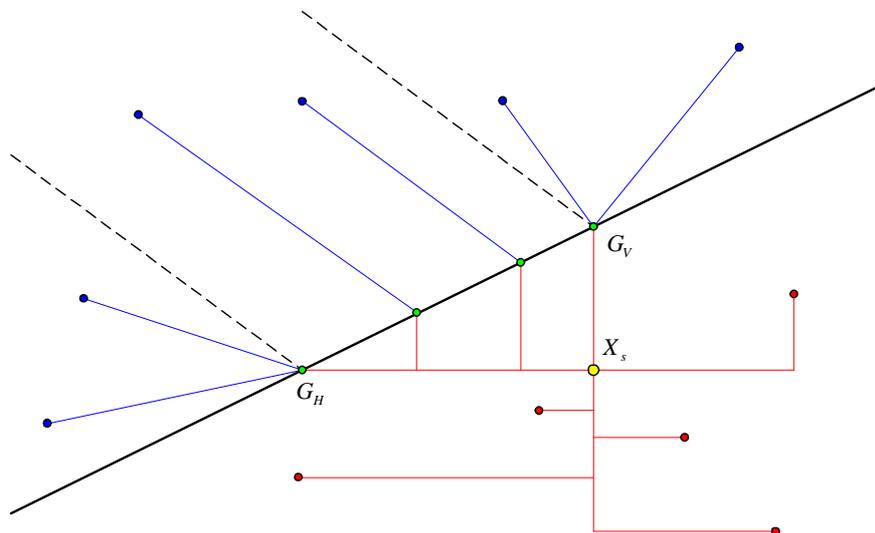




UNIVERSIDAD DE SEVILLA

DEPARTAMENTO DE ECONOMÍA APLICADA I

# LOCALIZACIÓN DE CENTROS DE SERVICIOS EN REGIONES CON MÉTRICAS DISTINTAS



**Luis Franco Martín**

Sevilla, Diciembre 2012





UNIVERSIDAD DE SEVILLA

DEPARTAMENTO DE ECONOMÍA APLICADA I

LOCALIZACIÓN DE CENTROS DE  
SERVICIOS EN REGIONES CON  
MÉTRICAS DISTINTAS

Trabajo presentado por **Luis Franco Martín** para la obtención del grado de doctor,

dirigido por los doctores

**D. Francisco Velasco Morente**

**D. Luis González Abril.**

Sevilla, 2012



## **Agradecimientos**

Sirvan las siguientes líneas como reconocimiento y sincero agradecimiento a todos los que en ellas figuran.

A mis directores de Tesis, Francisco Velasco Morente y Luis González Abril, de los que tanto he aprendido personal y profesionalmente. A pesar de sus múltiples ocupaciones han tenido una paciencia infinita conmigo y han soportado la ingrata tarea de lectura y corrección de mis múltiples borradores.

A mis compañeros docentes del Departamento de Economía Aplicada I que me han apoyado en la realización del trabajo, en especial a mi amiga Puri.

A Lola y Rosa, quienes desde sus cargos de administrativas del Departamento, han sido capaces de simplificar todas las gestiones.

A mi familia y amigos, por su fe inquebrantable en mis posibilidades.

Y muy especialmente a Fina, Luis, Antonio y Julio, a los que he echado mucho de menos durante todo el trabajo.



---

## ÍNDICE DE CONTENIDO

---

Índice General. . . . .	v
Índice de Tablas. . . . .	ix
Índice de Figuras. . . . .	xi

### 1

## El problema de localización simple en dos regiones con distintas métricas

1.1	Introducción. . . . .	1
1.1.1	Modelando con distancias. . . . .	5
1.1.2	Modelando objetivos. . . . .	9
1.2	Breve reseña histórica. . . . .	11
1.3	Localización simple en dos regiones del plano con diferentes normas. . . . .	14
1.4	Justificación y organización del trabajo. . . . .	20

**2**

**Líneas geodésicas entre regiones con diferentes métricas**

2.1	Introducción y resultados previos. . . . .	23
2.2	Puertas de acceso (Gates) y puntos Notables. . . . .	33
2.3	Estudio analítico de las puertas de acceso. . . . .	40
2.3.1	El caso de normas $\ell_1$ y $\ell_q$ . . . . .	45
2.3.2	El caso de normas $\ell_p$ y $\ell_\infty$ . . . . .	52
2.3.3	El portón de un punto situado en la recta. . . . .	60

**3**

**Los portones y la solución del problema de localización con dos regiones**

3.1	La solución del problema para dos puntos. . . . .	63
3.2	La solución del problema con las normas $\ell_p$ y $\ell_q$ . . . . .	68
3.2.1	La solución del problema con las normas $\ell_1$ y $\ell_q$ . . . . .	74
3.2.2	La solución del problema con las normas $\ell_1$ y $\ell_\infty$ . . . . .	83
3.3	Método de obtención en $\Omega_1$ de la solución del problema. . . . .	86

**4**

**Algoritmos de localización en dos regiones con métricas distintas**

4.1	Algoritmo GMFP( $p, q$ ). . . . .	94
4.2	Algoritmos Gate( $p, q$ ). . . . .	98

---

ÍNDICE GENERAL

---

4.3	Algoritmo de búsqueda en la recta. . . . .	100
4.4	Algoritmos de búsqueda en $\Omega_q$ ( $1 < q < \infty$ ). . . . .	101
4.5	Algoritmos de búsqueda en $\Omega_1$ . . . . .	104
4.6	Algoritmos de búsqueda en $\Omega_\infty$ . . . . .	107
4.7	El problema del punto fijo $X = WP_p(X)$ : El algoritmo de Weiszfeld. . . . .	109

**5**

**Resultados computacionales**

5.1	Consideraciones generales. . . . .	119
5.2	Programas y resultados. . . . .	120
5.2.1	Regiones con normas $l_1$ y $l_2$ . . . . .	121
5.2.2	Regiones con normas $l_1$ y $l_q$ . . . . .	123
5.2.3	Regiones con normas $l_1$ y $l_\infty$ . . . . .	136
5.2.4	Regiones con normas $l_p$ y $l_\infty$ . . . . .	138
5.2.5	Regiones con normas $l_p$ y $l_q$ . . . . .	141
5.3	Conclusiones. . . . .	146

**6**

**Conclusiones y trabajos futuros**

6.1	Conclusiones. . . . .	149
6.2	Trabajos futuros. . . . .	153
	Bibliografía. . . . .	155
	Anexos. . . . .	163



---

## ÍNDICE DE TABLAS

---

Tabla I:	Resultados para los puntos de Parlar y Zaferanieh, con normas $\ell_1$ y $\ell_2$ , variando la pendiente $m$ . . . . .	122
Tabla II:	Resultados para conjuntos de puntos aleatorios (pesos unitarios), con normas $\ell_1$ y $\ell_2$ , variando $m$ . . . . .	123
Tabla III:	Resultados para los puntos de Parlar y Zaferanieh, con normas $\ell_1$ y $\ell_q$ , variando $q$ y la pendiente $m$ . . . . .	124
Tabla IV:	Promedio de distancias en conjuntos de puntos aleatorios (pesos unitarios). . . . .	126
Tabla V:	Promedio de tiempos de CPU en conjuntos de puntos aleatorios (pesos unitarios). . . . .	130
Tabla VI:	Comparativa de distancias al aumentar el número máximo de iteraciones. . . . .	134
Tabla VII:	Promedio de distancias y tiempos para $\ell_1$ y $\ell_\infty$ . . . . .	136
Tabla VIII:	Promedio de distancias y tiempos para $\ell_p$ y $\ell_\infty$ . . . . .	138
Tabla IX:	MFP vs Gate(p,q) variando $p$ , $q$ y $m$ . . . . .	142



---

## ÍNDICE DE FIGURAS

---

1.1	Desplazamientos según la norma. . . . .	7
1.2	Representación gráfica de un problema de localización en regiones heterogéneas. . . . .	15
1.3	Punto solución en $\Omega_q$ . . . . .	17
1.4	Línea geodésica entre puntos de regiones con distintas métricas. . . . .	18
2.1	Proyecciones en rectas horizontales y verticales, según $p$ . . . . .	29
2.2	Proyecciones en rectas con pendientes $\pm 1$ . . . . .	29
2.3	Proyección de $P$ en $r$ , según $m \neq 0$ , con $r(P) < 0$ . . . . .	32
2.4	Puerta de acceso entre dos puntos de regiones con normas distintas. . . . .	33
2.5	Puertas de acceso para $L[P, Q]$ vertical y horizontal. . . . .	34
2.6	Posición inadmisibles de $X_s$ respecto de $Q_1$ . . . . .	35
2.7	Posición inadmisibles de $X_s$ respecto de $P_1$ . . . . .	36
2.8	Posición relativa de $X_s$ para $m \geq 0$ . . . . .	36
2.9	Coincidencia de signo de las pendientes. . . . .	37
2.10	Posición inadmisibles de $X_s$ respecto de $P_p$ y $Q_q$ . . . . .	38

---

ÍNDICE DE FIGURAS

---

2.11 Posición relativa de la puerta de acceso, según los puntos notables. . . . .	38
2.12 División de $\Omega_q$ según segmento de la recta donde se encuentra la puerta de acceso. . . . .	39
2.13 Posición inadmisibles de $X_s$ respecto de $P_0$ y $P_1$ , para $p=1$ . . . . .	40
2.14 Portón común para rayos con pendientes $m_p$ y $m_q$ . . . . .	43
2.15 Gráfica de $f_p$ para $k < 0$ . . . . .	44
2.16 Gráfica de $f_p$ para $k > 0$ . . . . .	44
2.17 Portón común para rectas horizontales y verticales con normas $\ell_1$ y $\ell_q$ . . . . .	46
2.18 Regiones iniciales en $\Omega_q$ determinadas por $P \in \Omega_1$ . . . . .	46
2.19 Regiones en $\Omega_q$ definidas por $P \in \Omega_1$ . . . . .	48
2.20 Regiones en $\Omega_1$ definidas por $Q \in \Omega_q$ . . . . .	48
2.21 Regiones en $\Omega_\infty$ definidas por $P \in \Omega_1$ . . . . .	51
2.22 Regiones en $\Omega_\infty$ definidas por $P \in \Omega_p$ . . . . .	53
2.23 Portón para $Q \in \Omega_H \subset \Omega_\infty$ y $P \in \Omega_p$ . . . . .	53
2.24 Portones para $Q \in \Omega_C \subset \Omega_\infty$ y $P \in \Omega_p$ . . . . .	54
2.25 Portón para $Q \in \Omega_V \subset \Omega_\infty$ y $P \in \Omega_p$ . . . . .	55
2.26 Portón para $Q \in \Omega_{V'} \subset \Omega_\infty$ y $P \in \Omega_p$ . . . . .	56
2.27 Portones para $Q \in \Omega_B \subset \Omega_\infty$ y $P \in \Omega_p$ . . . . .	56
2.28 Regiones en $\Omega_\infty$ definidas por $P \in \Omega_p$ , para $m > 1$ . . . . .	57
2.29 Portones para rectas horizontales y verticales. . . . .	57
2.30 Regiones en $\Omega_p$ definidas por $Q \in \Omega_\infty$ . . . . .	58
2.31 Posibles trayectorias entre $P$ y $Q$ , cuando $Q \in r$ . . . . .	60

---

ÍNDICE DE FIGURAS

---

2.32 Portones entre $P$ y $Q$ , y regiones definidas por $Q \in r$ . . . . .	62
3.1 Posición inadmisibles de $F$ , solución del problema (7). . . . .	65
3.2 Representación gráfica de una posible solución y el portón en recta horizontal. . . . .	67
3.3 Regiones definidas en $\Omega_1$ por la mediana de $\mathcal{P}$ y $\mathcal{Q}^*$ . . . . .	76
3.4 Desplazamiento de $WP_1(X_0)$ respecto de $X_0 \in (III)$ . . . . .	76
3.5a Posición de $WP_1(X_0)$ para $X_0 \in (I)$ . . . . .	77
3.5b Posición de $WP_1(M)$ . . . . .	77
3.6 División de $\Omega_q$ en regiones según $X \in \Omega_1$ . . . . .	78
3.7 Puntos de $\Omega_1$ sin derivada. . . . .	79
3.8 Cambio de posición de los portones al cambiar $X \in \Omega_1$ . . . . .	81
3.9 Cambio de los portones para $x \triangleleft x_0$ . . . . .	85
3.10 Posición relativa del punto solución $X_s \in \Omega_1$ . . . . .	87
3.11 Regiones factibles para $X_s \in \Omega_1$ . . . . .	90
4.1 Punto solución en $\Omega_q$ . . . . .	95
4.2 Punto solución $X_r$ . . . . .	99
4.3 Zona roja/azul solución no válida/válida. . . . .	105
4.4 Situación inicial y situación tras giro de $45^\circ$ . . . . .	107
4.5 Situación final tras deshacer giro. . . . .	108
.	



---

## EL PROBLEMA DE LOCALIZACIÓN EN DOS REGIONES CON DISTINTAS MÉTRICAS

---

### 1.1 Introducción

Un problema de localización consiste, esencialmente, en buscar la mejor ubicación de una o varias instalaciones, dentro de una región, que atienda las necesidades de otro conjunto de instalaciones ya existentes, referidas a veces como puntos de demanda, representadas tanto unas como otras por puntos de cierto espacio de trabajo.

El término instalación (*facility*) hay que entenderlo en un sentido muy amplio e incluso de distinta naturaleza en un mismo problema. Podría ser un centro de distribución que atiende a la demanda de una cadena de supermercados, un hospital que atiende a un conjunto de poblaciones, una fábrica que atiende a las fuentes de materias primas y a los centros de distribución, uno o varios concentradores que atiende a diversos ordenadores dentro de un edificio, una estación de bombeo o/y depuradora de agua que atiende las necesidades de núcleos de población de distinta naturaleza, etc.

Si los núcleos de población no desean la cercanía del centro de servicios porque su actividad puede ser molesta, o potencialmente perjudicial para sus habitantes,

hablaremos de problemas de localización de centros no deseados. Las primeras investigaciones en estos problemas se sitúan alrededor de 1980 y surgen de la inquietud que ha despertado en los ciudadanos las instalaciones potencialmente contaminantes, como son los vertederos, las centrales nucleares, etc.

En caso contrario, si el deseo es situar la instalación lo más cerca posible del conjunto de puntos de demanda, hablaremos de problemas de localización de centros deseados o simplemente problemas de localización que, con las restricciones que expondremos más adelante, serán los que estudiaremos en este trabajo doctoral.

De la definición anterior se desprende que, en primera instancia, varios son los factores a tener en cuenta:

- ¿Cuál es el espacio de trabajo?
- ¿Cuál es la región donde ubicar la nueva instalación y cuáles son los lugares posibles?
- ¿Con qué criterio decidimos que una ubicación es mejor que otra?

En referencia al espacio de trabajo, nos centraremos en un espacio bidimensional en el que los puntos se representan con dos coordenadas y se puede aplicar a regiones terrestres no muy amplias, donde no tiene mucha importancia la curvatura terrestre. En otras ocasiones la zona geográfica es tan amplia que, aunque sigamos representando los puntos con dos variables, el espacio de trabajo sería la superficie esférica del globo terráqueo.

El espacio bidimensional es el más habitual en los trabajos sobre localización pero también podríamos estar interesados en otros espacios de más dimensiones, como sería el caso de situar la instalación en un edificio de varias plantas, bajo el agua o en el aire, con lo que necesitaríamos una tercera coordenada para indicar la altura. En otras ocasiones el espacio de trabajo se limita a una línea y basta con una única variable para representar los puntos. En este caso, la dificultad del estudio se reduce considerablemente y se aplica en caso de que las instalaciones existentes, y las nuevas, sean contiguas a una carretera, línea férrea o corriente de agua. Como ejemplo,

podríamos estar interesados en ubicar un hospital junto a una carretera que de servicio a núcleos de población próximos a la misma.

En segundo lugar hemos de concretar la región donde se puede ubicar la nueva instalación. La naturaleza finita de las instalaciones existentes no conlleva necesariamente que la región de posibles soluciones sea también discreta. Aunque en ocasiones podemos modelar la región de forma discreta, por los nodos de una red o grafo, nos centraremos en los modelos que podemos describir a través de variables continuas, hablando así de lo que se conoce como problemas de localización continua.

En el caso discreto, hemos de enumerar todos los lugares en los que podemos ubicar las nuevas instalaciones, no así en el caso continuo en el que los posibles candidatos son todos los puntos de nuestro espacio de trabajo y nuestra labor será poner de relieve los mejores candidatos. De este modo, los modelos de localización continua actúan como “generadores de lugares” (Love et al. 1988), y siempre tendrán una marcada componente geométrica.

Hay que comprobar, sin embargo, que las posibles soluciones deben pertenecer a la *región factible*, región donde es posible ubicar las nuevas instalaciones. Es de esperar que dicha región esté descrita por restricciones que no sean demasiado complicadas, por ejemplo, la forma rectangular de una placa de circuito impreso. Cuando el espacio de ubicación es de naturaleza geográfica, la región factible casi siempre se describe como el interior de una región poligonal, cuyo límite se especifica por una secuencia de sucesivos puntos (*corners*), que van a ser conectados por segmentos, lo que suele ser frecuente cuando los datos provienen de los SIG (Sistemas de Información Geográfica). En otras ocasiones, motivos técnicos, económicos o políticos hacen que la región sea mucho más complicada.

En algunas ocasiones, como las situaciones estudiadas en este trabajo, el modelo no requiere de una región factible y hablaremos de un problema sin restricciones. Debe quedar claro, sin embargo, que tales modelos tienen naturaleza teórica y que prácticamente todas las aplicaciones al mundo real tienen restricciones espaciales. Dichas restricciones hacen que el modelo sea más complejo que el homólogo sin restricciones y a menudo no se tienen en cuenta. Con algo de suerte, el modelo sin restricciones produce una solución factible para las restricciones reales.

Generalmente esto no sucede, y hemos de tener en cuenta las restricciones en el proceso de solución. Sin embargo, el conocimiento de soluciones para el modelo sin restricciones es a menudo muy útil en la solución del problema con restricciones. Así, los modelos sin restricciones tienen su utilidad como un primer paso en el proceso de solución, y su estudio no debe ser descuidado.

El tercer factor a tener en cuenta consiste en la especificación de un objetivo, es decir, la definición de criterios para la elección de las mejores opciones o, al menos, las opciones que no se puedan mejorar fácilmente. Es frecuente que los modelos estén definidos como problemas de optimización, que tratan de reducir algunos costes, daños o molestias o/y maximizar alguna utilidad, calidad o bienestar.

En los problemas más fáciles, el objetivo se establece como un criterio único, pero en muchas aplicaciones (por no decir en todas) hay varios criterios simultáneamente (*modelos multicriterio*). Mientras que la primera situación conduce al problema clásico, pero de ninguna manera sencillo, de encontrar un punto mínimo o máximo de una función de las variables que intervienen, en el segundo caso el objetivo real es menos definido. Las opciones pueden ser tratadas de diferentes maneras, y lo que es tal vez peor, pueden dar lugar a diferentes soluciones.

El método más popular consiste en la determinación de soluciones eficientes, que son aquellas soluciones para las que no existe otra solución factible que sea, al mismo tiempo, tan buena para todos los criterios y mejor estrictamente por lo menos en uno de los criterios. Lamentablemente este conjunto suele ser difícil de determinar y, a menudo, demasiado grande para ser de utilidad directa en una toma de decisiones, ya que no apunta hacia una solución precisa.

Se puede argumentar, sin embargo, que esta última situación no es quizá tan mala en absoluto. En efecto, por un lado, los modelos continuos de localización se utilizan a menudo como una primera aproximación a los problemas del mundo real: incluso si sabemos que el modelo adecuado para su problema es discreto o se basa en los nodos de una red, estas representaciones suelen exigir mayor cantidad de datos, que no siempre pueden obtenerse fácilmente sin gran esfuerzo o coste. Podemos entonces utilizar una menor demanda de datos de descripción de un espacio continuo como un

“rápido pero sucio” método para determinar qué regiones son las más prometedoras para ser estudiadas con más precisión.

Por otro lado, los modelos de optimización están basados en formulaciones matemáticas precisas de las variables, limitaciones y objetivos, por lo que sólo describen los aspectos cuantitativos del problema en cuestión, sin tener en cuenta los aspectos cualitativos que se presentan con frecuencia, por lo que la evaluación de la mejor solución debería dejarse en manos de los tomadores de decisiones. Además, los datos en los que estos modelos se basan suelen ser aproximaciones, ya sea por datos no disponibles o por la incertidumbre, debido a los constantes cambios en el entorno que reflejan. En tales situaciones, el modelo de optimización debe ser considerado solamente como una herramienta de ayuda, ya que no se modela para determinar las decisiones precisas (soluciones óptimas), sino más bien sugerir las mejores decisiones por la eliminación de soluciones inadecuadas. Esto se puede realizar a través de un análisis de sensibilidad de las soluciones obtenidas que toma la forma de una región de soluciones casi-óptimas, o que contiene soluciones óptimas para cualquier situación que pueda surgir.

### **1.1.1 Modelando con distancias**

Aparte de características locales, como el costo de la tierra que pueden diferir de un lugar a otro, lo que siempre distingue a un lugar de otro es su posición relativa respecto de otros puntos, y una de las propiedades más básicas de la posición relativa es la distancia. Prácticamente todos los modelos de localización continua tienen que ver con las distancias entre los puntos. La distancia es la descripción matemática de la idea de proximidad, y esto puede adoptar muchas formas diferentes, dependiendo de la aplicación. Por ejemplo, en regiones “complicadas” como son las montañosas, pantanosas, marismas, e incluso las ciudades, sucede con frecuencia que dos lugares sean muy “cercaños”, pero el desplazamiento físico de un lugar a otro puede requerir un desvío grande o, en otras palabras, un gran coste debido al tiempo necesario para desplazarse por carreteras sinuosas o/y con pendientes variables o por calles en zigzag. Es decir, la distancia no sólo mide la cercanía espacial sino que mide también la cercanía temporal entre dos puntos. Por tanto, el conocimiento de la posición de dos

puntos (por sus coordenadas, por ejemplo) no es suficiente para calcular su distancia, se debe saber qué tipo de medida de distancia debe ser considerado, y son posibles muchos tipos.

En el contexto de dos o más dimensiones del espacio afín, la más conocida es la distancia euclídea, que mide la cercanía entre dos puntos considerando la longitud del segmento que los une, lo que tradicionalmente se declara como el camino más corto entre dos puntos es la línea recta. Si los dos puntos se describen, por ejemplo en el plano, por sus coordenadas  $X(x, y)$  y  $P(a, b)$ , relativas a un sistema de coordenadas ortonormal (los ejes son perpendiculares y utilizan la misma escala), entonces la distancia euclídea se obtiene como:

$$d_2(X, P) = \sqrt{(x-a)^2 + (y-b)^2}.$$

Este es quizás el ejemplo apropiado en la comunicación verbal, pero ciertamente no es adecuado para la distancia de viaje, excepto en el caso muy raro en el que uno puede desplazarse en todas direcciones con la misma facilidad, como a veces en un desierto o en el mar. Otras medidas, muy diferentes a la distancia euclídea surgen cuando sólo algunas direcciones de marcha están permitidas. Esto sucede en la práctica en las ciudades o con algunos tipos de maquinaria como, por ejemplo, un taladro automatizado. A menudo se mueve sobre una mesa de perforación por dos motores, uno en horizontal y otro en dirección vertical (sin tener en cuenta un tercer motor que lo mueve arriba y abajo).

Por razones técnicas, puede ser que los motores no puedan trabajar en paralelo, con lo que el *tiempo total* necesario para mover el taladro de una posición a otra es la suma de los movimientos horizontal y vertical (ver figura 1.1). En este caso la medida de distancia apropiada será la distancia rectangular, también conocida como la distancia de Manhattan, debido al trazado rectangular de sus calles, dada por:

$$d_1(X, P) = |x-a| + |y-b|.$$

Si ambos motores pudieran funcionar simultáneamente, y con el mismo nivel de potencia, el taladro se movería en un tiempo mucho más corto desde el punto  $X$  al punto  $P$ . Empezarían ambos motores juntos y cuando el taladro alcance la posición horizontal o vertical de  $P$  se pararía el motor correspondiente, por lo que el tiempo total

de viaje es el más grande entre los tiempos de viaje horizontal y vertical, y tenemos otra medida de la distancia:

$$d_{\max}(X, P) = \text{Max}\{|x - a|, |y - b|\}.$$

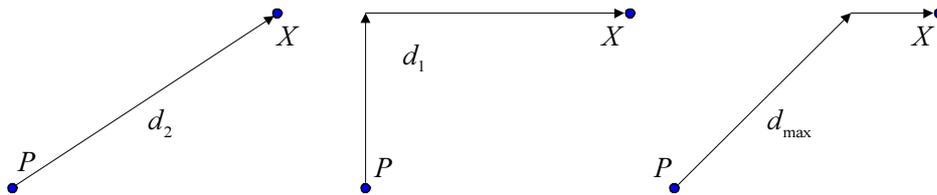


Figura 1.1: Desplazamientos según la norma.

Nótese que la trayectoria del taladro se compone de dos partes: una diagonal orientada, mientras que ambos motores trabajan juntos y, una parte horizontal o bien una vertical, dependiendo de qué motor se detuvo primero; también las velocidades reales a lo largo de estos caminos son diferentes: en la diagonal, la velocidad es  $\sqrt{2}$  veces mayor que en la última parte (Plastria, 2002).

Estos tres ejemplos son casos particulares, para  $p = 1, 2, \infty$ , de una distancia más general, conocida como distancia  $d_p$  ó “redonda”, cuya formulación es:

$$d_p(X, P) = \sqrt[p]{|x - a|^p + |y - b|^p}, \text{ con } p \in [1, \infty].$$

Esta situación puede ser ampliamente generalizada si admitimos movimientos en un número finito de direcciones fijas y sus velocidades correspondientes. Habrá muchas maneras posibles de pasar de un punto a otro, y se elige la que toma menor tiempo, lo que determina la distancia. De esta manera se obtiene una amplia clase de medidas llamadas *distancias block* o distancias poliédricas. Este nombre proviene del hecho de que cualquier “bola de radio  $r$ ”, es decir, el conjunto de todos los puntos que distan de otro (origen de la bola) a lo sumo  $r$ , es un polígono convexo (o poliedro en dimensiones más altas). Sus vértices pueden obtenerse desplazándose desde el mismo origen en cada dirección permitida, y a la velocidad correspondiente, en un tiempo  $r$ ; finalmente se toma la envolvente convexa de todos estos puntos que, en el caso del plano, es el menor polígono convexo que los encierra.

Aparte de las distancias rectangulares y máximo, distancias más generales que la euclídea, como son las distancias poliédricas, no suelen ser directamente aplicable en la práctica. Se utilizan, sin embargo, como aproximaciones a otras distancias, y a menudo conducen a problemas de localización que son más fáciles de resolver. Por ejemplo, se obtiene un valor aproximado bastante bueno de la distancia euclídea entre dos puntos en un mapa, utilizando una distancia hexagonal calculada por superposición de una malla hexagonal regular en el mapa y contando el número de celdas hexagonales que debe ser atravesada.

Dando otro salto hacia distancias más generales se llega a la familia de las distancias de Minkowski (*gauge*), que consiste en todas las nociones de distancia posibles que satisfacen la conocida desigualdad triangular (cada lado de un triángulo es menor que la suma de los otros dos) y son funciones convexas. Cada conjunto  $B$  convexo, cerrado y acotado, que contenga al origen, es la bola unidad para alguna medida de distancia. A partir de este conjunto se define la distancia de Minkowski, con bola unidad  $B$ , con la fórmula:

$$d_B(X, P) = d_B(0, X - P) = \min\{r > 0 \mid X - P \in rB\}.$$

En la práctica, existen obstáculos en los desplazamientos, como lagos, ríos, abismos, y las trayectorias o/y velocidades deben adaptarse a diferentes terrenos o a influencias externas, como las pendientes, los vientos o las corrientes. La influencia de una corriente constante (si no es demasiado fuerte) es relativamente fácil de tener en cuenta en las distancias  $d_p$ , simplemente mediante la adición de un término de perturbación lineal en la fórmula de la distancia (Plastria, 1992), destruyendo entonces la simetría de la distancia euclidiana: la distancia de ida ya no es igual a la distancia de vuelta

Los efectos que otras circunstancias tienen en el cálculo de la distancia entre dos puntos, es un problema difícil y exigen técnicas de computación muy complejos. Cuando la velocidad es constante en regiones poligonales y en sus segmentos fronterizos, los enfoques geométricos pueden ser útiles (Herschberger y Suri, 1999; Mitchell y Papadimitriou, 1991).

Un caso algo más simple de obstáculos surge cuando el espacio consta de una región poligonal y todo movimiento debe permanecer dentro de esta región (Guibas y Herschberger, 1989) o, por ejemplo, los desplazamientos se realizan en una isla o en un

lago, donde se puede viajar en todas direcciones con la misma facilidad, con la única condición de permanecer siempre en la isla o en el lago.

Una situación fácil se obtiene cuando el espacio de localización es una esfera. Aquí la noción habitual correspondiente a la distancia euclídea es la distancia ortodrómica, cuyas propiedades son bien conocidas. Esta es una descripción bastante adecuada de las distancias de vuelo, a pesar de que no tiene en cuenta los efectos perturbadores comunes, como los vientos o las fuerzas de Coriolis (Plastria, 2002).

En resumen, las distancias del mundo real son generalmente muy complicadas y, como regla general, incluyen obstáculos y no son invariantes con la traslación (Spiekerman y Wegener, 1994). Por lo tanto, si queremos una descripción muy cercana a la realidad hay que considerar nociones de distancia tan complejas que en la actualidad parecen estar fuera de alcance. Un campo de aplicación importante de tales distancias generalizadas se encuentra en la robótica, con los numerosos grados de libertad que tienen los brazos manipuladores en sus movimientos, y en el obstáculo que supone el complicado entorno dentro del cual deben funcionar. Aquí entramos en el terreno de los espacios de Riemann, las ecuaciones diferenciales parciales en conjunción con la combinatoria, y en los que la teoría de localización continua en realidad no ha entrado todavía.

### 1.1.2 Modelando los objetivos

En un modelo típico de localización continua tenemos un conjunto prefijado de puntos de *demanda*  $\{P_i\}_{i \in S}$ , donde  $S$  es un conjunto de índices que suele ser finito  $S = \{1, 2, \dots, M\}$ , aunque algunos modelos también consideran una distribución continua de la demanda.

Las posiciones  $X_1, X_2, \dots, X_N$  de las instalaciones que hemos de ubicar dependerán de su interacción con los puntos de demanda, y posiblemente entre sí, y los efectos de esta interacción dependerán principalmente de la distancia entre los puntos. El objetivo a considerar se resume entonces en una medida global de todos estos efectos.

Algunas interacciones son *atractivas*, lo que significa que el objetivo general se mejora cuando la distancia se reduce, en otras palabras, significan una atracción: “cuanto más cerca mejor”. Un ejemplo típico es el tiempo de servicio, que aumenta con la distancia y que normalmente debe ser minimizado. El problema de optimización habitual para una sola variable (*minisum*) es entonces:

$$\min_{X \in \Omega} \sum_{i \in S} w_i d(X, P_i),$$

donde  $\Omega$  es la región en la que queremos ubicar la instalación,  $d$  es la distancia más adecuada al problema y  $w_i \in \mathbb{R}^+$  son pesos que miden la demanda, población, costes unitarios, etc., de los puntos prefijados  $P_i$ .

Otras interacciones son *repulsivas* en el sentido de que es deseable ubicar los puntos que interactúan tan separados como sea posible. Esto sucede por ejemplo con instalaciones potencialmente peligrosas, que deben ser localizadas, en la medida de lo posible, lo más lejos posible de los lugares más sensibles y nos lleva a problemas de optimización tipo *maxisum*:

$$\max_{X \in \Omega} \sum_{i \in S} \alpha_i d(X, P_i).$$

Este modelo nos conduce a veces a soluciones que no son aceptables por su excesiva proximidad a algunos de los puntos prefijados y, por ello, se emplea el método *maximin* (Brimberg y Juel, 1998):

$$\max_{X \in \Omega} \left\{ \min_{i \in S} \{ \alpha_i d(X, P_i) \} \right\}.$$

Muchas aplicaciones, si no la mayoría, tendrán atracciones y repulsiones y, por lo tanto, requerirán un delicado acto de equilibrio entre estos dos efectos bastante contradictorios. Así, en Velasco (1991) se aborda el problema “maximin” con pesos, considerando la posibilidad de que el centro a instalar pueda ser deseado por algunas poblaciones y no deseado por otras. En general, la elección de un objetivo adecuado no es evidente, principalmente porque los dos efectos no son medibles de forma inmediata en escalas comparables. En tales casos, el objetivo general tal vez pueda ser mejor expresado como un conjunto de dos objetivos (Plastria y Carrizosa, 1999).

Los modelos de localización continua para una sola instalación pueden ser bastante difíciles de resolver, por lo que cabe esperar que sus homólogos de instalaciones

múltiples sean aún más difíciles. Probablemente, esta sea la razón por la que los modelos de localización continua de múltiples instalaciones sean menos corrientes. Pero esto está cambiando con el uso de métodos heurísticos, que comienzan a ser aplicados a dichos problemas que resultan irremediablemente complejos.

## 1.2 Breve reseña histórica

El planteamiento matemático del primer problema de localización conocido data de principios del siglo XVII. Kuhn (1967), proporciona un excelente esbozo histórico y atribuye su autoría a Pierre de Fermat (1601-1665) quien, en su ensayo sobre máximos y mínimos, lanza el siguiente reto:

*Aquel que no esté de acuerdo con mi método, que intente resolver el siguiente problema: dados tres puntos del plano, encontrar un cuarto punto de tal forma que la suma de las distancias a los tres puntos dados sea la menor posible.*

Otras fuentes que proporcionan detalles sobre el problema de Fermat son Zacharias (1913), Dörrie (1965) y Hosberger (1973). En ellas se atribuye su enunciado, y varias propuestas de resolución antes de 1640, a Torricelli (1608-1647) quien observó que las circunferencias circunscritas a los triángulos equiláteros construidos exteriormente sobre los lados del triángulo determinado por tres puntos, se cortan en el punto óptimo. De ahí que la solución del problema de Fermat sea conocida como el “punto de Torricelli”.

