a) Si el camino construido como solución del problema $\min h(R)$ s.a. : $d(p_i, R) \leq e^*$ cumple la restricción, entonces e^* es la solución.
(b) En caso contrario, continuar.
3. (a) Si $h_j \leq h - 2$, basta añadir dos segmentos horizontales en los extremos del camino de forma que el trazado resultante verifique la restricción y e^* es la solución (véase la Figura 4.11).
(b) Si $h_j = h$ o $h_j = h - 1$, se continua la búsqueda binaria sobre e_{j+1} . Hacer $e^* = e_{j+1}$ e ir al paso 2.

4.2 Restricción en la longitud

La restricción sobre el número de codos o segmentos del camino proviene del interés por conseguir trayectorias “simples” en el sentido de que la existencia de gran cantidad de codos es no deseable (en problemas de diseño de circuitos integrados por ejemplo). Sin embargo, en otros contextos, como en el trazado de redes de transporte, el número de vértices no es tan relevante como la

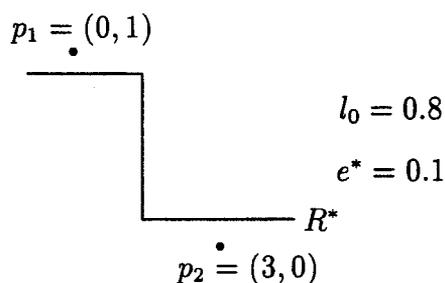


Figura 4.12: Los segmentos horizontales no son segmentos-centro.

longitud. En este caso, la poligonal representa un servicio para el cual la longitud es más costosa que el número de codos.

Si para el trazado de una trayectoria o cadena rectilínea x -monótona R a través de un conjunto de puntos, P , se dispone de una longitud máxima l_0 y marcamos como objetivo el hecho de que la cadena esté “lo menos alejada posible” de todos los puntos de la nube, debemos resolver el problema

$$\min_R \max_i d(p_i, R) \quad \text{s.a. : } l(R) \leq l_0 \quad (RCL).$$

Aunque la formulación de este problema difiere del anterior únicamente en el factor de restricción, es esencialmente distinto. Resulta fácil observar que, en este caso, puede no existir una cadena óptima para la cual los segmentos horizontales son segmentos-centro respecto del conjunto de puntos que toman la distancia sobre él (ver figura 4.12). Por tanto, podemos decir que el cambio en el factor de restricción resulta esencial.

Así pues, se requiere buscar otro tipo de caracterizaciones para una solución. Con este objetivo, definiremos los conceptos de “conjunto de puntos alternados de una nube” y “puntos extremales de una nube con respecto a una cadena”:

Definición 4.2.1 Llamaremos *conjunto de puntos alternados* de P (respecto de la ordenación de la nube P), al conjunto $A_P = \{a_1, a_2, \dots, a_m\}$ de los puntos correspondientes a los máximos y mínimos locales de la función cuya gráfica es el conjunto de puntos P .

Obsérvese que dos puntos consecutivos de A_P , a_i, a_{i+1} , determinan una banda horizontal en cuyo interior se encuentran todos los puntos comprendidos entre a_i y a_{i+1} . Por otra parte, puesto que un extremo local puede darse en varios puntos con la misma ordenada, diremos que A_P es único en el sentido de que sus puntos son únicos respecto a las ordenadas. El conjunto de puntos alternados para la nube de la Figura 4.13 es $A_P = \{p_1, p_2, p_3, p_4, p_7, p_9, p_{11}, p_{14}\}$.

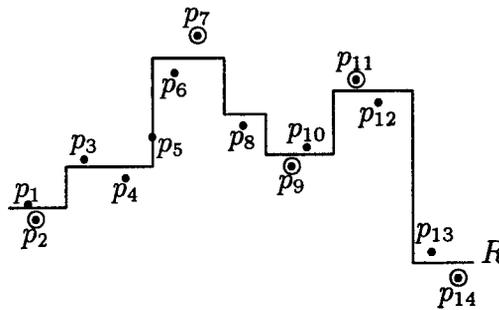


Figura 4.13: Puntos extremales de una cadena R .

Sea $\mathcal{A}(P)$ el conjunto de subconjuntos de A_P cuyos puntos están en posición alternada, i.e., $\mathcal{A}(P) = \{B \in \mathcal{P}(A_P) : A_B = B\}$, siendo A_B el conjunto de puntos alternados de B .

Una cadena rectilínea x -monótona R que atraviesa de izquierda a derecha una nube P , puede descomponerse en un conjunto de subcadenas maximales xy -monótonas (monótonas en las direcciones OX y OY), digamos R_1, R_2, \dots, R_t . Cada subcadena maximal empieza y termina con segmentos horizontales puesto que, sin pérdida de generalidad, podemos suponer este hecho sobre la cadena general R . Por tanto, podemos decir que la intersección entre dos subcadenas maximales consecutivas es un segmento horizontal.

Definición 4.2.2 Sea R_i una subcadena xy -monótona de R . Denotando por $P(R_i) \subset P$, $i = 1, \dots, t$ el conjunto de puntos p_i cuya abcisa pertenece al rango de abcisas de los segmentos que componen R_i , llamaremos *puntos extremales* de P con respecto a R_i , $E(R_i)$, a los elementos del par de puntos más alejados de $P(R_i)$ (respecto a la distancia vertical). Finalmente, sea $E(R) = \bigcup_{i=1}^t E(R_i)$

el conjunto de *puntos extremales* de P con respecto a R . (En la Figura 4.13 $E(R) = \{p_2, p_7, p_9, p_{11}, p_{14}\}$).

De la definición anterior se extrae directamente la siguiente propiedad:

Proposición 4.2.1 *El conjunto de puntos extremales de cualquier cadena R es un conjunto de puntos en posición alternada, en particular, $E(R^*) \in \mathcal{A}(P)$.*

4.2.1 Análisis de soluciones

A continuación, establecemos una condición necesaria para las soluciones del problema propuesto:

Proposición 4.2.2 *Si R^* es una cadena óptima para el problema RCL con excentricidad $e^* = e(R^*)$, entonces $d(q, R^*) = e^*$ para cada $q \in E(R^*)$.*

Demostración: Supongamos que existe $q \in E(R^*)$ cumpliéndose $e^* > d(q, R^*)$. En ese caso, el segmento de R^* sobre el que q toma la distancia, S_q , puede situarse a distancia e^* de q y obtenemos una nueva cadena con igual excentricidad global pero menor longitud. Esta diferencia de longitud puede usarse para situar a todos los segmentos correspondientes a puntos de $E(R^*)$ a una distancia menor que e^* y, por tanto, obtener una nueva cadena factible con menor excentricidad que R^* . Esto contradice el hecho de que R^* es óptima. \square

Puede observarse que en el cálculo de la longitud total de una cadena óptima no intervienen los puntos no extremales (la longitud de la poligonal situada entre dos puntos extremales consecutivos es igual a la suma de la diferencia absoluta de las abcisas mas la diferencia absoluta de las ordenadas de los segmentos correspondientes). Como consecuencia de esto y, teniendo en cuenta que $E(R^*) \in \mathcal{A}(P)$ (Proposición 4.2.1), buscaremos elementos de $\mathcal{A}(P)$ que verifiquen la condición necesaria de la Proposición 4.2.2. Llamaremos en esta ocasión *conjuntos candidatos* a dichos subconjuntos de $\mathcal{A}(P)$.

Puesto que, tanto la excentricidad como la longitud de una cadena rectilínea viene determinada por los puntos extremales, podemos obtener una cadena óptima (no es única) encontrando sus puntos extremales. Como sabemos, deben ser puntos en posición alternada y situados a la misma distancia de la cadena. Además, se verifica la siguiente propiedad:

Proposición 4.2.3 *Si R^* es una solución del problema (RCL), entonces se cumple $l(R^*) = l_0$.*

Demostración: Sea R^* una solución y $E(R^*)$ el conjunto de sus puntos extremales. Si $l(R^*) < l_0$, la cantidad de longitud sobrante, $l_0 - l(R^*) > 0$, permite construir otra poligonal deslizando los segmentos horizontales de R^* en dirección a los puntos de $E(R^*)$. De esta forma, se obtiene una poligonal con el mismo conjunto de puntos extremales y con menor excentricidad, lo que contradice la optimalidad de R^* . \square

Dado un conjunto candidato $Q = \{q_1, q_2, \dots, q_r\}$ y $k > 0$, pueden darse dos situaciones:

1. Que para cada par de puntos consecutivos $q_j, q_{j+1} \in Q$, se verifique que $d(q_j, q_{j+1}) > 2k$.
2. Que exista al menos un par de puntos consecutivos para el cual no se verifique la condición anterior.

En el primer caso y, teniendo en cuenta la condición necesaria de la proposición 4.2.2, podemos situar un segmento horizontal a distancia k de cada punto extremal para construir una posible solución. En otro caso, si $d(q_j, q_{j+1}) \leq 2k$ para algún j , entonces la cadena que contiene un sólo segmento horizontal para el par $\{q_j, q_{j+1}\}$ posee menos longitud que la que asigna un segmento a cada punto y, como consecuencia, aplicando la proposición 4.2.3, el conjunto Q no será el conjunto de puntos extremales de una solución. Deducimos de lo anterior la propiedad siguiente:

Proposición 4.2.4 Si $Q = \{q_1, q_2, \dots, q_r\}$ es el conjunto de puntos extremales de una cadena óptima R^* y $e(R^*) = k$, entonces se cumple que $d(q_j, q_{j+1}) > 2k$, $j = 1, \dots, r - 1$.

Por otro lado, puesto que la trayectoria buscada es x-monótona, la suma de las longitudes de los segmentos horizontales está fijada (es la diferencia absoluta entre las abscisas del primer y último punto de la nube). Por esta razón, es suficiente tener en cuenta la suma de las longitudes de los segmentos verticales, denominada *longitud vertical* de la cadena, $l_v(R)$. El problema a resolver quedaría formulado entonces:

$$\min_R \max_i d(p_i, R) \quad \text{s.a.} \quad l_v(R) \leq \bar{l} \quad (RCL_v).$$

Desde luego, el problema resulta trivial si $\bar{l} \geq \sum_{j=1}^{m-1} d(a_j, a_{j+1})$ siendo $\{a_1, a_2, \dots, a_m\} = A_P$ el conjunto de puntos alternados de P . En este caso, una solución viene dada por la trayectoria x-monótona que pasa por todos los puntos de P . Supondremos en lo que sigue que tal condición no se verifica para el factor de restricción \bar{l} .

Las dos proposiciones siguientes nos indican cuál debe ser la excentricidad de una cadena solución y cómo debe estar colocada con respecto a sus puntos extremales:

Proposición 4.2.5 Sea $Q = \{q_1, q_2, \dots, q_r\}$ el conjunto de puntos extremales de una cadena óptima R^* del problema (RCL_v) , entonces, se cumple que

$$e(R^*) = \frac{\sum_{j=1}^{r-1} d(q_j, q_{j+1}) - \bar{l}}{2(r-1)}.$$

Demostración: Denotemos $e(R^*) = k$ por comodidad. Teniendo en cuenta los resultados anteriores, para cada $j = 1, 2, \dots, r - 1$ debe cumplirse que $d(q_j, R^*) = k$ y $d(q_j, q_{j+1}) > 2k$. Así pues, la longitud de la cadena óptima vendrá dada por $l(R^*) = \sum_{j=1}^{r-1} [d(q_j, q_{j+1}) - 2k]$. Por otra parte, si R^* es

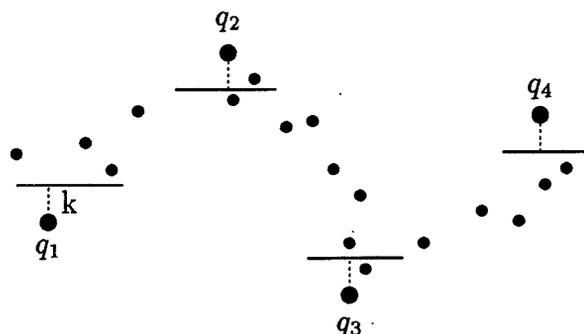


Figura 4.14: Situación de los segmentos horizontales.

óptima, $l(R^*) = \bar{l}$ por la proposición 4.2.3. Como consecuencia de lo anterior, se obtiene

$$k = \frac{\sum_{j=1}^{r-1} d(q_j, q_{j+1}) - \bar{l}}{2(r-1)}.$$

□

Proposición 4.2.6 Sea $Q = \{q_1, q_2, \dots, q_r\}$ el conjunto de puntos extremales de una solución R^* , k el valor expresado por la fórmula de la proposición anterior. Si se cumple que $d(q_j, q_{j+1}) > 2k$ para $j = 1, \dots, r-1$, entonces, los segmentos horizontales extremos de las subcadenas xy -monótonas de R^* se encuentran situados en el interior de las bandas horizontales determinadas por cada dos puntos consecutivos de Q y a distancia k de dichos puntos.