Sin embargo, como de costumbre, la historia de un problema es algo borrosa. Según las fuentes anteriores, otros matemáticos como Cavalieri, Viviani y Roberval trabajaron en esos momentos para encontrar otros métodos de solución del problema. Cavalieri (1598-1647) en su “Exerciones Geometricae” de 1647 demuestra que las rectas que unen el punto de Torricelli de un triángulo con sus vértices forman, dos a dos, ángulos de  $120^\circ$ . Según Melzak (1983) Cavalieri se lo propuso a Fermat y éste a su vez a Torricelli que obtuvo la solución aportada por el primero.

En 1834 Heinen prueba que el enunciado de Torricelli no es general, pues si los tres puntos forman un triángulo con un ángulo mayor o igual que  $120^\circ$ , el vértice de este ángulo es el punto óptimo. (Love et al. (1988)).

Posteriormente, en su libro “Doctrine and Application of Fluxions” (1750), Thomas Simpson (1710-1761) generaliza el problema y obtiene el punto que minimiza la suma ponderada de las distancias a tres puntos dados.

Los orígenes del problema dual se remontan a 1755. En la página 47 de la publicación “The Ladies diary” o “Woman’s Almanack”, aparece el siguiente problema propuesto por Thomas Moss:

*En los tres lados de un campo equiangular se encuentran tres árboles a distancias 10, 12 y 16 entre ellos. Encontrar el contenido del mayor campo que admite estos datos.*

Este problema geométrico fue enunciado de una forma más académica en la página 384 del Volumen I de Annales de Mathematiques Pures et Appliques (1810-11):

*Dado un triángulo cualquiera, circunscribirle el mayor triángulo equilátero posible.*

En Vol. II (1811-12) Rochat et al. aportan la solución:

*De este modo, el mayor triángulo equilátero que circunscribe a un triángulo tiene sus lados perpendiculares a las rectas que unen los vértices del triángulo dado con el punto tal que la suma de las distancias a esos vértices es mínima...Se puede concluir que la altura del mayor triángulo que se puede circunscribir a un triángulo dado es igual a la suma de las distancias desde los vértices del triángulo dado al punto que minimiza la suma de distancias.*

En el siglo XX, el problema y su solución se retomaron para utilizarlos en casos reales ante la necesidad de minimizar los costes de transporte o suministro a los puntos de demanda. Alfred Weber (1868-1958) en su libro “Uber den Standort der Industrien”

(1909) utilizó la versión ponderada del problema para tres puntos al buscar la ubicación de una fábrica que hiciera mínimos los costes de transporte de los tres puntos: dos de ellos eran fuentes de materias primas con diferentes pesos y el tercero era un punto de venta. En un apéndice de este libro, Georg Pick hace referencia a “un viejo aparato inventado por Varignon” y utiliza la analogía mecánica para describir el problema y resolverlo cuando el número de puntos sea mayor o igual que tres.

Estas primeras investigaciones no anticipaban el punto de vista iterativo habilitado por el advenimiento de la computadora electrónica. Esto explica por qué la mayoría de los trabajos sobre teoría de localización son recientes. La notable excepción es el artículo de E. Weiszfeld (1937), en el que propone un método iterativo de punto fijo para encontrar la mediana espacial o punto minisum para múltiples puntos con pesos positivos.

Dicho trabajo fue escrito en Francia en 1936, remitido desde Praga y publicado en una revista japonesa en 1937 permaneció virtualmente desconocido hasta los años sesenta del siglo XX. Por esta razón el método fue redescubierto años más tarde y de forma independiente por Miehe (1958), Kuhn y Kuenne (1962) y Cooper (1963).

Weiszfeld resolvió 25 años antes el objetivo principal de la investigación moderna en teoría de localización que comenzó a cobrar impulso al comienzo de la era de los ordenadores. De hecho, el problema fue descrito como no resuelto en *Progress in Operations Research*, Vol II, editado por Hertz y Eddison en 1964.

Aunque el problema de Weber es la piedra angular de la moderna y prolífica teoría de localización, innumerables autores publican sus trabajos haciendo referencia de este problema con variados nombres: el problema de Fermat, el problema generalizado de Fermat, el problema de Fermat-Torricelli, el problema de Steiner, el problema de Weber, el problema generalizado de Weber, el problema de Fermat-Weber, el problema minisum o el problema de la mediana espacial.

Podemos encontrar un instructivo recorrido histórico del problema de Weber en Love et al (1988), Francis et al. (1992), Wesolowsky (1993), Drezner et al. Capítulo 1 de Drezner, Z. y H.W. Hamacher (2002) y un breve resumen en Guerrero (2007).

### 1.3 Localización simple en dos regiones del plano con diferentes normas

En la práctica, suponer la existencia de una única norma que abarque a todo el plano, o solamente a la región que contiene a los puntos de demanda es una hipótesis muy restrictiva e irreal. Lo más común será que estos puntos, por motivos diversos, especialmente geográficos, se encuentren en regiones heterogéneas, en las que las distancias se midan a partir de distintas normas. Además, la frontera entre las distintas regiones también suelen ser de naturaleza geográfica y, por tanto, la expresión matemática de las mismas puede ser bastante complicada.

El objetivo fundamental de este trabajo doctoral es abordar una situación simplificada del problema anterior, en la que supondremos dos regiones separadas por una frontera rectilínea. El planteamiento del problema es el siguiente:

Sea  $r(x, y) \equiv \omega_2 y + \omega_1 x + \omega_0 = 0$  una recta con  $(x, y) \in \mathbb{R}^2$  y  $\omega_0, \omega_1, \omega_2 \in \mathbb{R}$ , que divide al plano  $\mathbb{R}^2$  en dos regiones  $\Omega_p$  y  $\Omega_q$  con normas  $\ell_p$  y  $\ell_q$ , respectivamente. Para  $p < q$  consideraremos  $r \subset \Omega_q$  ( $r \cap \Omega_p = \emptyset$ ).

Sean  $\mathcal{P} = \{P_i = (a_i, b_i)\}_{i \in S_p} \subset \Omega_p$  y  $\mathcal{Q} = \{Q_j = (c_j, d_j)\}_{j \in S_q} \subset \Omega_q$ , conjuntos de puntos de demanda (vértices) en  $\mathbb{R}^2$ . El problema de localización se establece como sigue:

$$\min_{X \in \mathbb{R}^2} \left\{ \sum_{i \in S_p} w_i d(P_i, X) + \sum_{j \in S_q} w_j d(X, Q_j) \right\} \quad (1)$$

donde  $w_i, w_j \in \mathbb{R}^+$ , son los pesos,  $S_p$  y  $S_q$  son dos conjuntos finitos de índices, y  $d$  es la distancia entre dos puntos.

Debemos señalar que si ambos puntos se encuentran en la misma región la distancia  $d(X, Y)$  se reduce a la norma  $\ell_p$  o  $\ell_q$ . Por otro lado, si los puntos no se encuentran en la misma región, entonces  $d(X, Y)$  es la longitud del camino más corto (*geodesic path*) que une a los puntos  $X$  e  $Y$  (Mitchell y Papadimitriou, 1991):

$$d(X, Y) = \min_{G \in r} \{d_p(X, G) + d_q(G, Y)\} \quad (2)$$

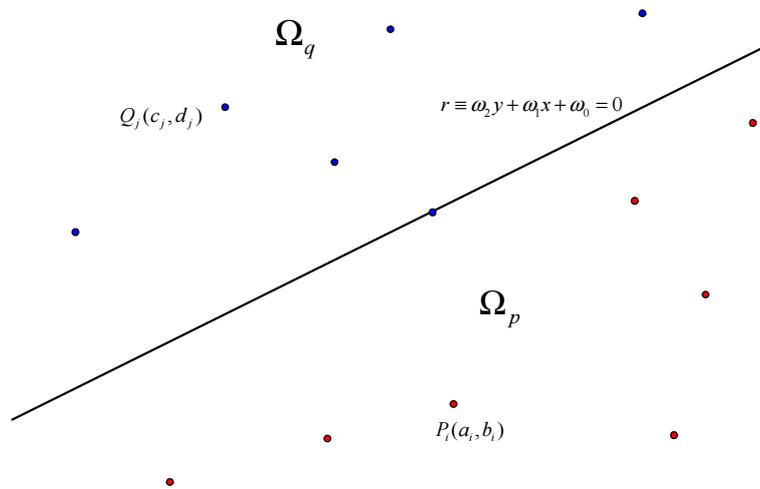


Figura 1.2: Representación gráfica de un problema de localización en regiones heterogéneas.

Una aplicación obvia del problema (1) es la ubicación de una instalación dentro o fuera de una zona urbana donde, debido a la distribución de las calles dentro de los límites de la ciudad ( $\Omega_p$ ), el movimiento es lento, mientras que fuera de esta frontera, en el área rural ( $\Omega_q$ ), el movimiento es rápido y en ese caso,  $p < q$  (Brimberg et al. 2003).

Otra aplicación del problema (1) podemos encontrarla en Love y Walker (1994), Fernández et al. (2002) y Brimberg et al. (2007), donde se estudian dos regiones de Londres (la “city” de Londres y el norte de Londres) y, usando los criterios AD y SD (Brimberg et al., 1996), se obtienen normas  $\ell_p$  y  $\ell_q$ , distintas para cada región.

El problema (1) ha sido estudiado para las normas  $\ell_1$  y  $\ell_2$  en Parlar (1994), aplicado a la distribución de gas natural a puntos de demanda situados en una ciudad donde las tuberías se encuentran bajo las calles horizontales y verticales, y a puntos de demanda situados en los alrededores de la misma, donde las tuberías pueden ir en línea recta en cualquier dirección. En este trabajo se prueba que la función objetivo:

$$D(X) = \sum_{i \in S_p} w_i d(P_i, X) + \sum_{j \in S_q} w_j d(X, Q_j),$$

no es una función convexa en  $X$ , y se propone un algoritmo heurístico denominado MWP (Modified Weiszfeld Procedure), cuyos resultados son comparados con los obtenidos con el método ARS (Adaptive Random Search) de Gottfried y Weisman, 1973. Establece el algoritmo creando una mixtura de distancias de la siguiente forma:

$$d(X, Y) = \begin{cases} d_1(X, Y) & \text{si } X, Y \in \Omega_1 \\ d_{12}(X, Y) & \text{si } X \in \Omega_1, Y \in \Omega_2 \\ d_2(X, Y) & \text{si } X, Y \in \Omega_2 \\ d_{21}(X, Y) & \text{si } X \in \Omega_2, Y \in \Omega_1 \end{cases} \quad \text{ó } (X \in \Omega_1, Y \in r) \quad \text{ó } (X \in r, Y \in \Omega_1)$$

donde  $d_{12}(X, Y) = \min_{G \in r} \{d_1(X, G) + d_2(G, Y)\}$  y  $d_{21}(X, Y) = \min_{G \in r} \{d_2(X, G) + d_1(G, Y)\}$ .

A continuación, aplica el algoritmo de Weiszfeld como si hubiera una “única” distancia  $d$  definida a trozos. Este algoritmo destaca por su rapidez y, en algunos casos, por la falta de precisión en la función objetivo, errando incluso en la región donde se encuentra el punto óptimo si la pendiente es inferior a 0.5. No obstante, si la pendiente de la recta es elevada se consiguen errores inferiores al 0,03%. Hemos implementado este algoritmo y sus resultados se muestran en las tablas I y II del capítulo V.

Antecedentes con medidas de distancia mixtas podemos encontrar en Planchart y Hurter (1975) y en Michelot y Lefebvre (1987). En sus modelos, la función “distancia” se obtiene, para cada punto de demanda, por adición de las distancias euclídea y rectangular:  $d(X, P_i) = d_1(X, P_i) + d_2(X, P_i)$ , en el primer caso y con mezclas de diferentes *gauges* en el segundo.

Batta y Palekar (1988) analizan un problema de localización similar en una ciudad que tiene sectores no planificados en la parte antigua, en la que los desplazamientos se realizan en una estructura en red, mientras que en la parte más moderna la estructura de las calles es rectilínea.

El problema (1) es resuelto en Brimberg et al. (2003) en el caso particular de normas  $\ell_1$  y  $\ell_2$ , y de recta vertical  $r(x, y) \equiv x = \alpha$  y, por un giro de amplitud  $\frac{\pi}{2}$ , que no altera las distancias, para las rectas horizontales  $r(x, y) \equiv y = \beta$ . Para ello, divide el problema no convexo en dos problemas convexos donde usa el algoritmo de Weiszfeld con suavizado hiperbólico y el algoritmo de cálculo de la mediana unidimensional.

Además, aborda el problema (1) para el caso general de normas  $\ell_p$  y  $\ell_q$ ,  $1 \leq p < q$  y recta oblicua  $r(x, y) \equiv y = mx$ , con  $m > 0$ , y considera que los puntos de la recta están en el semiplano con norma  $\ell_q$ .

### 1.3 Localización simple en dos regiones del plano con diferentes normas

Supongamos que la solución del problema (1) se encuentra en  $\Omega_q$ . En este caso, el problema (1) se puede reformular de la siguiente forma:

$$\min_{X \in \mathbb{R}^2} \left\{ \sum_{i \in S_p} w_i d(P_i, X) + \sum_{j \in S_q} w_j d_q(X, Q_j) \right\} = \min_{X \in \Omega_q, Z_i \in r} \left\{ \sum_{i \in S_p} w_i d_p(P_i, Z_i) + \sum_{i \in S_p} w_i d_q(Z_i, X) + \sum_{j \in S_q} w_j d_q(X, Q_j) \right\}$$

,

donde  $Z_i = (z_i, mz_i) \in r$  para cada  $i \in S_p$ .

Tenemos por tanto el problema siguiente:

$$\min_{(x,y) \in \Omega_q, z_i \in \mathbb{R}} \left\{ \sum_{i \in S_p} w_i^p \sqrt{|z_i - a_i|^p + |mz_i - b_i|^p} + \sum_{i \in S_p} w_i^q \sqrt{|z_i - x|^q + |mz_i - y|^q} + \sum_{j \in S_q} w_j^q \sqrt{|x - c_j|^q + |y - d_j|^q} \right\}$$

.

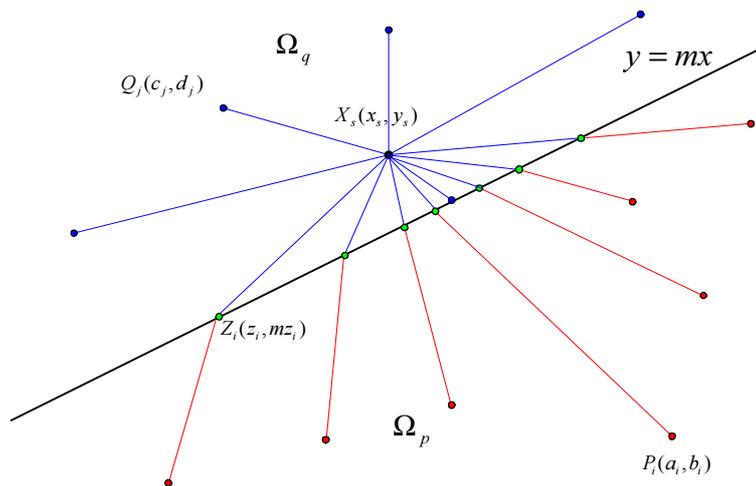


Figura 1.3: Punto solución en  $\Omega_q$ .

Este es un problema convexo sin restricciones que se puede resolver modificando el procedimiento de Weiszfeld de localización múltiple y suavizando con una aproximación hiperbólica. La principal modificación estriba en que tenemos dos normas en lugar de una única norma. La función objetivo suavizada con  $\varepsilon > 0$  queda:

$$\begin{aligned}
 DH(x, y, z_1, \dots, z_{|S_p|}) = & \sum_{i \in S_p} w_i^p \sqrt{\left( (z_i - a_i)^2 + \varepsilon \right)^{p/2} + \left( (mz_i - b_i)^2 + \varepsilon \right)^{p/2}} \\
 & + \sum_{i \in S_p} w_i^q \sqrt{\left( (x - z_i)^2 + \varepsilon \right)^{q/2} + \left( (y - mz_i)^2 + \varepsilon \right)^{q/2}} \\
 & + \sum_{j \in S_q} w_j^q \sqrt{\left( (x - c_j)^2 + \varepsilon \right)^{q/2} + \left( (y - d_j)^2 + \varepsilon \right)^{q/2}}.
 \end{aligned}$$

Para el algoritmo propuesto se igualan las  $|S_p| + 2$  derivadas parciales a cero y se obtiene un problema de punto fijo, resuelto de forma iterativa:

Análogamente se construye el procedimiento si suponemos que la solución se encuentra en  $\Omega_p$ .

Este procedimiento, que denominaremos MFP, ha sido implementado en este trabajo y sus resultados son comparados con los obtenidos con otros procedimientos.

Resaltamos ahora que el método MFP crea  $|S_p| + 1$  sucesiones de puntos:

$$X^k(x^k, y^k) \rightarrow X_s(x_s, y_s) ; Z_i^k(z_i^k, m z_i^k) \rightarrow Z_i^*(z_i^*, m z_i^*), i \in S_p.$$

Evidentemente, para cada  $i \in S_p$ ,  $Z_i^*$  ha de ser necesariamente la solución del problema (2) para los puntos  $X_s(x_s, y_s)$  y  $P_i(a_i, b_i)$ .

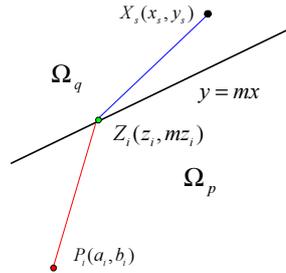


Figura 1.4: Línea geodésica entre puntos de regiones con distintas métricas.

$$d(P_i, X_s) = \min_{G \in r} \{d_p(P_i, G) + d_q(G, X_s)\} = d_p(P_i, Z_i^*) + d_q(Z_i^*, X_s).$$

Es decir, para cada  $i \in S_p$ ,  $Z_i^*$  ha de ser necesariamente el punto por el cual el camino más corto entre  $P_i$  y  $X_s$  cruza de un semiplano al otro (*crossing point*). Es plausible que el conocimiento previo de estos puntos, que en este trabajo denominamos puertas

de acceso o portones (*gates*), consiga mayor precisión y acelere de forma considerable cualquier algoritmo que pretenda obtener el punto óptimo en este tipo de problemas.

De hecho, Zaferanieh et al. (2008) caracterizan las puertas de acceso, pero solamente en el caso de las normas  $\ell_1$  y  $\ell_2$ :

*Supongamos que el plano  $\mathbb{R}^2$  está dividido por una recta  $y = mx$ , en dos regiones  $\Omega_1$  y  $\Omega_2$ , respectivamente. Supongamos, sin pérdida de generalidad, que  $m > 0$ . Entonces, para cualquier par de puntos  $P(a,b) \in \Omega_1$  y  $Q(c,d) \in \Omega_2$ , el camino más corto de  $P$  a  $Q$  cruza la recta por el punto  $(a, ma)$  ó  $(b/m, b)$  ó  $(g, mg)$  con  $a \leq g \leq b/m$  y*

$$g = \begin{cases} \frac{d+mc}{1+m^2} + \frac{d-mc}{m-1} \left( \frac{\sqrt{2m}}{-2m} + \frac{\sqrt{2m}}{1+m^2} \right), & m \neq 1 \\ \frac{c+d}{2}, & m = 1. \end{cases}$$

Sin embargo, no utilizan este resultado para mejorar el algoritmo MFP, en cambio, demuestran que la solución se encuentra en la envolvente rectangular de los puntos de demanda iniciales usando el Lema de Farkas (Bazaraa et al. 1993) y usan una versión del método BSSS (*Big Square Small Square*), adaptada al caso particular que nos ocupa, realizando divisiones sucesivas en rectángulos y obteniendo una cota inferior de la función objetivo en el centro de los mismos, ya que dichos centros siempre se encuentran en la región factible.

El método BSSS es un algoritmo geométrico de bifurcación y acotación (*branch and bound*) sugerido por Hansen et al. (1981) para resolver problemas de localización de centros no deseados. Consiste en dividir el cuadrado inicial en sucesivos cuadrados, sobre cada uno de los cuales se obtiene una cota superior (inferior) de la función objetivo. Si dado un cuadrado, la función objetivo es peor que la actual cota inferior (superior), el cuadrado es rechazado. Se continúa la subdivisión hasta alcanzar la tolerancia deseada.

Este algoritmo aparece otra vez en (Hansen et. al, 1985), para resolver el problema de Weber, eliminado en cada iteración aquellos cuadrados que no superen un determinado

test de “optimalidad”, consiguiendo de esta forma disminuir la sobrecargada lista de cuadrados del algoritmo inicial.

En (Plastria, 1992a) se construye un algoritmo generalizado, denominado GBSS (Generalized Big Square Small Square) sustituyendo de manera natural los cuadrados por rectángulos y en el que la función objetivo es evaluada, al menos una vez, en cada uno de ellos.

Entre los métodos de bifurcación y acotación señalamos, para finalizar, el algoritmo BTST (Big Triangle Small Triangle) propuesto en (Drezner y Suzuki, 2004), basado en la triangulación de Delaunay que se puede utilizar tanto para la localización de centros no deseados como para resolver el problema de Weber con pesos positivos y negativos.

## 1.4 Justificación y organización del trabajo

Como expusimos anteriormente, resolver un problema de localización simple en el plano entre regiones con diferentes normas nos conduce a resolver el problema de optimización (1).

Si la solución se encuentra en  $\Omega_p$ , este problema se puede reescribir como:

$$\min_{X \in \Omega_p, Z_j \in r} \left\{ \sum_{i \in S_p} w_i d_p(P_i, X) + \sum_{j \in S_q} w_j d_p(X, Z_j) + \sum_{j \in S_q} w_j d_q(Z_j, Q_j) \right\},$$

con  $Z_j = (z_j, mz_j) \in r$ ,  $j \in S_q$ .

Tenemos pues un problema de optimización no lineal sin restricciones con  $|S_q| + 2$  incógnitas  $\{x, y, z_1, z_2, \dots, z_{|S_q|}\}$ .

Reescribiendo nuevamente el problema (1) tenemos:

$$\min_{X \in \Omega_p} \left\{ \sum_{i \in S_p} w_i d_p(P_i, X) + \min_{Z_j \in r} \left\{ \sum_{j \in S_q} w_j d_p(X, Z_j) + \sum_{j \in S_q} w_j d_q(Z_j, Q_j) \right\} \right\}.$$

Si para cada  $X(x,y) \in \Omega_p$  y cada  $Q_j \in \Omega_q$  conocemos el valor de  $z_j = z_j(x,y,c_j,d_j)$  o, en otras palabras, si conocemos la solución del problema (2):

$$\min_{Z_j \in R} \left\{ \sum_{j \in S_q} w_j d_p(X, Z_j) + \sum_{j \in S_q} w_j d_q(Z_j, Q_j) \right\},$$

el problema de optimización (1) se habrá simplificado considerablemente ya que el número de incógnitas será solamente dos  $(x, y)$ . Por ello, el capítulo 2 estará dedicado a la obtención explícita de la solución del problema (2), en los casos que sea posible.

En el capítulo 3 obtendremos los principales resultados teóricos que, apoyándose en las soluciones del problema (2), nos permitan construir algoritmos de búsqueda de la solución del problema (1).

En el capítulo 4 detallaremos los algoritmos existentes sobre el problema (1) y expondremos los algoritmos resultantes de este trabajo doctoral, basados en los resultados teóricos del capítulo III.

El capítulo 5 presenta los resultados computacionales de los algoritmos propuestos en el capítulo anterior y se comparan, en cuanto a precisión y velocidad, con los propuestos por otros autores.

En el capítulo 6 se muestran las conclusiones del trabajo y las posibles investigaciones futuras.

El trabajo finaliza con la bibliografía empleada y dos anexos correspondientes a los datos utilizados (Anexo I) y las implementaciones en MATLAB de los algoritmos propuestos (Anexo II).



---

# LÍNEAS GEODÉSICAS ENTRE REGIONES CON DIFERENTES MÉTRICAS

---

### 2.1 Introducción y resultados previos

En el capítulo anterior hemos resaltado que el cálculo de la distancia entre dos puntos situados en regiones con distintas métricas (problema (2)) es una clave importante para resolver el problema (1). En este capítulo vamos a centrarnos en el problema (2), empleando una metodología basada en la pendiente de la recta de separación de las regiones y las pendientes de las rectas “geodésicas” de cada región. Estas últimas serán definidas en este capítulo como el camino desde un punto dado en una región al punto más próximo de la recta de separación.

En (Mitchell y Papadimitriou, 1991) se estudia este problema (2) al considerar una división del plano en polígonos convexos y, por tanto, las fronteras son rectilíneas, pero sólo considera distancias proporcionales a la distancia euclídea, con lo que obtiene resultados análogos a la ley de Snell sobre refracción de la luz.

En (Brimberg et al., 2003) se demuestra que el camino más corto entre dos puntos  $P$  y  $Q$  situados en regiones con normas  $\ell_1$  y  $\ell_2$ , respectivamente, es la poligonal

$\overline{PM} \cup \overline{MQ}$ , donde  $M$  es la proyección ortogonal de  $P$  en la recta de separación de las regiones, pero sólo tiene en cuenta rectas verticales y horizontales.

En (Brimberg et al., 2005) se denomina “gate point” al punto solución del problema (2) al considerar una región acotada del plano con norma distinta de la norma existente en el resto del plano.

En (Zaferanieh et al., 2008), se generaliza el resultado de 2003 para rectas cualesquiera y normas  $\ell_1$  y  $\ell_2$ . Nuestro objetivo en este capítulo es generalizar estos resultados para un abanico más amplio de normas  $\ell_p$  y  $\ell_q$ , para, en capítulos posteriores, construir algoritmos que resuelvan el problema (1).

Para ello vamos a exponer previamente unos resultados conocidos sobre distancias, que pueden encontrarse, por ejemplo, en Love et al. (1988) y en Beckenbach y Bellman (1961).

**Proposición 2.1.** Sean  $P(a,b)$  y  $Q(c,d)$  puntos cualesquiera de  $\mathbb{R}^2$ . Se verifica que:

1. Si  $a=c$  ó  $b=d$ , entonces  $d_p(P,Q) = d_q(P,Q)$ , para cualesquiera  $p, q \in [1, \infty]$ .
2. Si  $p \leq q$ , entonces  $d_p(P,Q) \geq d_q(P,Q)$ , para cualesquiera  $p, q \in [1, \infty]$ .
3. Si  $P'$  y  $Q'$  son los transformados de  $P$  y  $Q$ , por un movimiento que sea composición de simetrías horizontales o/y verticales, traslaciones o giros múltiples de  $90^\circ$ , entonces:  $d_p(P',Q') = d_p(P,Q)$ .
4. Si  $X = \mu P + (1-\mu)Q$ ,  $0 \leq \mu \leq 1$ , entonces  $d_p(P,Q) = d_p(P,X) + d_p(X,Q)$ , para cualquier  $p \in [1, \infty]$ .

Nótese que  $X = \mu P + (1-\mu)Q$  con  $0 \leq \mu \leq 1$ , es equivalente a afirmar que  $X$  pertenece al segmento que une  $P$  y  $Q$ , es decir:

$$X \in L[P,Q] = \{Y \in \mathbb{R}^2 / Y = \mu P + (1-\mu)Q, 0 \leq \mu \leq 1\}.$$

**Proposición 2.2.** Si  $X \in L[P,Q]$  y  $1 \leq p < q \leq \infty$ , entonces:

$$d_\infty(P,Q) \leq d_q(P,Q) \leq d_p(P,X) + d_q(X,Q) \leq d_p(P,Q) \leq d_1(P,Q)$$

Además, si  $\mathcal{A} \subset \mathbb{R}^2$  es un conjunto tal que  $\mathcal{A} \cap L[P,Q] \neq \emptyset$ , entonces:

$$d_\infty(P, Q) \leq d_q(P, Q) \leq \inf_{X \in \mathcal{A}} \{d_p(P, X) + d_q(X, Q)\} \leq d_p(P, Q) \leq d_1(P, Q)$$

**Demostración.** Sea  $X \in L[P, Q]$ , entonces, por la proposición anterior:

$$d_q(P, Q) = d_q(P, X) + d_q(X, Q) \leq d_p(P, X) + d_q(X, Q) \leq d_p(P, X) + d_p(X, Q) = d_p(P, Q)$$

Además, para cualquier  $X \in \mathcal{A}$ ,

$$\begin{aligned} d_q(P, Q) &\leq d_q(P, X) + d_q(X, Q) \leq d_p(P, X) + d_q(X, Q) \Rightarrow \\ d_q(P, Q) &\leq \inf_{X \in \mathcal{A}} \{d_p(P, X) + d_q(X, Q)\} \end{aligned}$$

Además, existe al menos un punto  $Y \in L[P, Q] \cap \mathcal{A}$ , para el cual se tiene:

$$\begin{aligned} d_p(P, Y) + d_q(Y, Q) &\leq d_p(P, Y) + d_p(Y, Q) = d_p(P, Q) \Rightarrow \\ \inf_{X \in \mathcal{A}} \{d_p(P, X) + d_q(X, Q)\} &\leq d_p(P, Q) \quad \blacksquare \end{aligned}$$

**Proposición 2.3.** Sea  $P(a, b) \in \mathbb{R}^2$  y  $r \equiv \omega_2 y + \omega_1 x + \omega_0 = 0$  una recta tal que  $P \notin r$ . La función  $f(X) = d_p(P, X)$ , con  $X \in r$ , es una función estrictamente convexa si  $p \in (1, \infty)$  y es convexa si  $p = 1$  ó  $p = \infty$ .

**Nota 2.4.** Además, en Love et al. (1988) se demuestra que la función anterior es estrictamente convexa para  $p = 1$  salvo que la recta tenga pendiente  $|m| = \left| \frac{\omega_2}{\omega_1} \right| = 1$  y también es estrictamente convexa para  $p = \infty$  salvo que la recta sea horizontal o vertical  $m = 0$  ó  $m = \infty$ .

**Definición 2.5.** Dado un punto  $P$  y una recta  $r$ , la  $\ell_p$ -distancia entre ambos, denotada como  $d_p(P, r)$ , se define como

$$d_p(P, r) = \min_{X \in r} \{d_p(P, X)\} = \min_{X \in r} f(X) \quad (3)$$

Nótese que la distancia  $d_p(P, r)$  está bien definida ya que  $f$  es convexa y está acotada inferiormente ( $f(X) \geq 0$ ).

A un punto en el que la función alcance el mínimo se le llama proyección sobre la recta del punto  $P$  según la dirección geodésica, es decir, la dirección del camino más corto

para la norma  $\ell_p$ , y lo denotaremos como  $P_p = \text{proy}_{(r, d_p)}(P)$ , o simplemente  $P_p = \text{proy}(P)$  si no hay ambigüedad.

La definición anterior es un caso particular del resultado siguiente (Mangasarian, 1999), en el que  $w' = (w_1, w_2, \dots, w_n)$  es el vector traspuesto de  $w$  y  $w'x$  es el producto escalar de  $w$  y  $x$ .

**Teorema 2.6.** *Consideremos un hiperplano  $\Pi = \{x \in \mathbb{R}^n / w'x + b = 0\}$ ,  $0 \neq w \in \mathbb{R}^n$ ,  $b \in \mathbb{R}$  y un punto  $P \in \mathbb{R}^n$ . La proyección de este punto sobre el hiperplano,  $\text{proy}(P) \in \Pi$ , según una norma arbitraria  $\|\cdot\|$  sobre  $\mathbb{R}^n$  es:*

$$\text{proy}(P) = P - \frac{w'P + b}{\|w\|'} y(w),$$

donde  $\|\cdot\|'$  es la norma dual de la norma  $\|\cdot\|$  e  $y(w) = \arg \max_{\|y\|=1} w'y$ .

Consecuentemente, la distancia entre  $P$  y su proyección  $\text{proy}(P) \in \Pi$  es:

$$d(P, \text{proy}(P)) = \|P - \text{proy}(P)\| = \frac{|w'P + b|}{\|w\|'}.$$

Un caso particular del teorema anterior, para normas  $\ell_p$ , y de los resultados obtenidos en (Dax, 2006) y (Plastria y Carrizosa, 2001), es el siguiente:

**Proposición 2.7.** *Consideremos un hiperplano  $\Pi \equiv w'x + b = 0$  y un punto  $P = (p_1, p_2, \dots, p_n)' \in \mathbb{R}^n$ . Entonces, la  $p$ -distancia entre  $P$  y el hiperplano  $\Pi$ , viene dada por:*

$$d_p(P, \Pi) = \min_{X \in \Pi} d_p(P, X) = \frac{|w'P + b|}{\|w\|_{\bar{p}}},$$

donde  $\bar{p}$  es el conjugado de  $p$ , es decir:  $\frac{1}{p} + \frac{1}{\bar{p}} = 1 \Rightarrow \bar{p} = \frac{p-1}{p}$  si

$p > 1$ .

Si  $p = 1$  entonces, su norma conjugada es la norma del máximo o, lo que es lo mismo,  $\bar{p} = \infty$  y viceversa.

Además, el mínimo se alcanza en  $P_p = (p_1^*, p_2^*, \dots, p_n^*) = \text{proy}(P) \in \Pi$ , donde,

$$\text{para } p \in (1, \infty], \quad p_i^* = \begin{cases} p_i & \text{si } w_i = 0 \\ p_i - \frac{w'P + b}{\|w\|_p^p} \frac{|w_i|^p}{w_i} & \text{si } w_i \neq 0 \end{cases},$$

$$\text{para } p = 1, \quad p_i^* = \begin{cases} p_i & \text{si } i \neq h \\ p_i - \frac{w'P + b}{w_i} & \text{si } i = h \end{cases},$$

para cada  $h$  tal que  $|w_h| = \max_{i=1, \dots, n} \{|w_i|\} = \|w\|_\infty$ .

**Nota 2.8.** Con respecto al número de soluciones del problema (3), podemos indicar que si  $p \in (1, \infty)$  existe una única solución, ya que la función  $f(X) = d_p(P, X)$  es estrictamente convexa.

Sin embargo, para  $p = 1$ , puede ocurrir que:

- exista un único  $h$  tal que  $|w_h| = \|w\|_\infty$  entonces la solución es única.
- existan  $k$  componentes de  $w$  tales que  $|w_{i_1}| = \dots = |w_{i_k}| = \|w\|_\infty$ , entonces existen infinitas soluciones  $P_p(\alpha) = (p_1^*, p_2^*, \dots, p_n^*)$ , definidas como sigue:

$$p_i^* = \begin{cases} p_i - \alpha_h \frac{w'P + b}{w_i} & \text{si } i = i_h, \quad h = 1, \dots, k \\ p_i & \text{en otro caso} \end{cases},$$

para cualquier  $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_k) \in \mathbb{R}^k$ , tal que  $\alpha_h \geq 0$ ,  $h = 1, \dots, k$ ,  $\sum_{h=1}^k \alpha_h = 1$ .

Para  $p = \infty$ , puede ocurrir que,

- no exista  $k$  tal que  $w_k = 0$  entonces la solución es única.
- existan  $k$  componentes de  $w$  tales que  $w_{i_1} = w_{i_2} = \dots = w_{i_k} = 0$ . Entonces, existen infinitas soluciones  $P_p(\alpha) = (p_1^*, p_2^*, \dots, p_n^*)$ , definidas como sigue:

$$P_i^* = \begin{cases} p_i + \alpha_h & \text{si } i = i_h, \quad h = 1, \dots, k \\ p_i - \frac{w'P + b}{\|w\|_1} \frac{|w_i|}{w_i} & \text{si } i \neq i_h \end{cases},$$

para cualquier  $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_k) \in \mathbb{R}^k$  tal que  $\max_{h=1, \dots, k} |\alpha_h| \leq \frac{|w'P + b|}{\|w\|_1}$ .

De todo ello, considerando que estamos en  $\mathbb{R}^2$ , se sigue:

**Proposición 2.9.** Dado un punto  $P(a, b)$  y una recta  $r(x, y) \equiv \omega_2 y + \omega_1 x + \omega_0 = 0$ , existe un punto  $P_p(x_p, y_p) \in r$  tal que  $d_p(P, r) = d_p(P, P_p)$  y sus coordenadas son:

$$\begin{cases} x_p = a - \frac{r(P)}{|\omega_1|^{\bar{p}} + |\omega_2|^{\bar{p}}} \frac{|\omega_1|^{\bar{p}}}{\omega_1} & \text{si } \omega_1 \neq 0 \quad \text{en otro caso } x_p = a \\ y_p = b - \frac{r(P)}{|\omega_1|^{\bar{p}} + |\omega_2|^{\bar{p}}} \frac{|\omega_2|^{\bar{p}}}{\omega_2} & \text{si } \omega_2 \neq 0, \quad \text{en otro caso } y_p = b \end{cases}, \quad \text{para } p \in (1, \infty]$$

$$\begin{cases} x_p = a - \frac{r(P)}{\omega_1} & \text{si } |\omega_1| > |\omega_2|, \quad \text{en otro caso } x_p = a \\ y_p = b - \frac{r(P)}{\omega_2} & \text{si } |\omega_2| \geq |\omega_1|, \quad \text{en otro caso } y_p = b \end{cases}, \quad \text{para } p = 1$$

donde  $r(P) \equiv \omega_2 b + \omega_1 a + \omega_0$  y  $\bar{p}$  es el conjugado de  $p$ . Además, el punto  $P_p$  es único si  $p \in (1, \infty)$ . ■

A partir de la proposición anterior, vamos a considerar algunos casos particulares, dependiendo de los valores de  $\omega_0, \omega_1, \omega_2$  y  $p$ , que nos serán útiles en los estudios que haremos posteriormente.

- Si  $\omega_1 = 0$ , entonces  $P_p = \left( a, -\frac{\omega_0}{\omega_2} \right)$ , y si  $\omega_2 = 0$ , entonces  $P_p = \left( -\frac{\omega_0}{\omega_1}, b \right)$ , es decir, para rectas horizontales y verticales existe un punto  $P_p$  que no depende del valor de  $p$ .

Ahora bien, si  $p = \infty$ , existen, según la nota 2.8, infinitos puntos en la recta cuya distancia al punto  $P$  es mínima. La figura 2.1 ilustra esta situación:

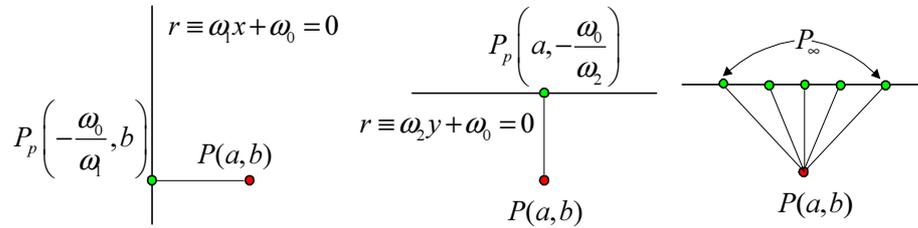


Figura 2.1: Proyecciones en rectas horizontales y verticales, según  $p$ .

- Si  $|\omega_1| = |\omega_2|$ , entonces  $P_p = \left( a - \frac{r(P)}{2\omega_1}, b - \frac{r(P)}{2\omega_2} \right)$ , es decir, para rectas con pendiente  $\pm 1$  existe un punto  $P_p$  que no depende del valor de  $p$ .

En el caso  $p = 1$ , existen, según la nota 2.8, infinitos puntos en la recta cuya distancia al punto  $P$  es mínima. La figura 2.2 ilustra esta situación:

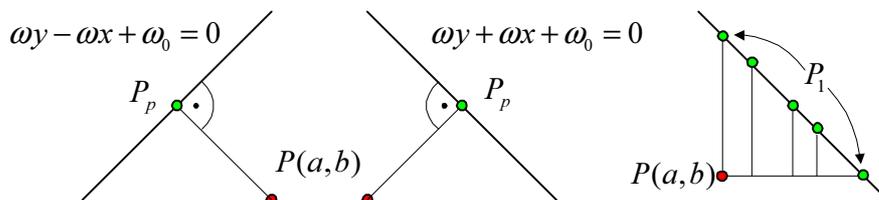


Figura 2.2: Proyecciones en rectas con pendientes  $\pm 1$ .

- En caso contrario, i.e.  $0 \neq |\omega_1| \neq |\omega_2| \neq 0$  existe un único punto  $P_p$  para cualquier valor de  $p$ .

La metodología que vamos a seguir en lo sucesivo utilizará la ecuación de la recta en la forma  $r \equiv y = mx + y_0$ , o bien  $r \equiv x = \alpha$ , si es vertical. Por ello, vamos a estudiar la expresión de la pendiente “geodésica”  $m_p^g$  de la recta  $r_p(P)$  que une  $P \notin r$  con el punto  $P_p = (x_p, y_p) \in r$ , es decir, el camino geodésico de  $P$  a  $r$  con respecto a la norma  $\ell_p$ . De esta forma, al igual que en los desarrollos anteriores, debemos de distinguir según los valores de  $p$ .

- Para  $p \in (1, \infty)$ 
  - Si  $\omega_1 \cdot \omega_2 \neq 0$ , entonces  $m = -\frac{\omega_1}{\omega_2}$  es la pendiente de la recta  $r$  y, usando las fórmulas de la proposición 2.9, tenemos:

$$m_p^g = \frac{y_p - b}{x_p - a} = \frac{\frac{r(P)}{|\omega_1|^{\bar{p}} + |\omega_2|^{\bar{p}}} \frac{|\omega_2|^{\bar{p}}}{\omega_2}}{\frac{r(P)}{|\omega_1|^{\bar{p}} + |\omega_2|^{\bar{p}}} \frac{|\omega_1|^{\bar{p}}}{\omega_1}} = \frac{|\omega_2|^{\bar{p}}}{\omega_2} \frac{\omega_1}{|\omega_1|^{\bar{p}}} = -m |m|^{-\bar{p}}.$$

Obsérvese que si  $|m| = 1$ , entonces  $m_p^g = -m$  con independencia del valor de  $p$ , es decir, la línea geodésica del punto a la recta es la perpendicular, para cualquier norma  $\ell_p$ .

- Si  $\omega_1 = 0$ , entonces  $r_p(P) \equiv x = a$  y  $m_p^g = \infty$ .
  - Si  $\omega_2 = 0$ , entonces  $r_p(P) \equiv y = b$  y  $m_p^g = 0$ .
- Para  $p = \infty$ 
    - Si  $\omega_1 \cdot \omega_2 \neq 0$  entonces  $m_\infty^g = -m |m|^{-1} = -\text{signo}(m)$ , ya que  $\bar{p} = 1$ . Por tanto, independientemente del valor de  $m$ , la línea geodésica  $r_\infty(P)$  tiene pendiente unitaria con signo opuesto a  $m$  (ver figura 2.3).
    - Si  $\omega_1 \cdot \omega_2 = 0$  entonces, hay infinitas líneas geodésicas  $r_\infty(P)$  (ver figura 2.1).
  - Para  $p = 1$

- Si  $0 \leq |m| < 1$  ( $0 \leq |\omega_1| < |\omega_2|$ ), existe un único punto  $P_1 = \left( a, b - \frac{r(P)}{2\omega_2} \right)$  y  $r_1(P) \equiv x = a$ , con lo que  $m_1^g = \infty$ . Si, además,  $\omega_1 \neq 0$  será relevante en los siguientes desarrollos considerar otro punto que denotamos con  $P_0$ , con coordenadas  $\left( a - \frac{r(P)}{2\omega_1}, b \right)$  (ver figura 2.3).

- Si  $|m|=1$  ( $0 < |\omega_1| = |\omega_2|$ ), cualquier punto del segmento  $L[P_0, P_1]$ , con  $P_0$  y  $P_1$  definidos anteriormente, minimiza la distancia del punto a la recta (ver figura 2.2).
- Si  $|m| > 1$  ( $0 \leq |\omega_2| < |\omega_1|$ ), existe un único punto  $P_1 \left( a - \frac{r(P)}{2\omega_1}, b \right)$  y  $r_1(P) \equiv y = b$ , con lo que  $m_1^g = 0$ . Si, además,  $\omega_2 \neq 0$  también será relevante otro punto, que denotamos con  $P_0$ , con coordenadas  $\left( a, b - \frac{r(P)}{2\omega_2} \right)$  (ver figura 2.3).

En caso de recta vertical u horizontal, consideraremos que los puntos  $P_0$  y  $P_1$  son coincidentes y, además, coinciden con algún  $P_p$ , cualquiera que sea  $p$  (ver figura 2.1).

En caso contrario,  $r \equiv y = mx + y_0$  con  $0 \neq m \in \mathbb{R}$ , y los puntos  $P_0$  y  $P_1$ , con coordenadas  $(a, ma + y_0)$  y  $\left( \frac{b-y_0}{m}, b \right)$ , son elementos fundamentales en los siguientes desarrollos pues delimitarán las posiciones de los puntos  $P_p$ , para  $p \in [1, \infty]$ , ordenados según los valores de  $p$ , como se deduce del siguiente resultado:

**Proposición 2.10.** Sea  $P(a, b) \in \mathbb{R}^2$  un punto exterior a la recta  $r \equiv y = mx + y_0$ , con  $0 \neq m \in \mathbb{R}$ . Entonces,  $P_p \in L[P_0, P_1]$ , cualquiera que sea  $p \in [1, \infty]$ .

**Demostración.** Si  $p = 1$ , es trivial que  $P_1 \in L[P_0, P_1]$ .

En otro caso, usando la fórmula de la proposición 2.9

$$P_p = (x_p, y_p) = \left( a - \frac{r(P)}{\omega_1} \frac{1}{1 + |m|^{-p}}, b - \frac{r(P)}{\omega_2} \frac{1}{1 + |m|^{-p}} \right),$$

y teniendo en cuenta que  $\omega_1 = -m$  y  $\omega_2 = 1$ , obtenemos que:

$$x_p = a + \frac{b - ma - y_0}{m} \frac{1}{1 + |m|^{-p}},$$

ya que las abscisas de  $P_0$  y  $P_1$  son  $(a, ma + y_0)$  y  $\left(\frac{b-y_0}{m}, b\right)$ , se tiene:

$$\begin{aligned} (x_p - a) \left( x_p - \frac{b-y_0}{m} \right) &= \left( \frac{b-ma-y_0}{m} \frac{1}{1+|m|^{-\bar{p}}} \right) \left( \frac{ma-b+y_0}{m} \frac{|m|^{-\bar{p}}}{1+|m|^{-\bar{p}}} \right) = \\ &= - \left( \frac{b-ma-y_0}{m} \frac{1}{1+|m|^{-\bar{p}}} \right)^2 |m|^{-\bar{p}} \leq 0 \end{aligned}$$

luego  $x_p \in I[a, (b-y_0)/m]$ , y por tanto  $P_p \in L[P_0, P_1]$ .

Además, para  $1 \leq p < q \leq \infty$  tenemos que  $-\infty \leq -\bar{p} < -\bar{q} \leq -1$ , con lo que un exhaustivo estudio, en función de  $m$  y de  $r(P) = b - ma - y_0$ , nos permite concluir que los puntos  $P_p$  están ordenados (ver figura 2.3) según se expresa a continuación:

- Si  $|m| < 1$  y  $r(P) < 0$  ó  $|m| > 1$  y  $r(P) > 0$ , entonces  $x_p > x_q$  para  $1 \leq p < q \leq \infty$ .

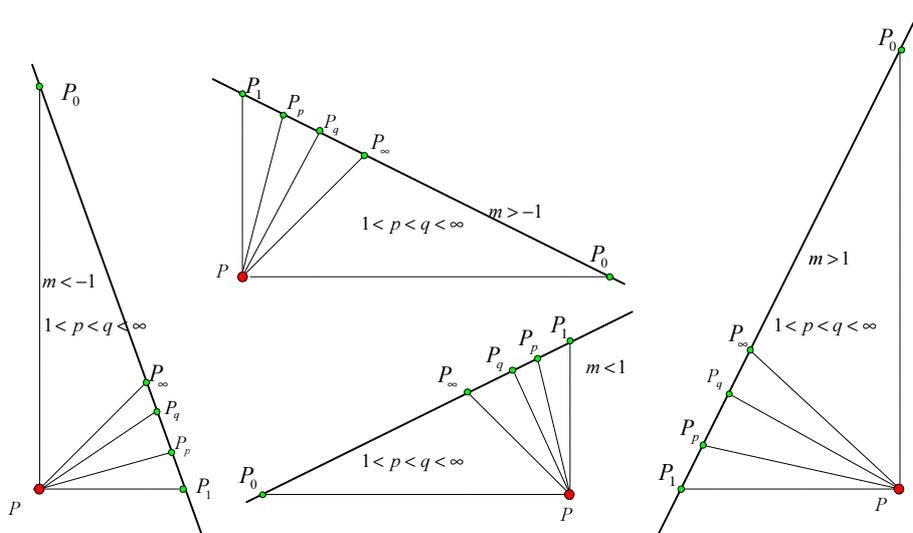


Figura 2.3: Proyección de  $P$  en  $r$ , según  $m \neq 0$ , con  $r(P) < 0$ .

- Si  $|m| < 1$  y  $r(P) > 0$  ó  $|m| > 1$  y  $r(P) < 0$ , entonces  $x_p < x_q$  para  $1 \leq p < q \leq \infty$ .
- Si  $|m| = 1$  entonces,  $x_p = x_q$  para cualesquiera  $1 \leq p < q \leq \infty$  (ver figura 2.2). ■

A partir de los resultados anteriores vamos a abordar el problema del cálculo de la distancia entre dos puntos separados por una recta que divide al plano en dos regiones con diferentes normas  $\ell_p$ .

## 2.2 Puertas de acceso (Gates) y puntos Notables

Sea  $r \equiv \omega_2 y + \omega_1 x + \omega_0 = 0$  una recta que divide al plano  $\mathbb{R}^2$  en dos regiones  $\Omega_p$  y  $\Omega_q$ , con normas  $\ell_p$  y  $\ell_q$ , respectivamente, y sean  $P(a,b) \in \Omega_p$  y  $Q(c,d) \in \Omega_q$ , con  $1 \leq p < q \leq \infty$ . El problema (2) quedó establecido en el capítulo 1 como el problema de optimización:

$$d(P,Q) = \min_{X \in r} \{d_p(P,X) + d_q(X,Q)\} \quad (2)$$

**Proposición 2.11.** Existe un punto  $X_s(x_s, y_s) \in r$  solución del problema (2). Este punto es único salvo que  $p=1$ ,  $q=\infty$  y,  $m=0$  ó  $m=\infty$ .

**Demostración.** Por la proposición 2.3,  $d_p(P,X) + d_q(X,Q)$  es un función estrictamente convexa, ya que al menos una de ellas es estrictamente convexa, y por tanto tiene un mínimo absoluto único, salvo que  $p=1$  y  $q=\infty$ . No obstante, la función  $d_1(P,X) + d_\infty(X,Q)$  también es estrictamente convexa si  $|m| \in (0, \infty)$ . ■

**Definición 2.12.** Al punto solución del problema (2), para los puntos  $P$  y  $Q$ , lo denominamos *puerta de acceso* o “gate” entre  $P$  y  $Q$ , y lo denotamos  $X_s = G(P,Q)$  ó simplemente  $G$  si no existe ambigüedad.

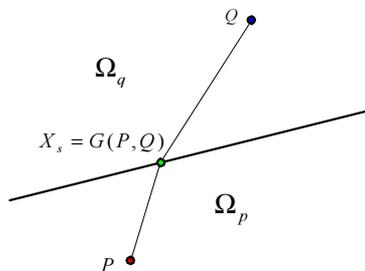


Figura 2.4: Puerta de acceso entre dos puntos de regiones con normas distintas.

**Nota 2.13.** Obsérvese que  $d_q(P, Q) \leq d_p(P, X_s) + d_q(X_s, Q) \leq d_p(P, Q)$ , según la proposición 2.2. Además, las desigualdades se convierten en igualdades si y solo si los puntos están alineados. Por tanto, si  $(a-c)(b-d) = 0$  la puerta de acceso está alineada con los dos puntos, es decir,  $X_s \in L[P, Q]$ . La figura 2.5 ilustra esta situación:

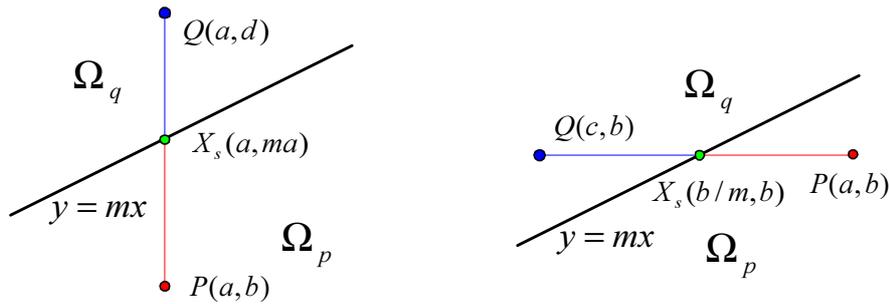


Figura 2.5: Puertas de acceso para  $L[P, Q]$  vertical y horizontal.

En el estudio geométrico que vamos a realizar a continuación, estableceremos relaciones entre los puntos  $P_0, P_1, P_p, Q_0, Q_1, Q_q$  (puntos *notables* de  $P$  y  $Q$ , respectivamente), estudiados en los desarrollos anteriores, y la puerta de acceso  $G(P, Q)$ , lo que constituye una de las aportaciones más importante de este trabajo.

Para ello, y como consecuencia de la nota 2.13, solamente consideraremos el caso  $(a-c)(b-d) \neq 0$ , y podemos suponer, sin pérdida de generalidad, que  $a < c$  y  $b < d$ , ya que esta situación puede conseguirse con un movimiento que, según la proposición 2.1 no altera las distancias. Si además hacemos una traslación o/y un giro de  $90^\circ$ , que dejan invariantes las distancias, conseguimos que la recta tenga de ecuación  $r \equiv y = mx$  y los desarrollos serán más sencillos. Esta segunda suposición será, a partir de ahora, permanente en todo el trabajo doctoral.

**Definición 2.14.** Sean  $P$  y  $Q$  puntos cualesquiera de  $\mathbb{R}^2$ . Diremos que  $P \prec Q$  ó  $Q \succ P$  si se verifica que  $a < c$  y  $b < d$ .

Geoméricamente,  $P \prec Q$  significa que  $P$  se encuentra a la izquierda y por debajo de  $Q$  y, además, si están separados por una recta  $r \equiv y = mx$ , se verifica que  $r(P) = b - ma < 0$  y  $r(Q) = d - mc > 0$ .

**Proposición 2.15.** Sea  $r \equiv y = mx$ , una recta que divide al plano  $\mathbb{R}^2$  en dos regiones  $\Omega_p$  y  $\Omega_q$ , con normas  $\ell_p$  y  $\ell_q$ , respectivamente, y sean  $P \in \Omega_p$  y  $Q \in \Omega_q$ , con  $1 \leq p < q \leq \infty$ . Si  $X_s = G(P, Q)$  entonces  $X_s \in L[P_1, Q_1] \cap L[P_0, Q_0]$ .

**Demostración.** Supongamos  $P \prec Q$ .

Si  $m > 0$  se cumple que  $b/m < a < c < d/m$  con lo que:

1. Si  $0 < m \leq 1$ , se sigue que  $P_0 \prec P_1 \prec Q_1 \prec Q_0$  (ver figura 2.3) y por tanto  $L[P_1, Q_1] \subset L[P_0, Q_0]$ . Además,

- si  $Q_1 \prec X_s$ , sea  $A$  el punto intersección de la recta que pasa por  $P$  y  $X_s$  con la recta que pasa por  $Q$  y  $Q_1$ .

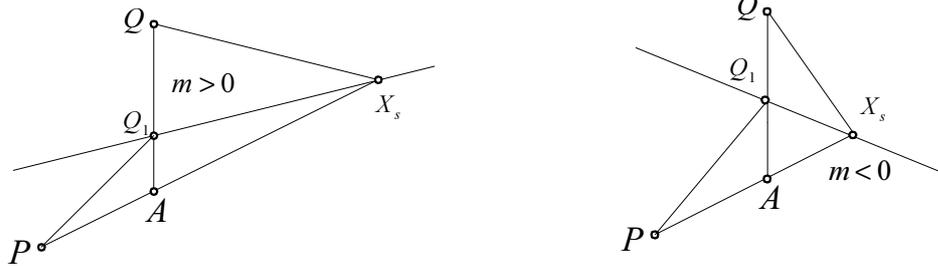


Figura 2.6: Posición inadmisibles de  $X_s$  respecto de  $Q_1$ .

Entonces, utilizando la desigualdad triangular y la proposición 2.1, tenemos:

$$d_p(P, Q_1) + d(Q_1, Q) \leq d_p(P, A) + d(A, Q_1) + d(Q_1, Q) = d_p(P, A) + d(A, Q)$$

$$\text{Además, } d(A, Q) = d_q(A, Q) \leq d_q(A, X_s) + d_q(X_s, Q) \leq d_p(A, X_s) + d_q(X_s, Q)$$

sigue que:

$$d_p(P, Q_1) + d(Q_1, Q) \leq d_p(P, A) + d_p(A, X_s) + d_q(X_s, Q) = d_p(P, X_s) + d_q(X_s, Q)$$

En contradicción con el hecho de que  $X_s$  es el único punto óptimo.

- si  $X_s \prec P_1$  la demostración sería similar al caso anterior.

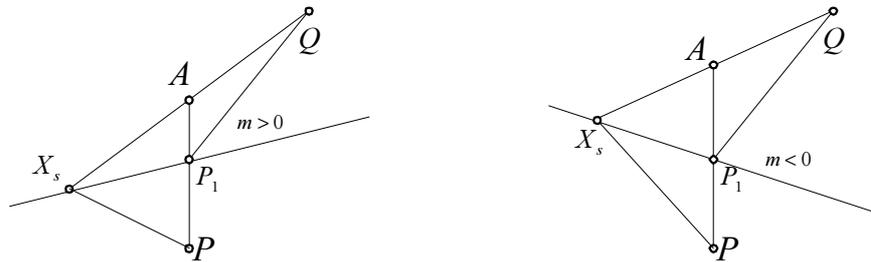


Figura 2.7: Posición inadmisble de  $X_s$  respecto de  $P_1$ .

Por tanto,  $X_s \in L[P_1, Q_1] = L[P_1, Q_1] \cap L[P_0, Q_0]$

2. Si  $1 < m$ ,  $P_1 < P_0 < Q_0 < Q_1$  (ver figura 2.3) y la demostración es análoga, intercambiando los papeles de  $P_1$  y  $P_0$ , así como  $Q_1$  y  $Q_0$ , y obtendríamos  $X_s \in L[P_0, Q_0] = L[P_1, Q_1] \cap L[P_0, Q_0]$ . La figura 2.8 ilustra las posiciones relativas de los puntos:

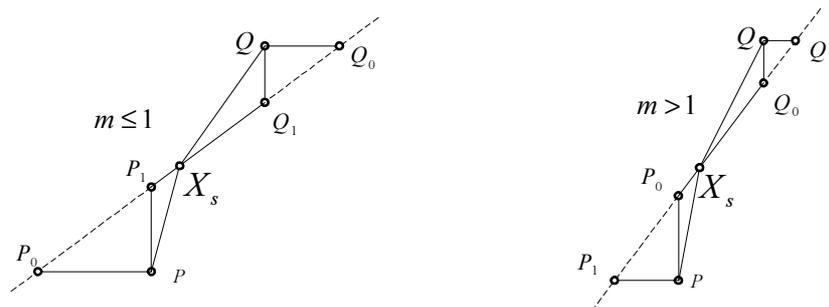


Figura 2.8: Posición relativa de  $X_s$  para  $m \geq 0$ .

3. Si  $m = 0$  ó  $m = \infty$ , la demostración anterior es válida también, pero en este caso  $X_s \in L[P_1, Q_1] \equiv L[P_0, Q_0]$  (ver figura 2.1).

Nótese que el signo de la pendiente es irrelevante en la demostración, como se indica en las figuras 2.6 y 2.7, si bien, el orden de los puntos  $\{P_1, P_0, Q_1, Q_0\}$  cambia según el valor de la pendiente  $m$ . ■

Como consecuencia geométrica de la proposición anterior obtenemos que el camino más corto entre dos puntos, estén o no en la misma región, no tiene “retroceso”, es decir: si  $(x, y) \in \overline{PX_s} \cup \overline{X_sQ}$  entonces,  $x \in I(a, c)$  e  $y \in I(b, d)$ . En particular, si  $G(P, Q) = (x_s, y_s)$  y  $P < Q$  entonces:  $a \leq x_s \leq c$  y  $b \leq y_s \leq d$ .

Además, si  $P \prec Q$  y denotamos con  $m_{pq}$ ,  $m_p$  y  $m_q$  las pendientes de los segmentos  $\overline{PQ}$ ,  $\overline{PX_s}$  y  $\overline{X_sQ}$ , respectivamente, tenemos que:

$$m_{pq} = \frac{d-b}{c-a} > 0 \quad m_p = \frac{y_s-b}{x_s-a} \geq 0 \quad m_q = \frac{y_s-d}{x_s-c} \geq 0.$$

En cualquier caso, el hecho de no tener retroceso implica que:

$$\text{signo}(m_{pq}) = \text{signo}(m_p) = \text{signo}(m_q) \text{ para } 1 < p < q \leq \infty.$$

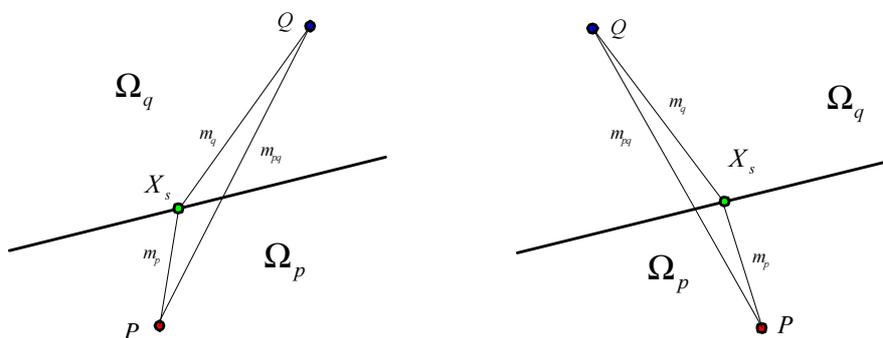


Figura 2.9: Coincidencia de signo de las pendientes.

En la sección siguiente estudiaremos que para  $1 = p < q \leq \infty$ ,  $m_p$  podría ser 0 ó  $\infty$ , permaneciendo cierta la otra igualdad.

Antes de obtener un resultado más general que el obtenido en la proposición 2.15 vamos a incluir en la acotación otros dos puntos notables.

**Proposición 2.16.** Sean  $P \in \Omega_p$  y  $Q \in \Omega_q$ , con  $1 \leq p < q \leq \infty$ . Si  $X_s = G(P, Q)$  entonces  $X_s \in L[P_p, Q_q]$ , donde  $P_p$  y  $Q_q$  son las proyecciones de  $P$  y  $Q$  sobre la recta  $r$ , según las normas  $\ell_p$  y  $\ell_q$ , respectivamente.

**Demostración.** Supongamos que  $X_s \notin L[P_p, Q_q]$ . Como  $X_s, P_p, Q_q \in r$  entonces, ó  $P_p \in L[X_s, Q_q]$  ó  $Q_q \in L[P_p, X_s]$ . Supongamos que estamos en la primera posibilidad, tal como indica la figura 2.10. Entonces:

- $d_p(P, P_p) \leq d_p(P, X_s)$  por definición de  $P_p$  y  $X_s \in r$ .
- $P_p = \lambda Q_q + (1-\lambda)X_s$  con  $\lambda \in [0, 1]$ .

- $d_q(X, Q)$  es una función convexa para  $X \in r$ , con un mínimo absoluto en  $X = Q_q$ , y por tanto:

$$d_q(P_p, Q) \leq \lambda d_q(Q_q, Q) + (1 - \lambda) d_q(X_s, Q) \leq \lambda d_q(X_s, Q) + (1 - \lambda) d_q(X_s, Q) = d_q(X_s, Q)$$

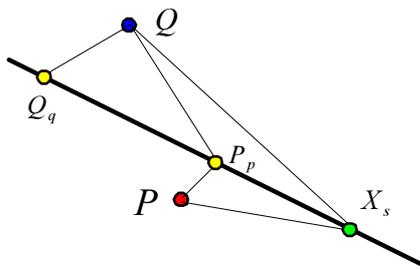


Figura 2.10: Posición inadmisibles de  $X_s$  respecto de  $P_p$  y  $Q_q$ .

De este modo,  $d_p(P, P_p) + d_q(P_p, Q) \leq d_p(P, X_s) + d_q(Q, X_s)$  y  $X_s$  no sería el único punto solución, lo que es una contradicción.

En el caso de  $Q_q \in L[P_p, X_s]$  la demostración sería análoga. ■

Nótese que el caso  $m = 0$  ó  $m = \infty$ ,  $X_s \in L[P_1, Q_1] \equiv L[P_0, Q_0] \equiv L[P_p, Q_q]$  (ver figura 2.1).

Como consecuencia de las últimas proposiciones obtenemos el siguiente resultado.

**Teorema 2.17.** Sean  $P \in \Omega_p$ ,  $Q \in \Omega_q$  y  $X_s = G(P, Q)$  el punto solución del problema (2). Si ordenamos los puntos  $P_0, P_1, P_p, Q_0, Q_1$  y  $Q_q$ , según sus abscisas, como  $P_{i_1} \ll P_{i_2} \ll P_{i_3} \ll P_{i_4} \ll P_{i_5} \ll P_{i_6}$ , entonces:  $X_s \in L[P_{i_3}, P_{i_4}]$ .

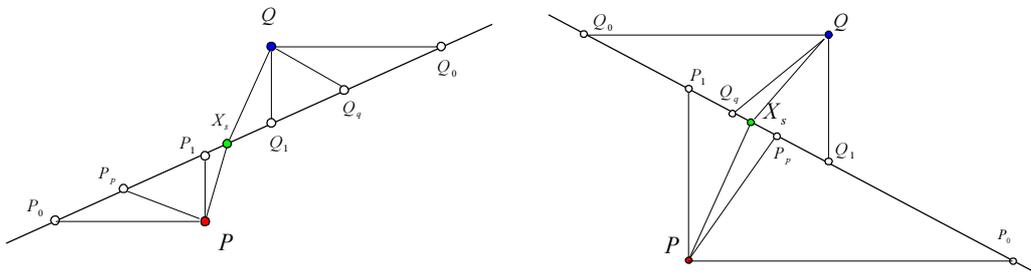


Figura 2.11: Posición relativa de la puerta de acceso, según los puntos notables.

Este teorema generaliza para cualesquiera  $p$  y  $q$ , los resultados obtenidos en Brimberg et al. (2003) y en Zaferanieh et al. (2008), ya que en estos trabajos sólo se estudia el caso  $p=1$  y  $q=2$ . Además, el segmento obtenido en la última proposición es más estricto que el obtenido en estos trabajos, en los que no tienen en cuenta que  $X_s \in L[P_p, Q_q]$ .

**Nota 2.18.** A partir de este teorema, dado  $P \in \Omega_p$  y  $r \equiv y = mx$ , la región  $\Omega_q$  puede dividirse con respecto a la localización del punto solución  $X_s$ . El caso  $-1 < m < 0$  se describe en la figura 2.12, donde  $m_p^g$  y  $m_q^g$  son las pendientes de los caminos geodésicos  $r_p(P)$  y  $r_q(Q)$ , respectivamente. En cada subregión de  $\Omega_q$  se indica el segmento en el que se encuentra  $X_s$ .

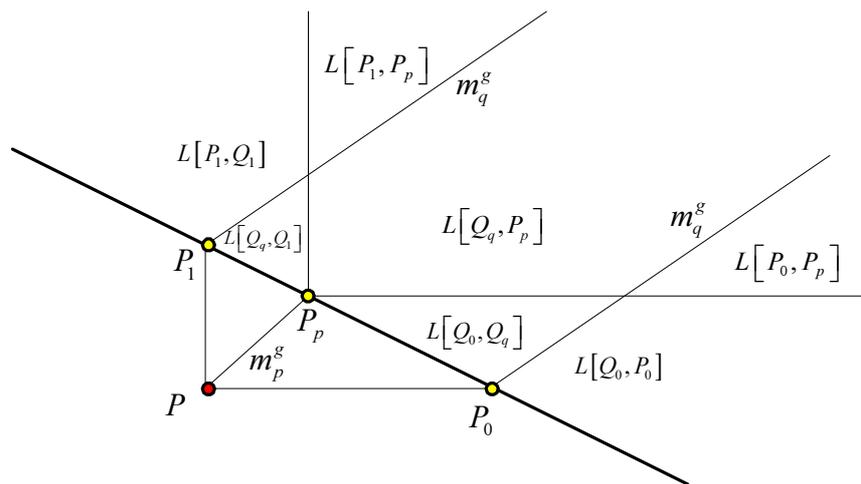


Figura 2.12: División de  $\Omega_q$  según segmento de la recta donde se encuentra la puerta de acceso.