Demostración: Puesto que dos puntos extremales consecutivos de R^* determinan una banda horizontal que contiene a todos los puntos comprendidos entre ellos, los segmentos horizontales sobre los que toman la distancia los puntos de Q , deben estar incluidos en cada banda (Figura 4.14). Por otro lado, si R^* es una solución, los segmentos horizontales (extremos de cada subcadena xy -monótona) deben estar a distancia menor o igual a k . Si los colocamos a distancia menos que k , la trayectoria no sería factible pues la longitud superaría el valor máximo l_0 . Por tanto, deben estar colocados a distancia justamente k . □

4.2.2 Descripción del método

El estudio de las propiedades de una solución descritas en la sección anterior sugiere como método el ir generando conjuntos candidatos y construyendo una cadena rectilínea para cada uno, de forma que ésta se encuentre a una distancia común de cada uno de los puntos del conjunto. El primer conjunto candidato a considerar será el conjunto de puntos extremales correspondientes a la cadena que pasa a través de todos los puntos de P , R_0 . Es fácil observar que esos puntos extremales coinciden precisamente con el conjunto de puntos alternados de P , esto es, $E(R_0) = A_P = \{q_1^0, q_2^0, \dots, q_{r_0}^0\}$. Entonces, si $d(q_j^0, q_{j+1}^0) > 2k_0$, $\forall j = 1, \dots, r_0 - 1$, con el valor

$$k_0 = \frac{\sum_{j=1}^{r_0-1} d(q_j^0, q_{j+1}^0) - \bar{l}}{2(r_0 - 1)},$$

se podría construir una cadena situando los segmentos horizontales extremos de cada subcadena xy -monótona a distancia k_0 de cada punto $q_j \in E(R_0)$ y el resto de los segmentos horizontales de las subcadenas sobre cada punto intermedio. La poligonal así construida será óptima por la proposición 4.2.6. Si por el contrario, para algún $q_j \in E(R_0)$ se tiene $d(q_j, q_{j+1}) \leq 2k_0$, entonces la cadena así construida no es óptima debido a la Proposición 4.2.4. En este caso, debemos buscar el siguiente conjunto candidato. Obsérvese que, para mejorar la excentricidad de la cadena anterior debemos asignar un único segmento horizontal a cada par $\{q_j^0, q_{j+1}^0\}$ de $E(R_0)$ que verifique $d(q_j^0, q_{j+1}^0) \leq 2k_0$. De hecho, más de un par de puntos consecutivos pueden verificar la desigualdad anterior. Esto nos lleva a la consideración de familias de puntos consecutivos de $E(C_0)$ cumpliendo que la distancia para cada par de puntos de la familia no supere la cantidad $2k_0$. De este modo, dado k_0 , se puede obtener una partición disjunta de $E(R_0)$,

$$E(R_0) = \cup_{\nu=1}^{\gamma(k_0)} F_\nu(k_0), \quad F_\nu(k_0) = \{p_{i(\nu)}, p_{i(\nu)+1}, \dots, p_{i(\nu)+l}\}$$

de forma que existen $q, q' \in F_\nu(k_0)$ con $d(q, p_{i(\nu)-1}) > 2k_0$ y $d(q', p_{i(\nu)+l+1}) > 2k_0$.

Con objeto de elaborar un nuevo conjunto candidato, tomaremos de cada familia $F_\nu(k_0)$ el punto más alejado respecto a la familia anterior $F_{\nu-1}(k_0)$,

obteniendo así un conjunto de puntos que denominaremos *conjunto de representantes* de la partición $\{F_\nu\}_{\nu=1}^{\gamma(k_0)}$ que lo denotaremos por $\mathcal{R}(k_0)$ (para comenzar este proceso, elijeremos los puntos representantes de las dos primeras familias como los más alejados entre sí). Puesto que el conjunto candidato debe estar en posición alternada, esto es, $E(R^*) \in \mathcal{A}(P)$, el nuevo candidato vendrá dado por el subconjunto de puntos en posición alternada de $\mathcal{R}(k_0)$, $A_{\mathcal{R}(k_0)}$.

Para el nuevo candidato $E_1 = \{q_1^1, \dots, q_{r_1}^1\}$, obtendremos el valor

$$k_1 = \frac{\sum_{j=1}^{r_1-1} d(q_j^1, q_{j+1}^1) - \bar{l}}{2(r_1 - 1)}.$$

Razonando de igual forma, si $d(q_j^1, q_{j+1}^1) > 2k_1, j = 1, \dots, r_1 - 1$, entonces k_1 corresponde a una trayectoria óptima ($k_1 = e(R^*)$). En otro caso, pasaremos a construir un nuevo conjunto candidato E_2 .

En general, dado $E_{\alpha-1}$ and $k_{\alpha-1}$, siendo

$$k_{\alpha-1} = \frac{\sum_{j=1}^{|E_{\alpha-1}|-1} d(q_j^{\alpha-1}, q_{j+1}^{\alpha-1}) - \bar{l}}{2(|E_{\alpha-1}| - 1)},$$

podemos obtener una cadena óptima de excentricidad $k_{\alpha-1}$ cuando $\forall j = 1, \dots, |E_{\alpha-1}|$, se cumple $d(q_j^{\alpha-1}, q_{j+1}^{\alpha-1}) > 2k_{\alpha-1}$. En otro caso, creamos la partición de conjuntos maximales $F_\nu(k_{\alpha-1})$, $\nu = 1, 2, \dots, \gamma(k_{\alpha-1})$, de "altura" no superior a $2k_{\alpha-1}$, así como el conjunto de representantes $\mathcal{R}(k_{\alpha-1})$ de la forma ya indicada anteriormente. $E_\alpha = A_{\mathcal{R}(k_{\alpha-1})}$ es el siguiente conjunto candidato y

$$k_\alpha = \frac{\sum_{j=1}^{|E_\alpha|-1} d(q_j^\alpha, q_{j+1}^\alpha) - \bar{l}}{2(|E_\alpha| - 1)}$$

es el nuevo valor posible para la excentricidad buscada.

Este proceso da lugar a una secuencia finita

$$(E_0, k_0), (E_1, k_1), \dots, (E_{\alpha-1}, k_{\alpha-1}), (E_\alpha, k_\alpha), \dots, (E_{\alpha^*}, k_{\alpha^*}),$$

cumpléndose $E_\alpha \in \mathcal{A}(P)$, $E_{\alpha+1} \subset E_\alpha$ y $k_\alpha < k_{\alpha+1}$. El par $(E_{\alpha^*}, k_{\alpha^*})$ será el primero que cumpla $d(q_j^{\alpha^*}, q_{j+1}^{\alpha^*}) > 2k_{\alpha^*}$ para cada $j = 1, \dots, |E_{\alpha^*}| - 1$.

Puesto que esta construcción barre todos los posibles candidatos que verifican las condiciones de las proposiciones de la sección anterior, el par $(E_{\alpha^*}, k_{\alpha^*})$ da la solución al problema.

Basado en la descripción anterior, proponemos el siguiente algoritmo que proporciona una solución al problema:

Algoritmo RCL

Entrada: $P = \{p_i, i = 1, \dots, n\}$ con $x(p_1) < x(p_2) < \dots < x(p_n)$ y la cota de longitud l_0 .

Salida: $k^* = e(R^*)$ y $E(R^*)$.

1. Obtener el conjunto de puntos alternados de P , $A_P = \{a_1, \dots, a_m\}$, y denotar $\bar{l} = l_0 - (x(p_n) - x(p_1))$.
2. Si $\bar{l} \geq \sum_{j=1}^{m-1} d(a_j, a_{j+1})$, salir con $k^* = 0$ y $E(R^*) = A_P$.
En caso contrario, hacer $\alpha = 0$, $q_j^\alpha = a_j$, $E_\alpha = E_0$ y pasar a 3.
3. Mientras para algún $j \in \{1, 2, \dots, |E_\alpha| - 1\}$, $d(q_j^\alpha, q_{j+1}^\alpha) \leq 2k_\alpha$ con

$$k_\alpha = \frac{\sum_{j=1}^{|E_\alpha|-1} d(q_j^\alpha, q_{j+1}^\alpha) - \bar{l}}{2(|E_\alpha| - 1)}, \quad q_j^\alpha \in E_\alpha,$$

hacer $\alpha = \alpha + 1$ y $E_\alpha = A_{\mathcal{R}(k_{\alpha-1})}$.

Salir con $k^* = k_\alpha$ y $E(R^*) = E_\alpha$.

A continuación hacemos un análisis de la complejidad del método propuesto: dado un conjunto de puntos ordenados, podemos obtener el conjunto de puntos alternados, A_P , en tiempo lineal, usando sin más un algoritmo incremental que decide en tiempo $O(1)$ si un punto es máximo o mínimo local. La unicidad del conjunto (respecto a las ordenadas), permite usar el barrido de izquierda a derecha. Puesto que el test del paso 2 es $O(n)$, concluimos que los dos primeros pasos requieren tiempo $O(n)$. Sin embargo, el paso 3 conlleva

el cálculo de un nuevo conjunto candidato y podemos descomponerlo en tres apartados:

- Construcción de la partición $\{F_\nu\}_{\nu=1}^{\gamma(k_\alpha)}$.
- Determinar el conjunto representante $\mathcal{R}(k_\alpha)$ de la partición.
- Seleccionar el subconjunto de $\mathcal{R}(k_\alpha)$ que están en posición alternada, $A_{\mathcal{R}(k_\alpha)}$.

Los dos primeros apartados pueden hacerse a la vez con barrido plano de izquierda a derecha de la siguiente forma: se toma una recta vertical en la que se consideran dos punteros que indican la mayor y menor ordenada de los puntos que se vayan incluyendo en cada familia y que iremos actualizando cada vez que se introduzca un nuevo punto. Así, para cada punto bajo consideración se decide si pertenece o no a la familia de la partición en tiempo $O(1)$ comparando con los puntos de mayor y menor ordenadas de dicha familia. De esta manera, se obtendrán a la misma vez los conjuntos $\{F_\nu\}_{\nu=1}^{\gamma(k_\alpha)}$ y $\mathcal{R}(k_\alpha)$ en tiempo $O(n)$. Puesto que los puntos extremales de una solución son únicos (con respecto a sus ordenadas), se puede realizar el barrido de derecha a izquierda o en sentido contrario. La partición puede ser distinta aunque el nuevo conjunto candidato $A_{\mathcal{R}(k_\alpha)}$ es el mismo.

Por otra parte, como ya se ha indicado, la selección de los puntos alternados de un conjunto es lineal, por lo que $A_{\mathcal{R}(k_\alpha)}$ se realiza en tiempo $O(n)$. Por tanto, cada iteración del paso 3 del algoritmo se hace en tiempo $O(n)$. Finalmente, como el número de etapas que tendremos que recorrer es, en el peor caso, del orden de n (puesto que el cardinal de los sucesivos conjuntos candidatos va disminuyendo al menos en un punto), el algoritmo es de complejidad $O(n^2)$ y podemos establecer el siguiente resultado:

Teorema 4.2.1 *El problema de encontrar una trayectoria rectilínea con restricción en la longitud que se ajuste a una nube de puntos (no degenerada) en el sentido de Chebichev (RCL) puede resolverse en tiempo $O(n^2)$.*

Para el caso degenerado puede usarse el mismo método teniendo en cuenta la siguiente consideración. Puesto que el conjunto de puntos alternados de un conjunto depende de la ordenación de éste, se considerará en todo momento, la ordenación de los puntos que proporciona el camino rectilíneo de mínima longitud que pasa por cada uno de ellos. El algoritmo (RMCD) (capítulo 3) resuelve dicho problema en tiempo lineal y, por consiguiente, la complejidad final $O(n^2)$ no se altera.

4.3 Conclusiones y problemas abiertos

En este capítulo se ha abordado la localización óptima de caminos rectilíneos u ortogonales para el criterio minimax. Se ha supuesto que el acceso de los puntos de demanda al camino se realiza en la dirección OY, esto es, hemos considerado la distancia vertical de punto a trayectoria definida en el capítulo anterior.

El problema resulta trivial si no se consideran restricciones sobre el camino. Aquí se han considerado restricciones en términos del número de codos y la longitud del camino. Los problemas resultantes son esencialmente distintos puesto que, las soluciones correspondientes no verifican las mismas propiedades. Sin embargo, el método usado en ambos casos es similar: analizar propiedades de una solución, con objeto de acotar el rango de las mismas. Desde luego, la propia naturaleza de cada problema sugiere después usarlas de una u otra forma.

Cuando la restricción aparece sobre el número de codos de la trayectoria, basta buscar una solución en el conjunto de caminos cuyos segmentos horizontales son segmentos-centro respecto del conjunto de puntos que toman la distancia sobre él y, cuyos segmentos verticales se encuentran sobre la abscisa de algún punto de la nube. Este hecho sugiere un procedimiento de obtención y comparación sobre todos los caminos en esas condiciones. Desafortunadamente, la complejidad del método aumenta exponencialmente con el máximo número de codos permitido, por lo que reduce su aplicabilidad al caso de

caminos con reducido número de codos. Sin embargo, para el caso no degenerado (no existen dos puntos con la misma abcisa), proponemos un algoritmo eficiente de complejidad $O(n^2)$ no aplicable para el caso general. Constituye un ejemplo para el que la posición degenerada de los puntos es esencial. Resultados parciales de este problema han sido presentados en congresos de ámbito nacional [20] e internacional [21].