Vamos a enunciar y demostrar otra acotación, para el caso particular de  $p=1$ , que se puede ver demostrada para  $q=2$  en Zaferanieh et al. (2008).

**Proposición 2.19.** Sean  $P \in \Omega_1$  y  $Q \in \Omega_q$ , con  $1 < q \leq \infty$ . Si  $X_s = G(P, Q)$  entonces  $X_s \in L[P_0, P_1]$ .

**Demostración.** Supongamos que  $X_s \notin L[P_0, P_1]$ . El siguiente gráfico ilustra las dos posibles posiciones de  $X_s$ :

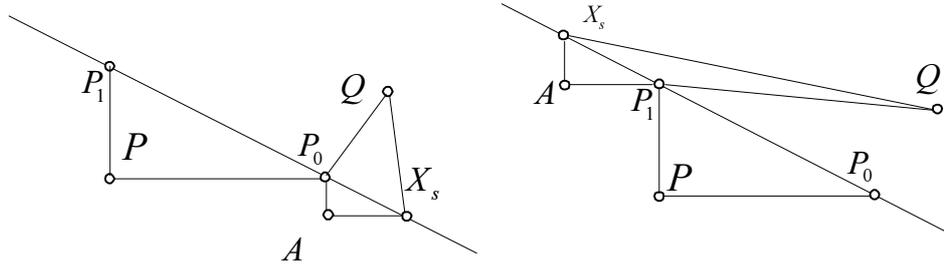


Figura 2.13: Posición inadmisibles de  $X_s$  respecto de  $P_0$  y  $P_1$ , para  $p=1$ .

Supongamos, sin pérdida de generalidad, que  $P_0 \in L[X_s, P_1]$ . Entonces:

$$\begin{aligned} d_1(P, X_s) + d_q(X_s, Q) &= d_1(P, P_0) + d_1(P_0, A) + d_1(A, X_s) + d_q(X_s, Q) \geq \\ &\geq d_1(P, P_0) + d_q(P_0, A) + d_q(A, X_s) + d_q(X_s, Q) \geq d_1(P, P_0) + d_q(P_0, Q) \end{aligned}$$

Con lo que  $X_s$  no sería el único punto óptimo.

Análogamente si  $P_1 \in L[X_s, P_0]$ . ■

## 2.3 Estudio analítico de las puertas de acceso

Las propiedades de las puertas de acceso estudiadas en la sección anterior, han sido demostradas de forma geométrica, aunque podríamos demostrarlas de forma analítica. En esta sección vamos a caracterizar estas puertas de acceso de forma analítica en algunos casos especiales que se presentan para normas comunes como la rectangular ( $\ell_1$ ), euclídea ( $\ell_2$ ) y la norma del supremo ( $\ell_\infty$ ). En el caso general no es posible obtener una expresión en cuadratura de las coordenadas de las puertas de acceso, pero encontraremos expresiones con las que obtenerlas a partir de métodos de análisis numérico y los implementaremos utilizando el programa informático MATLAB.

Sean  $P(a, b) \in \Omega_p$  y  $Q(c, d) \in \Omega_q$ , con  $1 \leq p < q < \infty$ , puntos separados por la recta  $r \equiv y = mx$ .

Para encontrar el punto de acceso hemos de minimizar, para  $X \in r$ , la función:

$$D(X) = d(P, Q) = d_p(P, X) + d_q(X, Q) = \sqrt[p]{|x-a|^p + |y-b|^p} + \sqrt[q]{|x-c|^p + |y-d|^p},$$

que se puede escribir, a partir de  $y = mx$ , como,

$$D(X) \equiv d(x) = \sqrt[p]{|x-a|^p + |mx-b|^p} + \sqrt[q]{|x-c|^q + |mx-d|^q}.$$

Esta función es derivable salvo, a lo sumo, en el conjunto  $A = \{a, c, b/m, d/m\}$ , que contiene las abscisas de los puntos  $P_0, P_1, Q_0$  y  $Q_1$ . Si  $x \notin A$  la función tiene como derivada:

$$d'(x) = \frac{\operatorname{sgn}(x-a)|x-a|^{p-1} + \operatorname{sgn}(mx-b)|mx-b|^{p-1} m}{\left(\sqrt[p]{|x-a|^p + |mx-b|^p}\right)^{p-1}} + \frac{\operatorname{sgn}(x-c)|x-c|^{q-1} + \operatorname{sgn}(mx-d)|mx-d|^{q-1} m}{\left(\sqrt[q]{|x-c|^q + |mx-d|^q}\right)^{q-1}},$$

donde  $\operatorname{sgn}$  es la función signo, es decir,  $\operatorname{sgn}(u) = \begin{cases} 1 & \text{si } u > 0 \\ 0 & \text{si } u = 0 \\ -1 & \text{si } u < 0 \end{cases}$

Este reducido análisis nos permite enunciar el siguiente resultado:

**Teorema 2.20.** Sean  $P(a,b) \in \Omega_p$  y  $Q(c,d) \in \Omega_q$ , puntos de regiones con normas  $\ell_p$  y  $\ell_q$ , respectivamente, separados por la recta  $r \equiv y = mx$ . Entonces, la puerta de acceso  $G(P,Q) = X_s(x_s, y_s) \in r$  es, o bien  $x_s = \alpha$ , o bien  $x_s = \beta$ , donde  $\alpha$  y  $\beta$  son los extremos del intervalo  $I[a,c] \cap I[b/m, d/m]$ , o bien es el único punto que verifica el siguiente sistema de ecuaciones:

$$\begin{cases} y_s = mx_s \\ y_s - b = m_p(x_s - a) \\ y_s - d = m_q(x_s - c) \\ \frac{1 + \operatorname{sgn}(m_p)|m_p|^{p-1} m}{\left(\sqrt[p]{1 + |m_p|^p}\right)^{p-1}} = \frac{1 + \operatorname{sgn}(m_q)|m_q|^{q-1} m}{\left(\sqrt[q]{1 + |m_q|^q}\right)^{q-1}} \end{cases}$$

**Demostración.** Por la proposición 2.15,  $X_s \in L[P_1, Q_1] \cap L[P_0, Q_0]$  y por tanto  $x_s \in I[a,c] \cap I[b/m, d/m] = [\alpha, \beta]$ . Por la convexidad de la función objetivo:

- Si  $d'(x) \geq 0$  en  $(\alpha, \beta)$ , entonces:  $x_s = \alpha$ .
- Si  $d'(x) \leq 0$  en  $(\alpha, \beta)$ , entonces:  $x_s = \beta$ .
- En caso contrario, para  $x \in (\alpha, \beta) \subset I(a, c)$ :

$$d'(x) = \frac{\operatorname{sgn}(x-a) + \operatorname{sgn}(mx-b) |m_p|^{p-1} m}{\left(\sqrt[p]{1+|m_p|^p}\right)^{p-1}} + \frac{\operatorname{sgn}(x-c) + \operatorname{sgn}(mx-d) |m_q|^{q-1} m}{\left(\sqrt[q]{1+|m_q|^q}\right)^{q-1}},$$

$$\text{donde } m_p = \frac{mx-b}{x-a} \text{ y } m_q = \frac{mx-d}{x-c}.$$

Dado que  $x \in I(a, c)$ , entonces  $(x-a)(x-c) < 0$  y sigue que:

$$d'(x) = \operatorname{sgn}(x-a) \left( \frac{1 + \operatorname{sgn}(m_p) |m_p|^{p-1} m}{\left(\sqrt[p]{1+|m_p|^p}\right)^{p-1}} - \frac{1 + \operatorname{sgn}(m_q) |m_q|^{q-1} m}{\left(\sqrt[q]{1+|m_q|^q}\right)^{q-1}} \right).$$

$$\text{Luego, } d'(x) = 0 \text{ si y solo si } \frac{1 + \operatorname{sgn}(m_p) |m_p|^{p-1} m}{\left(\sqrt[p]{1+|m_p|^p}\right)^{p-1}} = \frac{1 + \operatorname{sgn}(m_q) |m_q|^{q-1} m}{\left(\sqrt[q]{1+|m_q|^q}\right)^{q-1}}, \quad (4)$$

$$\text{en cuyo caso, } m_p = \frac{y_s - b}{x_s - a}, \quad m_q = \frac{y_s - d}{x_s - c} \text{ e } y_s = mx_s. \quad \blacksquare$$

En los desarrollos anteriores hemos utilizado la terminología “puerta de acceso” para denominar el punto de cruce con la recta de separación, de la trayectoria más corta entre dos puntos situados en regiones separadas por dicha recta. Inicialmente, hay una puerta de acceso diferente para cada par de puntos  $P$  y  $Q$ . Realmente, hay una puerta diferente para cada par de direcciones  $m_p$  y  $m_q$ , solución de la ecuación (4), lo que trae consigo que haya una infinidad de puntos en  $\Omega_p$  cuya trayectoria geodésica a una infinidad de puntos en  $\Omega_q$  corta a la recta en la misma puerta de acceso, como pone de manifiesto el siguiente resultado:

**Proposición 2.21.** Con las hipótesis del teorema 2.20, para cualesquiera puntos  $P' = \lambda P + (1-\lambda)X_s \in \Omega_p$  y  $Q' = \mu Q + (1-\mu)X_s \in \Omega_q$ , con  $\lambda, \mu \geq 0$ , se verifica que  $X_s = G(P', Q')$ .

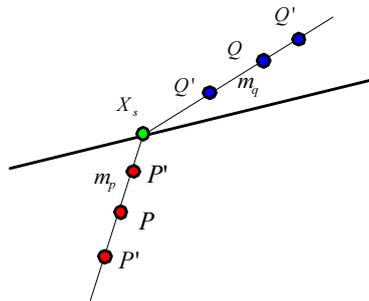


Figura 2.14: Portón común para rayos con pendientes  $m_p$  y  $m_q$ .

**Demostración.** Por la definición de los puntos  $P'$  y  $Q'$ , se tiene que  $m_{\overline{PX_s}} = m_{\overline{P'X_s}} = m_p$  y  $m_{\overline{QX_s}} = m_{\overline{Q'X_s}} = m_q$  (ver figura 2.14), luego el mismo punto  $X_s$  verifica las condiciones de primer orden del teorema 2.20, que son suficientes dado que la función  $d(x)$  es convexa. ■

Si bien esta infinidad de puntos es de medida nula, en la sección 2.3.1 vamos a demostrar que para determinados valores de  $p$  y  $q$  esta infinidad tiene medida positiva y cambiaremos la terminología de puerta de acceso por la de “portón”, en el sentido de punto de acceso de un punto de una región a muchos puntos de la otra región. De hecho, en el diccionario de la RAE se define portón como puerta de acceso del zaguán (sala única) al resto de estancias de la casa.

Nos interesa para los siguientes desarrollos estudiar la función que se obtiene de la fórmula (4), haciendo  $k = \text{sgn}(m_p)m$  y variable no negativa  $x = |m_p|$

$$f_p(x) = \frac{1 + kx^{p-1}}{\left(\sqrt[p]{1 + x^p}\right)^{p-1}}, \quad x \geq 0, \quad p > 1.$$

**Proposición 2.22.** La función  $f_p$  tiene las siguientes propiedades:

- 1) Si  $k \leq 0$ , entonces  $f_p$  es monótona estrictamente decreciente, con valores en el intervalo  $(k, 1]$  (ver figura 2.15).

- 2) Si  $k > 0$ ,  $f_p$  toma valores en el intervalo  $\left(k, \sqrt[p]{1+k^p}\right]$ , tiene un máximo en  $x = k$  y es monótona estrictamente decreciente para  $x > k$ . Además, existe un único  $x_0 > 0$  tal que  $f_p(x_0) = 1$  y  $f_p(x) < 1$  para  $x > x_0$  (ver figura 2.16).
- 3) Si  $k > 0$  y  $q > p$ , existen  $x_1$  y  $x_2$ , tales que  $0 < x_1 < x_2$  y  $f_p(x_1) = f_p(x_2) = \sqrt[q]{1+k^q}$ .

**Demostración.** Es inmediata a partir de que  $f_p'(x) = \frac{(p-1)x^{p-2}}{\left(\sqrt[p]{1+x^p}\right)^{2p-1}}(k-x)$  y de que

$$\lim_{x \rightarrow +\infty} f_p(x) = k.$$

Además, si  $1 < p < q$ , entonces  $f_p(0) = 1 < \sqrt[q]{1+k^q} < \sqrt[p]{1+k^p} = \max_{x \geq 0} f_p(x)$  y la función alcanza el valor  $\sqrt[q]{1+k^q}$  en dos puntos distintos de  $\mathbb{R}^+$ .

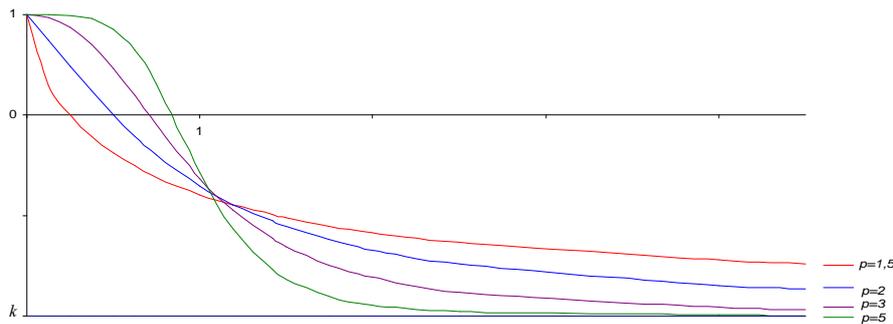


Figura 2.15: Gráfica de  $f_p$  para  $k < 0$ .

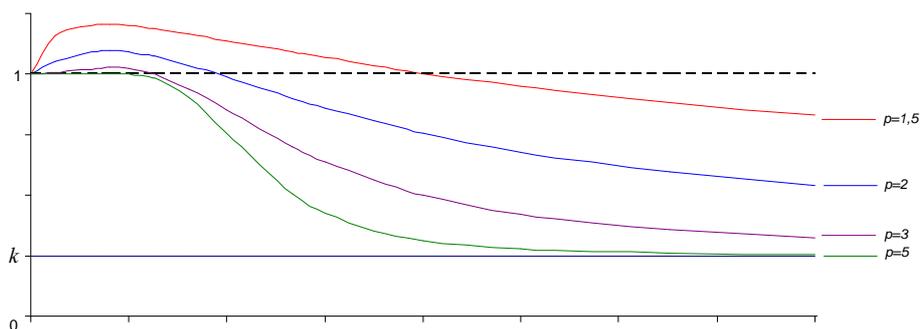


Figura 2.16: Gráfica de  $f_p$  para  $k > 0$ .

Cuando la recta tiene pendiente  $m$ , con  $|m| < 1$ , el punto  $P_1$  es la proyección vertical de  $P$ , es decir,  $(a, ma)$  y  $P_0$ , de existir, es la proyección horizontal  $(b/m, b)$ . Sin embargo,

### 2.3.1 El caso de normas $\ell_1$ y $\ell_q$

si la pendiente es tal que  $|m| > 1$  las posiciones de ambos puntos notables se intercambian. Por ello, y para simplificar, vamos a cambiar la notación y denotaremos a los puntos notables como  $P_V = (a, ma)$  y  $P_H = (b/m, b)$ , y de forma análoga los puntos  $Q_V = (c, mc)$  y  $Q_H = (d/m, d)$ , todos ellos pertenecientes a la recta  $r \equiv y = mx$ .

### 2.3.1 El caso de normas $\ell_1$ y $\ell_q$

**Teorema 2.23.** Sean  $P(a, b) \in \Omega_1$  y  $Q(c, d) \in \Omega_q$ , puntos de regiones con normas  $\ell_1$  y  $\ell_q$ , respectivamente, con  $q \in (1, \infty)$ , separados por la recta  $r \equiv y = mx$ . Entonces, la puerta de acceso  $G(P, Q) = X_s(x_s, y_s) \in r$  es, o bien  $(a, ma)$ , o bien  $(b/m, b)$ , o bien

$\left( \frac{m^*c - d}{m^* - m}, \frac{m(m^*c - d)}{m^* - m} \right)$  donde  $m^*$  es la única solución de la ecuación:

$$1 - |m| = \frac{1 - |m| |m^*|^{q-1}}{\sqrt[q]{(1 + |m^*|^q)^{q-1}}}, \text{ con } m \cdot m^* < 0.$$

#### Demostración.

- Si  $m = 0$ , usando (4):

$$d'(x) = \text{sgn}(x - a) \left( 1 - \frac{1}{\left( \sqrt[q]{1 + |m_q|^q} \right)^{q-1}} \right) \Rightarrow d'(a^+) \cdot d'(a^-) < 0 \Rightarrow X_s = P_V = (a, ma),$$

(ver figura 2.17).

Análogamente si la recta fuese vertical.

- Si  $m > 0$ , el punto  $P$  determina tres regiones en  $\Omega_q$ , según se indica en la figura 2.18.

Por la proposición 2.15,  $X_s \in L[P_H, Q_H] \cap L[P_V, Q_V]$ . Además, según la proposición 2.19,  $X_s \in L[P_H, P_V]$  (ver también Zaferanieh et al. (2008)).

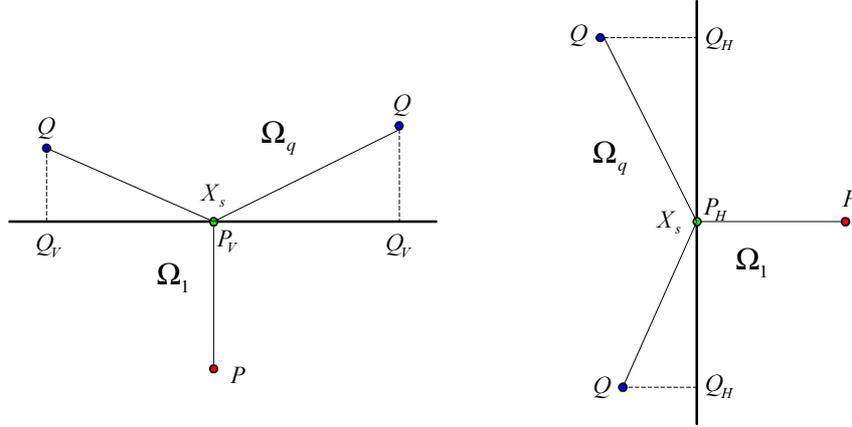


Figura 2.17: Portón común para rectas horizontales y verticales con normas  $\ell_1$  y  $\ell_q$ .

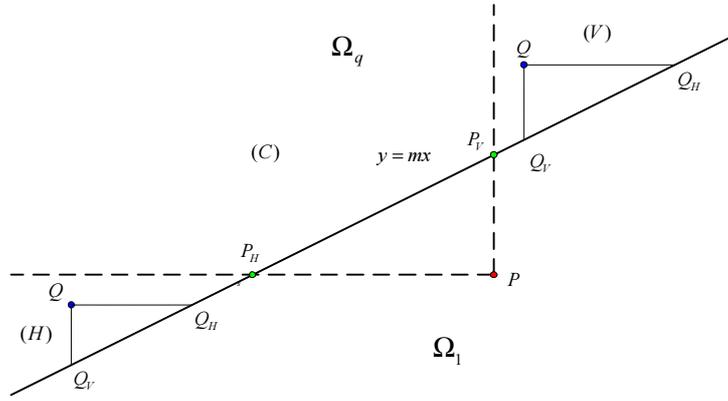


Figura 2.18: Regiones iniciales en  $\Omega_q$  determinadas por  $P \in \Omega_1$ .

Si  $Q \in (V)$ , entonces  $L[P_H, Q_H] \cap L[P_v, Q_v] \cap L[P_H, P_v] = \{P_v\}$  y por tanto:  
 $X_s = P_v = (a, ma)$ .

Si  $Q \in (H)$ , entonces  $L[P_H, Q_H] \cap L[P_v, Q_v] \cap L[P_H, P_v] = \{P_H\}$  y por tanto:  
 $X_s = P_H = (b/m, b)$ .

Si  $Q \in (C)$  y  $P_v \neq X_s \neq P_H$ , entonces  $\text{sgn}(m_p) = \text{sgn}(m_q) = -1$ , y usando (4), el punto solución verifica:

$$d'(x) = 1 + \text{sgn}(m_p)m - \frac{1 + \text{sgn}(m_q)|m_q|^{q-1}m}{\left(\sqrt[q]{1 + |m_q|^q}\right)^{q-1}} = 1 - |m| - \frac{1 - |m_q|^{q-1}|m|}{\left(\sqrt[q]{1 + |m_q|^q}\right)^{q-1}} = 0$$

Por la proposición 2.22,  $m_q = m^*$  es la solución única de la ecuación:

$$\frac{1 - |m| \cdot |m^*|^{q-1}}{\sqrt[q]{(1 + |m^*|^q)^{q-1}}} = 1 - |m| \text{ con } m \cdot m^* < 0.$$

- Si  $m < 0$ , mediante una simetría axial de eje  $x = a$  tendremos una recta con pendiente positiva en la que son válidos los razonamientos anteriores.

Además, en dicho caso  $\text{sgn}(m_p) = \text{sgn}(m_q) = 1$  y usando (4):

$$d'(x) = 1 + \text{sgn}(m_p)m - \frac{1 + \text{sgn}(m_q)|m_q|^{q-1}m}{\left(\sqrt[q]{1 + |m_q|^q}\right)^{q-1}} = 1 - |m| - \frac{1 - |m_q|^{q-1}|m|}{\left(\sqrt[q]{1 + |m_q|^q}\right)^{q-1}} = 0$$

con  $m \cdot m^* < 0$ . ■

**Nota 2.24. (Posición del portón según  $Q$ )** Fijado  $P \in \Omega_1$ , para determinar el portón (gate o puerta de acceso)  $X_s$  para  $Q \in \Omega_q$ , hemos de proyectar el punto  $Q$  en la recta  $r$ , según la pendiente  $m_q = m^*$ . Para ello, consideremos las rectas  $r_H \equiv y = m^*(x - b/m) + b$  y  $r_V \equiv y = m^*(x - a) + ma$ , con pendiente  $m^*$ , que pasan por los puntos  $P_H$  y  $P_V$ , respectivamente (ver figura 2.19).

De esta manera, si:

- $Q$  se encuentra en la región  $\Omega_{qV}$ , definida por la recta  $r$  y la recta  $r_V$ , la proyección  $Q^* = (c^*, d^*) = \left( \frac{m^*c - d}{m^* - m}, \frac{m(m^*c - d)}{m^* - m} \right)$  en  $r$ , según  $m^*$ , no es solución admisible porque  $Q^* \notin L[P_H, P_V]$ . Como  $d'(c^*) = 0$  y la función es convexa,  $d'(x) < 0$  para  $x < c^*$ , luego el mínimo se obtiene en el mayor valor admisible, es decir, el portón es  $P_V$ .
- $Q$  se encuentra en la región  $\Omega_{qH}$ , definida por la recta  $r$  y la recta  $r_H$ , la proyección  $Q^* = (c^*, d^*)$ , según  $m^*$ , no es solución admisible porque  $Q^* \notin L[P_H, P_V]$ . Como  $d'(c^*) = 0$  y la función es convexa,  $d'(x) > 0$  para  $x > c^*$ , luego el mínimo se obtiene en el menor valor admisible, es decir, el portón es  $P_H$ .

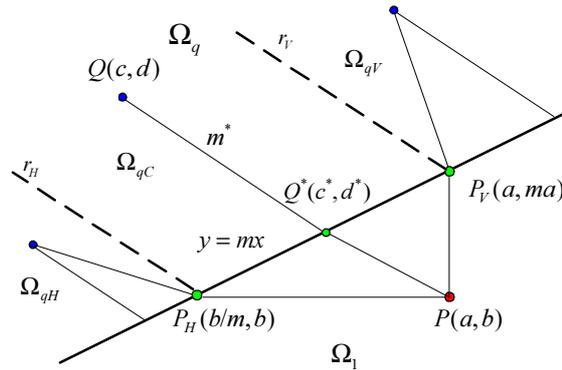


Figura 2.19: Regiones en  $\Omega_q$  definidas por  $P \in \Omega_1$ .

- $Q$  se encuentra en la banda  $\Omega_{qC}$ , definida por las rectas  $r$ ,  $r_H$  y  $r_V$ , el camino más corto es la poligonal  $\overline{PX_sQ}$ , con  $X_s = Q^*$ .

Obsérvese que si  $Q \in r_V$  es indiferente considerar que  $Q \in \Omega_{qV}$  o que  $Q \in \Omega_{qC}$  ya que, en el primer caso el portón admisible es  $P_V$  y en el segundo caso  $d'(c^*) = 0$  con  $c^* = a$ .

No obstante, en lo sucesivo  $r_V$  no se considera incluida en  $\Omega_{qV}$ .

Análoga observación si  $Q \in r_H$  y también se considera que  $r_H$  no se incluye en  $\Omega_{qH}$ . ■

**Nota 2.25. (Posición del portón según  $P$ )** Si ahora fijamos el punto  $Q \in \Omega_q$ , su proyectado en  $r$  según  $m^*$ ,  $Q^* = (c^*, d^*)$ , determina tres zonas en  $\Omega_1$ . Para delimitar estas zonas, consideramos las semirrectas  $r_H \equiv y = d^*$  y  $r_V \equiv x = c^*$  en  $\Omega_1$ , según ilustra la figura 2.20.

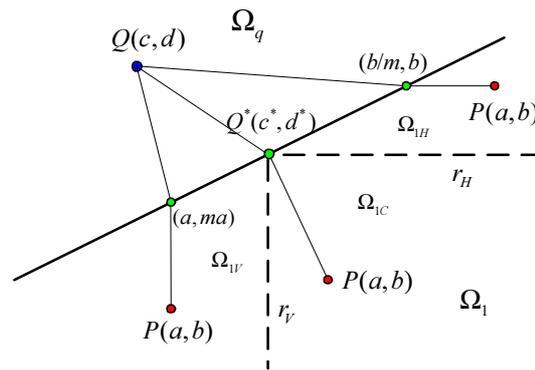


Figura 2.20: Regiones en  $\Omega_1$  definidas por  $Q \in \Omega_q$ .

Procediendo de esta forma, si:

2.3.1 El caso de normas  $\ell_1$  y  $\ell_q$

- $P$  se encuentra en la región  $\Omega_{IV}$ , definida por la recta  $r$  y la recta  $r_V$ , existe un portón  $P_V(a, ma)$  para cada punto.
- $P$  se encuentra en la región  $\Omega_{IH}$ , definida por la recta  $r$  y la recta  $r_H$ , existe un portón  $P_H(b/m, b)$  para cada punto.
- $P$  se encuentra en el cono  $\Omega_{IC}$ , definido por las rectas  $r_H$  y  $r_V$ , el camino más corto es la poligonal  $\overline{PX_sQ}$ , con  $X_s = Q^*$ .

Obsérvese que si  $P \in r_V$ , es indiferente considerar que  $P \in \Omega_{IV}$  o que  $P \in \Omega_{IC}$  ya que, en el primer caso el portón admisible es  $P_V$  y en el segundo caso  $d'(c^*) = 0$  con  $c^* = a$ .

Análoga consideración si  $P \in r_H$ . ■

**Corolario 2.26.** En las condiciones del teorema 2.23, si  $q = 2$ , entonces el portón entre  $P$  y  $Q$  es, o bien  $(a, ma)$ , o bien  $(b/m, b)$ , o bien  $\left( \frac{m^*c - d}{m^* - m}, \frac{m(m^*c - d)}{m^* - m} \right)$  donde

$$m^* = -\frac{m}{|m|} \frac{\sqrt{|2m|}(\sqrt{|2m|} + 2)}{2(\sqrt{|2m|} + 1)}.$$

**Demostración.** Hemos de resolver la ecuación  $\frac{1 - |m| |m^*|^{q-1}}{\sqrt[q]{(1 + |m^*|^q)^{q-1}}} = 1 - |m|$  para  $q = 2$ .

Denotando con  $k = -|m|$  y  $x = |m^*|$  la ecuación resultante queda:  $\frac{1 + kx}{\sqrt{(1 + x^2)}} = 1 + k$ ,

que tendrá solución única según la proposición 2.22.

$$(1+k)^2(1+x^2) = (1+kx)^2 \Rightarrow (2k+1)x^2 - 2kx + k(k+2) = 0,$$

$$\text{si } k = -\frac{1}{2} \Rightarrow x = \frac{k(k+2)}{2k} = \frac{3}{4},$$

$$\text{si } k \neq -\frac{1}{2} \Rightarrow x = \frac{k \pm (k+1)\sqrt{-2k}}{2k+1}.$$

De las dos soluciones posibles sólo  $x = \frac{k + (k+1)\sqrt{-2k}}{2k+1}$  satisface la ecuación.

$$\text{Además: } x = \frac{(k + (k+1)\sqrt{-2k})(k - (k+1)\sqrt{-2k})}{(2k+1)(k - (k+1)\sqrt{-2k})} = \frac{k(k+2)}{(k - (k+1)\sqrt{-2k})}.$$

$$\text{Si hacemos } k = -2t^2, \text{ tenemos } x = \frac{4t^4 - 4t^2}{4t^3 - 2t^2 - 2t} = \frac{2t(t+1)}{2t+1} = \frac{\sqrt{-2k}(\sqrt{-2k}+2)}{2(\sqrt{-2k}+1)}.$$

$$\text{Deshaciendo el cambio, } |m^*| = \frac{\sqrt{|2m|}(\sqrt{|2m|}+2)}{2(\sqrt{|2m|}+1)}.$$

Obsérvese que la solución obtenida, a diferencias del resultado obtenido por Zaferanieh et al. (2008), es válida siempre, ya que el denominador no se anula nunca.

$$\text{Como } m \cdot m^* < 0, \quad m^* = -\frac{m}{|m|}|m^*| = -\frac{m}{|m|} \frac{\sqrt{|2m|}(\sqrt{|2m|}+2)}{2(\sqrt{|2m|}+1)}.$$

Nótese que la fórmula dada para  $m^*$  no presenta ambigüedad, ya que si  $m = 0$  el portón es  $(a, 0)$  y si la recta es vertical el portón es  $(0, b)$ , según el teorema 2.23. ■

**Corolario 2.27.** En las condiciones del teorema 2.23, si  $q = \infty$ , entonces el portón entre

$P$  y  $Q$  es, o bien  $(a, ma)$ , o bien  $(b/m, b)$ , o bien  $Q_\infty = \left( \frac{m^*c - d}{m^* - m}, \frac{m(m^*c - d)}{m^* - m} \right)$  donde

$$m^* = -\frac{m}{|m|}.$$

**Demostración.**

- Supongamos en primer lugar que  $m > 0$  y dividamos la región  $\Omega_\infty$  en tres zonas según las rectas  $r_H$  y  $r_V$ , que pasan por los puntos  $P_H$  y  $P_V$ , respectivamente, definidas por las ecuaciones:  $r_H \equiv y = -(x - b/m) + b$  y  $r_V \equiv y = -(x - a) + ma$ .

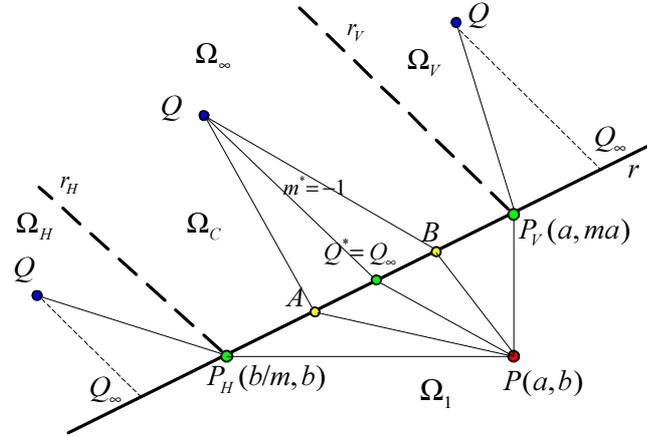


Figura 2.21: Regiones en  $\Omega_\infty$  definidas por  $P \in \Omega_1$ .

Entonces, si

- $Q$  se encuentra en la región  $\Omega_C$ , definida por las rectas  $r$ ,  $r_H$  y  $r_V$ , la proyección  $Q^* = \left( \frac{m^*c - d}{m^* - m}, \frac{m(m^*c - d)}{m^* - m} \right)$  en  $r$ , según  $m^* = -1$ , es el punto  $Q_\infty$  y, además, vamos a probar que es el portón entre  $P$  y  $Q$ .

Para ello consideremos la función objetivo:

$$d(x) = |x - a| + |mx - b| + \max\{|x - c|, |mx - d|\}$$

y cualesquiera puntos  $A$  y  $B$  tales que  $A \prec Q_\infty \prec B$  (ver figura 2.21).

- En  $A$ ,  $d(x) = a - x + mx - b + d - mx \Rightarrow d'(x) = -1 < 0$
- En  $B$ ,  $d(x) = a - x + mx - b + x - c \Rightarrow d'(x) = m > 0$ . Por tanto,  $X_s = Q_\infty$ .
- $Q$  se encuentra en la región  $\Omega_V$ , definida por la recta  $r$  y la recta  $r_V$ , entonces,  $Q^* = Q_\infty \notin L[P_V, Q_V]$  y no puede ser el punto solución.

Si  $c \geq a$ , entonces  $a \leq x_s \leq c$ , pero  $X_s \in L[P_H, P_V]$ , luego  $X_s = P_V$ .

Si  $c < a$ , entonces  $c \leq x_s \leq a$  luego  $d'(x_s) = -1 + m - m < 0$  y, por la convexidad de la función objetivo,  $x_s$  ha de ser el mayor valor admisible, es decir:  $X_s = P_V$

- $Q$  se encuentra en la región  $\Omega_H$ , definida por la  $r$  y la recta  $r_H$ , con un razonamiento análogo, obtenemos que  $X_s = P_H$ .
- La demostración para  $m < 0$  consiste en realizar una simetría axial de eje  $x = a$ , con lo que se tiene una recta con pendiente positiva en la que son válidos los razonamientos anteriores. Tras deshacer la simetría, tenemos que  $m^* = 1$ .