El método usado para resolver el problema con restricción sobre la longitud del camino, se basa, en sentido intuitivo, en la continuidad de la función *longitud* (frente a la no continuidad de la función *número de codos*) de la trayectoria. De hecho, se trata de *ajustar* a una nube de puntos una trayectoria de longitud dada, de forma que se encuentre a una distancia común de una serie de puntos de la nube (*extremos* de la trayectoria). El algoritmo propuesto requiere tiempo $O(n^2)$ cuando los puntos están en posición no degenerada. Por su parte, el caso degenerado puede resolverse con una variante del algoritmo propuesto (la degeneración no es esencial para este problema). Una versión de este trabajo aparece en [22].

En la Tabla 4.3 se resumen los resultados obtenidos para los problemas propuestos de camino rectilíneos en los capítulos 3 y 4.

Servicios rectilíneos	mín $l(R)$	mín máx $d_v(p_i, R)$
$d_\infty(p_i, R) \geq k$	$O(n^3)$	N.A.
$d_v(p_i, R) \geq k$	$\theta(n \log(n))$	N.A.
$d_v(p_i, R) \leq k$	$O(n \log(n))$	N.A.
$ R = k$	N.A.	$O(n^2)$
$l(R) = l_0$	N.A.	$O(n^2)$

Tabla 4.3. Algunos problemas de localización para servicios rectilíneos en el plano.

Capítulo 5

El problema minimax para poligonales

5.1 Antecedentes

Trataremos en este capítulo el trazado de caminos poligonales genéricos en el plano que sean óptimos para el criterio minimax. La búsqueda de poligonales óptimas (atendiendo a distintos criterios) ha constituido un tema de interés en las dos últimas décadas habiendo sido tratada por gran grupo de investigadores de distintos campos (véase [48]). En la mayoría de los textos recientes del área de Geometría Computacional aparece un problema íntimamente relacionado con la búsqueda de cadenas poligonales óptimas: la *aproximación de curvas poligonales*, que posee múltiples aplicaciones. De hecho, las curvas lineales a trozos han sido usadas a menudo a la hora de aproximar fronteras de complicadas figuras en cartografía, reconocimiento de patrones, diseño por ordenador [71], [9], [11].

En los trabajos realizados en este campo, entre los que debemos destacar los de Imai e Iri [45], [46], Toussaint [87], Melkman y O'Rourke [62], Chan y Chin [10], Guibas et al. [39], entre otros, se plantea el problema de aproximar una poligonal dada por otra cuyos vértices comparten abcisa o bien, son un

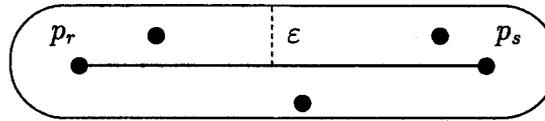


Figura 5.1: Interpretación del error correspondiente a un segmento.

subconjunto de los de la poligonal de entrada. Formalmente, sea una curva poligonal C , definida por el conjunto de sus vértices $V(C) = \{p_1, p_2, \dots, p_n\}$, esto es, p_1, p_2, \dots, p_n es una secuencia de puntos en el plano y cada par de puntos p_i, p_{i+1} , $i = 1, \dots, n - 1$ están conectados por un segmento. El segmento $\overline{p_r p_s}$ puede ser usado para aproximar la curva poligonal dada por p_r, p_{r+1}, \dots, p_s y el error de $\overline{p_r p_s}$ puede definirse como la máxima distancia entre el segmento $\overline{p_r p_s}$ y cada punto p_k comprendido entre p_r y p_s (i.e. $r \leq k \leq s$). La distancia $d(\overline{p_r p_s}, p_k)$ entre el segmento $\overline{p_r p_s}$ y el punto p_k se define como $d(\overline{p_r p_s}, p_k) = \min_{p \in \overline{p_r p_s}} \{d(p, p_k)\}$ siendo $d(p, p_k)$ la distancia euclídea entre puntos. De esta manera, el *error* del segmento $\overline{p_r p_s}$ (denotado $e(r, s)$) se define como $\max_{r \leq k \leq s} \{d(\overline{p_r p_s}, p_k)\}$. En este contexto, se dice que C' es una poligonal aproximante de C si su conjunto de vértices, $V(C') = \{p_{i_1}, p_{i_2}, \dots, p_{i_m}\}$ es un subconjunto de $V(C)$ siendo $i_1 = 1$, $i_m = n$ y $1 < m \leq n$. El *error* de la curva aproximante C' se define como el máximo error de los segmentos que la componen, esto es, $e(C') = \max_{1 \leq k \leq m} \{e(i_k, i_{k+1})\}$. Nótese que $d(\overline{p_r p_s}, p_k) \leq \epsilon$ si y sólo si p_k está contenido en el entorno mostrado en la Figura 5.1.

En los artículos mencionados se consideran dos tipos de problemas de optimización bajo el marco de aproximación de una poligonal:

1. **min-#**: Dado $\epsilon \geq 0$, construir una poligonal aproximante con error no superior a ϵ y con el mínimo número de codos.
2. **min- ϵ** : Dado un valor natural $k > 0$, construir una curva poligonal aproximante con un número de codos no superior a k y con mínimo error.

Lógicamente este tipo de problemas permite múltiples variantes si imponemos

o no restricciones en la ubicación de los vértices o bien, si consideramos un error u otro en la aproximación (véase el artículo de Imai e Iri [46]).

Justamente el concepto de error definido coincide con el de excentricidad de una cadena poligonal que atraviesa una nube de puntos $P = \{p_1, \dots, p_n\}$, $e(C) = \max_{p_i \in P} \{d_2(p_i, C)\}$, donde $d_2(p, C)$ denota la distancia de punto a poligonal inducida por la métrica euclídea. Por consiguiente el problema **min-#** se interpreta como un problema de localización-rutas con cobertura, donde se pretende minimizar el número de codos de la poligonal (en el capítulo 3 se definió este problema para poligonales rectilíneas de mínima longitud) y, por otra parte, el problema **min- ε** corresponde a un problema minimax para poligonales genéricas. En ambos problemas, la distancia considerada es la euclídea y los algoritmos más eficientes obtenidos por los autores referenciados son:

1. Un algoritmo de tiempo $O(n^2)$ para el problema **min-#** [10].
2. Un algoritmo $O(n^2 \log(n))$ para el problema **min- ε** [10].

Sin embargo, hay un problema íntimamente conectado con los anteriores que surge de forma natural en contextos de Estadística, Investigación Operativa o Inteligencia Artificial: el ajuste de una poligonal a una nube de puntos. Es sabido que en este tipo de problemas de *regresión o ajuste* se usa la distancia vertical en la definición del error. Precisamente, en un trabajo reciente de Hakimi y Schmeichel [40] se plantea el siguiente problema: dado un conjunto de datos (puntos en \mathbb{R}^2) $P = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ con $x_1 < x_2 < \dots < x_n$, se trata de construir una función poligonal (o sea, continua y lineal a trozos) f con un número limitado de vértices (puntos no diferenciables) que ajuste el conjunto P . Y para medir la calidad del ajuste emplean dos criterios:

- El número de vértices de la gráfica de f .
- $d_\infty(f, P) = \max_{1 \leq i \leq n} |y_i - f(x_i)|$ (error de Chebychev del ajuste).

Precisamente, la herramienta clave que usan para diseñar sus algoritmos es la propuesta por Imai e Iri [46]: un algoritmo lineal que obtiene el polígono de visibilidad desde un eje en un polígono monótono. Concretamente adaptan el método de Imai e Iri al problema con distancia vertical y obtienen algoritmos de tiempo $O(n^2 \log(n))$ para el problema minimax siguiente:

Dado P y un entero $k \geq 0$, construir una función poligonal f con k o menos codos de forma que $d_\infty(f, P)$ sea lo menor posible.

Aunque Hakimi y Schmeichel [40] resuelven el problema tanto si los vértices de la función poligonal aproximante son puntos de la entrada o puntos cualesquiera, en las hipótesis del problema aparece la nube de puntos en “posición general”, esto es, sin degeneraciones (así se consideraba igualmente en los trabajos de Imai e Iri [45], [46]). No contemplan por tanto el caso de puntos con la misma abcisa, situación muy común si los puntos representan demanda.

Nosotros centraremos la atención en este capítulo sobre los problemas de criterio minimax para una poligonal x -monótona con cota en el número de vértices y , al igual que se hizo en el capítulo anterior, sustituiremos el factor de la restricción número de codos por longitud de la poligonal, puesto que en nuestro contexto de Localización de Servicios, C interpreta un camino a través del conjunto de puntos P . Además, planteamos los problemas con puntos origen y final correspondientes a los extremos del trayecto que determina C . La distancia utilizada va a ser (como en el trabajo de Hakimi y Schmeichel [40]), la distancia vertical de punto a poligonal definida como se hizo en capítulos anteriores para poligonales ortogonales: dado un punto $r \in \mathbb{R}^2$ comprendido entre a y b y una poligonal monótona C , se define la distancia vertical como la distancia euclídea entre r y su proyección vertical sobre C :

$$d_v(r, C) = \min\{d(r, s) / s \in C, x(s) = x(r)\}$$

Es usual en problemas de regresión, y aquí haremos lo mismo, abusar de la notación designando por $d(r, C)$, la distancia vertical y llamando distancia a la distancia vertical. En cada discusión se señalará si la nube de puntos puede contener degeneraciones y propondremos, en cualquier caso, algoritmos que

mejoran la complejidad de algunos de los obtenidos por Hakimi y Schmeichel [40].

5.2 Definición de los problemas

Sean $a, b \in \mathbb{R}^2$ los puntos entre los cuales se encuentra incluida la nube de puntos $P = \{p_1, \dots, p_n\}$. Las coordenadas cartesianas de p_i se designarán como se hizo a lo largo de toda la memoria por $(x(p_i), y(p_i))$, donde $x(a) \leq x(p_i) \leq x(b)$, para todo $i = 1, \dots, n$. Diremos que $C = \{c_0, c_1, \dots, c_k, c_{k+1}\}$ es una *cadena poligonal* que *une* a y b si $c_0 = a$ y $c_{k+1} = b$. Llamaremos a los puntos c_1, \dots, c_k *codos* de la cadena poligonal y denotaremos por $|C| = k$ el número de codos. Al igual que se hizo en los problemas análogos de cadenas ortogonales consideraremos caminos monótonos con respecto al eje x .

Las poligonales que unen a y b se pueden clasificar en dos tipos:

- *poligonales discretas*, cuando cada codo esté situado sobre algún punto de la nube P .
- *poligonales libres*, si los codos están situados en puntos arbitrarios comprendidos entre las rectas $x = x(a)$ y $x = x(b)$.

Basándonos en dicha clasificación, dividimos el capítulo en dos secciones en las que se plantean los siguientes problemas de optimización:

$\mathcal{P}1$: Encontrar una poligonal discreta con un codo que minimice la máxima distancia de los puntos a la poligonal:

$$\min_C \{ \max_{p_i \in P} \{ d(p_i, C) \} \} \quad \text{s.a. : } |C| = 1 \quad \text{y} \quad c_1 \in P.$$

Véase Figura 5.2 (a).

$\mathcal{P}2$: Dados dos convexos P_1 y P_2 , encontrar una poligonal discreta con un codo que empiece en P_1 , acabe en P_2 , y que minimice la máxima

distancia de los puntos a la poligonal:

$$\min_C \{ \max_{p_i \in P} \{ d(p_i, C) \} \} \quad \text{s.a. : } |C| = 1, c_1 \in P, a \in P_1, b \in P_2.$$

Véase Figura 5.2 (d).

$\mathcal{P}3$: Encontrar una poligonal discreta con k codos que minimice la máxima distancia de los puntos a la poligonal:

$$\min_C \{ \max_{p_i \in P} \{ d(p_i, C) \} \} \quad \text{s.a. : } |C| = k \quad \text{y} \quad c_i \in P, \quad i = 1, \dots, k$$

Véase Figura 5.2 (b).

$\mathcal{P}4$: Dado $l_0 > 0$, encontrar una poligonal discreta cuya longitud no supere el valor l_0 y que minimice la máxima distancia de los puntos a la poligonal:

$$\min_C \{ \max_{p_i \in P} \{ d(p_i, C) \} \} \quad \text{s.a. : } l(C) \leq l_0.$$

$\mathcal{P}5$: Encontrar una poligonal libre con un codo que minimice la máxima distancia de los puntos a la poligonal:

$$\min_C \{ \max_{p_i \in P} \{ d(p_i, C) \} \} \quad \text{s.a. : } |C| = 1 \quad \text{y} \quad c_1 \in \mathbb{R}^2.$$

Véase Figura 5.2 (c).

Concretamente, nuestra aportación respecto a algunos de los algoritmos aparecidos en [40] es:

- Un algoritmo $O(n \log(n))$ para calcular una poligonal discreta de un codo, que mejora el existente de tiempo $O(n^2)$.
- Un algoritmo $O(n^2)$ para calcular una poligonal discreta de k codos, que mejora el actual de $O(n^2 \log(n))$.
- Un algoritmo $O(n^2)$ para calcular una poligonal libre de un codo, que es mejora el propuesto en [40] de tiempo $O(n^2 \log(n))$.

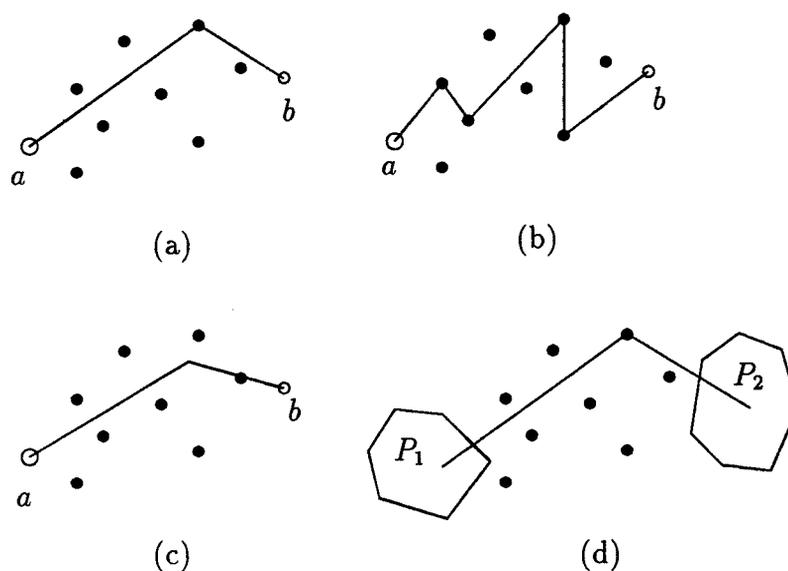


Figura 5.2: Distintos problemas de optimización.

Además, los algoritmos que se proponen son válidos para el caso de que el conjunto de puntos P esté en posición degenerada. Por otra parte, se obtienen algoritmos de tiempo $O(n^2 \log(n))$ para los problemas $\mathcal{P}2$ y $\mathcal{P}5$ y se plantean problemas abiertos que aparecen de forma natural por proximidad en la formulación con los definidos aquí.

5.3 Poligonales discretas

5.3.1 Poligonales de un solo codo

El problema de hallar la mejor poligonal discreta con un codo correspondiente al problema \mathcal{P}_1 admite una solución directa por fuerza bruta. En efecto, el número de cadenas candidatas es $\theta(n)$, una por cada punto p_1, \dots, p_n . Para una candidata $\{a, c_i, b\}$ en concreto, hay que calcular $n - 1$ distancias al resto de los puntos, resultando una complejidad total de $O(n^2)$.

Sin embargo, se puede conseguir un algoritmo más eficiente haciendo

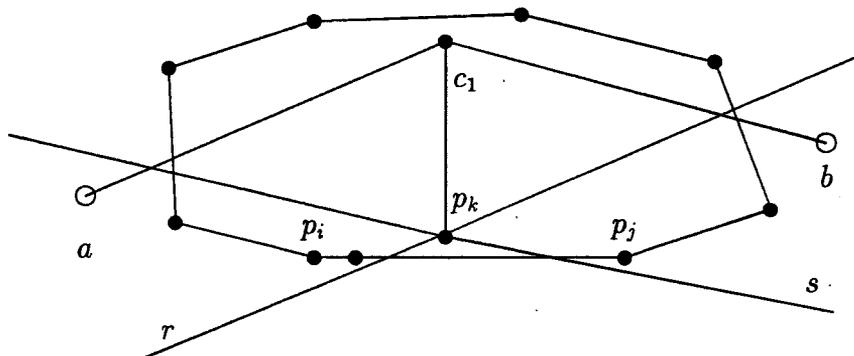


Figura 5.3: El máximo de la distancia no se alcanza en el cierre convexo.

uso del cierre convexo. Si disponemos del cierre convexo de $\{a, p_1, \dots, p_n, b\}$ podemos usar las siguientes propiedades, que permiten bajar la complejidad a $O(n \log(n))$.

Lema 5.3.1 *Sea $C = \{a, c_1, b\}$ una cadena poligonal discreta que une a y b . Entonces el máximo de la distancia $d(p_i, C)$, para todo i , se alcanza sobre un vértice del cierre convexo, o bien sobre un vértice interior situado en la recta vertical $x = x(c_1)$.*

Demostración: Dada una cadena $C = \{a, c_1, b\}$, el máximo de la distancia $d(p_i, C)$ no es necesariamente un vértice del cierre convexo de $\{a, p_1, \dots, p_n, b\}$, como se muestra en la Figura 5.3.

Refiriéndonos a dicha figura, se demuestra fácilmente la máxima distancia se alcanza en el punto p_k que, sin embargo, no pertenece al cierre convexo. En efecto, el punto p_i tiene menos distancia a la poligonal $C = \{a, c_1, b\}$ que p_k puesto que se encuentra incluido en la banda dada por las rectas ac_1 y r , y a la izquierda de la recta $x = x(c_1)$. Lo mismo le ocurre a p_j , situado en la banda dada por las rectas c_1b y s , y a la derecha de la recta $x = x(c_1)$. El resto de los puntos del cierre convexo se halla a menor distancia de la cadena que p_k .

Lo que sí es cierto es que, si una recta cualquiera interseca a un polígono

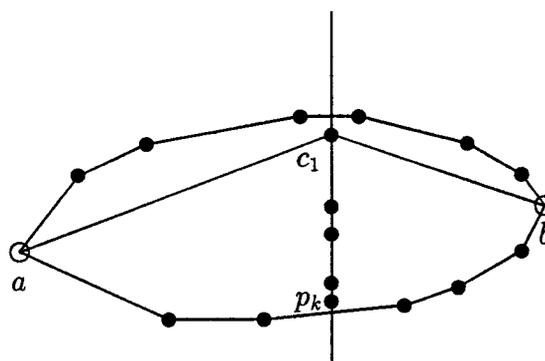


Figura 5.4: Descomposición del cierre convexo por la recta $x = x(c_1)$.

convexo, entonces la máxima distancia vertical entre los puntos (cuyo cierre convexo es el polígono) y la recta se alcanza en un vértice del polígono debido a la convexidad de éste. Para el caso de nuestra poligonal $\{a, c_1, b\}$, descomponemos el cierre convexo de $\{a, p_1, \dots, p_n, b\}$ trazando la recta $x = x(c_1)$, como muestra la Figura 5.4.

La recta $x = x(c_1)$ da lugar a dos subpolígonos convexos cuyos vértices coinciden con los del cierre convexo salvo, posiblemente, dos de ellos (las intersecciones de $x = x(c_1)$ con el cierre convexo). En virtud de la observación anterior, el máximo de la distancia de los puntos a la cadena se alcanzará en alguno de los vértices de los subpolígonos convexos. Si esto ocurre en un vértice del cierre convexo original, hemos acabado. Cuando, por el contrario, el máximo se alcance en uno de los nuevos vértices, lo despreciamos y buscamos el máximo en el resto de los vértices situados sobre la recta $x = x(c_1)$. En todo caso, hay que calcular la máxima distancia con respecto a los puntos de P situados sobre la recta $x = x(c_1)$. Entre ambas distancias, la dada por los vértices originales del cierre convexo y la dada por los puntos de P en $x = x(c_1)$, elegimos, de nuevo, la mayor. \square

Lema 5.3.2 *Sea R un polígono convexo y t una recta que interseca a R . La función distancia es unimodal (alcanza un único valor máximo) en cada una de las dos cadenas en que t divide a R .*

Demostración: Sea $\{r_1, \dots, r_n\}$ el conjunto de los vértices de R dados en orden positivo y designemos por $\{r_1, \dots, r_i\}$ una de las cadenas formadas por la intersección de R y t . Supongamos que en $r_j, 1 \leq j \leq i$, se produce un máximo local. Probaremos que r_j es, en realidad, el único máximo posible y, por lo tanto, el máximo global. Si hubiese otro máximo, digamos $r_l, j < l \leq i$, necesariamente habría un mínimo r_k entre r_j y r_l . Pero en r_k se rompería la convexidad de la cadena $\{r_1, \dots, r_i\}$, ya que el ángulo dado por $r_{k-1}r_kr_{k+1}$ habría de ser mayor que π para que r_k sea mínimo. Se obtiene así una contradicción. \square

Con la ayuda de los dos lemas anteriores podemos probar el siguiente teorema, que resuelve $\mathcal{P}1$.

Teorema 5.3.1 *El problema $\mathcal{P}1$, encontrar una poligonal discreta de un codo, puede resolverse en tiempo $O(n \log(n))$ y espacio lineal.*

Demostración: A continuación se hace la descripción del algoritmo que resuelve el problema y que se compone de dos pasos principales: primero, el cálculo del cierre convexo de $\{a, p_1, \dots, p_n, b\}$ y de otra estructura de datos auxiliar D y, segundo, el cómputo, para cada punto $p_i \in P$, de la máxima distancia de la nube a $\{a, p_i, b\}$. Finalmente se selecciona la mínima entre todas las distancias máximas.

Como se en muestra el lema 5.3.1, para una cadena $\{a, p_i, b\}$, el máximo no se alcanza necesariamente en el cierre convexo. Con el fin de tener en cuenta esa situación, se construye una matriz D que contiene al conjunto $\{a, p_1, \dots, p_n, b\}$ ordenado lexicográficamente. Además, en cada elemento de la matriz se guardan punteros a los puntos de menor y mayor ordenada de entre todos los puntos con la misma abcisa x . Construir D lleva tiempo $O(n \log(n))$.

El siguiente paso es mostrar cómo, dada una cadena $\{a, c_1, b\}$, donde $c_1 = p_i$ para algún i , se puede calcular en tiempo $O(\log(n))$ la máxima distancia a esa cadena. Suponemos que el cierre convexo está almacenado en sentido positivo en una matriz. Para descomponer el cierre convexo en dos subpolígonos

convexos como los que aparecen en el lema 5.3.1 en tiempo $O(\log(n))$, se procede de la forma siguiente. Tomando la recta $x = x(c_1)$ orientada hacia $+\infty$, se realiza búsqueda dicotómica en cada una de las dos cadenas monótonas con respecto al eje x en que se divide el polígono. Accediendo en tiempo constante al elemento mitad y a su anterior y se comprueba a qué lado de la recta se encuentran ambos. Si están uno a cada lado, hemos encontrado un punto de intersección; si ambos están a la derecha, se busca recursivamente en la submatriz izquierda; y si no es así, miramos en la submatriz derecha. Este procedimiento garantiza que las intersecciones entre la recta $x = x(c_1)$ y el cierre convexo se hallan en tiempo $O(\log(n))$.

Una vez descompuesto el cierre convexo en dos subpolígonos convexos, se calcularán las distancias máximas en cada subpolígono. La máxima distancia de los puntos de cada subpolígono a la recta correspondiente se calcula en tiempo $O(\log(n))$ con búsqueda binaria sobre los vértices ordenados del subpolígono. Si se alcanza en un vértice que proviene de la intersección de la recta vertical con el cierre inicial, se debe buscar el máximo sobre los puntos de la nube contenidos en el segmento común a ambos subpolígonos. Llamamos la atención sobre el hecho de que, en el segmento común a ambos subpolígonos, puede haber varios puntos de P . No obstante, de todos esos puntos, sólo es relevante el que está a más distancia de c_1 y la estructura D proporciona ese punto en tiempo $O(1)$. \square

5.3.2 Poligonales de un codo apoyadas en convexos

En algunos problemas reales, el camino poligonal que se pretende trazar no está restringido a un origen y destino puntual, sino a regiones de salida y llegada. Concretamente, planteamos el caso para el que las regiones de la restricción son polígonos convexos. Así pues, aparece el problema que hemos denominado *poligonales apoyadas en convexos*. Nosotros proponemos un algoritmo eficiente para el caso de poligonales de un solo codo.

Como paso intermedio para la solución del problema de las poligonales de un codo apoyadas en convexos, consideraremos el problema previo siguiente:

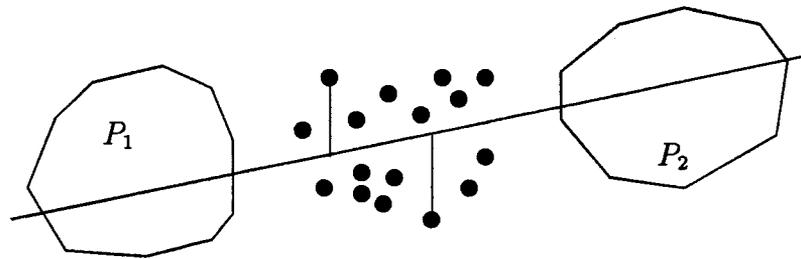


Figura 5.5: Cálculo de la recta óptima que pasa por dos polígonos convexos.

$\mathcal{P}2.1$: Dada una nube de puntos y dos polígonos convexos P_1 y P_2 , encontrar la recta que minimiza la máxima distancia vertical a los puntos y que tiene intersección no vacía con P_1 y P_2 .

La primera observación que hacemos sobre $\mathcal{P}2.1$ es a propósito de una propiedad de la solución. La recta solución tiene que equidistar de dos puntos situados en semiplanos distintos (véase la Figura 5.5). Si no fuese así, podríamos mover infinitesimalmente la recta y la solución óptima sería mejorada, lo cual es una contradicción.