Nótese nuevamente que la fórmula dada para  $m^*$  no presenta ambigüedad, ya que si  $m = 0$  un portón es  $(a, 0)$  y si la recta es vertical un portón es  $(0, b)$  (ver teorema 2.23). ■

### 2.3.2 El caso de normas $\ell_p$ y $\ell_\infty$

**Teorema 2.28.** Sean  $P(a, b) \in \Omega_p$  y  $Q(c, d) \in \Omega_\infty$ , puntos de regiones con normas  $\ell_p$  y  $\ell_\infty$ , respectivamente,  $p > 1$ , separados por la recta  $r \equiv y = mx$ . Entonces, la puerta de acceso  $G(P, Q) = X_s(x_s, y_s) \in r$  es, o bien  $(a, ma)$ , o bien  $(b/m, b)$ , o bien

$$\left( \frac{m_q^* c - d}{m_q^* - m}, \frac{m(m_q^* c - d)}{m_q^* - m} \right), \text{ o bien } \left( \frac{m_p^* a - b}{m_p^* - m}, \frac{m(m_p^* a - b)}{m_p^* - m} \right) \text{ donde } m_q^* = \text{sgn}(m_{pq}) \text{ y } m_p^* \text{ es}$$

la única solución de la ecuación:

$$\frac{1 + |m| |m_p^*|^{p-1}}{\sqrt[p]{(1 + |m_p^*|^p)^{p-1}}} = 1, \text{ con } \text{sgn}(m_p^*) = \text{sgn}(m). \quad (5)$$

**Demostración.** En este caso la función objetivo es:

$$d(x) = \sqrt[p]{|x-a|^p + |mx-b|^p} + \max\{|x-c|, |mx-d|\}$$

$$d'(x) = \text{sgn}(x-a) \left( \frac{1 + \text{sgn}(m_p) |m_p|^{p-1} m}{\left( \sqrt[p]{1 + |m_p|^p} \right)^{p-1}} \right) + \delta = \text{sgn}(x-a) f(\alpha) + \delta,$$

2.3.2 El caso de normas  $\ell_p$  y  $\ell_\infty$

donde  $\delta = \frac{d}{dx} \max\{|x-c|, |mx-d|\}$ , con  $\delta \in \{\pm 1, \pm m\}$  y  $f(\alpha) = \frac{1+k\alpha^{p-1}}{(\sqrt[p]{1+\alpha^p})^{p-1}}$ , con

$\alpha = |m_p|$  y  $k = \text{sgn}(m_p)m$ .

Además, por la proposición 2.22, 
$$\begin{cases} f(\alpha) \in (k, 1] & \text{si } k \leq 0 \\ f(\alpha) \in \left(k, \sqrt[p]{1+k^p}\right] & \text{si } k \geq 0 \end{cases} \quad (6)$$

De aquí se sigue que:

- si  $0 < m \leq 1$ , consideremos  $r_H \equiv y = -(x-b/m)+b$ ,  $r_V \equiv y = -(x-a)+ma$ ,  $r_{V1} \equiv y = (x-a)+ma$  y  $r_{V2} \equiv y = (x-x^*)+mx^*$ , la rectas que delimitan las regiones  $\Omega_H$ ,  $\Omega_C$ ,  $\Omega_V$ ,  $\Omega_B$  y  $\Omega_{V'}$  (ver figura 2.22).

Vamos a estudiar la ubicación de los portones según donde se encuentre  $Q$ .

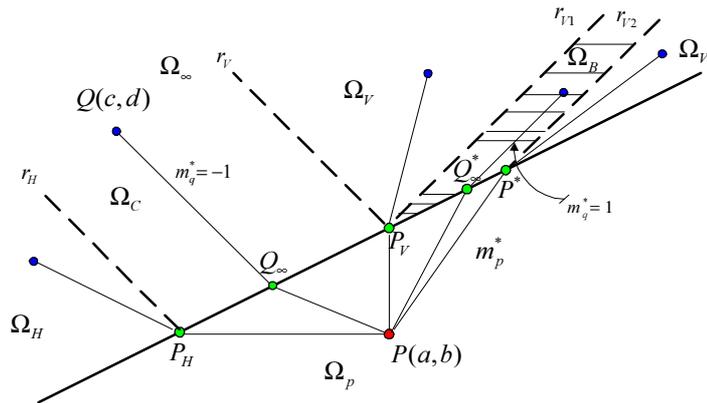


Figura 2.22: Regiones en  $\Omega_\infty$  definidas por  $P \in \Omega_p$ .

- Si  $Q \in \Omega_H$ , entonces, para  $(x, mx) \in L[P_H, Q_H]$ , se tiene que  $\delta = 1$  y  $d'(x) = -f(\alpha) + 1$ .

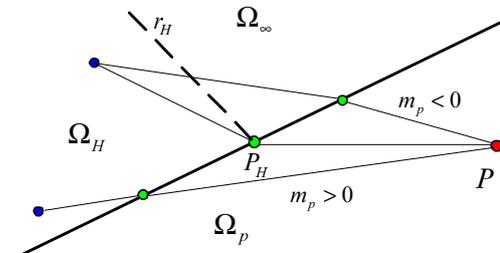


Figura 2.23: Portón para  $Q \in \Omega_H \subset \Omega_\infty$  y  $P \in \Omega_p$ .

Si  $m_p > 0$ , entonces  $0 < k < f(\alpha)$ , luego,  $d'(x) < -k + 1 = -m \operatorname{sgn}(m_p) - 1 < 0$ .

Si  $m_p < 0$ , entonces  $k < f(\alpha) \leq 1$ , y por tanto,  $d'(x) \geq -1 + 1 = 0$ .

Por tanto, la función objetivo  $d(x)$  alcanza el mínimo para un valor de  $x$  tal que  $m_p = 0$ , lo que implica que  $X_s = P_H$ .

- Si  $Q \in \Omega_C$ , entonces, para  $(x, mx) \in L[P_V, Q_V] \cap L[P_H, Q_H]$ , se tiene que  $c \leq x \leq a$  y por tanto:  $d'(x) = -f(\alpha) + \delta$ , con  $\delta \in \{1, -m\}$ .

Consideremos cualesquiera puntos  $A$  y  $B$  tales que  $A \prec Q_\infty \prec B$ .

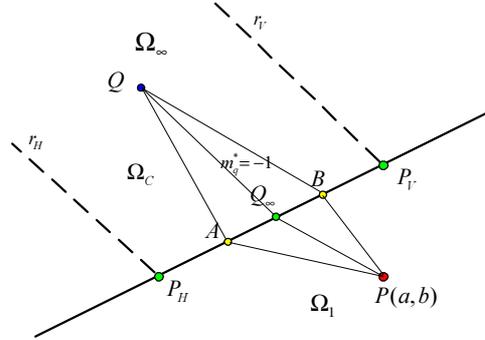


Figura 2.24: Portones para  $Q \in \Omega_C \subset \Omega_\infty$  y  $P \in \Omega_p$ .

Tanto en  $A$  como en  $B$ , se tiene que  $m_p < 0$  y, según (6):

- en  $A$ ,  $\delta = -m \Rightarrow d'(x) = -f(\alpha) - m \Rightarrow d'(x) \leq -m \operatorname{sgn}(m_p) - m = 0$ .

- en  $B$ ,  $\delta = 1 \Rightarrow d'(x) = -f(\alpha) + 1 \Rightarrow d'(x) \geq -1 + 1 = 0$ .

Por tanto, la función objetivo alcanza el mínimo en  $X_s = Q_\infty$ .

- Si  $Q \in \Omega_V$ , entonces, para  $(x, mx) \in L[P_V, Q_V]$ , se tiene que  $\delta = -m$  y  $d'(x) = \operatorname{sgn}(x - a)f(\alpha) - m$ . De aquí se sigue

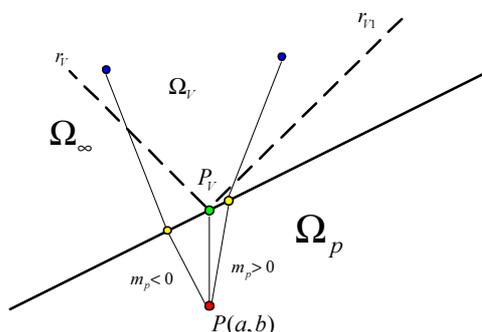


Figura 2.25: Portón para  $Q \in \Omega_V \subset \Omega_\infty$  y  $P \in \Omega_p$ .

- Si  $x < a$ , entonces,  $d'(x) = -f(\alpha) - m < -m \operatorname{sgn}(m_p) - m = 0$ .
- Si  $x > a$ , entonces,  $d'(x) = f(\alpha) - m > m \operatorname{sgn}(m_p) - m = 0$ .

Por tanto, el mínimo de la función objetivo se alcanza en  $X_s = P_V$  (ver figura 2.25).

- En otro caso,  $X_s \in L[P_V, Q_V] \Rightarrow a \leq x_s \leq c$  y  $b \leq y_s \leq d$ , por tanto:

$$d'(x) = f(\alpha) + \delta \quad \text{con} \quad \delta \in \{-1, -m\} \quad \text{y} \quad m_p > 0, \quad \text{y} \quad \text{según} \quad (6),$$

$$m < f(\alpha) \leq \sqrt[p]{1 + m^p}.$$

Si  $\delta = -1$ , resolvemos la ecuación  $d'(x) = f(\alpha) - 1 = \frac{1 + m|m_p^*|^{p-1}}{\sqrt[p]{(1 + |m_p^*|^p)^{p-1}}} - 1 = 0$ ,

que, según proposición 2.22, tiene solución positiva única  $m_p^*$ . Denotemos con  $P^*$  el punto de la recta tal que  $m_{pP^*} = m_p^*$ , es decir, la pendiente de la recta determinada por  $P$  y  $P^*$  (ver figura 2.26).

De aquí se sigue que:

- Si  $Q \in \Omega_V$ , entonces  $0 < m_{p^*Q} \leq 1 \Rightarrow \delta = -1$ , y por tanto  $X_s = P^*$

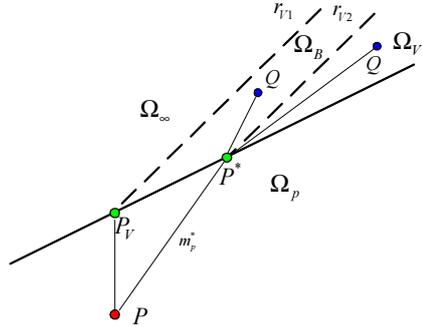


Figura 2.26: Portón para  $Q \in \Omega_{V'} \subset \Omega_{\infty}$  y  $P \in \Omega_p$ .

- En otro caso,  $Q \in \Omega_B$  y  $P^*$  no puede ser el portón entre  $P$  y  $Q$ , ya que  $m_{P^*Q} \geq 1 \Rightarrow \delta = -m$  y  $d'(x) = f(\alpha) + \delta > m - m = 0$  (ver figura 2.27).

De forma análoga a la región  $\Omega_C$ , consideremos los puntos  $A$  y  $B$  tales que  $A \prec Q_{\infty}^* \prec B$ , con  $Q_{\infty}^* = \left( \frac{c-d}{1-m}, \frac{m(c-d)}{1-m} \right)$ , proyección de  $Q$  sobre  $r$  con pendiente unitaria.

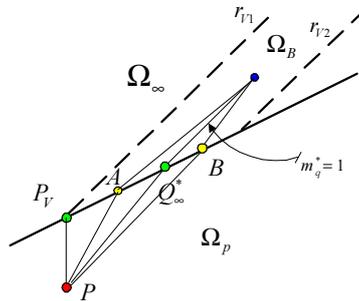


Figura 2.27: Portones para  $Q \in \Omega_B \subset \Omega_{\infty}$  y  $P \in \Omega_p$ .

Si  $A \prec Q_{\infty}^*$ , entonces  $\delta = -1$ ,  $m_{PA} > m_p^*$  y, según la proposición 2.22,  $f(m_{PA}) < 1$ , con lo que  $d'(x) = f(m_{PA}) - 1 < 1 - 1 = 0$ .

Si  $Q_{\infty}^* \prec B$ , entonces  $\delta = -m$ , luego  $d'(x) = f(\alpha) + m > m + m > 0$ .

De ambas desigualdades se sigue que  $X_s = Q_{\infty}^*$ .

- Completamos el estudio de pendientes positivas con el caso  $m > 1$ . En esta situación, la banda  $\Omega_B$  y el cono  $\Omega_{V'}$  se ubican en la parte inferior izquierda del punto  $P$ , denominando a  $\Omega_{V'}$  como  $\Omega_{H'}$ , tal como indica la figura 2.28. Los

2.3.2 El caso de normas  $\ell_p$  y  $\ell_\infty$

portones siguen siendo  $P_V$ ,  $Q_\infty$ ,  $P_H$ ,  $Q_\infty^*$  y  $P^*$  obtenidos con las mismas fórmulas del enunciado del teorema.

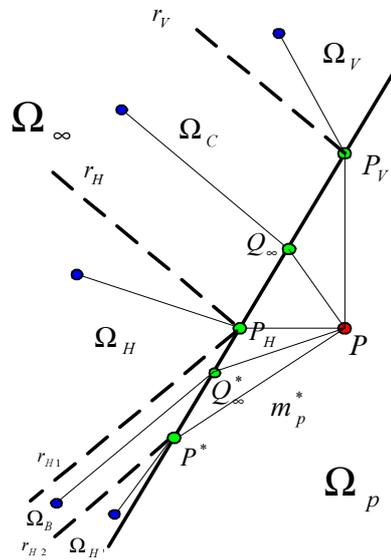


Figura 2.28: Regiones en  $\Omega_\infty$  definidas por  $P \in \Omega_p$ , para  $m > 1$

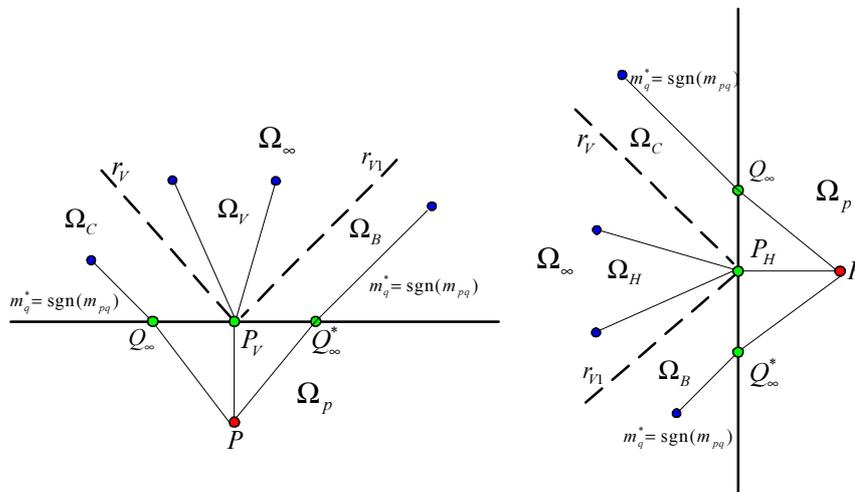


Figura 2.29: Portones para rectas horizontales y verticales.

- En el caso  $m = 0$ , la ecuación  $\frac{1 + |m| \cdot |m_p^*|^{p-1}}{\sqrt[p]{(1 + |m_p^*|^p)^{p-1}}} = 1$  sólo tiene como solución  $m_p^* = 0$ , es decir, el punto  $P^*$  se “aleja” al infinito, de manera que los portones son,

ó bien  $P_V = (a, 0)$ , ó bien  $\left( \frac{m_q^* c - d}{m_q^* - m}, \frac{m(m_q^* c - d)}{m_q^* - m} \right)$ , donde  $m_q^* = \text{sgn}(m_{pq})$ , según

ilustra la figura 2.29. Análogamente, si  $m = \infty$  sólo cambia el portón vertical por el horizontal  $P_H = (0, b)$ .

- Para finalizar, si  $m < 0$ , mediante una simetría axial de eje  $x = a$  tenemos una recta con pendiente positiva en la que son válidos tanto los razonamientos anteriores como los portones del enunciado. ■

**Nota 2.29. (Posición del portón según  $P$ )** Dado un punto  $Q \in \Omega_\infty$ , sus proyectados en  $r$ ,  $Q^*$  y  $Q_\infty$  según pendientes 1 y -1, respectivamente, determinan cinco regiones en

$\Omega_p$ , delimitadas por la propia recta  $r$  y las semirrectas  $r_H \equiv y = \frac{m(c+d)}{m+1}$ ,

$r_V \equiv x = \frac{(c+d)}{m+1}$ ,  $r_{V1} \equiv x = \frac{c-d}{1-m}$  y  $r_{V2} \equiv y - \frac{m(c-d)}{1-m} = m_p^* \left( x - \frac{c-d}{1-m} \right)$ , según ilustra la

figura 2.30.

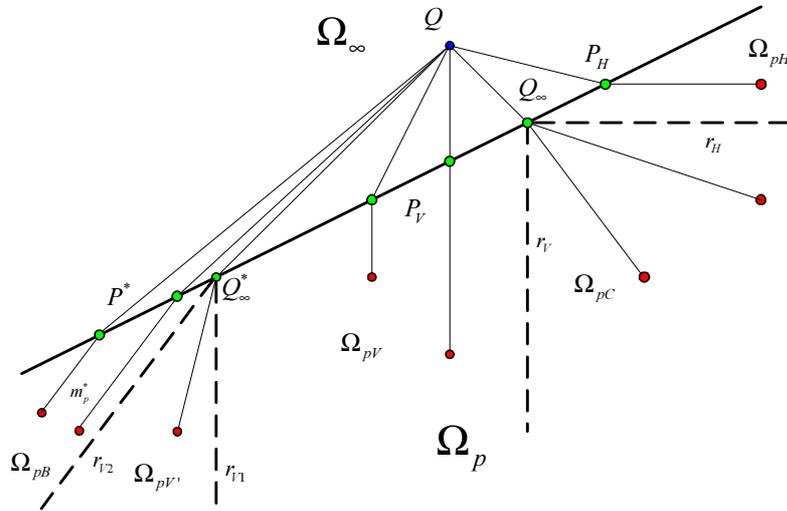


Figura 2.30: Regiones en  $\Omega_p$  definidas por  $Q \in \Omega_\infty$ .

Sigue que, si:

- $P \in \Omega_{pH}$ , entonces  $G(P, Q) = P_H$ .
- $P \in \Omega_{pC}$ , entonces  $G(P, Q) = Q_\infty$ .

### 2.3.2 El caso de normas $\ell_p$ y $\ell_\infty$

- $P \in \Omega_{pV}$ , entonces  $G(P, Q) = P_V$ .
- $P \in \Omega_{pV'}$ , entonces  $G(P, Q) = Q^*$ .
- $P \in \Omega_{pB}$ , entonces  $G(P, Q) = P^*$ , proyección de  $P$  en  $r$ , según la pendiente  $m_p^*$ , solución de la ecuación (5).

**Nota 2.30.** Si  $|m|=1$ , entonces, al no tener solución la ecuación (5), sólo existen tres portones:  $P_V$  si  $Q \in \Omega_V$ ,  $P_H$  si  $Q \in \Omega_H$  y  $Q_\infty$  si  $Q \in \Omega_C$ .

En los siguientes resultados vemos dos casos particulares de especial significación ya que están involucradas la norma rectangular y la norma euclídea, en cada caso.

**Corolario 2.31.** Sean  $P(a, b) \in \Omega_1$  y  $Q(c, d) \in \Omega_\infty$ , puntos del plano separados por la recta  $r \equiv y = mx$ . Entonces, la puerta de acceso  $G(P, Q) \in r$  es, o bien  $(a, ma)$ , o bien

$$(b/m, b), \text{ o bien } \left( \frac{m_q^*c - d}{m_q^* - m}, \frac{m(m_q^*c - d)}{m_q^* - m} \right), \text{ donde } m_q^* = \text{sgn}(m_{pq}).$$

**Demostración.** Si  $p = 1 + \varepsilon$ , la ecuación (5) queda  $\frac{1 + |m| |m_p^*|^\varepsilon}{(1 + |m_p^*|^{1+\varepsilon})^{\frac{\varepsilon}{1+\varepsilon}}} = 1$ .

Si  $\varepsilon \rightarrow 0$  entonces  $m_p^* \rightarrow +\infty$ , en virtud de la proposición 2.22, y, por lo tanto, el portón  $P^*$  coincide con el portón  $P_V$ . De este modo, solamente tenemos 3 regiones en  $\Omega_\infty$  y 3 portones:  $P_V$  si  $Q \in \Omega_V$ ,  $P_H$  si  $Q \in \Omega_H$ ,  $Q_\infty$  si  $Q \in \Omega_C$ . ■

**Corolario 2.32.** Sean  $P(a, b) \in \Omega_2$  y  $Q(c, d) \in \Omega_\infty$ , puntos del plano separados por la recta  $r \equiv y = mx$ . Entonces, la puerta de acceso  $G(P, Q) \in r$  es, o bien  $(a, ma)$ , o bien

$$(b/m, b), \text{ o bien } \left( \frac{m_q^*c - d}{m_q^* - m}, \frac{m(m_q^*c - d)}{m_q^* - m} \right), \text{ o bien } \left( \frac{m_p^*a - b}{m_p^* - m}, \frac{m(m_p^*a - b)}{m_p^* - m} \right), \text{ donde}$$

$$m_q^* = \text{sgn}(m_{pq}) \text{ y } m_p^* = \frac{2m}{1 - m^2}.$$

**Demostración** Si  $p = 2$ , la ecuación (5) queda como  $\frac{1 + k\alpha}{\sqrt{(1 + \alpha^2)}} = 1$ , con  $\alpha = |m_p^*|$  y

$$k = |m|.$$

Resolviendo la ecuación de segundo grado resultante, se obtiene: 
$$\begin{cases} \alpha = 0 \\ \alpha = \frac{2k}{1-k^2} \end{cases}$$

Excluimos  $\alpha = 0$ , ya que proporciona un punto que no está en  $L[P_V, Q_V]$ .

Por tanto, la única solución es:  $m_p^* = \frac{2m}{1-m^2}$ , ya que  $\text{sgn}(m_p^*) = \text{sgn}(m)$ . ■

### 2.3.3 El portón de un punto situado en la recta

Sean  $P(a, b) \in \Omega_p$  y  $Q(c, d) \in \Omega_q$ ,  $1 < p < q < \infty$ , puntos del plano separados por la recta  $r \equiv y = mx$ , con  $m \geq 0$ . Supongamos, además, que  $Q \in r \subset \Omega_q$  (ver figura 2.31) y, por tanto,  $d = mc$ . Pretendemos encontrar el portón  $G = G(P, Q) = (x_s, mx_s)$  que minimice la distancia entre  $P$  y  $Q$  que, sorprendentemente, no será siempre la línea recta salvo que  $G \equiv Q$ , como vamos a poner de manifiesto a continuación.

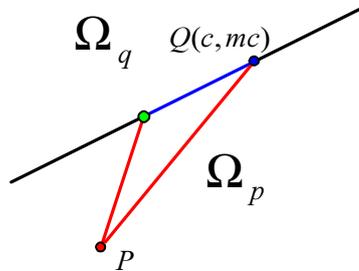


Figura 2.31: Posibles trayectorias entre  $P$  y  $Q$ , cuando  $Q \in r$ .

Como  $X, Q \in r$ , hemos de hallar:

$$\begin{aligned} d(P, Q) &= \min_{X \in r} \{d_p(P, X) + d_q(X, Q)\} = \\ &= \min_{x \in \mathbb{R}} \left\{ \sqrt[p]{|x-a|^p + |mx-b|^p} + \sqrt[q]{|x-c|^q + |mx-mc|^q} \right\} = \min D(x). \end{aligned}$$

Usando el teorema 2.20 tenemos que  $x_s \in I[a, c] \cap I[b/m, c] = [\alpha, \beta]$ , es decir:  $x_s = a$ , ó  $x_s = c$ , ó bien  $x_s = b/m$ , en caso contrario, usando la fórmula (4), tenemos:

$$D'(x) = \operatorname{sgn}(x-a) \left( \frac{1 + \operatorname{sgn}(m_p) |m_p|^{p-1} m}{\left( \sqrt[p]{1 + |m_p|^p} \right)^{p-1}} - \frac{1 + \operatorname{sgn}(m_q) |m_q|^{q-1} m}{\left( \sqrt[q]{1 + |m_q|^q} \right)^{q-1}} \right),$$

con  $m_q = m$  y  $\operatorname{sgn}(m_p) = \operatorname{sgn}(m_q) = 1$ .

Por tanto,

$$D'(x) = \operatorname{sgn}(x-a) \left( \frac{1 + m |m_p|^{p-1}}{\left( \sqrt[p]{1 + |m_p|^p} \right)^{p-1}} - \frac{1 + m^q}{\left( \sqrt[q]{1 + m^q} \right)^{q-1}} \right) = \operatorname{sgn}(x-a) \left( f_p(m_p) - \sqrt[q]{1 + m^q} \right).$$

Se pueden dar los siguientes casos:

- Si  $m = 0$ , necesariamente  $x_s = c$ , ó  $x_s = a$ , ó bien  $m_p = 0$ . Las dos últimas soluciones son inadmisibles ya que  $m_p = 0$  conduce a la ausencia de portón, y

$$D(a) = \sqrt[p]{|b|^p} + \sqrt[q]{|a-c|^q} = |b| + |a-c| > \sqrt[p]{|c-a|^p} + |b|^p = D(c) \Rightarrow x_s = c,$$

es decir:  $G(P, Q) = Q$ .

- En caso contrario, en virtud de la proposición 2.22, existen  $0 < m_{p1} < m_{p2}$  tales que

$$f_p(m_{p1}) = f_p(m_{p2}) = \sqrt[q]{1 + m^q} \text{ y, por tanto, se anula la derivada.}$$

Razonando de forma análoga para  $m < 0$ , obtenemos el siguiente resultado:

**Proposición 2.33.** Sean  $P(a, b) \in \Omega_p$  y  $Q(c, d) \in \Omega_q$ ,  $1 < p < q < \infty$ , puntos del plano separados por la recta  $r \equiv y = mx$ , con  $m \geq 0$ . Supongamos, además, que  $Q \in r \subset \Omega_q$  y

sea  $G = G(P, Q) = (x_s, mx_s)$ . Entonces, ó bien  $x_s = c$  ó bien  $x_s = \frac{b - m_p a}{m - m_p}$ , donde  $m_p$

es una de las dos soluciones de la ecuación:



---

### LOS PORTONES Y LA SOLUCIÓN DEL PROBLEMA DE LOCALIZACIÓN CON DOS REGIONES

---

En este capítulo vamos a exponer los principales resultados teóricos de este trabajo doctoral, que están basados en el uso de los portones descritos en el capítulo anterior. Así, en la sección 2 estudiamos el problema (1) teniendo en cuenta los portones y las diferentes métricas aplicadas en cada subregión obtenida al dividir el plano en dos partes. Debemos de tener en cuenta que para cada par de métricas utilizadas, tendremos que resolver problemas diferentes, ya que en cada caso la solución en cada subregión se ha de encontrar utilizando diferentes métodos. Es por ello por lo que tratamos los diferentes problemas expuestos en la sección segunda. Terminamos este capítulo estudiando la obtención de la solución al problema 2 en la subregión  $\Omega_1$ .

#### **3.1 La solución del problema para dos puntos**

Supongamos ahora que  $P \in \Omega_p$  y  $Q \in \Omega_q$  son puntos de regiones con normas  $\ell_p$  y  $\ell_q$ , respectivamente, separados por la recta  $r \equiv y = mx$ . Consideremos el problema

$$\min_{X \in \mathbb{R}^2} \{w_p d(P, X) + w_q d(X, Q)\}, \text{ donde } w_p, w_q \in \mathbb{R}^+ \quad (7)$$

Nótese que el problema (7) en realidad es el problema (1) cuando  $|S_p| = |S_q| = 1$ . La siguiente proposición proporciona la solución a este problema.

**Proposición 3.1.** Sean  $P \in \Omega_p$  y  $Q \in \Omega_q$ , puntos de regiones con normas  $\ell_p$  y  $\ell_q$ , respectivamente, separados por la recta  $r \equiv y = mx$ . Sea  $d = \min\{D_p, D_q\}$ , donde

$$D_p = \min_{F \in \Omega_p, X \in r} \{w_p d_p(P, F) + w_q d_p(F, X) + w_q d_q(X, Q)\}$$

$$D_q = \min_{F \in \Omega_q, X \in r} \{w_p d_p(P, X) + w_p d_q(X, F) + w_q d_q(F, Q)\}$$

Si  $R \in \mathbb{R}^2$  y  $G \in r$  son los puntos en los que  $d$  alcanza el mínimo, entonces  $G = G(P, Q)$ , portón entre  $P$  y  $Q$ , y además,

- $R = P$  si  $w_p > w_q$
- $R = Q$  si  $w_q > w_p$
- $R$  es cualquier punto de la poligonal  $L[P, G] \cup L[G, Q]$  si  $w_p = w_q$

**Demostración.** Dado que la función objetivo  $d = \min_{X \in \mathbb{R}^2} \{w_p d(P, X) + w_q d(X, Q)\}$ , no es convexa, se ha dividido el problema (7), en dos problemas convexos  $D_p$  y  $D_q$ , según el semiplano en el que busquemos la solución.

1) Sea  $F \in \Omega_p$ .

- Si  $w_p > w_q$ , entonces, para cualquier  $X \in r$ :

$$w_p d_p(P, F) + w_q d_p(F, X) \geq w_q d_p(P, F) + w_q d_p(F, X) \geq w_q d_p(P, X), \text{ luego}$$

$$w_p d_p(P, F) + w_q d_p(F, X) + w_q d_q(X, Q) \geq w_q d_p(P, X) + w_q d_q(X, Q).$$

De este modo, el mínimo se alcanza en  $F = P$  y  $G = G(P, Q)$

$$D_p = \min_{F \in \Omega_p, X \in r} \{w_p d_p(P, F) + w_q d_p(F, X) + w_q d_q(X, Q)\} =$$

$$= \min_{X \in r} \{w_q d_p(P, X) + w_q d_q(X, Q)\} = w_q d_p(P, G) + w_q d_q(G, Q) = w_q d(P, Q)$$

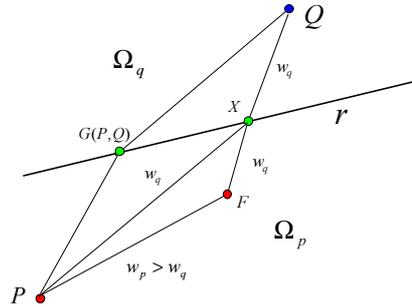


Figura 3.1: Posición inadmisibles de  $F$ , solución del problema (7).

- Si  $w_p = w_q = w$ , entonces, para cualquier  $F \in L[P, X]$

$$w d_p(P, F) + w d_p(F, X) + w d_q(X, Q) = w d_p(P, X) + w d_q(X, Q)$$

Por tanto, el mínimo  $D_p = w d(P, Q)$  se alcanza en  $G = G(P, Q)$  y en cualquier punto de  $L[P, G]$

- Si  $w_p < w_q$ , entonces, para cualquier  $X \in r$ :

$$\begin{aligned} w_p d_p(P, F) + w_q d_p(F, X) &\geq w_p d_p(P, F) + w_p d_p(F, X) \geq w_p d_p(P, X), \text{ luego} \\ w_p d_p(P, F) + w_q d_p(F, X) + w_q d_q(X, Q) &\geq w_p d_p(P, X) + w_q d_q(X, Q) \geq \\ &\geq w_p (d_p(P, X) + d_q(X, Q)) \end{aligned}$$

Por lo tanto,

$$\begin{aligned} D_p &= \min_{F \in \Omega_p, X \in r} \{w_p d_p(P, F) + w_q d_p(F, X) + w_q d_q(X, Q)\} = \\ &= \min_{X \in r} \{w_p d_p(P, X) + w_q d_q(X, Q)\} \geq w_p (d_p(P, G) + d_q(G, Q)) = w_p d(P, Q) \end{aligned}$$

2) Sea  $F \in \Omega_q$ , por simetría,

- Si  $w_p > w_q$ , entonces  $D_q \geq w_q d(P, Q)$
- Si  $w_p = w_q = w$ , entonces  $D_q = w d(P, Q)$ , se alcanza en  $G = G(P, Q)$  y en cualquier punto de  $L[G, Q]$
- Si  $w_p < w_q$ , entonces  $D_q = w_p d(P, Q)$  se alcanza en  $R = Q$  y  $G = G(P, Q)$

En conclusión:

- Si  $w_p > w_q$ , entonces  $d = \min\{D_p, D_q\} = D_p = w_q d(P, Q) \leq D_q$ , se alcanza en  $R = P$  y  $G = G(P, Q)$
- Si  $w_p = w_q = w$ , entonces  $d = \min\{D_p, D_q\} = D_q = D_p = wd(P, Q)$ , se alcanza en  $G = G(P, Q)$  y en cualquier punto de  $L[P, G] \cup L[G, Q]$ .
- Si  $w_p < w_q$ , entonces  $d = \min\{D_p, D_q\} = D_q = w_p d(P, Q) \leq D_p$ , se alcanza en  $R = Q$  y  $G = G(P, Q)$ . ■

En esta proposición vemos que tanto el portón  $G = G(P, Q)$  como el peso de los puntos en cada región juegan un papel fundamental para encontrar la solución del problema (1). Mientras que los portones siguen jugando este papel en el caso de múltiples puntos en cada región, la distribución geométrica de los pesos gana importancia en detrimento del peso total de los puntos en cada región.

No obstante, hay situaciones en las que el peso total de los puntos determina la región en la que se encuentra el óptimo del problema (1). En Brimberg et al. (2003) podemos encontrar el siguiente resultado:

**Proposición 3.2.** Sean  $\{P_i\}_{i \in S_1} \subset \Omega_1$  y  $\{Q_j\}_{j \in S_2} \subset \Omega_2$ , puntos de regiones con normas  $\ell_1$  y  $\ell_2$ , respectivamente, separados por la recta  $r \equiv y = y_0$ , y sean  $W_1 = \sum_{i \in S_1} w_i$  y  $W_2 = \sum_{j \in S_2} w_j$ . Si  $W_2 \geq W_1$ , entonces el punto solución del problema (1) se encuentra en  $\Omega_2 \cup r$ . El resultado también es válido para rectas verticales. ■

Vamos a enunciar y demostrar un resultado más general que el anterior, basado en el conocimiento preciso de los portones:

**Proposición 3.3.** Sean  $\{P_i\}_{i \in S_1} \subset \Omega_1$  y  $\{Q_j\}_{j \in S_q} \subset \Omega_q$ , puntos de regiones con normas  $\ell_1$  y  $\ell_q$ , respectivamente, separados por la recta  $r \equiv y = y_0$ , y sean  $W_1 = \sum_{i \in S_1} w_i$  y  $W_q = \sum_{j \in S_q} w_j$ . Si  $W_q \geq W_1$ , entonces el punto solución del problema (1) se encuentra en  $\Omega_q \cup r$ . El resultado también es válido para rectas verticales.