¿Qué significado tiene la recta solución en el dual? La recta solución se transforma en un punto y los puntos de los que equidista, en sendas rectas. La aplicación dual polar [70] $(a, b) \mapsto y = 2ax - b$ no conserva la distancia euclídea, pero sí conserva en cambio la distancia vertical [12]. Entonces, en el arreglo de rectas, la recta solución corresponde al punto medio del menor de los mayores segmentos verticales que se pueden inscribir en una cara del arreglo.

Sin embargo, lo anterior no tiene en cuenta que la recta deba intersectar a los dos polígonos convexos. En el dual un polígono convexo se transforma en dos cadenas convexas. La región comprendida entre ellas representa las rectas que intersecan al polígono. Dado que hay dos polígonos convexos, salen cuatro cadenas y las rectas que intersecan a ambos polígonos convexos se identifican

con los puntos que pertenecen al exterior de las cuatro cadenas.

La porción del arreglo que se debe considerar es la que se encuentra en la región descrita. Calcular el arreglo entero de rectas costaría $\Theta(n^2)$ en tiempo y espacio; sin embargo, se puede evitar esa complejidad y hacerlo en tiempo $O(n \log(n))$ y espacio $O(n)$. Para ello, basta hacer uso de la siguiente propiedad: toda recta solución determina un par antipodal, a saber, el dado por los dos puntos que realizan la mínima distancia (véase de nuevo la Figura 5.5). Como hay un número lineal de pares antipodales, se obtienen también un número lineal de candidatos. En el dual, esto se traduce en que el número de intersecciones de las rectas comprendidas entre las cadenas convexas es lineal.

Finalmente, mediante un algoritmo de barrido se puede recorrer la parte del arreglo que nos interesa. Este algoritmo puede diseñarse adaptando el de Bentley y Ottman [6] y el de Nievergelt y Preparata [66].

Concluimos con el siguiente teorema:

Teorema 5.3.2 *El problema $\mathcal{P}2.1$ puede resolverse en tiempo $O(n \log(n))$ y espacio $O(n)$.*

Una vez resuelto el problema $\mathcal{P}2.1$ es sencillo resolver el problema $\mathcal{P}2$, esto es, el problema de encontrar la poligonal discreta de un codo que se apoya en los dos convexas y que minimiza la máxima distancia. Dado que cada punto de la nube es un convexo, podemos calcular todas las posibles poligonales que se apoyan en P_1 , un punto de la nube y, en este punto y en P_2 . Como hay un número lineal de esas poligonales, y para cada una de ellas, se emplea tiempo $O(n \log(n))$, en total, tenemos tiempo $O(n^2 \log(n))$ y espacio lineal.

Concluimos la sección con este teorema:

Teorema 5.3.3 *El problema $\mathcal{P}2$ puede resolverse en tiempo $O(n^2 \log(n))$ y espacio lineal.*

5.3.3 Poligonales de k codos

Para el problema $\mathcal{P3}$, cálculo de una poligonal discreta de k codos que minimiza la máxima distancia a los puntos, recurrimos a la técnica de programación dinámica. La razón para ello es que las soluciones del problema para valores menores que k participan en la solución final. Introduzcamos previamente algunas definiciones que harán más cómoda la descripción del algoritmo de programación dinámica.

Definición 5.3.1 Sean $a = p_0, p_1, p_2, \dots, p_n, p_{n+1} = b$ los puntos de la nube P ordenados lexicográficamente.

1. Denotaremos por $\mathcal{C}_j(a, p_i)$ el conjunto de las cadenas de j codos que unen a con $p_i, i = 1, \dots, n$.
2. Si $C \in \mathcal{C}_j(a, p_i)$, la *excentricidad* de C , $e(C)$, por:

$$e(C) = \text{máx}\{d(p_1, C), \dots, d(p_{i-1}, C)\}.$$

3. El *coste* desde a hasta $p_i, i = 1, \dots, n+1$, es el valor dado por la fórmula:

$$\gamma_j(a, p_i) = \text{mín}\{e(C) / C \in \mathcal{C}_j(a, p_i)\}, \quad j = 0, \dots, k.$$

Usando esta terminología, buscar una solución al problema $\mathcal{P3}$ es equivalente a calcular el coste $\gamma_k(a, b)$. El lema siguiente muestra la relación entre el coste para el punto p_i y los costes para puntos anteriores.

Lema 5.3.3 *El coste $\gamma_j(a, p_i), j < i$, es igual al menor valor positivo del siguiente conjunto de costes:*

$$\{\gamma_{j-1}(a, p_j), \gamma_0(p_j, p_i), \gamma_{j-1}(a, p_{j+1}), \gamma_0(p_{j+1}, p_i), \dots, \gamma_{j-1}(a, p_{i-1}), \gamma_0(p_{i-1}, p_i)\}.$$

Demostración: Sea $C = \{a, p_{k_1}, \dots, p_{k_j}, p_i\}$ la cadena de coste $\gamma_j(a, p_i)$, esto es, la cadena con excentricidad óptima. Esta cadena puede descomponerse en dos

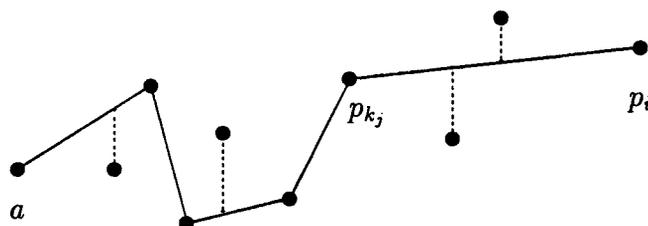


Figura 5.6: Una de las dos subcadenas es óptima.

subcadenas, a saber, $\{a, p_{k_1}, \dots, p_{k_j}\}$ y $\{p_{k_j}, p_i\}$; la primera subcadena tiene $j - 1$ codos y la segunda, cero codos (Figura 5.6). Puesto que una de estas dos subcadenas debe tener excentricidad $\gamma_j(a, p_i)$, el valor mínimo positivo entre los costes $\gamma_{j-1}(a, p_{k_j})$ y $\gamma_0(p_{k_j}, p_i)$ será el coste de $\gamma_j(a, p_i)$. Como el índice k_j puede variar entre j e $i - 1$, han de considerarse todas esas cadenas candidatas, por tanto se obtiene la fórmula expresada. \square

Una vez probado el lema anterior podemos continuar con el siguiente teorema:

Teorema 5.3.4 *El problema $\mathcal{P}3$, esto es, encontrar una cadena discreta de k codos, puede resolverse en tiempo $O(n^2)$ y espacio lineal.*

Demostración: El primer paso, ordenar los puntos por coordenada x lleva tiempo $O(n \log(n))$. Después, se requiere hacer el cálculo de cada uno de los costes $\gamma_0(p_k, b), \dots, \gamma_0(p_n, b)$, lo cual lleva tiempo lineal para cada uno de ellos, y por tanto, $O(n^2)$ en total.

Finalmente, el cálculo recursivo de los costes γ_k usando el lema 5.3.3, da lugar a la ecuación $T(n) = T(n - 1) + O(n)$. El término $O(n)$ es debido al cálculo del valor mínimo positivo de entre los costes. La solución de la ecuación es $T(n) = O(n^2)$, lo que proporciona la cota para el tiempo establecida por el teorema. En cada paso sólo es necesario guardar los valores de los costes del paso anterior. El número de dichos costes son a lo más n , lo cual prueba la cota para el espacio. \square

A modo de observación, cabe destacar que los algoritmos propuestos en estas dos primeras secciones (poligonales con uno y k codos respectivamente) contemplan la posibilidad de que dos puntos puedan tener la misma abcisa. Por tanto, no sólo mejoran la complejidad de los algoritmos hasta ahora conocidos, sino que son válidos también para el caso degenerado.

5.3.4 Poligonales con longitud restringida

Como ya se hizo en el capítulo anterior, consideramos aquí el problema minimax para poligonales (genéricas en este caso) cuando se dispone de una longitud máxima a invertir en el trazado del camino. Supongamos que los puntos de P están en posición no degenerada, esto es, $x(a) = x(p_0) < x(p_1) < \dots < x(p_n) < x(p_{n+1}) = x(b)$. El problema en cuestión puede formularse como

$$\min_C \{ \max_{1 \leq i \leq n} \{ d(p_i, C) / l(C) \leq l_0 \} \} \quad (\mathcal{P4})$$

Resolveremos el problema en esta ocasión basándonos en un problema previo que proviene de alterar el orden de los factores objetivo y restricción y cuya formulación es la siguiente:

P2.1: Dado $\lambda > 0$, encontrar una poligonal discreta C de mínima longitud que conecte los puntos a y b y cumpliendo $d(p_i, C) \leq \lambda$, para $i = 1, \dots, n$.

Por analogía con un problema similar planteado en el capítulo 3, podemos denominarlo *poligonal discreta de mínima longitud con cobertura indirecta*:

$$\min_C l(C) \quad \text{s.a. : } d(p_i, C) \leq \lambda \quad (\text{PDMCI}).$$

Para la resolución de este problema usaremos un método tradicional (véase [46]) que consta de dos pasos. Primero se construye un grafo dirigido, que llamaremos $G(\lambda)$, de forma que sus vértices son los puntos de la entrada $a = p_0, p_1, \dots, p_n, p_{n+1} = b$ y sus aristas son los pares (p_i, p_j) que verifican que $d(p_k, \overline{p_i p_j}) \leq \lambda$, para todo punto p_k comprendido entre p_i y p_j .

Si ponderamos cada arista del grafo (p_i, p_j) con la longitud del segmento $\overline{p_i p_j}$, un camino de mínima longitud sobre el grafo que conecte los puntos a y b ,

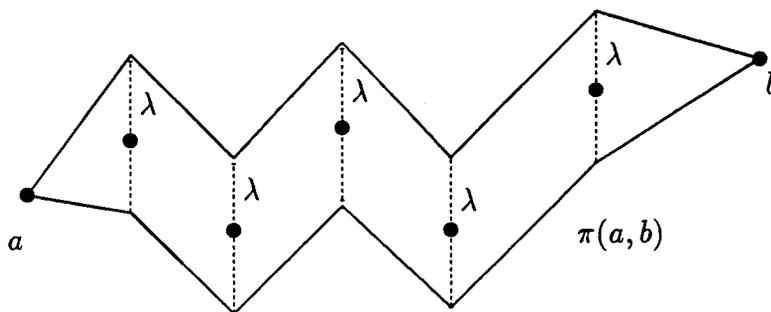


Figura 5.7: Construcción del polígono monótono $\pi(a, b)$.

proporciona la solución al problema. Así pues, la resolución del problema pasa por la construcción del grafo mencionado mas la aplicación de un algoritmo de camino más corto sobre grafos.

Lema 5.3.4 *Se puede construir el grafo $G(\lambda)$ en tiempo $O(n^2)$.*

Demostración: Dados los puntos $a, p_1, p_2, \dots, p_n, b$, considerar para cada $i = 1, 2, \dots, n$ los puntos $(x(p_i), y(p_i) + \lambda)$ y $(x(p_i), y(p_i) - \lambda)$ y construir el polígono monótono (ordenado) conectando los puntos $a, (x(p_1), y(p_1) + \lambda), (x(p_2), y(p_2) + \lambda), \dots, (x(p_n), y(p_n) + \lambda), b, (x(p_n), y(p_n) - \lambda), \dots, (x(p_1), y(p_1) - \lambda), a$ como indica la Figura 5.7. Denotaremos por $\pi(a, b)$ dicho polígono.

Para determinar el conjunto de puntos conectados en $G(\lambda)$ con cada punto p_i , consideraremos el polígono de visibilidad sobre $\pi(a, b)$ desde el punto $p_i, i = 1, \dots, n$, que denominamos $V(p_i)$. De esta manera, (p_i, p_j) es una arista de $G(\lambda)$ si y sólo si p_j está en el polígono de visibilidad $V(p_i)$.

Puesto que $V(p_i)$ puede construirse en tiempo $O(n)$ (véase [45]), podemos determinar los vertices conectados con p_i en tiempo lineal. Como consecuencia directa, tenemos construido el grafo en tiempo $O(n^2)$. \square

Finalmente, podemos usar el algoritmo de Dijkstra para encontrar el camino más corto desde a hasta b en el grafo $G(\lambda)$. Nuestro grafo no tiene ciclos, está dirigido y los pesos son positivos, luego podemos encontrar el camino

más corto en tiempo $\theta(|V| + |A|)$ donde $|V|$ y $|A|$ denotan el número de vértices y aristas respectivamente del grafo ([14]). $G(\lambda)$ contiene n vértices y, en el peor caso, $O(n^2)$ aristas, por lo que el algoritmo de Dijkstra adaptado al grafo dirigido sin ciclos corre en tiempo $\theta(n^2)$. Por tanto, podemos establecer el siguiente resultado:

Teorema 5.3.5 *El problema (PDMCI) puede resolverse en tiempo $O(n^2)$.*

Una vez resuelto el problema considerado como previo $\mathcal{P}4.1$, abordamos la resolución del problema principal $\mathcal{P}4$, cuya formulación es

$$\min_C \{ \max_{1 \leq i \leq n} \{ d(p_i, C) / l(C) \leq l_0 \} \} \quad (\mathcal{P}4).$$

Un método usual para los problemas de tipo minimax aparecidos en esta memoria ha sido la consideración de los posibles valores de la excentricidad que corresponden a una solución, llamados *valores candidatos*. En nuestro caso, la solución al problema, digamos $e(C^*)$ (excentricidad de la cadena óptima), sólo puede tomar un número finito de valores. De hecho, existen dos puntos $p_i, p_j, i < j$, de tal manera que $e(C^*) = \max_{i < k < j} \{ d(p_k, \overline{p_i p_j}) \}$. Por tanto, pasaremos a la construcción del conjunto de valores candidatos.