### 3.1 La solución del problema para dos puntos

**Demostración.** Supongamos que el punto solución  $X_s(x_s, y_s)$  se encuentra en  $\Omega_1$ . Entonces, según el teorema 2.23, el portón correspondiente al punto  $X_s$  y a todos los puntos de  $\Omega_q$  es el punto  $X_V(x_s, y_0) \in r$  (ver figura 3.2). Por tanto, la función objetivo en  $X_s$  es:

$$\begin{aligned} D(X_s) &= \sum_{i \in S_1} w_i d_1(P_i, X_s) + \sum_{j \in S_q} w_j d_1(X_s, X_V) + \sum_{j \in S_q} w_j d_q(X_V, Q_j) = \\ &= \sum_{i \in S_1} w_i d_1(P_i, X_s) + W_q d_1(X_s, X_V) + \sum_{j \in S_q} w_j d_q(X_V, Q_j) \end{aligned}$$

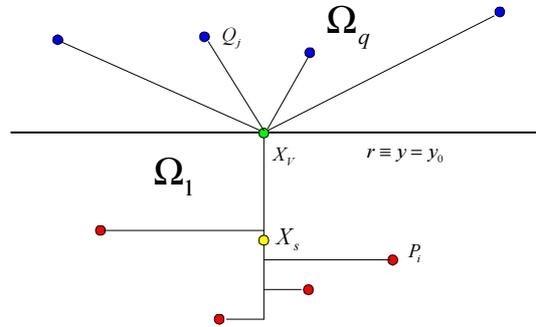


Figura 3.2: Representación gráfica de una posible solución y el portón en recta horizontal.

Sin embargo, en  $X_V$  toma el valor:

$$\begin{aligned} D(X_V) &= \sum_{i \in S_1} w_i d_1(P_i, X_V) + \sum_{j \in S_q} w_j d_q(X_V, Q_j) \leq \\ &\leq \sum_{i \in S_1} w_i d_1(P_i, X_s) + \sum_{i \in S_1} w_j d_1(X_s, X_V) + \sum_{j \in S_q} w_j d_q(X_V, Q_j) = \\ &= \sum_{i \in S_1} w_i d_1(P_i, X_s) + W_1 d_1(X_s, X_V) + \sum_{j \in S_q} w_j d_q(X_V, Q_j) \end{aligned}$$

Dado que  $W_1 \leq W_q$ , se tiene  $D(X_V) \leq D(X_s)$  y por tanto,  $X_s \in \Omega_1$  no es el único punto óptimo, que habrá de estar en  $\Omega_q \cup r$ . ■

Este resultado no se puede generalizar para rectas oblicuas, ni tampoco para rectas horizontales cuando  $p > 1$  (Brimberg et al., 2003).

### 3.2 La solución del problema con las normas $\ell_p$ y $\ell_q$

**Definición 3.4.** Sean  $\mathcal{P} = \{P_i\}_{i \in S_p} \subset \Omega_p$  y  $\mathcal{Q} = \{Q_j\}_{j \in S_q} \subset \Omega_q$ , con  $1 \leq p, q \leq \infty$ , dos conjuntos de puntos separados por una recta  $r$ , con pesos  $\{w_i\}_{i \in S_p} \subset \mathbb{R}^+$  y  $\{w_j\}_{j \in S_q} \subset \mathbb{R}^+$ , respectivamente. Para cualquier  $X \in \Omega_q$ , se define el conjunto  $WP_q(X; \mathcal{P}, \mathcal{Q}) \subset \Omega_q$ , abreviadamente  $WP_q(X)$ , como el conjunto de puntos que son solución del siguiente problema de localización simple (problema de Weber-Fermat):

$$\min_{Y \in \mathbb{R}^2} \left\{ \sum_{i \in S_p} w_i d_q(G(X, P_i), Y) + \sum_{j \in S_q} w_j d_q(Y, Q_j) \right\} \quad (8)$$

Nótese que el problema (8) tiene solución única para  $q \in (1, \infty)$ . Más aún, la solución se encuentra en  $\Omega_q$  ya que se encuentra en la envolvente convexa de los puntos de  $\mathcal{Q}$  y de  $\{G(P_i, X)\}_{i \in S_p} \subset r$ . Si  $q \notin (1, \infty)$  la función objetivo del problema (8) no es estrictamente convexa y puede tener varias soluciones (Love et al. 1988).

**Teorema 3.5 (Condición Necesaria y Suficiente en  $\Omega_q$ ).** Sean  $\mathcal{P} = \{P_i\}_{i \in S_p} \subset \Omega_p$  y  $\mathcal{Q} = \{Q_j\}_{j \in S_q} \subset \Omega_q$ , con  $1 < q < \infty$ , dos conjuntos de puntos separados por la recta  $r \equiv y = mx$ . Supongamos que la solución del problema (1) se encuentra en  $\Omega_q$ . Entonces,  $X_s \in \Omega_q$  es solución del problema (1) si y solo si  $WP_q(X_s) = X_s$ .

**Demostración:**

(Condición Necesaria): Si  $WP_q(X_s) = X' \neq X_s$ , entonces

$$\sum_{i \in S_p} w_i d_q(G(X_s, P_i), X') + \sum_{j \in S_q} w_j d_q(X', Q_j) < \sum_{i \in S_p} w_i d_q(G(X_s, P_i), X_s) + \sum_{j \in S_q} w_j d_q(X_s, Q_j)$$

y por tanto,

$$\begin{aligned} & \sum_{i \in S_p} w_i d_p(P_i, G(X_s, P_i)) + \sum_{i \in S_p} w_i d_q(G(X_s, P_i), X') + \sum_{j \in S_q} w_j d_q(X', Q_j) < \\ & < \sum_{i \in S_p} w_i d_p(P_i, G(X_s, P_i)) + \sum_{i \in S_p} w_i d_q(G(X_s, P_i), X_0) + \sum_{j \in S_q} w_j d_q(X_s, Q_j) \end{aligned}$$

con lo que  $X_0$  no sería solución del problema (2).

(Condición suficiente): Sea  $WP_q(X_s) = X_s = (x_s, y_s)$  la solución del problema de Weber-Fermat, y denotemos con  $G_i = G(P_i, X) = (g_i, mg_i) \in r$  al portón correspondiente a cada par de puntos  $P_i$  y  $X$ .

Vamos a probar que  $X_s = (x_s, y_s)$  es el mínimo absoluto de la función objetivo del problema (1), es decir, el mínimo de:

$$D(x, y) = \sum_{j \in S_q} w_j d_q(Q_j, X) + \sum_{i \in S_p} w_i d_q(X, G_i) + \sum_{i \in S_p} w_i d_p(G_i, P_i) = D_q(x, y) + D_{pq}(x, y) + D_p(x, y).$$

Por una parte:

$$D_q(x, y) = \sum_{j \in S_q} w_j d_q(Q_j, X) \Rightarrow \frac{\partial}{\partial x} D_q(x, y) = \sum_{j \in S_q} w_j \frac{\partial}{\partial x} d_q(Q_j, X),$$

Por otra parte:

$$D_{pq}(x, y) = \sum_{i \in S_p} w_i d_q(X, G_i) = \sum_{i \in S_p} w_i \sqrt[q]{|x - g_i|^q + |y - mg_i|^q}, \text{ y por tanto,}$$

$$\frac{\partial}{\partial x} D_{pq}(x, y) = \sum_{i \in S_p} w_i \frac{\operatorname{sgn}(x - g_i) |x - g_i|^{q-1} \left(1 - \frac{\partial}{\partial x} g_i\right) + \operatorname{sgn}(y - mg_i) |y - mg_i|^{q-1} \left(-m \frac{\partial}{\partial x} g_i\right)}{\sqrt[q]{\left(|x - g_i|^q + |y - mg_i|^q\right)^{q-1}}}.$$

Por último,

$$D_p(x, y) = \sum_{i \in S_p} w_i d_p(G_i, P_i) = \sum_{i \in S_p} w_i \sqrt[p]{|a_i - g_i|^p + |b_i - mg_i|^p}, \text{ si } P_i = (a_i, b_i), \quad i \in S_p.$$

Por lo tanto,

$$\frac{\partial}{\partial x} D_p(x, y) = \sum_{i \in S_p} w_i \frac{\operatorname{sgn}(a_i - g_i) |a_i - g_i|^{p-1} \left(-\frac{\partial}{\partial x} g_i\right) + \operatorname{sgn}(b_i - mg_i) |b_i - mg_i|^{p-1} \left(-m \frac{\partial}{\partial x} g_i\right)}{\sqrt[p]{\left(|a_i - g_i|^p + |b_i - mg_i|^p\right)^{p-1}}}.$$

Sumando y agrupando tenemos:

$$\begin{aligned} \frac{\partial}{\partial x} D(x, y) &= \sum_{j \in S_q} w_j \frac{\partial}{\partial x} d_q(Q_j, X) + \sum_{i \in S_p} w_i \frac{\operatorname{sgn}(x - g_i) |x - g_i|^{q-1}}{\sqrt[q]{(|x - g_i|^q + |y - mg_i|^q)^{q-1}}} + \\ &+ \sum_{i \in S_p} w_i \left( \frac{\operatorname{sgn}(x - g_i) |x - g_i|^{q-1} + m \operatorname{sgn}(y - mg_i) |y - mg_i|^{q-1}}{\sqrt[q]{(|x - g_i|^q + |y - mg_i|^q)^{q-1}}} \right) \left( -\frac{\partial}{\partial x} g_i \right) + \\ &+ \sum_{i \in S_p} w_i \left( \frac{\operatorname{sgn}(a_i - g_i) |a_i - g_i|^{p-1} + m \operatorname{sgn}(b_i - mg_i) |b_i - mg_i|^{p-1}}{\sqrt[p]{(|a_i - g_i|^p + |b_i - mg_i|^p)^{p-1}}} \right) \left( -\frac{\partial}{\partial x} g_i \right). \end{aligned}$$

Entonces:

$$\frac{\partial}{\partial x} D(x_s, y_s) = \frac{\partial}{\partial x} F(x_s, y_s) + \sum_{i \in S_p} w_i H_i(x_s, y_s) \left( -\frac{\partial}{\partial x} g_i \right), \text{ donde}$$

$$\begin{aligned} H_i(x_s, y_s) &= \frac{\operatorname{sgn}(x_s - g_{is}) |x_s - g_{is}|^{q-1} + m \operatorname{sgn}(y_s - mg_{is}) |y_s - mg_{is}|^{q-1}}{\sqrt[q]{(|x_s - g_{is}|^q + |y_s - mg_{is}|^q)^{q-1}}} + \\ &\frac{\operatorname{sgn}(a_i - g_{is}) |a_i - g_{is}|^{p-1} + m \operatorname{sgn}(b_i - mg_{is}) |b_i - mg_{is}|^{p-1}}{\sqrt[p]{(|a_i - g_{is}|^p + |b_i - mg_{is}|^p)^{p-1}}}, \end{aligned}$$

$$y \ F(x, y) = \sum_{j \in S_q} w_j d_q(Q_j, X) + \sum_{i \in S_p} w_i d_q(X, G_{is}), \text{ con } G_{is} = G(P_i, X_s) = (g_{is}, mg_{is}).$$

Para cada  $i \in S_p$  se tiene que:

- Si  $p, q \in (1, \infty)$ , entonces  $H_i(x_s, y_s) = 0$ , en virtud del teorema 2.20.
- Si  $1 = p < q < \infty$ , entonces según las posiciones relativas de  $P_i$  y  $X_s$  se tienen los siguientes casos (ver teorema 2.23 y figura 2.19):
  - Si  $X_s \in \Omega_{qC_i}$ , entonces  $H_i(x_s, y_s) = 0$ .
  - Si  $X_s \notin \Omega_{qC_i}$ , entonces,  $\frac{\partial}{\partial x} g_i(x_s, y_s) = 0$  ya que el portón no varía, siendo éste el vertical  $(a_i, ma_i)$  o el horizontal  $(b_i/m, b_i)$ .

$$\text{Por tanto, } H_i(x_s, y_s) = 0 \text{ o/y } \left( -\frac{\partial}{\partial x} g_i(x_s, y_s) \right) = 0.$$

- Si  $1 < q < p = \infty$ , entonces según las posiciones relativas de  $P_i$  y  $X_s$  (ver teorema 2.28 y figura 2.22) se tiene:
  - Si  $P_i \in \Omega_C \cup \Omega_B$ , entonces  $H_i(x_s, y_s) = 0$ .
  - Si  $P_i \in \Omega_H \cup \Omega_V \cup \Omega_{V'}$ , entonces,  $\frac{\partial}{\partial x} g_i(x_s, y_s) = 0$  ya que el portón no varía, siendo éste el vertical, el horizontal o el punto  $X_s^*$ .

$$\text{Por tanto, } H_i(x_s, y_s) = 0 \text{ o/y } \left( -\frac{\partial}{\partial x} g_i(x_s, y_s) \right) = 0.$$

Además, dado que  $WP_q(X_s) = (x_s, y_s)$  se tiene que:

$$(x_s, y_s) = \arg \min_{(x,y) \in \mathbb{R}^2} F(x, y) = \arg \min_{X \in \mathbb{R}^2} \left\{ \sum_{j \in \mathcal{S}_q} w_j d_q(Q_j, X) + \sum_{i \in \mathcal{S}_p} w_i d_q(X, G_{is}) \right\},$$

$$\text{luego, } \frac{\partial}{\partial x^-} F(x_s, y_s) \cdot \frac{\partial}{\partial x^+} F(x_s, y_s) \leq 0 \text{ y por tanto, } \frac{\partial}{\partial x^-} D(x_s, y_s) \cdot \frac{\partial}{\partial x^+} D(x_s, y_s) \leq 0.$$

Análogamente, tenemos que:  $\frac{\partial}{\partial y^-} D(x_s, y_s) \cdot \frac{\partial}{\partial y^+} D(x_s, y_s) \leq 0$  y, como la función objetivo  $D(x, y)$  es convexa, tenemos un mínimo absoluto en  $(x_s, y_s)$ . ■

Nótese que la hipótesis del teorema anterior también es necesaria para  $q=1$  ó  $q=\infty$ , ya que en la demostración no se ha tenido en cuenta el valor de  $q$ . Sin embargo, dicha condición no es suficiente en estos casos, como pone de manifiesto los siguientes ejemplos.

**Ejemplo.** Sean  $P_1 = (3, 1)$ ,  $P_2 = (6, 2)$  y  $P_3 = (9, 1)$ , puntos de  $\Omega_1$  con pesos unitarios y sea  $Q = (3, 7) \in \Omega_2$  con peso  $w = 2$ . Supongamos además que  $r \equiv y = x/2$  y sea  $X = (7, 2) \in \Omega_1$ .

Según el corolario 2.26,  $m^* = -0,75$  y, por tanto,  $Q^* = (7.4, 3.7) \in r$ . Dado que  $Q^* \notin L[X_H, X_V]$  no es el portón entre  $Q$  y  $X$ , el portón es  $G(Q, X) = X_V = (7, 3.5)$ , con lo que la función objetivo,

$$D(X) = \sum_{i=1}^3 d_1(P_i, X) + 2d_1(X, G(X, Q)) + 2d_2(G(X, Q), Q), \text{ toma el valor:}$$

$$D(X) = |3-7| + |1-2| + |6-7| + |2-2| + |9-7| + |1-2| + 2(|7-7| + |3.5-2|) + 2\sqrt{(7-3)^2 + (3.5-7)^2}, \text{ y por tanto,}$$

$$D(X) = 12 + 2\sqrt{28.25} \simeq 22.63.$$

Además,  $X = (7, 2)$  verifica que  $WP_1(X) = X$  ya que las medianas de  $\{3, 6, 7, 7, 9\}$  y  $\{1, 1, 2, 3.5, 3.5\}$  son 7 y 2, respectivamente.

Sin embargo, si consideramos el punto  $Y = (6, 2)$ , entonces  $G(Q, Y) = Y_v = (6, 3)$ , y además:

$$D(Y) = |3-6| + |1-2| + |6-6| + |2-2| + |9-6| + |1-2| + 2(|6-6| + |3-2|) + 2\sqrt{(6-3)^2 + (3-7)^2} \Rightarrow D(Y) = 10 + 2\sqrt{25} = 20 < D(X)$$

Por lo tanto,  $X = (7, 2)$  no es la solución del problema (1). ■

**Ejemplo.** Sean  $Q_1 = (4, 2)$ ,  $Q_2 = (8, 4)$  y  $Q_3 = (10, 8)$ , puntos de  $\Omega_\infty$  con pesos unitarios y sea  $P = (11, 2) \in \Omega_2$  con peso  $w = 2$ . Supongamos además que  $r \equiv y = x/3$  y sea  $X = (9, 5)$ . Según el corolario 2.31,  $m^* = -1$  y entonces  $G(P, X) = (10.5, 3.5)$ , con lo que  $X = WP_\infty(X)$ . Además, la función objetivo toma el valor,  $D(X) \simeq 15.16$ .

Sin embargo, si consideramos el punto  $Y = (8, 4) \in \Omega_\infty$ , tenemos que  $D(Y) \simeq 14.47$  y por lo tanto,  $X = (9, 5)$  no es la solución del problema (1). ■

La condición necesaria y suficiente del teorema 3.5 nos indica que el punto solución del problema (1), caso de estar en  $\Omega_q$ , es el único punto fijo de la función continua  $Y = WP_q(X)$ . El teorema siguiente, además de demostrar la existencia y unicidad del punto fijo, nos permite construir un algoritmo para encontrarlo.

**Teorema 3.6.** Sean  $\mathcal{P} \subset \Omega_p$  y  $\mathcal{Q} \subset \Omega_q$ , con  $1 < q < \infty$ , dos conjuntos de puntos separados por la recta  $r \equiv y = mx$ . Si  $X_s \in \Omega_q$  es la solución del problema (1), entonces  $X_s = \lim_{k \rightarrow \infty} X_k$ , donde la sucesión  $\{X_k\}$  es tal que  $X_0$  es cualquier punto de  $\Omega_q$  y  $X_{k+1} = WP_q(X_k)$ .

**Demostración.** Dado  $X \in \Omega_q$ , la función objetivo es:

$$\begin{aligned} D(X) &= \sum_{j \in S_q} w_j d_q(Q_j, X) + \sum_{i \in S_p} w_i d(X, P_i) = \\ &= \sum_{j \in S_q} w_j d_q(Q_j, X) + \sum_{i \in S_p} w_i d_q(X, G(P_i, X)) + \sum_{i \in S_p} w_i d_p(G(P_i, X), P_i) \end{aligned}$$

Veamos la existencia de la solución:

- Si existe  $k$  tal que  $X_k = WP_q(X_k)$ , entonces  $X_k$  es un punto fijo y por lo tanto es la solución del problema (1), en virtud del teorema 3.5.
- Si para todo  $k$  se cumple que  $X_k \neq X_{k+1} = WP_q(X_k; \mathcal{P}, \mathcal{Q})$ , entonces:

$$\begin{aligned} D(X_{k+1}) &= \sum_{j \in S_q} w_j d_q(Q_j, X_{k+1}) + \sum_{i \in S_p} w_i d_q(X_{k+1}, G(P_i, X_{k+1})) + \sum_{i \in S_p} w_i d_p(G(P_i, X_{k+1}), P_i) \\ &< \sum_{j \in S_q} w_j d_q(Q_j, X_{k+1}) + \sum_{i \in S_p} w_i d_q(X_{k+1}, G(P_i, X_k)) + \sum_{i \in S_p} w_i d_p(G(P_i, X_k), P_i) \\ &< \sum_{j \in S_q} w_j d_q(Q_j, X_k) + \sum_{i \in S_p} w_i d_q(X_k, G(P_i, X_k)) + \sum_{i \in S_p} w_i d_p(G(P_i, X_k), P_i) \\ &= D(X_k) \end{aligned}$$

Por lo tanto,  $\{X_k\}_{k \in \mathbb{N}}$  es una sucesión infinita de puntos en un compacto y tiene una subsucesión convergente a un punto  $\lim_{n \rightarrow \infty} X_{k_n} = X^* \in \Omega_q$ .

Dado que la sucesión  $D(X_{k_n})$  es estrictamente decreciente de términos positivos se cumple que  $\lim_{n \rightarrow \infty} [D(X_{k_n}) - D(WP_q(X_{k_n}))] = 0$ .

Por la continuidad en  $\Omega_q$  de la función  $WP_q$  tenemos  $WP_q(\lim_{n \rightarrow \infty} X_{k_n}) = WP_q(X^*)$

Por tanto,  $D(X^*) - D(WP_q(X^*)) = 0$ .

Sigue que  $X^* = WP_q(X^*)$  pues, de lo contrario, según la anterior cadena de desigualdades, tendríamos que  $D(WP_q(X^*)) < D(X^*)$ , en contra de la igualdad demostrada. Por lo tanto, según el teorema 3.5,  $X^*$  es la solución del problema (1).

Además, si existe otra subsucesión convergente a  $X'$ , razonando de forma análoga, tenemos que  $X' = WP_q(X')$ , y  $X'$  también es solución del problema (1).

Como la función objetivo es estrictamente convexa tiene un mínimo absoluto único y, en consecuencia,  $X^* = X' = \lim_{k \rightarrow \infty} X_k$  es el único punto que verifica la condición necesaria y suficiente, y por tanto es la solución en  $\Omega_q$  del problema (1). ■

Nótese que si  $1 < p < q < \infty$  este razonamiento es válido para  $\Omega_p$  y nos permite construir un algoritmo de búsqueda de la solución en ambas regiones. Sin embargo, como pusimos de manifiesto con los ejemplos anteriores, no es posible aplicarlo en regiones con norma  $\ell_1$  o  $\ell_\infty$ , y tendremos que estudiar estas situaciones como casos particulares. Comenzaremos estudiando el caso  $1 = p < q < \infty$ .

### 3.2.1 La solución del problema con las normas $\ell_1$ y $\ell_q$

Anteriormente hemos puesto un contraejemplo que ilustra la no validez del teorema 3.5 para la región con norma  $\ell_1$ . Este hecho se debe a dos causas:

- En primer lugar,  $WP_1(X)$  es el conjunto de soluciones de un problema de localización simple en una región con norma  $\ell_1$ , que se resuelve hallando las medianas de los puntos de  $\mathcal{P} \subset \Omega_1$  y  $\{G(X, Q_j)\}_{j \in S_q} \subset r$ . Dado que la mediana no es necesariamente única, el conjunto de soluciones es, en general, un intervalo bidimensional que denotaremos  $[m_X, M_X] \times [m_Y, M_Y]$  y no sabremos que punto elegir, pues cualquier punto  $Y \in [m_X, M_X] \times [m_Y, M_Y]$  cumple que  $WP_1(Y) = Y$ .
- En segundo lugar, aunque el intervalo mediano anterior se reduzca a un solo punto, tendremos una infinidad de puntos para los que se verifica  $WP_1(Y) = Y$ . La elección

### 3.2.1 La solución del problema con las normas $\ell_1$ y $\ell_q$

---

arbitraria de un punto inicial  $X_0 \in \Omega_1$  que genere una sucesión  $\{X_k\}_{k \in \mathbb{N}}$ , según el teorema 3.6, conduce indefectiblemente (con probabilidad 1) a obtener un punto que no es solución del problema (1) en  $\Omega_1$  y sin embargo verifica que  $WP_1(X_1) = X_1$ .

Vamos a justificar a continuación estas observaciones, que son consecuencia de la proposición 3.8.

**Definición 3.7.** Sean  $\mathcal{P} \subset \Omega_1$  y  $\mathcal{Q} \subset \Omega_q$ , con  $1 < q \leq \infty$ , dos conjuntos de puntos separados por la recta  $r \equiv y = mx$ . Sea  $\mathcal{Q}^* \subset r$  el conjunto de proyecciones de los puntos de  $\mathcal{Q}$  según la pendiente  $m^*$  del teorema 2.23, es decir:

$$\mathcal{Q}_j^* = \left( \frac{m^* c_j - d_j}{m^* - m}, \frac{m(m^* c_j - d_j)}{m^* - m} \right), \text{ donde } (c_j, d_j) = \mathcal{Q}_j, j \in S_q.$$

Se define entonces  $WP_1(\mathcal{P}, \mathcal{Q}^*) \subset \Omega_1 \cup r$  como el conjunto de puntos que son solución del problema de Weber-Fermat siguiente:

$$\min_{X \in \mathbb{R}^2} \left\{ \sum_{i \in S_1} w_i d_1(P_i, X) + \sum_{j \in S_q} w_j d_1(X, \mathcal{Q}_j^*) \right\}.$$

Nótese que, para la norma  $\ell_1$ , este conjunto de puntos está definido por las medianas de las coordenadas de los puntos de  $\mathcal{P}$  y  $\mathcal{Q}^*$ . Dicho intervalo lo denotamos como:

$$WP_1(\mathcal{P}, \mathcal{Q}^*) = [med_x, Med_x] \times [med_y, Med_y].$$

Nótese también que si  $q = \infty$ , entonces  $m^* = -\text{signo}(m)$ , en virtud del teorema 2.28.

**Proposición 3.8.** Sean  $\mathcal{P} \subset \Omega_1$  y  $\mathcal{Q} \subset \Omega_q$ , con  $1 < q \leq \infty$ , dos conjuntos de puntos separados por la recta  $r \equiv y = mx$ , y supongamos que  $WP_1(\mathcal{P}, \mathcal{Q}^*) = \{M\} \subset \Omega_1$ . Entonces, el punto  $M = (M_x, M_y)$  determina en  $\Omega_1$  seis regiones, según ilustra la figura 3.3, tales que:

- Si  $X_0 \in (I)$ , entonces  $WP_1(X_0) \in (I)$ .
- Si  $X_0 \notin (I)$ , entonces  $WP_1(X_0) \in (I)$ .

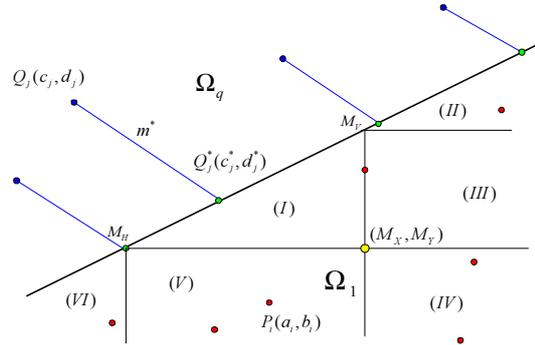


Figura 3.3: Regiones definidas en  $\Omega_1$  por la mediana de  $\mathcal{P}$  y  $\mathcal{Q}^*$ .

**Demostración.**

Sea  $\mathbf{u}^* = \{U_h^*(u_h^*, v_h^*)\}_{h \in S} = \mathcal{P} \cup \mathcal{Q}^*$ , con  $S = S_1 \cup S_q$ , y sea  $W = \sum_{i \in S_1} w_i + \sum_{j \in S_q} w_j$ .

Supongamos que  $X_0 = (x_0, y_0)$  se encuentra en (II), (III) ó (IV), regiones que no incluyen la línea vertical que pasa por  $M$ . En estos casos,  $M_X < x_0$  y entonces

$$\sum_{h \in S, u_h^* > x_0} w_h < W/2, \text{ ya que } M_X \text{ es la mediana de las abscisas } u_h^*.$$

Consideremos el conjunto  $\mathbf{u} = \{U_h(u_h, v_h)\}_{h \in S} \in \mathcal{P} \cup \mathcal{Q}$ , donde  $\mathcal{Q} = \{G(Q_j, X_0)\}_{j \in S_q}$ .

Teniendo en cuenta que los puntos de  $\mathcal{P}$  son fijos y que los puntos  $Q_j$  tales que  $u_j^* \leq x_0$ , verifican que  $u_j \leq x_0$  si  $G(Q_j, X_0) = (u_j, v_j)$  (ver figura 3.4), si  $WP_1(X_0)$  se encontrase en (II), (III) ó (IV), tendríamos una contradicción ya que

$$\sum_{h \in S, u_h > x_0} w_h \leq \sum_{h \in S, u_h^* > x_0} w_h < W/2 \text{ y } WP_1(X_0) \text{ no es mediana de los puntos } \mathbf{u}.$$

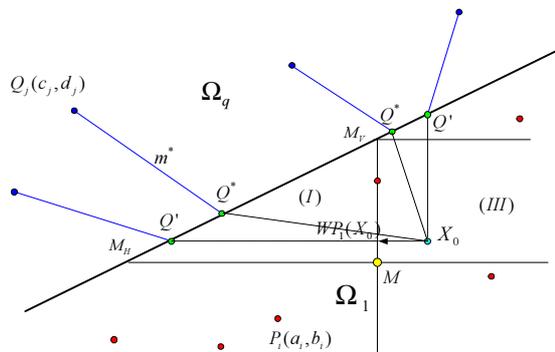


Figura 3.4: Desplazamiento de  $WP_1(X_0)$  respecto de  $X_0 \in (III)$ .

3.2.1 La solución del problema con las normas  $\ell_1$  y  $\ell_q$

En la figura 3.4, en la que suponemos pesos unitarios,  $WP_1(X_0)$  es el punto señalado en la recta vertical que pasa por  $M$ .

Si  $X_0$  se encuentra en (V) ó (VI) hacemos un razonamiento análogo al anterior, trabajando con las ordenadas en lugar de las abscisas. ■

**Nota 3.9.** Si  $X_0 \in (I)$ , entonces  $WP_1(X_0) = X_0$  y, por tanto,  $WP_1(M) = M$  (ver figuras 3.5a y 3.5b, por lo que usar el método expuesto en el teorema 3.6 nos puede conducir a un “agujero negro” como  $X_0$  o  $M$  y no obtenemos la solución del problema (1).

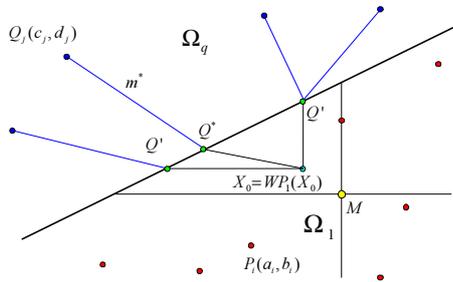


Figura 3.5a: Posición de  $WP_1(X_0)$  para  $X_0 \in (I)$ .

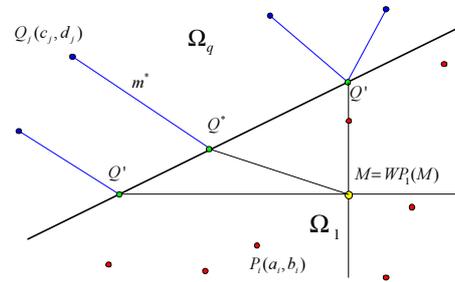


Figura 3.5b: Posición de  $WP_1(M)$ .

Notemos también que hemos supuesto que  $WP_1(\mathcal{P}, \mathcal{Q}^*) = \{M\} \subset \Omega_1$ . Puede ocurrir que  $WP_1(\mathcal{P}, \mathcal{Q}^*) = \{M\} \subset r \not\subset \Omega_1$ , con lo que el problema (1) no tiene solución en  $\Omega_1$ . En este caso se puede informar que la mejor solución en  $\Omega_1$  es un punto suficientemente cercano a  $M$ , pero no es la solución en  $\Omega_1$ .

Cuando la solución del problema (1) se encuentra en  $\Omega_1$  las consideraciones anteriores nos abocan a buscar la solución atacando directamente la derivada de la función objetivo en los puntos de  $\Omega_1$ .

Sean  $\mathcal{P} \subset \Omega_1$  y  $\mathcal{Q} \subset \Omega_q$ , con  $1 < q < \infty$ , dos conjuntos de puntos separados por la recta

$r \equiv y = mx$ , con  $m > 0$ , y sean  $W_1 = \sum_{i \in S_1} w_i$  y  $W_q = \sum_{j \in S_q} w_j$ . La función objetivo para un

punto  $X(x, y) \in \Omega_1$  es, según el Teorema 2.23:

$$D(X) = \sum_{i \in S_1} w_i d_1(X, P_i) + \sum_{j \in S_H} w_j d_1(X, G_H) + \sum_{j \in S_H} w_j d_q(G_H, Q_j) + \sum_{j \in S_P} w_j d_1(X, Q_j^*) + \sum_{j \in S_P} w_j d_q(Q_j^*, Q_j) + \sum_{j \in S_V} w_j d_1(X, G_V) + \sum_{j \in S_V} w_j d_q(G_V, Q_j),$$

donde  $S_H$ ,  $S_C$  y  $S_V$  son subconjuntos de índices de  $S_q$  tales que si  $j \in S_H$ , entonces  $Q_j \in \Omega_{qH}$ , si  $j \in S_C$ , entonces  $Q_j \in \Omega_{qC}$  y si  $j \in S_V$ , entonces  $Q_j \in \Omega_{qV}$ , según las consideraciones de la nota 2.24 (ver figura 3.6).

La función  $D$  es estrictamente convexa y, usando la definición de las distancias  $d_1$  y  $d_q$ , es una función de variables “separables” que podemos expresar explícitamente como sigue:

$$D(X) = D(x, y) = g(x) + h(y) + K$$

donde

$$g(x) = \sum_{i \in S_1} w_i |x - a_i| + \sum_{j \in S_H} w_j x + \sum_{j \in S_C} w_j (x - c_j^*) + m \sum_{j \in S_V} w_j x + \sum_{j \in S_V} w_j \sqrt{|x - c_j|^q + |mx - d_j|^q},$$

$$h(y) = \sum_{i \in S_1} w_i |y - b_i| - \frac{1}{m} \sum_{j \in S_H} w_j y + \sum_{j \in S_H} w_j \sqrt{\left|c_j - \frac{y}{m}\right|^q + |d_j - y|^q} + \sum_{j \in S_C} w_j (d_j^* - y) - \sum_{j \in S_V} w_j y$$

$$\text{y } K = \sum_{j \in S_C} w_j \sqrt{|c_j - c_j^*|^q + |d_j - d_j^*|^q}.$$

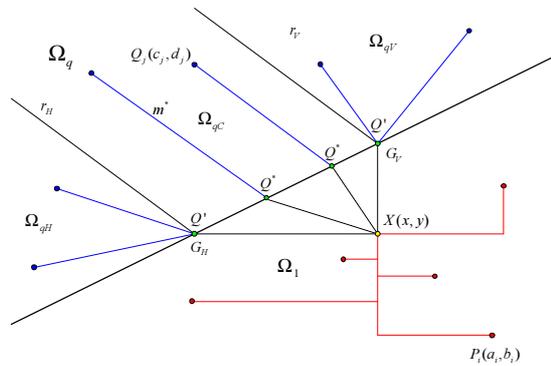


Figura 3.6: División de  $\Omega_q$  en regiones según  $X \in \Omega_1$ .

Nótese que estas tres funciones dependen implícitamente de ambas coordenadas, ya que  $X(x, y) \in \Omega_1$  determina las regiones  $\Omega_{qH}$ ,  $\Omega_{qC}$  y  $\Omega_{qV}$ . Si en un entorno de este punto las regiones no cambian, tampoco cambia la expresión de la función  $D$  y por

3.2.1 La solución del problema con las normas  $\ell_1$  y  $\ell_q$

tanto, tener un mínimo absoluto cuando  $\frac{\partial D(x, y)}{\partial x^+} \cdot \frac{\partial D(x, y)}{\partial x^-} \leq 0$  y

$\frac{\partial D(x, y)}{\partial y^+} \cdot \frac{\partial D(x, y)}{\partial y^-} \leq 0$  es equivalente a  $g'(x^+) \cdot g'(x^-) \leq 0$  y  $h'(x^+) \cdot h'(x^-) \leq 0$ , ya que

$K$  es constante.

Vamos a formalizar y precisar estas consideraciones:

**Proposición 3.10.** Sean  $\mathcal{P} \subset \Omega_1$  y  $\mathcal{Q} \subset \Omega_q$ , con  $1 < q < \infty$ , dos conjuntos de puntos separados por la recta  $r \equiv y = mx$ , con  $m > 0$ , y sea  $\mathcal{N} = \Omega_1 \cap (A \times B)$ , donde  $A = \{a_i\}_{i \in S_1} \cup \{c_j \mid d_j = mc_j\}_{j \in S_q}$  y  $B = \{b_i\}_{i \in S_1} \cup \{d_j \mid d_j = mc_j\}_{j \in S_q}$ . Entonces, la función objetivo  $D$  es derivable en  $\Omega_1 \setminus \mathcal{N}$  y tiene derivadas parciales con salto finito en  $\mathcal{N}$  (ver figura 3.7).

**Demostración.** Por la definición de la función objetivo,  $\frac{\partial D(x_0, y_0)}{\partial x} = g'(x_0)$  existe salvo en los puntos en los que el valor absoluto o la raíz se anule, es decir: si  $x_0 = a_i$  para algún  $i \in S_1$  o/y  $x_0 = c_j$  para algún  $j \in S_q$  tal que  $d_j = mc_j$ .

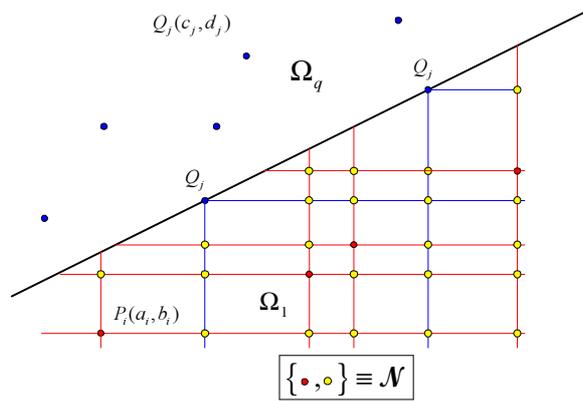


Figura 3.7: Puntos de  $\Omega_1$  sin derivada.

Sea  $(x_0, y_0) \in \Omega_1$  tal que  $x_0$  se encuentra en las condiciones anteriores, y sean  $\Omega_{qH}$ ,  $\Omega_{qC}$  y  $\Omega_{qV}$  las regiones que determina este punto en virtud del teorema 2.23.

Existe un  $\delta > 0$  tal que si  $x \in (x_0 - \delta, x_0 + \delta)$  se verifica:

- Las tres regiones que determina  $(x, y_0)$  contienen los mismos puntos que las anteriores salvo, a lo sumo, los puntos  $Q_j(c_j, d_j)$  tales que  $c_j^* = x_0$ .
- No existe  $Q_j(c_j, d_j)$  tal que  $c_j^* = x$ , ni existe  $P_i(a_i, b_i)$  tal que  $a_i = x$ .

Sea  $I_A(x)$  la función indicadora del conjunto  $A$ , y supongamos  $x > x_0$ , entonces:

$$g'(x) = \sum_{i \in S_1, a_i \neq x_0} w_i \operatorname{sgn}(x - a_i) + \sum_{i \in S_1} w_i I_{\{x_0\}}(a_i) + \sum_{j \in S_H} w_j + \sum_{j \in S_C} w_j + \sum_{j \in S_V} mw_j$$

$$+ \sum_{j \in S_V} w_j \frac{\operatorname{sgn}(x - c_j) |x - c_j|^{q-1} - m |mx - d_j|^{q-1}}{\left( \sqrt[q]{|x - c_j|^q} + |mx - d_j|^q \right)^{q-1}}.$$

$$g'(x) = \sum_{i \in S_1, a_i \neq x_0} w_i \operatorname{sgn}(x - a_i) + \sum_{i \in S_1} w_i I_{\{x_0\}}(a_i) + W_q +$$

$$\sum_{j \in S_V} w_j \left( m - 1 + \frac{\operatorname{sgn}(x - c_j) |x - c_j|^{q-1} - m |mx - d_j|^{q-1}}{\left( \sqrt[q]{|x - c_j|^q} + |mx - d_j|^q \right)^{q-1}} \right),$$

ya que  $\sum_{j \in S_H} w_j + \sum_{j \in S_C} w_j + \sum_{j \in S_V} mw_j = W_q + \sum_{j \in S_V} (m-1)w_j$ .

Por tanto,

$$g'(x_0^+) = \sum_{i \in S_1} w_i \operatorname{sgn}(x_0 - a_i) + \sum_{i \in S_1} w_i I_{\{x_0\}}(a_i) + W_q +$$

$$\sum_{j \in S_V} w_j \left( m - 1 + \frac{\operatorname{sgn}(x_0 - c_j) |x_0 - c_j|^{q-1} - m |mx_0 - d_j|^{q-1}}{\left( \sqrt[q]{|x_0 - c_j|^q} + |mx_0 - d_j|^q \right)^{q-1}} \right)$$

Sea  $x < x_0$ , y supongamos que existe  $h \in S_q$  tal que  $Q_h(c_h, d_h) \in r_V \subset \Omega_{qC}$  con  $r_V \equiv y - mx_0 = m^*(x - x_0)$ , entonces  $Q_h(c_h, d_h) \in \Omega'_{qV}$ , es decir,  $Q_h$  cambia de la región  $\Omega_{qC}$ , definida por  $(x_0, y_0)$ , a la región  $\Omega'_{qV}$ , definida por  $(x, y_0)$  (ver figura 3.8).

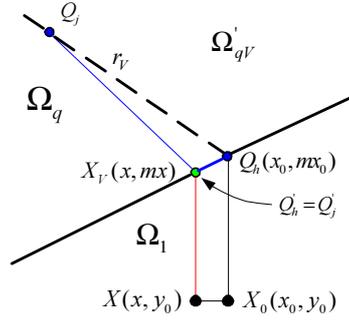


Figura 3.8: Cambio de posición de los portones al cambiar  $X \in \Omega_1$ .

En este caso,  $G((x, y_0), Q_h) = X_V = (x, mx)$ . Además,

- Si  $Q_h(c_h, d_h) \in r_V \cap r$ , entonces  $d_h = mc_h$  y por tanto,

$$d(X, Q_h) = mx - y_0 + \sqrt[p]{(c_h - x)^p + (mc_h - mx)^p} = mx - y_0 + (c_h - x)\sqrt[p]{1 + m^p},$$

y su derivada es:  $m - \sqrt[p]{1 + m^p}$ .

- Si  $Q_h(c_h, d_h) \in r_V \setminus r$ ,  $d(X, Q_h) = mx - y_0 + \sqrt[p]{|c_h - x|^p + |d_h - mx|^p}$ , y su derivada es:

$$m + \frac{\text{sgn}(x - c_h)|x - c_h|^{q-1} - m|mx - d_h|^{q-1}}{\left(\sqrt[q]{|x - c_h|^q + |mx - d_h|^q}\right)^{q-1}}.$$

Por tanto,

$$\begin{aligned} g'(x) = & \sum_{i \in S_1, a_i \neq x_0} w_i \text{sgn}(x - a_i) - \sum_{i \in S_1} w_i I_{\{x_0\}}(a_i) + \sum_{j \in S_H} w_j + \sum_{j \in S_C} w_j + \sum_{j \in S_V} mw_j + \\ & + \sum_{j \in S_V} w_j \frac{\text{sgn}(x - c_j)|x - c_j|^{q-1} - m|mx - d_j|^{q-1}}{\left(\sqrt[q]{|x - c_j|^q + |mx - d_j|^q}\right)^{q-1}} \\ & + \sum_{Q_j \in \eta_V \setminus r} w_j \left( m + \frac{\text{sgn}(x - c_h)|x - c_h|^{q-1} - m|mx - d_h|^{q-1}}{\left(\sqrt[q]{|x - c_h|^q + |mx - d_h|^q}\right)^{q-1}} \right) + \sum_{Q_j \in \eta_V \cap r} w_j \left( m - \sqrt[p]{1 + m^p} \right), \end{aligned}$$

donde  $S_{C'} = S_C \setminus \{j \in S_C \mid Q_j \in r_V\}$ . Agrupando:

$$\begin{aligned}
 g'(x) = & \sum_{i \in S_1, a_i \neq x_0} w_i \operatorname{sgn}(x - a_i) - \sum_{i \in S_1} w_i I_{\{x_0\}}(a_i) + \sum_{j \in S_H} w_j + \sum_{j \in S_C} w_j + \sum_{j \in S_V} w_j + \sum_{Q_j \in r_r} w_j + \\
 & + \sum_{j \in S_V} w_j \left( m - 1 + \frac{\operatorname{sgn}(x - c_j) |x - c_j|^{q-1} - m |mx - d_j|^{q-1}}{\left( \sqrt[q]{|x - c_j|^q + |mx - d_j|^q} \right)^{q-1}} \right) \\
 & + \sum_{Q_j \in r_r \setminus r} w_j \left( m - 1 + \frac{\operatorname{sgn}(x - c_j) |x - c_j|^{q-1} - m |mx - d_j|^{q-1}}{\left( \sqrt[q]{|x - c_j|^q + |mx - d_j|^q} \right)^{q-1}} \right) + w_h \left( m - 1 - \sqrt[q]{1 + m^q} \right).
 \end{aligned}$$

Teniendo en cuenta que,  $w_h = \sum_{j \in S_q} w_j I_{\{x_0\}}(c_j) I_{\{mx_0\}}(d_j)$  y que, si  $Q_j \in r_r \setminus r$ , entonces

$\frac{|mx_0 - d_j|}{|x_0 - c_j|} = |m^*|$ . En virtud del teorema 2.23, tenemos:

$$\lim_{x \rightarrow x_0} m - 1 + \frac{\operatorname{sgn}(x - c_j) |x - c_j|^{q-1} - m |mx - d_j|^{q-1}}{\left( \sqrt[q]{|x - c_j|^q + |mx - d_j|^q} \right)^{q-1}} = m - 1 + \frac{1 - m |m^*|^{q-1}}{\sqrt[q]{(1 + |m^*|^q)^{q-1}}} = 0,$$

con lo que,

$$\begin{aligned}
 g'(x_0^-) = & \sum_{i \in S_1} w_i \operatorname{sgn}(x_0 - a_i) - \sum_{i \in S_1} w_i I_{\{x_0\}}(a_i) + W_q + \sum_{j \in S_2} w_j \left( m - 1 - \sqrt[q]{1 + m^q} \right) I_{\{x_0\}}(c_j) I_{\{mx_0\}}(d_j) \\
 & + \sum_{j \in S_V} w_j \left( m - 1 + \frac{\operatorname{sgn}(x_0 - c_j) |x_0 - c_j|^{q-1} - m |mx_0 - d_j|^{q-1}}{\left( \sqrt[q]{|x_0 - c_j|^q + |mx_0 - d_j|^q} \right)^{q-1}} \right).
 \end{aligned}$$

Análogamente:

$$\begin{aligned}
 h'(y_0^+) = & \sum_{i \in S_1} w_i \operatorname{sgn}(y_0 - b_i) + \sum_{i \in S_1} w_i I_{\{y_0\}}(b_i) - W_q + \sum_{j \in S_q} w_j I_{\{y_0/m\}}(c_j) I_{\{y_0\}}(d_j) \left( 1 - \frac{1}{m} + \sqrt[q]{\frac{1}{m^q} + 1} \right) \\
 & + \sum_{j \in S_H} w_j \left( 1 - \frac{1}{m} + \frac{\frac{1}{m} \left| \frac{y_0}{m} - c_j \right|^{q-1} + \operatorname{sgn}(y_0 - d_j) |y_0 - d_j|^{q-1}}{\left( \sqrt[q]{\left| \frac{y_0}{m} - c_j \right|^q + |y_0 - d_j|^q} \right)^{q-1}} \right).
 \end{aligned}$$

### 3.2.2 La solución del problema con las normas $\ell_1$ y $\ell_\infty$

$$h'(y_0^-) = \sum_{i \in S_1} w_i \operatorname{sgn}(y_0 - b_i) - \sum_{i \in S_1} w_i I_{\{y_0\}}(b_i) - W_q$$

$$+ \sum_{j \in S_{II}} w_j \left( 1 - \frac{1}{m} + \frac{\frac{1}{m} \left| \frac{y_0}{m} - c_j \right|^{q-1} + \operatorname{sgn}(y_0 - d_j) |y_0 - d_j|^{q-1}}{\left( \sqrt[q]{\left| \frac{y_0}{m} - c_j \right|^q} + |y_0 - d_j|^q \right)^{q-1}} \right).$$

En conclusión,

$$\frac{\partial D(x_0, y_0)}{\partial x^+} = \frac{\partial D(x_0, y_0)}{\partial x^-} + 2 \cdot \sum_{i \in S_1} w_i I_{\{x_0\}}(a_i) - \sum_{j \in S_q} w_j I_{\{x_0\}}(c_j) I_{\{m \cdot x_0\}}(d_j) \left( m - 1 - \sqrt[q]{1 + m^q} \right), \text{ y}$$

$$\frac{\partial D(x_0, y_0)}{\partial y^+} = \frac{\partial D(x_0, y_0)}{\partial y^-} + 2 \cdot \sum_{i \in S_1} w_i I_{\{y_0\}}(b_i) + \sum_{j \in S_q} w_j I_{\{y_0/m\}}(c_j) I_{\{y_0\}}(d_j) \left( 1 - \frac{1}{m} + \sqrt[q]{\frac{1}{m^q} + 1} \right).$$

Si  $(x_0, y_0) \in \Omega_1 \setminus \mathcal{N}$  las derivadas laterales coinciden.

$$\text{En caso contrario, } \frac{\partial D(x_0, y_0)}{\partial x^+} > \frac{\partial D(x_0, y_0)}{\partial x^-} \text{ y } \frac{\partial D(x_0, y_0)}{\partial y^+} > \frac{\partial D(x_0, y_0)}{\partial y^-},$$

ya que,  $m - 1 - \sqrt[q]{1 + m^q} < 0$  y  $1 - \frac{1}{m} + \sqrt[q]{\frac{1}{m^q} + 1} > 0$ . ■

**Nota 3.11.** Para  $m < 0$  hacemos una simetría axial respecto del eje de ordenadas y obtenemos fórmulas análogas. Para rectas horizontales y verticales el problema está resuelto en Brimberg et al. (2003).

### 3.2.2 La solución del problema con las normas $\ell_1$ y $\ell_\infty$

Vamos a extender la proposición 3.10 al caso de norma  $\ell_\infty$ .

**Proposición 3.12.** Sean  $\mathcal{P} \subset \Omega_1$  y  $\mathcal{Q} \subset \Omega_\infty$ , dos conjuntos de puntos separados por la recta  $r \equiv y = mx$ , con  $0 < m \leq 1$ . Sea  $\mathcal{N}_\infty = \Omega_1 \cap (A \times B')$ , donde

$$A' = \{a_i\}_{i \in S_1} \cup \left\{ c_j^* \mid c_j^* = \frac{c-d}{1-m} \right\}_{j \in S_\infty} \cup \left\{ c_j^* \mid c_j^* = \frac{c+d}{1+m} \right\}_{j \in S_\infty} \text{ y}$$

$$B' = \{b_i\}_{i \in S_1} \cup \left\{ d_j^* \mid d_j^* = \frac{m(c+d)}{1+m} \right\}_{j \in S_\infty}.$$

Entonces, la función objetivo  $D$  es derivable en  $\Omega_1 \setminus \mathcal{N}_\infty$  y tiene derivadas parciales con salto finito en  $\mathcal{N}_\infty$ .

**Demostración.**

Consideremos en  $\Omega_\infty$  las regiones disjuntas  $\Omega_H, r_h, \Omega_C, r_1, \Omega_V, r_2$  y  $\Omega_{V'}$ , con sus correspondientes conjuntos de índices  $S_H, S_{rh}, S_C, S_{r1}, S_V, S_{r2}$  y  $S_{V'}$ , (ver figura 3.9). Sea  $S_H^* = S_{rh} \cup S_H$ ,  $S_V^* = S_{r1} \cup S_V \cup S_{r2} \cup S_{V'}$ , y  $S_{r1}'$  el conjunto  $S_{r1}$  del que hemos excluido el punto  $Q_0 = (x_0, mx_0)$ , con peso  $w_0$ , si éste pertenece a  $\mathcal{Q}$ , en caso contrario  $w_0 = 0$ .

Con esta notación la función objetivo es:

$$D(X) = D(x, y) = g(x) + h(y) + K,$$

donde

$$g(x) = \sum_{i \in S_1} w_i |x - a_i| + \sum_{j \in S_H^*} w_j x + \sum_{j \in S_C} w_j (x - c_j^*) + \sum_{j \in S_V^*} w_j \left( m + \max\{|x - c_j|, |mx - d_j|\} \right),$$

$$h(y) = \sum_{i \in S_1} w_i |y - b_i| + \sum_{j \in S_H^*} w_j \left( -\frac{y}{m} + \max\left\{ \left| c_j - \frac{y}{m} \right|, |d_j - y| \right\} \right) + \sum_{j \in S_C} w_j (d_j^* - y) - \sum_{j \in S_V^*} w_j y,$$

y  $K = \sum_{j \in S_C} w_j (c_j - c_j^*)$ .

Supongamos  $x > x_0$ . Si se cumple que:

$$Q_j \in r_1, \text{ entonces } \max\{|x - c_j|, |mx - d_j|\} = x - c_j.$$

$$Q_j \in (\Omega_V \cup r_2) \setminus r_1, \text{ entonces } \max\{|x - c_j|, |mx - d_j|\} = d_j - mx.$$

$$Q_j \in \Omega_{V'}, \text{ entonces } \max\{|x - c_j|, |mx - d_j|\} = c_j - x.$$

Sea  $I_A(x)$  la función indicadora del conjunto  $A$ . Derivamos y obtenemos:

### 3.2.2 La solución del problema con las normas $\ell_1$ y $\ell_\infty$

$$g'(x) = \sum_{i \in S_1, a_i \neq x_0} w_i \operatorname{sgn}(x - a_i) + \sum_{i \in S_1} w_i I_{\{x_0\}}(a_i) + \sum_{j \in S_{rh}} w_j + \sum_{j \in S_H} w_j + \sum_{j \in S_C} w_j \\ + \sum_{j \in S^*} m w_j + \sum_{j \in S_{r_1}} w_j - m \sum_{j \in S_{r_1} \cup S_{r_2}} w_j - \sum_{j \in S_{r'}} w_j$$

$$\frac{\partial D(x_0, y_0)}{\partial x^+} = \sum_{i \in S_1} w_i \operatorname{sgn}(x_0 - a_i) + \sum_{i \in S_1} w_i I_{\{x_0\}}(a_i) + W_{rh} + W_H + W_C + W_{r_1} + (m-1)W_{r'}$$

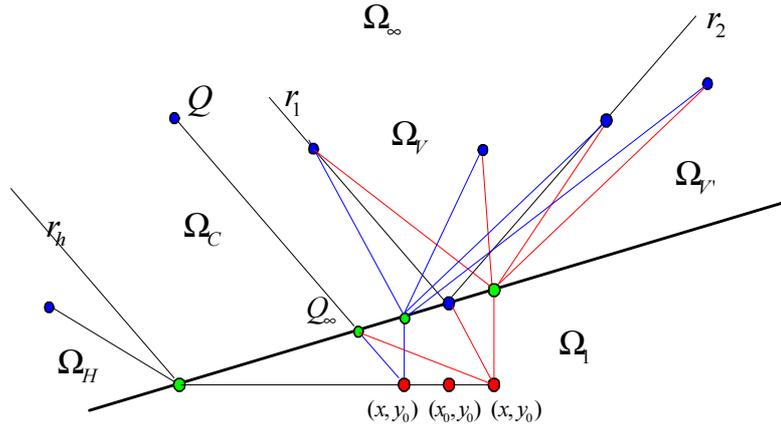


Figura 3.9: Cambio de los portones para  $x > x_0$ .

Supongamos ahora que  $x < x_0$ . Si se cumple que:

$$Q_j \in r_1 \setminus r, \text{ entonces } \max\{|x - c_j|, |mx - d_j|\} = d_j - mx.$$

$$Q_j \in r_1 \cap r, \text{ entonces } Q_j \equiv Q_0, \text{ y por tanto: } \max\{|x - c_0|, |mx - d_0|\} = c_0 - x.$$

$$Q_j \in \Omega_{r'}, \text{ entonces } \max\{|x - c_j|, |mx - d_j|\} = d_j - mx.$$

$$Q_j \in \Omega_{r'} \cup r_2, \text{ entonces } \max\{|x - c_j|, |mx - d_j|\} = c_j - x.$$

Luego:

$$g'(x) = \sum_{i \in S_1, a_i \neq x_0} w_i \operatorname{sgn}(x - a_i) - \sum_{i \in S_1} w_i I_{\{x_0\}}(a_i) + \sum_{j \in S_{rh}} w_j + \sum_{j \in S_H} w_j + \sum_{j \in S_C} w_j \\ + \sum_{j \in S^*} m w_j - m \sum_{j \in S_{r_1}} w_j - w_0 - m \sum_{j \in S_{r'}} w_j - \sum_{j \in S_{r'} \cup S_{r_2}} w_j.$$

$$\frac{\partial D(x_0, y_0)}{\partial x^-} = \sum_{i \in S_1} w_i \operatorname{sgn}(x_0 - a_i) - \sum_{i \in S_1} w_i I_{\{x_0\}}(a_i) + W_{rh} + W_H + W_C + (m-1)(w_0 + W_{r_2} + W_{r'}).$$

De manera análoga, obtenemos para  $y$ ,

$$\frac{\partial D(x_0, y_0)}{\partial y^-} = \sum_{i \in S_1} w_i \operatorname{sgn}(y_0 - b_i) - \sum_{i \in S_1} w_i I_{\{y_0\}}(b_i) + W_H - W_\infty.$$

$$\frac{\partial D(x_0, y_0)}{\partial y^+} = \sum_{i \in S_1} w_i \operatorname{sgn}(y_0 - b_i) + \sum_{i \in S_1} w_i I_{\{y_0\}}(b_i) + W_{rh} + W_H - W_\infty.$$

En conclusión,

$$\frac{\partial D(x_0, y_0)}{\partial x^+} = \frac{\partial D(x_0, y_0)}{\partial x^-} + 2 \cdot \sum_{i \in S_1} w_i I_{\{x_0\}}(a_i) + W_{r1} - (m-1)(w_0 + W_{r2}), \text{ y}$$

$$\frac{\partial D(x_0, y_0)}{\partial y^+} = \frac{\partial D(x_0, y_0)}{\partial y^-} + 2 \cdot \sum_{i \in S_1} w_i I_{\{y_0\}}(b_i) + W_{rH}.$$

Si  $(x_0, y_0) \in \Omega_1 \setminus \mathcal{N}_\infty$  las derivadas laterales coinciden.

En caso contrario,  $\frac{\partial D(x_0, y_0)}{\partial x^+} > \frac{\partial D(x_0, y_0)}{\partial x^-}$  y  $\frac{\partial D(x_0, y_0)}{\partial y^+} > \frac{\partial D(x_0, y_0)}{\partial y^-}$ ,

ya que,  $m-1 < 0$ , y  $W_{r1}, w_0$  y  $W_{r2}$  es suma de pesos de puntos de  $A'$  y  $W_{rH}$  es suma de pesos de puntos de  $B'$ . ■

### 3.3 Método de obtención en $\Omega_1$ de la solución del problema

El conocimiento de las derivadas laterales en los puntos de  $\Omega_1$  será de gran utilidad en los algoritmos que construiremos en el capítulo 4. A continuación vamos a utilizar dicho conocimiento para delimitar en  $\Omega_1$  una región en la que se pueda encontrar la solución del problema (1).

**Proposición 3.13.** Sean  $\mathcal{P} \subset \Omega_1$  y  $\mathcal{Q} \subset \Omega_q$ , con  $1 < q \leq \infty$ , dos conjuntos de puntos separados por la recta  $r \equiv y = mx$ , con  $m > 0$ . Consideremos también el conjunto  $WP_1(\mathcal{P}, \mathcal{Q}^*) = [\operatorname{med}_x, \operatorname{Med}_x] \times [\operatorname{med}_y, \operatorname{Med}_y]$  según la definición 3.7. Entonces, si

### 3.3 Método de obtención en $\Omega_1$ de la solución del problema

$X_s = (x_s, y_s) \in \Omega_1$  es solución del problema (1) se verifica que,  $x_s \leq med_x$  e  $y_s \geq Med_y$  (ver figura 3.10).

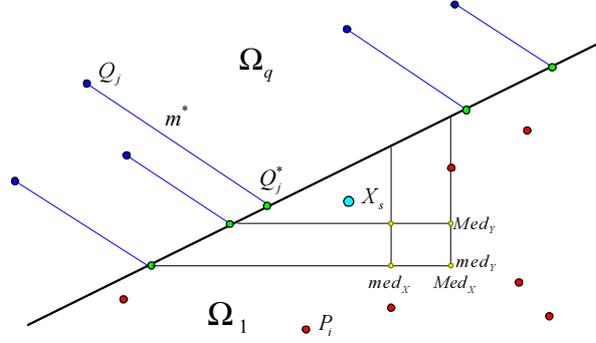


Figura 3.10: Posición relativa del punto solución  $X_s \in \Omega_1$ .

**Demostración.** Sea  $X_0 = (x_0, y_0) \in \Omega_1$  y  $1 < q < \infty$ . En virtud de la proposición 3.10:

$$\frac{\partial D(x_0, y_0)}{\partial x^+} = \sum_{a_i \leq x_0} w_i - \sum_{a_i > x_0} w_i + W_q + \sum_{j \in S_f} w_j \left( m-1 + \frac{\text{sgn}(x_0 - c_j) |x_0 - c_j|^{q-1} - m |mx_0 - d_j|^{q-1}}{\left( \sqrt[q]{|x_0 - c_j|^q + |mx_0 - d_j|^q} \right)^{q-1}} \right).$$

$$\text{Si } c_j > x_0, \text{ entonces } \frac{\text{sgn}(x_0 - c_j) |x_0 - c_j|^{q-1} - m |mx_0 - d_j|^{q-1}}{\left( \sqrt[q]{|x_0 - c_j|^q + |mx_0 - d_j|^q} \right)^{q-1}} = -\frac{1 + m \cdot m_j^{q-1}}{\left( \sqrt[q]{1 + m_j^q} \right)^{q-1}},$$

$$\text{con } m_j = \left| \frac{mx_0 - d_j}{x_0 - c_j} \right| \geq m > 0.$$

Según la proposición 2.22, se verifica que  $\frac{1 + m \cdot x^{q-1}}{\left( \sqrt[q]{1 + x^q} \right)^{q-1}} \leq \sqrt[q]{1 + m^q}$ , y por tanto,

$$-\frac{1 + m \cdot m_j^{q-1}}{\left( \sqrt[q]{1 + m_j^q} \right)^{q-1}} \geq -\sqrt[q]{1 + m^q}.$$

$$\text{Si } c_j = x_0, \text{ entonces } \frac{\text{sgn}(x_0 - c_j) |x_0 - c_j|^{q-1} - m |mx_0 - d_j|^{q-1}}{\left( \sqrt[q]{|x_0 - c_j|^q + |mx_0 - d_j|^q} \right)^{q-1}} = -m > -\sqrt[q]{1 + m^q},$$

ya que el denominador no se anula en  $\Omega_{qV}$ .

Análogamente, si  $c_j < x_0$ , entonces

$$\frac{\operatorname{sgn}(x_0 - c_j) |x_0 - c_j|^{q-1} - m |mx_0 - d_j|^{q-1}}{\left( \sqrt[q]{|x_0 - c_j|^q + |mx_0 - d_j|^q} \right)^{q-1}} = \frac{1 - m \cdot m_j^{q-1}}{\left( \sqrt[q]{1 + m_j^q} \right)^{q-1}} > \frac{-1 - m \cdot m_j^{q-1}}{\left( \sqrt[q]{1 + m_j^q} \right)^{q-1}} \geq -\sqrt[q]{1 + m_j^q}.$$

De este modo,

$$\frac{\partial D(x_0, y_0)}{\partial x^+} \geq \sum_{a_i \leq x_s} w_i - \sum_{a_i > x_s} w_i + W_q + \sum_{j \in S_r} w_j \left( m - 1 - \sqrt[q]{1 + m_j^q} \right), \text{ con } m - 1 - \sqrt[q]{1 + m_j^q} > -2.$$

Por lo tanto,

$$\frac{\partial D(x_0, y_0)}{\partial x^+} \geq \sum_{a_i \leq x_0} w_i - \sum_{a_i > x_0} w_i + \sum_{c_j^* \leq x_0} w_j + \sum_{c_j^* > x_0} w_j - 2 \sum_{j \in S_r} w_j = \sum_{a_i \leq x_0} w_i - \sum_{a_i > x_0} w_i + \sum_{c_j^* \leq x_0} w_j - \sum_{c_j^* > x_0} w_j,$$

sigue que:

$$\frac{\partial D(\operatorname{med}_X, y_0)}{\partial x^+} \geq \sum_{a_i \leq \operatorname{med}_X} w_i + \sum_{c_j^* \leq \operatorname{med}_X} w_j - \sum_{a_i > \operatorname{med}_X} w_i - \sum_{c_j^* > \operatorname{med}_X} w_j \geq \frac{W}{2} - \frac{W}{2} = 0,$$

con  $W = \sum_{i \in S_1} w_i + \sum_{j \in S_q} w_j$ . Por la convexidad de  $D$ , la solución óptima es  $x_s \leq \operatorname{med}_X$ .

Análogamente se demuestra que  $y_s \geq \operatorname{Med}_Y$ .

Supongamos ahora que  $q = \infty$ . En virtud de la proposición 3.12:

$$\frac{\partial D(x_0, y_0)}{\partial x^+} = \sum_{i \in S_1} w_i \operatorname{sgn}(x_0 - a_i) + \sum_{i \in S_1} w_i I_{\{x_0\}}(a_i) + W_{rh} + W_H + W_C + W_{r1} + (m-1)W_{V2}.$$

$$\frac{\partial D(x_0, y_0)}{\partial x^+} = \sum_{a_i \leq x_0} w_i - \sum_{a_i > x_0} w_i + \sum_{c_j^* \leq x_0} w_j + (m-1)W_{V2} \geq \sum_{a_i \leq x_0} w_i - \sum_{a_i > x_0} w_i + \sum_{c_j^* \leq x_0} w_j - W_{V2}.$$

Como  $W_{V2} \leq \sum_{c_j^* > x_0} w_j$ , se cumple que  $\frac{\partial D(x_0, y_0)}{\partial x^+} \geq \sum_{a_i \leq x_0} w_i - \sum_{a_i > x_0} w_i + \sum_{c_j^* \leq x_0} w_j - \sum_{c_j^* > x_0} w_j$ .

### 3.3 Método de obtención en $\Omega_1$ de la solución del problema

---

Por tanto:

$$\frac{\partial D(\text{med}_X, y_0)}{\partial x^+} \geq \sum_{a_i \leq \text{med}_X} w_i + \sum_{c_j^* \leq \text{med}_X} w_j - \sum_{a_i > \text{med}_X} w_i - \sum_{c_j^* > \text{med}_X} w_j \geq \frac{W}{2} - \frac{W}{2} = 0 \Rightarrow x_s \leq \text{med}_X.$$

Por otra parte,

$$\frac{\partial D(x_0, y_0)}{\partial y^-} = \sum_{i \in S_1} w_i \text{sgn}(y_0 - b_i) - \sum_{i \in S_1} w_i I_{\{y_0\}}(b_i) + W_H - W_\infty.$$

$$\begin{aligned} \frac{\partial D(x_0, y_0)}{\partial y^-} &= \sum_{b_i < y_0} w_i - \sum_{b_i \geq y_0} w_i + \sum_{d_j^* < y_0} w_j - \sum_{j \in S_\infty} w_j = \sum_{b_i < y_0} w_i - \sum_{b_i \geq y_0} w_i - \sum_{d_j^* \geq y_0} w_j \\ &\leq \sum_{b_i < y_0} w_i - \sum_{b_i \geq y_0} w_i - \sum_{d_j^* \geq y_0} w_j + \sum_{d_j^* < y_0} w_j. \end{aligned}$$

$$\frac{\partial D(x_0, \text{Med}_Y)}{\partial y^-} \leq \sum_{b_i < \text{Med}_Y} w_i - \sum_{b_i \geq \text{Med}_Y} w_i - \sum_{d_j^* \geq \text{Med}_Y} w_j + \sum_{d_j^* < \text{Med}_Y} w_j \leq \frac{W}{2} - \frac{W}{2} = 0.$$

Por la convexidad de la función objetivo, se tiene  $y_s \geq \text{Med}_Y$ . ■

La idea que preside los resultados que vamos a exponer a continuación consiste en dividir la región  $\Omega_1$  en rectángulos (regiones factibles) en los que la función objetivo tiene una expresión analítica única con variables separables y además sea diferenciable con derivadas parciales respecto de cada variable independiente de la otra variable, con lo que podremos aplicar algún algoritmo estándar para encontrar el mínimo absoluto de una función convexa. Las proposiciones 3.10 y 3.12 nos proporcionan los puntos en los que la función no es diferenciable y por tanto, serán vértices de dichos rectángulos. Además, tendremos que añadir los vértices correspondientes a valores de  $x$  e  $y$  en los que la función objetivo cambia de expresión analítica.