En primer lugar, se consideran los conjuntos de puntos consecutivos $S_{ij} = \{p_i, p_{i+1}, \dots, p_j\}, i < j$. Para cada conjunto S_{ij} definimos $e(S_{ij}) = e_{ij} = \max_{i < k < j} \{ d(p_k, \overline{p_i p_j}) \}$. Así, tendremos un conjunto de valores $E = \{e_{12}, e_{13}, \dots, e_{1n}, e_{23}, \dots, e_{2n}, \dots, e_{n-1n}\}$ que llamaremos conjunto candidato (obsérvese que $e(C^*) \in E$). Puesto que tenemos $O(n^2)$ conjuntos S_{ij} con $i < j$, el conjunto de candidatos E contiene $O(n^2)$ elementos.

Tras una ordenación sobre el conjunto E obtenemos una lista ordenada de valores candidatos $e_1 < e_2 < \dots < e_p$ con p de orden n^2 sobre la que vamos a aplicar búsqueda binaria. Desde luego, un valor e_j de la lista ordenada corresponde a una solución si existe una poligonal de longitud no superior a l_0 (factible) cuya máxima distancia a los puntos es e_j y no existe otra poligonal factible con máxima distancia inferior a e_j .

Para cada valor e_j , $j = 1, \dots, p$ de la lista ordenada anterior podemos resolver el problema previo

$$\text{mín } l(C) \quad \text{s.a. : } d(p_i, C) \leq e_j$$

y obtener en tiempo $O(n^2)$ el valor de longitud mínima l_j correspondiente al candidato e_j . Puesto que para $e_r < e_s$ se obtiene $l_r \geq l_s$, resulta una lista de pares

$$(e_1, l_1), (e_2, l_2), \dots, (e_p, l_p)$$

de forma que $e_1 < e_2 < \dots < e_p$ y $l_1 \geq l_2 \geq \dots \geq l_p$.

Finalmente, aplicando búsqueda binaria sobre la lista ordenada $e_1 < e_2 < \dots < e_p$, la solución vendrá dada por el primer valor candidato e_j cuyo correspondiente valor l_j cumpla la restricción ($l_j \leq l_0$).

La complejidad del método descrito puede descomponerse en dos operaciones: construcción del conjunto de candidatos y búsqueda binaria sobre la lista ordenada.

Lema 5.3.5 *Dados $a = p_0, p_1, \dots, p_n, p_{n+1} = b$, podemos construir el conjunto de candidatos E en tiempo $O(n^2 \log(n))$ y espacio $O(n^2)$.*

Demostración: Puesto que hay $O(n^2)$ candidatos, centrémonos en el cálculo de cada uno de ellos. El cálculo de $e_{ij} = e(S_{ij}) = \max_{i < k < j} \{d(p_k, \overline{p_i p_j})\}$ puede realizarse en tiempo $O(n)$. Sin embargo, haciendo uso del cierre convexo de S_{ij} , $CC(S_{ij})$, podemos encontrar e_{ij} de forma más eficiente como se describe a continuación. Primero, el cálculo de $CC(S_{ij})$ puede realizarse de forma incremental usando el cierre del conjunto anterior $CC(S_{ij-1})$ en tiempo $O(\log(n))$ [74]. Por tanto, podemos tomar como preprocesado la obtención de todos los cierres $CC(S_{ij})$, $i < j$, en tiempo $O(n \log(n))$.

Si conocemos el cierre convexo $CC(S_{ij})$, podemos calcular el valor candidato $e_{ij} = \max_{i < k < j} \{d(p_k, \overline{p_i p_j})\}$ en tiempo $O(\log(n))$ (Figura 5.8) puesto que la función distancia de los vértices al segmento $\overline{p_i p_j}$ es unimodal en las dos subcadenas de $CC(S_{ij})$ determinadas por dicho segmento (véase el Lema 5.3.2).

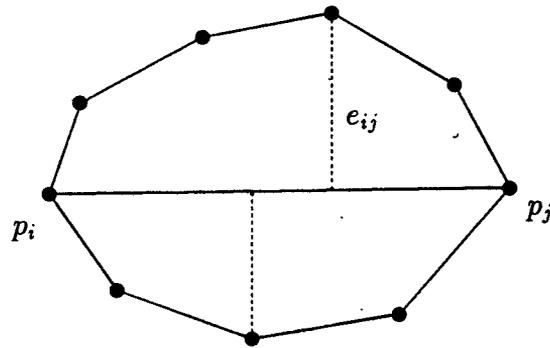


Figura 5.8: Obtención de e_{ij} en tiempo $O(\log(n))$

Como consecuencia, el conjunto

$$E = \{e_{12}, e_{13}, \dots, e_{1n}, e_{23}, \dots, e_{2n}, \dots, e_{n-1n}\}$$

puede calcularse en tiempo $O(n^2 \log(n))$ y espacio $O(n^2)$. \square

La segunda parte del método es la búsqueda binaria sobre la lista $e_1 < e_2 < \dots < e_p$ donde p es del orden de n^2 . Puesto que en el proceso de búsqueda visitaremos $O(\log(n))$ candidatos, para cada uno de los cuales debemos resolver el problema previo (P4.1) correspondiente en tiempo $O(n^2)$, se requiere en total tiempo $O(n^2 \log(n))$. Así pues, ya estamos en condiciones de enunciar el teorema de resolución:

Teorema 5.3.6 *El problema minimax para poligonales discretas con restricción sobre la longitud puede resolverse en tiempo $O(n^2 \log(n))$ y espacio $O(n^2)$ cuando el conjunto de puntos de la entrada no contiene degeneraciones.*

Para el caso degenerado, puede usarse el mismo procedimiento pero se requiere resolver el problema previo (PDMCI) para una nube degenerada. Dicho problema posee un interesante enunciado “geométrico” (Figura 5.9):

Dados dos puntos a y b en el plano y un conjunto de segmentos verticales situados entre a y b (puede haber varios segmentos sobre la misma vertical),

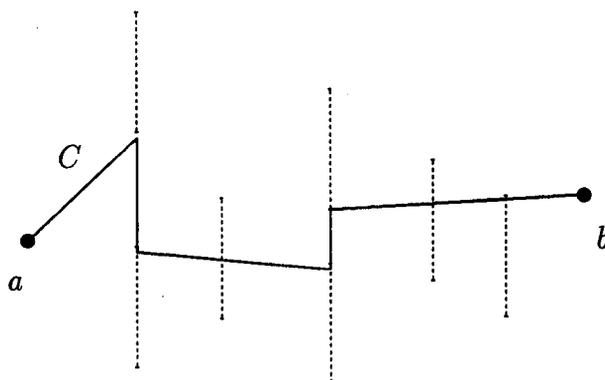


Figura 5.9: Interpretación geométrica de un problema

encontrar una poligonal de mínima longitud que conecte a y b e intersecte a todos los segmentos

Para la resolución de dicho problema puede adaptarse el algoritmo de programación dinámica propuesto para la misma cuestión con poligonales rectilíneas. En aquella ocasión, el método se basaba en una propiedad geométrica crucial: existe un camino de cobertura indirecta mínimo que cubre a los puntos situados en cada vertical entrando (o saliendo) por el punto posible de mayor abcisa y saliendo (o entrando) por el punto permitido de menor abcisa de dicha vertical. Puesta que dicha propiedad se mantiene para el caso de caminos x-monótonos genéricos, se puede diseñar un algoritmo lineal semejante a (RMCI) una vez que se tengan ordenados los puntos lexicográficamente. A pesar de que se reduce la complejidad del problema previo, y como consecuencia, la búsqueda binaria se realiza en tiempo $O(n \log^2(n))$, el cálculo de los valores candidatos $\{e_1, e_2, \dots, e_p\}$ sigue tomando tiempo $O(n^2 \log(n))$.

5.4 Poligonales libres

En los problemas de esta sección aparecen las poligonales x-monótonas con codos libres, o sea, pueden ser puntos cualesquiera del plano. Nosotros resolveremos el problema minimax para poligonales con un solo codo en tiempo

$O(n^2)$. Para poligonales con k codos tomamos como referencia el algoritmo de tiempo $O(n^2 \log(n))$ de Hakimi y Schmeichel [40] válido para un conjunto de puntos sin degeneraciones. Para un conjunto de puntos con degeneraciones el problema está abierto.

5.4.1 Poligonales libres de un codo

Sea el conjunto $P = \{p_1, p_2, \dots, p_n\}$ ordenados lexicográficamente, situados entre a y b , pudiendo contener degeneraciones. Se trata de resolver el problema

$$\min_C \{ \max_{p_i} \{ d(p_i, C) \} \} \quad \text{s.a. : } |C| = 1 \quad \text{y} \quad c_1 \in \mathbb{R}^2 \quad (\mathcal{P}5).$$

La idea principal de la resolución que proponemos es la consideración de un conjunto de problemas previos, cuyas soluciones determinan la solución de $\mathcal{P}5$.

Consideremos las abcisas no repetidas de los puntos de P , denotadas por a_1, a_2, \dots, a_m (supongamos están ordenadas). Vamos a definir con B_j la banda vertical determinada por a_j y a_{j+1} , $B_j = B(a_j, a_{j+1}) = \{r \in \mathbb{R}^2 / a_j \leq x(r) \leq a_{j+1}\}$. Llamamos $\mathcal{P}5j$ al problema de encontrar una poligonal de a a b , con un codo en B_j , que minimice la máxima distancia a los puntos de P :

$$\min \{ \max \{ d(p_i, C) \} \} \quad \text{s.a. : } C = \{a, c_1, b\}, c_1 \in B_j \quad (\mathcal{P}5j).$$

La observación crucial es que si resolvemos este problema para cada banda B_j , podemos determinar una solución de $\mathcal{P}5$ sin más que tomar la mejor de todas las soluciones parciales obtenidas para cada problema $\mathcal{P}5j$ (ver Figura 5.10).

Lema 5.4.1 *Existe una solución $C^* = \{a, c_1, b\}$ para $\mathcal{P}5$ cumpliendo que:*

$$\max \{ d(p_i, \overline{ac_1}) / x(p_i) \leq x(c_1) \} = \max \{ d(p_i, \overline{c_1b}) / x(p_i) \geq x(c_1) \}$$

Demostración: Supongamos que, por ejemplo:

$$\max \{ d(p_i, \overline{ac_1}) / x(p_i) \leq x(c_1) \} < \max \{ d(p_i, \overline{c_1b}) / x(p_i) \geq x(c_1) \}$$

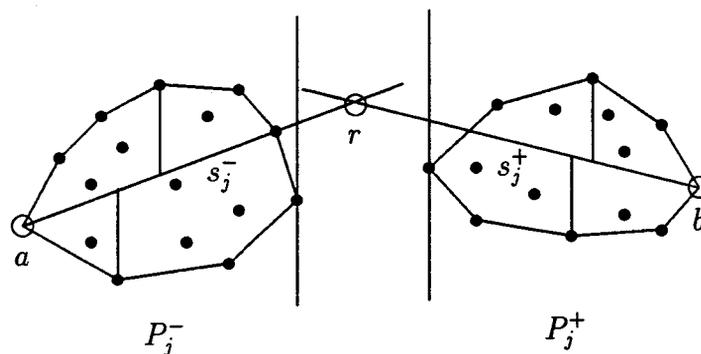


Figura 5.10: Resolución del problema en cada banda.

Entonces moviendo c_1 infinitesimalmente sobre la recta ac_1 en alguno de sus dos sentidos, la máxima distancia $d(p_i, \overline{c_1b})$ disminuye. En algún momento ambos mínimos se igualan. \square

El lema siguiente muestra cómo obtener, para cada banda, la solución en ella. La solución final será la mínima de las soluciones obtenidas en cada banda.

Lema 5.4.2 *Se puede encontrar una poligonal óptima para el problema $\mathcal{P}5j$ en las condiciones del lema 5.4.1 en tiempo $O(n)$ con preprocesamiento $O(n \log(n))$.*

Demostración: Para una banda $B_j = [a_j, a_{j+1}]$ definimos: $\mathcal{F}_j^- = \{p_i : x(p_i) \leq a_j\}$ y $\mathcal{F}_j^+ = \{p_i : x(p_i) \geq a_{j+1}\}$, así como sus cierres convexos, $P_j^- = CC(\mathcal{F}_j^-)$ y $P_j^+ = CC(\mathcal{F}_j^+)$.

Sean s_j^-, s_j^+ rectas que, pasando por a y b , respectivamente, minimizan la máxima distancia a los puntos de $\mathcal{F}_j^-, \mathcal{F}_j^+$, respectivamente. Es claro que, s_j^-, s_j^+ deben encontrarse a máxima distancia de al menos dos puntos (que están en semiplanos diferentes de los determinados por s_j^-, s_j^+), (véase Figura 5.10), puesto que, en caso contrario, se puede encontrar otras rectas que mejoren la máxima distancia efectuando un deslizamiento infinitesimal.

Sea r el punto intersección de las dos rectas, $r = s_j^- \cap s_j^+$. Si $r \in B_j$ hemos terminado; por tanto, supongamos $x(r) < a_j$ sin pérdida de generalidad.

Por otra parte, sean e^- y e^+ los valores que dan la máxima distancia de los puntos de $\mathcal{F}_j^-, \mathcal{F}_j^+$ a s_j^-, s_j^+ , respectivamente. Supongamos, sin pérdida de generalidad, que $e = \max\{e^-, e^+\} = e^+$. En ese caso, podemos modificar convenientemente la recta s_j^- de forma que se verifique:

$$e^- = \max_{x(p_i) \leq a_j} d(p_i, s_j^-) = \max_{x(p_i) \geq a_{j+1}} d(p_i, s_j^+) = e^+,$$

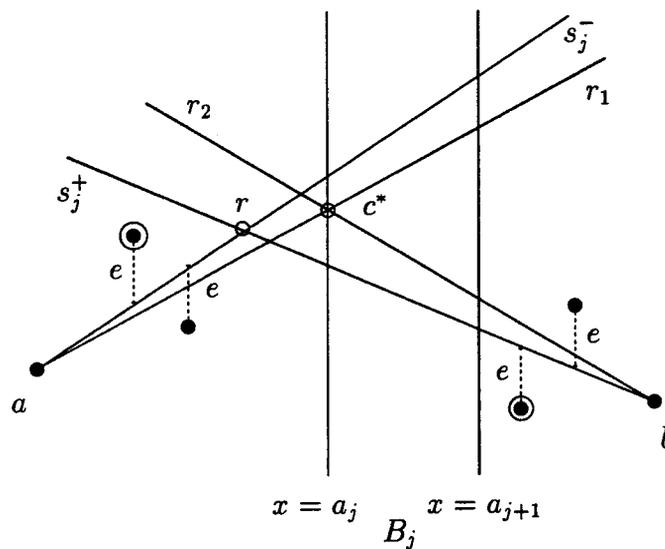
y el valor de e no se modifica. Si el nuevo punto de intersección de las rectas, r , se encuentra en la banda B_j hemos terminado, por lo que suponemos, sin pérdida de generalidad, que el nuevo punto de corte cumple igualmente $x(r) < a_j$.

En esas condiciones sean y_r^-, y_r^+ las ordenadas de los puntos de intersección de s_j^-, s_j^+ con la recta vertical $x = a_j$ y c^* el punto de abcisa a_j y ordenada entre y_j^- e y_j^+ cumpliendo

$$\max\{d(p_i, \overline{ac^*}) / x(p_i) \leq a_j\} = \max\{d(p_i, \overline{c^*b}) / x(p_i) \geq a_{j+1}\}$$

Es fácil ver que c^* es único (se obtiene resolviendo una ecuación) y que da una solución para la banda B_j en las condiciones del lema 5.4.1, puesto que tomando como codo cualquier otro punto de la banda no se mejora la máxima distancia a ambos lados de B_j . En efecto, en la Figura 5.11 se puede observar que si el codo c^* se mueve sobre la recta $x = a_j$ el valor de la excentricidad total no se mejora. Asimismo, en el interior de la banda B_j , podemos hacer un análisis según el codo esté situado en algunas de las tres regiones determinadas por las semirrectas r_1 y r_2 (que se cortan en el punto c^*) y se observa que la excentricidad total se empeora cuando se sitúa el codo c^* en cualquiera de dichas regiones (obsérvese que los puntos señalados con doble círculo en la Figura 5.11 determinan la excentricidad de las rectas r_1 y r_2).

Las rectas s_j^-, s_j^+ se obtienen en tiempo $O(n)$ si tenemos calculados P_j^-, P_j^+ adaptando el método que Lee y Wu [53] usan para el problema restringido de recta centro con distancia euclídea. Por otra parte, el cálculo de c^* es constante. \square

Figura 5.11: Obtención del codo c^* .

Teorema 5.4.1 *Se puede calcular una solución de $\mathcal{P5}$ en tiempo $O(n^2)$ y espacio lineal.*

Demostración: En virtud del lema anterior, obtenemos la solución para cada banda B_j en tiempo $O(n)$ con preprocesamiento $O(n \log(n))$ y seleccionamos la mínima de todas ellas. Puesto que hay $n - 1$ bandas, la complejidad es $O(n^2)$. Queda por precisar un detalle: en el cálculo de la solución en B_j se suponía que el cierre convexo de \mathcal{F}_j^- y \mathcal{F}_j^+ estaba ya dado de antemano. Para poder aplicar el lema necesitamos construir los cerrajes convexos de \mathcal{F}_j^- y \mathcal{F}_j^+ . Esto puede hacerse adaptando el algoritmo de barrido del cierre convexo [74]. Las rectas $x = a_j$ y $x = a_{j+1}$ son las rectas de barrido. Cuando la recta $x = a_j$ avanza una posición, actualiza el cierre convexo a su izquierda (insertando el punto de P con coordenada a_{j+1}); cuando la recta $x = a_{j+1}$ avanza, actualiza el cierre convexo a su derecha borrando el punto de P con coordenada a_{j+1} . La complejidad de este paso es $O(n \log(n))$ por la ordenación del barrido más el coste de las actualizaciones, que es lineal. \square

5.5 Conclusiones y problemas abiertos

Han sido resueltos en el presente capítulo varios problemas concernientes a la búsqueda de servicios poligonales genéricos bajo en criterio minimax, tanto para el caso de poligonales discretas como para poligonales libres. Sin embargo, han quedado otros sin resolver que surgen, bien de la necesidad de mejorar computacionalmente los algoritmos propuestos (incluso buscar cotas inferiores), bien de generalizaciones de los problemas tratados, o bien de cambiar el objetivo minimax por otros tópicos de Localización de Servicios como maxmin, minsum, etc.

Principalmente, los resultados que se recogen en este capítulo se refieren a poligonales discretas. Dos factores fundamentales a tener en cuenta en el diseño de un camino poligonal son el número de codos y la longitud. Se han propuesto, en este sentido, algoritmos eficientes para poligonales discretas de uno o k codos, mejorando la complejidad de los algoritmos existentes en la actualidad ($O(n^2 \log(n))$ en ambos casos) [40]. Además, se ha planteado la posibilidad de que el camino buscado empiece y termine en ciertas regiones convexas dadas en el *input* como polígonos convexos ordenados. Se ha resuelto el problema para poligonales de un solo codo, quedando como problema abierto la resolución para poligonales discretas de k codos.

Con respecto a la restricción en la longitud, se ha resuelto un interesante problema, tanto por el resultado en sí, como por el método de resolución usado. En dicho método se ha resuelto asimismo un bonito problema geométrico, pudiéndose considerar de localización-ruta, donde se busca la poligonal más corta con cobertura indirecta sobre los puntos de la nube. La resolución de dicho problema, denominado previo, se basa en la consideración de un grafo dirigido y la búsqueda del camino más corto entre dos puntos de dicho grafo. Hay que destacar, sin embargo, que el método se ha desarrollado para un conjunto de puntos sin degeneraciones. Para el caso degenerado, puede pensarse en usar el mismo procedimiento, aunque para ello se requiere resolver el correspondiente problema previo con degeneraciones.

En el fondo de la cuestión, se observa cómo la propiedad de que los co-

dos sean puntos de la entrada, determina un conjunto discreto de poligonales posibles sobre el que tenemos que trabajar. Sin embargo, cuando los codos pueden ser puntos cualesquiera situados entre las verticales determinadas por los puntos origen y final del trayecto, podemos decir que el problema los problemas son de naturaleza “continua”, que son en general, de de mayor dificultad intrínseca. Una técnica adecuada es la búsqueda de propiedades geométricas de una solución. Así, se ha resuelto el problema minimax de poligonales libres con un codo en tiempo $O(n^2)$, tanto si la nube es degenerada como si no lo es. Para el caso de k codos se puede adaptar el algoritmo de Hakimi y Schmeichel [40] ($O(n^2 \log(n))$) para el caso de poligonales con puntos origen y final. Sin embargo, dicho algoritmo no es aplicable para puntos con degeneraciones, por lo que está abierto, tanto la resolución de este caso, como la mejora de la complejidad del método referenciado. La mayor parte de los algoritmos aquí propuestos han sido presentados recientemente en un congreso español [19]. La Tabla 5.5 refleja los resultados computacionales obtenidos para los problemas aquí planteados.

Cadenas poligonales	mín máx $d_v(p_i, C)$
$ C = 1$ discreto	$O(n \log(n))$
$ C = k$ discreto	$O(n \log(n))$
$l(C) = l_0$ discreto	$O(n^2 \log(n))$
$ C = 1$ libre	$O(n^2)$
$ C = k$	$O(n^2 \log(n))^*$

Tabla 5.5. Problemas de localización minimax para poligonales en el plano. * Hakimi y Schmeichel [40].

Finalmente, consideramos oportuno plantear en este capítulo, y más aún en el terreno propio de Localización de Servicios, los problemas que surgen de manera natural por su proximidad en la formulación con los aquí resueltos. Nos referimos a la búsqueda de poligonales bajo el criterio maxmin que ha sido resueltos para el caso euclídeo [27]. No cabe duda de que, modificando los ingredientes del problema, aparecen una serie de cuestiones interesantes, tanto desde la óptica de Localización de Servicios como de Geometría Computacional, tanto desde el punto de vista teórico como del punto de vista práctico.

A modo de conclusión final, podemos decir que se han resuelto un conjunto de problemas y que, en realidad, se han sentado las bases para el desarrollo e investigación de un apartado de matemática aplicada situado entre dos campos de investigación que poseen un evidente solapamiento.

Bibliografía

- [1] AGARWAL, P.K., EFRAT, A., SHARIR, M. Y TOLEDO, S., Computing a Segment Center for Planar Point Set. *Journal of Algorithms*, 15, 1993, 314–323.
- [2] AGARWAL, P.K. Y SHARIR, M., Planar Geometric Location Problems. *Algorithmica*, 11, 1994, 185–195.
- [3] AGARWAL, P.K., SHARIR, M. Y TOLEDO, S. Applications of parametrics searching in geometric optimization. *Journal of Algorithms*, 17, 1994, 292–318.
- [4] ASANO, T., GUIBAS, L., HERSHBERGER J. E IMAI H., Visibility of disjoint polygons. *Algorithmica*, 1, 1986, 49–63.
- [5] BELLMAN R., *Dynamic Programming*, Princeton University Press, 1957.
- [6] BENTLEY, J.L. Y OTTMANN, T.A., Algorithms for reporting and counting geometric intersections. *IEEE Trans. Comput.*, 8, 1979, 300–307.
- [7] BHATTACHARYA B.K. Y TOUSSAINT G.T., On geometric algorithms that use the furthest-point Voronoi-diagram, *Computational Geometry*, G. T. Toussaint, editor (North Holland, 1985), 43–61.
- [8] BOSE P. Y TOUSSAINT G.T., Computing the Constrained Euclidean, Geodesic and Link Centre of a Simple Polygon with Applications. Aceptado en *Studies in Locational Analysis*, 1998.
- [9] BURT, P.J., Fast filter transforms for image processing. *Computer Graphics and Image Processing.*, 16, 1979, 20–51.

- [10] CHAN, W. S. Y CHIN, F., Approximation of Polygonal Curves with minimum number of line segments or minimum error. *International Journal of Computational Geometry & Applications*, 6, No. 1 (1996) 59–77.
- [11] CHIN, F., CHOI, A. Y LUO, Y., Optimal Generating Kernel for Image Pyramids by Piecewise Fitting. *IEEE Trans. Pattern Anal. Machine Intell.*, 14, No. 12, 1992, 1190–1198.
- [12] CHAZELLE, B., GUIBAS, L. J. Y LEE, D. T., The power of geometric duality. *BIT*, 25, 1985, 76–90.
- [13] CLARKSON, K. L., KAPOOR, S. Y VAIDYA, P. M., Rectilinear shortest paths through polygonal obstacles in $O(n \log^{\frac{3}{2}}(n))$ time. En *Proceedings of the 3rd Annual ACM Symposium on Computational Geometry* (Waterloo, Ontario, Canada). ACM, New York, 1987, 251–257.
- [14] CORMEN, T., LEISERSON, C. Y RIVEST, R., *Introduction to Algorithms*, McGraw-Hill, 1990.
- [15] COURANT, R. Y ROBBINS, H., “What is Mathematics?,” Oxford University Press, New York, 1941.
- [16] DEREZENDE, P. J., LEE, D.T. Y WU, Y.F., *Rectilinear shortest paths with rectangular barriers*, *Discrete and Computational Geometry*, 4, 1989, 41–53.
- [17] DÍAZ, J.M. Y DÍAZ, P., La semirrecta centro con métrica $l_{\infty}(l_1)$. En las *Actas de los VII Encuentros de Geometría Computacional*, Madrid, 1997, 223–237.
- [18] DÍAZ, J.M. Y DÍAZ, P., The Segment Center Problem with $l_{\infty}(l_1)$ Metrics. Aceptado en *Studies in Locational Analysis*, 1998.
- [19] DÍAZ, J.M., GÓMEZ F. Y HURTADO, F., Localización de servicios poligonales con restricción en el número de codos. En las *Actas de los VII Encuentros de Geometría Computacional*, Madrid, 1997, 239–250.

- [20] DÍAZ, J.M. Y MESA, J.A., Localización óptima de una trayectoria rectilínea centro. En las Actas de los *VI Encuentros de Geometría Computacional*, Barcelona, 1995, 140–147.
- [21] DÍAZ J.M. Y MESA J.A., Location of Rectilinear Center Trajectories. *Optimization Days*, Montreal, 1996.
- [22] DÍAZ J.M. Y MESA J.A., Minimax Rectilinear Polygons with Length Constraint. En *Proceedings Journées Franco-Espagnoles de Géométrie Algorithmique*. Barcelona, 1997, 30–34.
- [23] DREZNER, Z., *Facility Location: A Survey of Applications and Methods*, Editor Zvi Drezner, Springer, 1995.
- [24] DREZNER, Z., GAVISH, B., ϵ -approximations for Multidimensional Weighted Location Problem. *Operations Research*, 33, 1985, 772–783.
- [25] DREZNER, Z., STEINER, G. Y WESOŁOWSKY, G.O., One Facility Location with Rectilinear Tour Distances. *Naval Research Logistics Quarterly*, 32, 1985, 391–405.
- [26] DREZNER, Z. Y WESOŁOWSKY, G.O., A New Method for the Multi-facility Minimax Location Problem. *Journal Operational Research Society*, 29, 1978, 1095–1101.
- [27] DREZNER, Z. Y WESOŁOWSKY, G.O., Location of an obnoxious route. *Journal Operational Research Society*, 40, 1989, 1011–1018.
- [28] DYER, M.E., Linear time algorithms for two- and three-variable linear programs *SIAM Journal on Computing*, 13, 1984, 31–45.
- [29] EFRAT A. Y SHARIR M., A Near-Linear Algorithm for the Planar Segment-Center Problem. *Discrete and Computational Geometry*, 16, 1996, 239–257.
- [30] ELZINGA J. Y HEARN D.W., Geometrical solutions to some minimax location problems. *Trasp. Sci.*, 6, 1972, 379–394.

-
- [31] ELZINGA J. Y HEARN D.W., The Minimum Covering Sphere Problem. *Management Science*, 19, 1972, 96–104.
- [32] FORTUNE, S., Progress in Computational Geometry. *Antony Rowe Ltd., bumper's Farm*, Chippenham, England, 1993.
- [33] FRANCIS, R.L., HAMACHER, H.M., LEE, C.L. Y YERALAN, S., Finding Placement Sequences and Bin Locations for Cartesian Robots. *IIE Transactions*, 26, 1994, 47–59.
- [34] FRANCIS, R.L. Y WHITE, J.A., *Facility Layout and Location*. Prentice Hall, Englewood Cliffs, NJ, 1974.
- [35] GAREY, M.R. Y JOHNSON, D.S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freedman and Company, NJ, 1979.
- [36] GOODMAN, F. Y O'ROURKE J., *Handbook of Discrete and Computational Geometry*. Editores Goodman, F. y O'Rourke J, 1997.
- [37] GÓMEZ, F., *Proyecciones de calidad*. Tesis doctoral, Madrid, 1996.
- [38] GÓMEZ, F., RAMASWAMI, S. Y TOUSSAINT, G.T. *On removing non-degeneracy assumptions in Computational Geometry*. En Proceedings ESA'96, 1996.
- [39] GUIBAS, L.J., HERSHBERGER, J.E., MITCHELL, J.S.B. Y SNOEYINK, J.S., Approximating Polygons and Subdivisions with Minimum-Link Paths. *International Journal of Computational Geometry & Applications*, 3, 1993, 383–415.
- [40] HAKIMI, S.L. Y SCHMEICHEL, E.F., Fitting polygonal functions to a set of points in the plane. *Graphical models y image processing*, 53, N. 2, 1991, 132–136.
- [41] HEARN, W.H. Y VIJAY, J., Efficient algorithms for the weighted minimum circle problem. *Operations Research*, 12, 30, 1982, 777–795.

-
- [42] HURTADO, F., SACRISTÁN, V. Y TOUSSAINT, G.T., Constrained minimax facility location. *Tech. Report MA2-IR-96-0010*, Universidad Politécnica de Cataluña, Octubre de 1996.
- [43] HOULE, M.E. Y TOUSSAINT, G.T., Computing the width of a set, *ACM Symp. on Computational Geometry, Baltimore, Maryland*, 1985, 1-7.
- [44] HOULE, M.E., IMAI H., IMAI K., ROBERT J.-M. Y YAMAMOTO P. Orthogonal weighted linear l_1 and l_∞ approximation and applications, *Discrete Applied Mathematics*, 43, 1993, 217-232.
- [45] IMAI H. Y IRI M., An optimal Algorithm for Approximating a Piecewise Linear Function. *Journal of Information Processing*, 9, 3, 1986, 159-162.
- [46] IMAI H. Y IRI M., Polygonal Approximations of a Curve-Formulations and Algorithms. *Computational Morphology*, G. T. Toussaint, editor (North Holland, 1988).
- [47] IMAI, H., LEE, D. T. Y YANG, CH., 1-Segment Center Problem. *ORSA Journal on Computing*, 4, 1992, 426-434.
- [48] KUROZUMI Y. Y DAVIS, W.A., Polygonal Approximation by the Minimax Method. *Computer Graphics and Image Processing*, 19, 1982, 248-264.
- [49] LEE D. T. Two-Dimensional Voronoi Diagrams in L_p -Metric. *Journal of the Association for Computing Machinery*, 27, 1980, 604-618.
- [50] LEE D. T. Y LEE V. B. Out-of-roundness problem revisited. *IEEE Transactions on Pattern Analysis and Machinery Intelligence*, 13, 1991, 217-223.
- [51] LEE D. T. Y PREPARATA F. P. Euclidean Shortest paths in the presence of rectilinear barriers. *Networks*, 14, 1984, 393-410.
- [52] LEE, D.T. Y WONG, C.K. Voronoi Diagrams in $L_1(L_\infty)$ Metrics with 2-Dimensional Storage Applications. *SIAM J. Comput.*, 9, 1980, 200-211.

- [53] LEE, D.T. Y WU, Y.F. Geometric Complexity of Some Location Problems. *Algorithmica*, 1, 1986, 193–211.
- [54] LOVE, R.F., MORRIS J.G Y WESOLOWSKY, G.O., *Facilities Location* (North-Holland, 1988).
- [55] LOVE, R.F., WESOLOWSKY, G.O. Y KRAEMER, A Multifacility Minimax Location Method for Euclidean Distances. *International Journal of Production Research* 11, 1988, 32–40.
- [56] KORNEENKO, N. Y MARTINI, H., Hyperplane Approximation and Related Topics. *New Trends in Discrete and Computational Geometry* (Ed. János Pach). (Springer-Verlag, 1993).
- [57] MARTOS, B., *Nonlinear Programming Theory and Methods*. North-Holland. 1975.
- [58] MEGIDDO, N., The Weighted Euclidean 1-center Problem. *Mathematics of Operations Research*, 8, 1983, 498–504.
- [59] MEGIDDO, N., Applying parallel computation algorithms in the desing of serial algorithms. *J. Assoc. Comput. Mach.*, 30, 1983, 852–865.
- [60] MEGIDDO, N., Linear-time Algorithms for Linear Programming in \mathbb{R}^3 and Related Problems. *SIAM Journal on Computing*, 12, 1983, 759–776.
- [61] MEGIDDO, N. Y TAMIR, A., On the Complexity of Locating Linear Facilities in the Plane. *Opns. Res. Letters*, 1, 1982, 194–197.
- [62] MELKMAN, A. Y O'ROURKE, J., On Polygonal Chain Approximation. *Computational Morphology*, G. T. Toussaint, editor (North Holland, 1988).
- [63] MITCHELL, J.S.B. Y PAPADIMITRIOU, C.H., Planning shortest paths. *SIAM Conference on Geometric Modeling and Robotics* (Albany, N.Y.), SIAM, New York, 1985, 1–21.
- [64] MITCHELL, J.S.B., SNOYEINK, J., SUNDARAM, G. Y ZHU, B. Generating random x-monotone polygons with given vertices. *Proc. Sixth Canadian Conference on Computational Geometry*, 1994, 189–194.

- [65] MORRIS, J.G. Y NORBACK, J.P., Linear Facility Location - Solving Extensions on the Basic Problems. *European Journal of Operational Research*, 12, 1983, 90-94.
- [66] NIEVERGELT, J. Y PREPARATA, F., Plane-sweep algorithms for intersecting geometric figures. *Proc. Communications of ACM*, 25, 1982, 739-747.
- [67] OKABE, A., BOOTS, B Y SUGIHARA, K., *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, John Wiley & Sons, 1992.
- [68] OKABE, A. Y SUZUKI, A., Locational optimization problems solved through Voronoi diagrams. *European Journal of Operational Research*, 98, 1997, 445-456.
- [69] O'ROURKE J., An on-line algorithm for fitting straight lines between data ranges. *Comm. ACM*, 24, 1981, 574-578.
- [70] O'ROURKE J., *Computational Geometry in C*, Cambridge University Press, 1994.
- [71] PAVLIDIS T., Algorithms for shape analysis of contours and waveforms. *IEEE Trans. Pattern Anal. Mach. Intell.*, Pami-2, 1980, 301-312.
- [72] PELEGRIN B. Y CÁNOVAS L., Primal and Dual Algorithms for the Minimum Covering l_p -hypersphere Problem. *Studies in Locational Analysis*, 7, 1994, 153-169.
- [73] PLASTRIA F., Fully Geometric Solutions to Some Planar Minimax Location Problem. *Studies in Locational Analysis*, 7, 1994, 171-183.
- [74] PREPARATA, F.P. Y SHAMOS, M.I., *Computational Geometry: An Introduction*. Springer-Verlag. New York. 1985.
- [75] RADEMACHER, H.O. Y TOEPLITZ, O., "The Enjoyment of Mathematics." Princeton University Press, Princeton, NJ, 1957.
- [76] RATLIFF, H.D. Y ROSENTHAL, A.S., Order-Picking in a Rectangular Warehouse: A Solvable Case of the Traveling Salesman Problem. *Opns. Res.*, 31, 1983, 507-521.

- [77] RICE, J., *The Approximation of Functions*, Vol. 4, (Addison-Wesley, 1994).
- [78] ROBERT, J.M. Y TOUSSAINT, G.T., Linear approximation of simple objects. *Computational Geometry*, 4, 1994, 27–52.
- [79] SHAMOS, M.I., *Computational Geometry*. Ph. D. Thes., Dept. Computer Sci., Yale Univ., New Haven, 1978.
- [80] SHAMOS, M.I. Y HOEY, D., Closest-point Problems. *Proc. 16th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Los Angeles*, 1975, 151–162.
- [81] SHARIR, M. Y SHORR A., On shortest paths in polyedral spaces. En *Proc. 16th Symposium on the Theory of Computing, Washington, D.C.*, ACM, New York, 1984, 144–153.
- [82] SCHÖBEL, A., Locating Least-Distant Lines with Block Norms. *Studies in Locational Analysis*, 10, 1996, 139–150.
- [83] SHÖMER, E., SELLEN, J., TEICHMANN, M. Y YAP, CH. Efficient Algorithms for the Smallest Enclosing Cylinders Problem. *Proc. 8th Canadian conference on Computational Geometry*, 1996, 264–269.
- [84] SYLVESTER, J.J., A Question in the Geometry of Situation. *Quarterly Journal of Pure and Applied Mathematics*, 1, 1857, 79.
- [85] TOUSSAINT G. T., Solving geometric problems with the rotating calipers. *Proc. of IEEE MELECON*, Athens, Greece, 1983, A10.02-A10.05.
- [86] TOUSSAINT G. T., Computing Largest Empty Circles with Location Constraints. *International Journal of Computer and Information Sciences*, 12, 1983, 347–358.
- [87] TOUSSAINT G. T., On the Complexity of Approximating Polygonal Curves in the Plane. *Proc. IASTED, International Symposium on Robotics and Automation*, Lugano, Switzerland, 1985.

-
- [88] VACCARO H., Alternative Techniques for Modeling Travel Distances. Thesis in Civil Engineering, Massachusetts Institute of Technology, 1974.
- [89] WESOLOWSKY G.O., Rectangular Distance Location Under the Minimax Optimality Criterion. *Transportation Science*, 6, 1972, 103-113.
- [90] WU, Y.F., WIDMAYER, P., SCHALG, M.D.F. Y WONG C.K., Rectilinear shortest paths and minimum spanning trees in presence of rectilinear obstacles. *IEEE Trans. Comput.*, 1987, 321-331.
- [91] YANG, CH., On Rectilinear Paths Among Rectilinear Obstacles. Ph. D. Dissertation, Northwestern University, 1994.