Para reducir considerablemente el número de rectángulos a estudiar, usaremos la proposición 3.13 junto con el hecho de que la solución del problema (1) se encuentra en la envolvente convexa rectangular de los puntos de  $\mathcal{P}$  y  $\mathcal{Q}$  (Brimberg et al. 2008), es decir, si  $X_s(x_s, y_s) \in \Omega_1$  es la solución del problema (1), entonces  $x_s \leq \text{med}_X$ ,

$y_s \geq Med_Y$ , y además,  $x_m \leq x_s \leq x_M$  e  $y_m \leq y_s \leq y_M$ , donde  $x_m = \min\{a_i, c_j\}_{i \in S_1, j \in S_q}$ ,

$x_M = \max\{a_i, c_j\}_{i \in S_1, j \in S_q}$ ,  $y_m = \min\{b_i, d_j\}_{i \in S_1, j \in S_q}$ , y  $y_M = \max\{b_i, d_j\}_{i \in S_1, j \in S_q}$ .

Sean  $\mathcal{A} = A_1 \cup A_2 \cup A_3$  y  $\mathcal{B} = B_1 \cup B_2 \cup B_3$  subconjuntos de  $\mathbb{R}$  definidos como sigue:

$A_1 = \{x_1, x_{\max}\}$ ,  $A_2 = \{a_i \mid x_1 \leq a_i \leq x_{\max}\}_{i \in S_1}$ ,  $A_3 = \{c_j^* \mid x_1 \leq c_j^* \leq x_{\max}\}_{j \in S_q}$ ,

$B_1 = \{Med_Y, m \cdot med_X\}$ ,  $B_2 = \{b_i \mid Med_Y \leq b_i \leq m \cdot med_X\}_{i \in S_1}$ ,

$B_3 = \{d_j^* \mid Med_Y \leq d_j^* \leq m \cdot med_X\}_{j \in S_q}$ ,

donde  $x_1 = \max\left\{\frac{Med_Y}{m}, x_m\right\}$  y  $x_{\max} = \min\{med_X, x_M\}$  (ver figura 3.11).

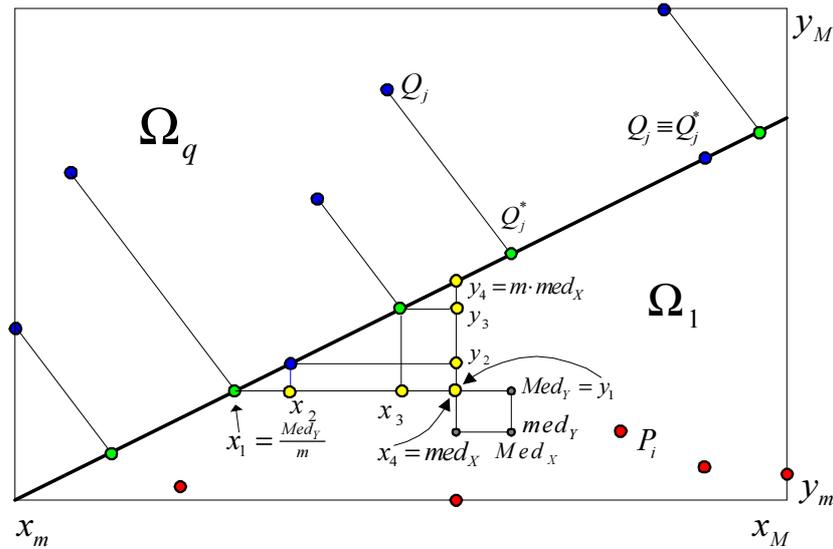


Figura 3.11: Regiones factibles para  $X_s \in \Omega_1$ .

**Teorema 3.14.** Sean  $\mathcal{P} \subset \Omega_1$  y  $\mathcal{Q} \subset \Omega_q$ , con  $1 < q \leq \infty$ , dos conjuntos de puntos separados por la recta  $r \equiv y = mx$ , con  $0 < m \leq 1$ . Supongamos que la solución  $X_s(x_s, y_s)$  del problema (1) se encuentra en  $\Omega_1$ , entonces existen dos índices  $h$  y  $t$  tales que:

$$x_h, x_{h+1} \in \mathcal{A}, y_{t-1}, y_t \in \mathcal{B} \text{ y } X_s \in I = (x_h, x_{h+1}] \times [y_{t-1}, y_t).$$

### 3.3 Método de obtención en $\Omega_1$ de la solución del problema

Además, la función objetivo tiene expresión analítica única y es dos veces derivable en el intervalo  $I$ .

#### **Demostración.**

Sea  $A^* = \{(x_1, Med_Y), (x_2, Med_Y), \dots, (x_k, Med_Y), (x_{max}, Med_Y)\} = \mathcal{A} \times \{Med_Y\}$  puntos de  $\Omega_1$  tales que  $x_1 < x_2 < \dots < x_k < x_{k+1} = x_{max}$ . Teniendo en cuenta que  $X_s \in \Omega_1$  y debido a la convexidad estricta y separabilidad de la función objetivo,

- $x_s = x_1$ , si  $\frac{\partial D(x_1, Med_Y)}{\partial x^+} \geq 0$ .
- En caso contrario  $x_s \in (x_h, x_{h+1}]$ , donde  $x_h = \max_{i=1, \dots, k} \left\{ x_i; \frac{\partial D(x_i, Med_Y)}{\partial x^+} < 0 \right\}$ .

Análogamente, consideremos ahora el conjunto de puntos de  $\Omega_1$ ,

$B^* = \{(med_X, y_1), (med_X, y_2), \dots, (med_X, y_l), (med_X, y_{l+1})\} = \{med_X\} \times \mathcal{B}$ , tales que:

$Med_Y = y_1 < y_2 < \dots < y_l < y_{l+1} = m \cdot x_{max}$ . Teniendo en cuenta que  $X_s \in \Omega_1$  y debido a la convexidad estricta y separabilidad de la función objetivo,

- $y_s = y_{l+1}$ , si  $\frac{\partial D(x_{max}, y_{l+1})}{\partial y^-} \leq 0$ .
- En caso contrario  $y_s \in [y_{l-1}, y_l)$ , donde  $y_l = \min_{j=2, \dots, l+1} \left\{ y_j; \frac{\partial D(x_{max}, y_j)}{\partial y^-} > 0 \right\}$ .

Además, la función es dos veces derivable y no cambia de expresión por la construcción de los intervalos. ■

**Nota 3.15.** En la implementación informática se calculan ambas derivadas laterales en cada punto de  $A^*$  y de  $B^*$ , de tal manera que podemos limitar la región factible para  $X_s \in \Omega_1$  a un rectángulo bidimensional, a un rectángulo unidimensional (segmento) e incluso a un único punto con lo que tendremos la solución exacta del problema (1) si ésta se encuentra en  $\Omega_1$ .



---

### ALGORITMOS DE LOCALIZACIÓN EN DOS REGIONES CON MÉTRICAS DISTINTAS

---

En el presente capítulo vamos a detallar diversos algoritmos, basados en los resultados teóricos de los capítulos anteriores sobre la caracterización de los portones entre dos regiones con métricas diferentes. Además, las expresiones de las derivadas parciales laterales obtenidas en el capítulo anterior nos permitirán encontrar la solución del problema (1) cuando intervienen regiones con normas  $\ell_1$  y  $\ell_\infty$ .

En todo el capítulo supondremos que  $\mathcal{P} = \{P_i = (a_i, b_i)\}_{i \in S_p} \subset \Omega_p$  y  $\mathcal{Q} = \{Q_j = (c_j, d_j)\}_{j \in S_q} \subset \Omega_q$ , son dos conjuntos de puntos con pesos  $w_i, w_j \in \mathbb{R}^+$ , separados por una recta  $r$ , tales que  $r \subset \Omega_q$ , región con norma  $\ell_q$ , y que  $r \cap \Omega_p = \emptyset$ , región con norma  $\ell_p$ , con  $1 \leq p < q \leq \infty$ . Supongamos también que  $r \equiv \omega_2 y + \omega_1 x + \omega_0 = 0$ , con  $\omega_1 \cdot \omega_2 \neq 0$  (recta oblicua).

Mediante un movimiento (traslación o/y simetría), que deja invariantes las distancias, podemos escribir  $r \equiv y = mx$ , con  $0 < m \leq 1$ , y supondremos que  $r(P_i) < 0$ ,  $i \in S_p$  y que  $r(Q_j) \geq 0$ ,  $j \in S_q$ , es decir,  $\Omega_p$  se encuentra por debajo de la recta y de  $\Omega_q$ .

## 4.1 Algoritmo GMFP (1, q)

El algoritmo GMFP (Gate MultiFacility Procedure) es una ligera modificación del propuesto por Brimberg et al. (2003), que denominamos MFP, en el que  $1 \leq p < q < \infty$ . En el capítulo I hicimos una somera introducción al algoritmo, y ahora vamos a detallarlo para cualquier valor de  $p$ , y así comprender mejor el nuevo algoritmo propuesto para el caso particular de  $p = 1$ .

Pretendemos resolver el problema de optimización:

$$\min_{X \in \mathbb{R}^2} \left\{ \sum_{i \in S_p} w_i d(P_i, X) + \sum_{j \in S_q} w_j d(X, Q_j) \right\}, \quad (1)$$

donde  $\{w_i\}_{i \in S_1} \subset \mathbb{R}^+$  y  $\{w_j\}_{j \in S_q} \subset \mathbb{R}^+$ .

Dado que este problema de optimización no es convexo (Parlar, 1994), los autores del método MFP descomponen el problema en dos problemas de optimización, suponiendo que la solución se encuentra en  $\Omega_q$  y suponiendo que se encuentra en  $\Omega_p$ .

Bajo el primer supuesto, es decir, la solución del problema (1) se encuentra en  $\Omega_q$ , el problema de optimización (1) se puede reformular de la siguiente forma:

$$\begin{aligned} D_q &= \min_{X \in \Omega_q} \left\{ \sum_{i \in S_p} w_i d(P_i, X) + \sum_{j \in S_q} w_j d_q(X, Q_j) \right\} \\ &= \min_{X \in \Omega_q, Z_i \in r} \left\{ \sum_{i \in S_p} w_i d_p(P_i, Z_i) + \sum_{i \in S_p} w_i d_q(Z_i, X) + \sum_{j \in S_q} w_j d_q(X, Q_j) \right\}, \end{aligned}$$

donde  $Z_i = (z_i, mz_i) \in r$  para cada  $i \in S_p$ . Tenemos por tanto el problema siguiente:

$$\min_{(x,y) \in \Omega_q, z_i \in \mathbb{R}} \left\{ \sum_{i \in S_p} w_i^p \sqrt{|z_i - a_i|^p + |mz_i - b_i|^p} + \sum_{i \in S_p} w_i^q \sqrt{|z_i - x|^q + |mz_i - y|^q} + \sum_{j \in S_q} w_j^q \sqrt{|x - c_j|^q + |y - d_j|^q} \right\},$$

Este es un problema convexo sin restricciones que se puede resolver modificando el procedimiento de Weiszfeld de localización múltiple y suavizando con una aproximación hiperbólica, ya que en los problemas localización múltiple el suavizado es prácticamente una necesidad (Love et al., 1988). La principal modificación respecto

#### 4.1 Algoritmo GMFP(1,q)

del método tradicional de Weiszfeld estriba en que tenemos dos normas en lugar de una única norma. La función objetivo suavizada,  $DH$ , con  $\varepsilon > 0$  queda:

$$DH(x, y, z_1, \dots, z_{|S_p|}) = \sum_{i \in S_p} w_i^p \sqrt{\left((z_i - a_i)^2 + \varepsilon\right)^{p/2} + \left((mz_i - b_i)^2 + \varepsilon\right)^{p/2}} \\ + \sum_{i \in S_p} w_i^q \sqrt{\left((x - z_i)^2 + \varepsilon\right)^{q/2} + \left((y - mz_i)^2 + \varepsilon\right)^{q/2}} \\ + \sum_{j \in S_q} w_j^q \sqrt{\left((x - c_j)^2 + \varepsilon\right)^{q/2} + \left((y - d_j)^2 + \varepsilon\right)^{q/2}}.$$

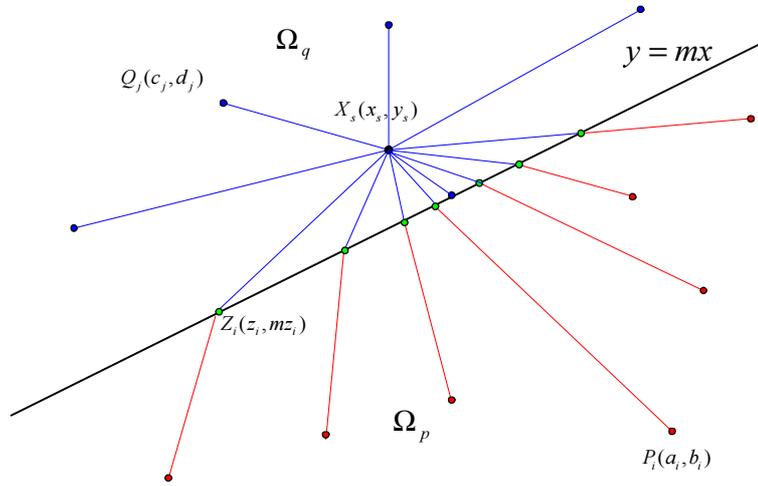


Figura 4.1: Punto solución en  $\Omega_q$ .

Para el algoritmo propuesto se igualan las  $|S_p| + 2$  derivadas parciales a cero y se obtiene un problema de punto fijo, resuelto de forma iterativa:

- Se eligen  $|S_p| + 2$  valores iniciales arbitrarios:  $x^0, y^0, z_i^0$ ,  $i \in S_p$  y  $\varepsilon > 0$ , con  $(x^0, y^0) \in \Omega_q$ . Para  $k = 0, 1, \dots$ , se consideran los siguientes valores:
- $A_{1i}^k = (z_i^k - a_i)^2 + \varepsilon$ ,  $A_{2i}^k = (mz_i^k - b_i)^2 + \varepsilon$ ,  $A_i^k = A_{1i}^{k \frac{p}{2}} + A_{2i}^{k \frac{p}{2}}$ ,  $i \in S_p$ ,
- $B_{1i}^k = (x^k - z_i^k)^2 + \varepsilon$ ,  $B_{2i}^k = (y^k - mz_i^k)^2 + \varepsilon$ ,  $B_i^k = B_{1i}^{k \frac{q}{2}} + B_{2i}^{k \frac{q}{2}}$ ,  $i \in S_p$ ,
- $C_{1j}^k = (x^k - c_j)^2 + \varepsilon$ ,  $C_{2j}^k = (y^k - d_j)^2 + \varepsilon$ ,  $C_j^k = C_{1j}^{k \frac{q}{2}} + C_{2j}^{k \frac{q}{2}}$ ,  $j \in S_q$ ,

y se obtienen de forma iterada las siguientes coordenadas:

$$\bullet \quad x^{k+1} = \frac{\left( \sum_{i \in S_p} z_i^k w_i B_i^{k^{\frac{1}{q}}} B_{1i}^{k^{\frac{q-1}{2}}} + \sum_{j \in S_q} c_j w_j C_j^{k^{\frac{1}{q}}} C_{1i}^{k^{\frac{q-1}{2}}} \right)}{\left( \sum_{i \in S_p} w_i B_i^{k^{\frac{1}{q}}} B_{1i}^{k^{\frac{q-1}{2}}} + \sum_{j \in S_q} w_j C_j^{k^{\frac{1}{q}}} C_{1i}^{k^{\frac{q-1}{2}}} \right)}, \quad (9)$$

$$y^{k+1} = \frac{\left( \sum_{i \in S_p} m z_i^k w_i B_i^{k^{\frac{1}{q}}} B_{2i}^{k^{\frac{q-1}{2}}} + \sum_{j \in S_q} d_j w_j C_j^{k^{\frac{1}{q}}} C_{2i}^{k^{\frac{q-1}{2}}} \right)}{\left( \sum_{i \in S_p} w_i B_i^{k^{\frac{1}{q}}} B_{2i}^{k^{\frac{q-1}{2}}} + \sum_{j \in S_q} w_j C_j^{k^{\frac{1}{q}}} C_{2i}^{k^{\frac{q-1}{2}}} \right)}, \quad (10)$$

$$z_i^{k+1} = \frac{A_i^{k^{\frac{1}{p}}} \left( a_i A_{1i}^{k^{\frac{p-1}{2}}} + m b_i A_{2i}^{k^{\frac{p-1}{2}}} \right) + B_i^{k^{\frac{1}{q}}} \left( x^k B_{1i}^{k^{\frac{q-1}{2}}} + m y^k B_{2i}^{k^{\frac{q-1}{2}}} \right)}{A_i^{k^{\frac{1}{p}}} \left( A_{1i}^{k^{\frac{p-1}{2}}} + m^2 A_{2i}^{k^{\frac{p-1}{2}}} \right) + B_i^{k^{\frac{1}{q}}} \left( B_{1i}^{k^{\frac{q-1}{2}}} + m^2 B_{2i}^{k^{\frac{q-1}{2}}} \right)}, \quad i \in S_p. \quad (11)$$

De esta forma, se genera una sucesión  $\{X^k, Z_i^k\}_{i \in S_p} \in \mathbb{R}^{2|S_p|+2}$ ,  $k = 0, 1, \dots$ , escrito de forma abreviada como la transformación:

$$\begin{aligned} MFP: \quad \mathbb{R}^{2|S_p|+2} &\rightarrow \mathbb{R}^{2|S_p|+2} \\ (X^k, Z_i^k)_{i \in S_p} &\rightarrow (X^{k+1}, Z_i^{k+1})_{i \in S_p} \end{aligned}$$

con  $X^k = (x^k, y^k) \in \Omega_q$  y  $Z_i^k = (z_i^k, m z_i^k) \in r$ ,  $i \in S_p$ , a partir de  $|S_p|+1$  puntos arbitrarios  $X^0 = (x^0, y^0) \in \Omega_q$  y  $Z_i^0 = (a_i, m a_i)$ ,  $i \in S_p$ , convergentes al punto fijo  $(X^*, Z_i^*)_{i \in S_p}$ . Si hubiésemos tomado  $\varepsilon = 0$  y la sucesión generada fuese no singular, es decir, no se anulan los denominadores de (9), (10) y (11), entonces el punto fijo sería  $(X_q, G(X_q, P_i))_{i \in S_p}$ , donde  $X_q$  es la solución en  $\Omega_q$  del problema (1) y  $\{G(X_q, P_i)\}_{i \in S_p}$  son las puertas de acceso correspondientes. Dado que es necesario tomar  $\varepsilon > 0$ , obtenemos una aproximación de la función objetivo como  $D_q \simeq DH(X^*, Z_1^*, \dots, Z_{|S_p|}^*)$ .

#### 4.1 Algoritmo GMFP(1,q)

---

Sin embargo, es conocido que si  $q > 2$ , el procedimiento iterativo puede no converger (Morris, 1981), (Brimberg y Love, 1992) y (Brimberg y Love, 1993). Por ello, en este trabajo consideramos un factor  $\lambda = \min\{1, 1/(q-1)\}$  que, para cualquier  $q > 1$ , asegura la convergencia usando las fórmulas de recurrencia:

$$\begin{aligned}x^{k+1} &:= \lambda x^{k+1} + (1-\lambda)x^k, \\y^{k+1} &:= \lambda y^{k+1} + (1-\lambda)y^k, \\z_i^{k+1} &:= \lambda z_i^{k+1} + (1-\lambda)z_i^k, \quad i \in S_p.\end{aligned}\tag{12}$$

Una discusión más detallada de la convergencia se estudia en la sección 4.7, al final de este capítulo.

Por otro lado, si suponemos que la solución se encuentra en  $\Omega_p$  procedemos de manera análoga y obtenemos el punto aproximado a  $X_p$  y un valor aproximado de la función objetivo  $D_p$ . Para finalizar obtenemos como punto solución del problema (1)

$$X_s = \arg \min_{X \in \mathbb{R}^2} \{D_p(X), D_q(X)\}.$$

Hasta aquí el método general, pero hemos de hacer notar que, en el caso  $p=1$ , el método MFP puede resultar excesivamente lento, sin embargo, el conocimiento inmediato y exacto de los portones (ver teorema 2.23) nos permite acelerar la convergencia de este método, lo que nos lleva a diseñar el algoritmo GMFP(1,q).

A partir de lo expuesto anteriormente, y suponiendo que la solución se encuentra en  $\Omega_q$  (análogamente en  $\Omega_1$ ), el pseudocódigo para obtener la solución del problema (1) queda como sigue:

- 1) Fijamos una tolerancia  $tol > 0$ , un número máximo de iteraciones  $N$ , un valor  $\varepsilon > 0$  y se elige un punto inicial arbitrario  $X^0 = (x^0, y^0) \in \Omega_q$ .
- 2) Se actualizan los portones  $Z_i^0 = G(P_i, X^0) = G_i^0$ , de acuerdo al teorema 2.23, se evalúa la función objetivo.  $DH_0 = DH(X^0, \mathbf{G}^0)$  e inicializamos  $k = 0$ .

3) Usando las fórmulas (9) y (10) obtenemos  $X^{k+1}$ , actualizamos los portones

$Z_i^{k+1} = G(P_i, X^{k+1}) = G_i^{k+1}$  y se evalúa la función objetivo

$$DH_{k+1} = DH(X^{k+1}, \mathbf{G}^{k+1}).$$

4) Si  $DH_k - DH_{k+1} < tol$  vamos al paso 6.

5) Si  $k+1 < N$ , hacemos  $k := k+1$  y vamos al paso 3.

6) Finaliza mostrando solución  $X_q \approx X^{k+1}$ , y el valor de la función objetivo

$$D_q \approx DH_{k+1}.$$

La diferencia sustancial del algoritmo GMFP con el original MFP estriba en el punto 3, ya que MFP utiliza en cada iteración la fórmula (11), mientras que este nuevo algoritmo calcula directamente el portón, con lo que se obtiene un valor inferior de la función objetivo:

$$DH\left(MFP\left(X^k, Z_i^k\right)_{i \in S_1}\right) = DH\left(\left(X^{k+1}, Z_i^{k+1}\right)_{i \in S_1}\right) > DH\left(\left(X^{k+1}, G_i^{k+1}\right)_{i \in S_1}\right) = DH\left(GMFP\left(X^k, Z_i^k\right)_{i \in S_1}\right)$$

## 4.2 Algoritmos Gate( $p, q$ )

En todos los algoritmos que exponemos a continuación vamos a descomponer el problema (1) en tres problemas:

P1) Buscamos la solución en la recta, es decir:

$$D_r = D(X_r) = \min_{X \in r} \left\{ \sum_{i \in S_p} w_i d_p(P_i, G(P_i, X)) + \sum_{i \in S_p} w_i d_q(G(P_i, X), X) + \sum_{j \in S_q} w_j d_q(X, Q_j) \right\}$$

Nótese que la trayectoria de cada  $P_i$  a  $X$  no se realiza necesariamente por  $\Omega_p$ , ya que, al usar el portón entre  $P_i$  y  $X$ , una parte del trayecto se realiza en  $\Omega_p$ , y se mide en norma  $\ell_p$ , mientras que la otra parte del trayecto se realiza en la recta y se mide en norma  $\ell_q$ , según vimos en la proposición 2.33 (ver figura 4.2).

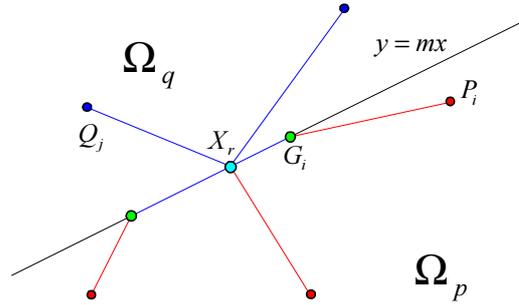


Figura 4.2: Punto solución  $X_r$ .

P2) Buscamos la solución en la región  $\Omega_q$  :

$$D_q = D(X_q) = \min_{X \in \Omega_q} \left\{ \sum_{i \in S_1} w_i d(P_i, X) + \sum_{j \in S_q} w_j d(X, Q_j) \right\}, \text{ es decir,}$$

$$D_q = \min_{X \in \Omega_q} \left\{ \sum_{i \in S_1} w_i d_1(P_i, G(P_i, X)) + \sum_{i \in S_1} w_i d_q(G(P_i, X), X) + \sum_{j \in S_q} w_j d_q(X, Q_j) \right\}.$$

P3) Buscamos la solución en la región  $\Omega_p$  :

$$D_p = D(X_p) = \min_{X \in \Omega_p} \left\{ \sum_{i \in S_p} w_i d(P_i, X) + \sum_{j \in S_q} w_j d(X, Q_j) \right\}, \text{ es decir,}$$

$$D_p = \min_{X \in \Omega_p} \left\{ \sum_{i \in S_p} w_i d_p(P_i, X) + \sum_{j \in S_q} w_j d_p(X, G(Q_j, X)) + \sum_{j \in S_q} w_j d_q(G(Q_j, X), Q_j) \right\}.$$

Una vez resueltos los tres problemas anteriores y, para finalizar, el punto solución del problema (1) es  $X_s = \arg \min \{D_p(X_p), D_q(X_q), D_r(X_r)\}$ .

Aunque se considera que la recta está incluida en  $\Omega_q$ , la experiencia nos demuestra que si la solución está en la recta, ésta no coincide con la encontrada en  $\Omega_q$ , ya que son métodos de aproximación que paran al alcanzar la tolerancia prefijada. Además, los tiempos de cómputo en la recta son muy reducidos y nos permiten alcanzar una mayor precisión con poco esfuerzo computacional.

A continuación desarrollamos diferentes algoritmos según donde consideremos que se encuentra la solución y los valores de  $p$  y  $q$ .

### 4.3 Algoritmo de búsqueda en la recta

La función objetivo tiene como expresión:

$$\begin{aligned}
 D(X) = D(x, mx) = & \sum_{j \in S_q} w_j \sqrt[q]{|x - c_j|^q + |mx - d_j|^q} + \sum_{i \in S_{pC}} w_i \sqrt[q]{|x - a_i|^p + |mx - b_i|^p} + \\
 & + \sum_{i \in S_{pV}} w_i \sqrt[q]{|x - a_i|^p + |mx - ma_i|^p} + \sum_{i \in S_{pV}} w_j \sqrt[q]{|x - a_i|^q + |mx - ma_i|^q} + \\
 & + \sum_{i \in S_{pH}} w_i \sqrt[q]{|x - a_i|^p + |mx - ma_i|^p} + \sum_{j \in S_{pH}} w_j \sqrt[q]{|x - b_j/m|^q + |mx - b_j|^q},
 \end{aligned}$$

donde  $S_{pH}$ ,  $S_{pC}$  y  $S_{pV}$  son los conjuntos de índices correspondientes a las regiones de  $\Omega_p$  definidas por  $X$  (ver proposición 2.33).

Para esta situación hemos realizado un algoritmo *ad hoc*, adaptando el método ARS a una recta y una función convexa, de la siguiente forma:

1. Fijamos una tolerancia  $tol > 0$ , un número máximo de iteraciones  $N$ , un paso  $\lambda > 0$  suficientemente grande, dos direcciones en la recta:  $u_1 = (1, m)$  y  $u_2 = (-1, -m)$ , un punto inicial arbitrario  $X_0 = (x_0, mx_0)$ , hacemos  $X_r = X_0$ , hallamos portones y evaluamos la función objetivo  $D_0 = D(X_0) = D_r$  e inicializamos  $k, h = 0$ .
2. Hacemos  $X_{h+1, k+1} = X_k + \lambda \cdot u_{h+1}$ , actualizamos portones y evaluamos la función objetivo en dicho punto:  $D_{k+1} = D(X_{h+1, k+1})$ .
3. Si  $D_{k+1} < D_k$  y  $D_k - D_{k+1} < tol$ , hacemos  $X_r = X_{h+1, k+1}$ ,  $D_r = D_{k+1}$  y vamos al paso 8.
4. Si  $D_{k+1} < D_k$  y  $k+1 = N$ , hacemos  $X_r = X_{h+1, k+1}$ ,  $D_r = D_{k+1}$  y vamos al paso 8.
5. Si  $D_{k+1} < D_k$ , hacemos  $k := k+1$ ,  $h = 0$  y vamos al paso 2.

#### 4.4 Algoritmos de búsqueda en $\Omega_q$

---

6. Si  $h+1 < 2$ , hacemos  $h := h+1$  y vamos al paso 2.
7. Si  $h+1 = 2$ , hacemos  $\lambda := \lambda/2$ ,  $h = 0$  y vamos al paso 2.
8. Finaliza mostrando solución  $X_r$  y el valor de la función objetivo  $D_r = D(X_r)$ .

#### Notas:

1) En la implementación tomamos como punto inicial  $X_0 = (x_0, mx_0)$ , donde  $x_0$  es una mediana de los valores  $\{a_i^*, c_j^*\}_{i \in S_p, j \in S_q}$ , siendo  $a_i^*$  y  $c_j^*$  las abscisas de las proyecciones de  $P_i$  y  $Q_j$  en  $r$ , según  $m_p^g = -m|m|^{-p}$  y  $m_q^g = -m|m|^{-q}$ , respectivamente. Justificamos esta elección por dos motivos: en primer lugar, las líneas geodésicas o de mínima distancia tienen estas pendientes y en segundo lugar, el punto que minimiza la suma de distancias ponderadas en una recta es independiente de la métrica, por lo que tomamos el punto que minimiza dicha suma en  $\ell_1$ , es decir, la mediana.

2) Para obtener los portones, utilizaremos la proposición 2.33, lo que ha dado lugar a tres programas, según los valores de  $p$  y  $q$ :

$$\text{rectarspinf} \quad \text{si } 1 \leq p < q = \infty$$

$$\text{rectarslq} \quad \text{si } 1 = p < q < \infty$$

$$\text{rectarspq} \quad \text{si } 1 < p < q < \infty$$

## 4.4 Algoritmos de búsqueda en $\Omega_q$ ( $1 < q < \infty$ )

Mostramos a continuación dos algoritmos de búsqueda de la solución en  $\Omega_q$ , ambos están basados en el uso de los portones. El primer método se basa en el uso del teorema 3.5 (Condición necesaria y suficiente en  $\Omega_q$ ), mientras que el segundo método es una aplicación del método ARS construida con el objeto de comprobar la eficiencia del primer método.

Método 1:

Cualquiera que sea el valor de  $p \in [1, \infty]$ , con  $p \neq q$ , el pseudocódigo para encontrar la posible solución del problema (1) en  $\Omega_q$  es el siguiente:

1. Fijamos una tolerancia  $tol > 0$ , un número máximo de iteraciones  $N$ , y se elige un punto inicial arbitrario  $X_0 = (x_0, y_0) \in \Omega_q$ .
2. Se actualizan los portones  $G(P_i, X_0)$ , se evalúa la función objetivo.  $D_q(X_0)$  e inicializamos  $k = 0$ .
3. Hacemos  $X_{k+1} = WP_q(X_k)$ , actualizamos portones y evaluamos la función objetivo:  $D_q(X_{k+1})$ .
4. Si  $D_q(X_k) - D_q(X_{k+1}) < tol$  vamos al paso 6.
5. Si  $k+1 < N$ , hacemos  $k := k+1$  y vamos al paso 3.
6. Finaliza mostrando solución  $X_q \approx X_{k+1}$ , y el valor de la función objetivo  $D_q \approx D_q(X_{k+1})$ .

La convergencia del algoritmo está garantizada por la condición necesaria y suficiente de solución en  $\Omega_q$  (teoremas 3.5 y 3.6).

Notas:

1) En la implementación del algoritmo tomamos  $X_0 = \left( \frac{W_q \bar{x} + W_1 x_r}{W}, \frac{W_q \bar{y} + W_1 m x_r}{W} \right)$ , si

$$p=1, \text{ donde } W_1 = \sum_{i \in S_1} w_i, \quad W_q = \sum_{j \in S_q} w_j, \quad W = W_1 + W_q, \quad \bar{x} = \frac{\sum_{i \in S_1} w_i a_i^* + \sum_{j \in S_q} w_j c_j}{\sum_{i \in S_1} w_i + \sum_{j \in S_q} w_j} \text{ y}$$

$$\bar{y} = \frac{\sum_{i \in S_1} w_i m a_i^* + \sum_{j \in S_q} w_j d_j}{\sum_{i \in S_1} w_i + \sum_{j \in S_q} w_j}, \text{ es decir, } (\bar{x}, \bar{y}) \text{ es el centro de gravedad de los puntos}$$

$Q_j$  y  $P_i^* = (a_i^*, m a_i^*)$ , proyectados de los puntos  $P_i$  en la recta, según la pendiente  $m_p^g = -m|m|^{-p}$  de la línea geodésica. Además,  $x_r$  es la abscisa de la solución  $X_r \in r$ .

#### 4.4 Algoritmos de búsqueda en $\Omega_q$

---

La elección de la semilla ha surgido de la experimentación ya que tomamos inicialmente  $X_0 = (\bar{x}, \bar{y})$  pero las pruebas mostraban el acercamiento de la sucesión a la recta por lo que se consideró un promedio con la solución obtenida previamente en la recta, y se observó que con esta nueva semilla se aceleraba considerablemente la convergencia.

En el caso  $p > 1$ , simplificamos la elección de la semilla tomando un promedio entre

$$(\bar{x}, \bar{y}) \text{ y su proyección vertical en la recta, en concreto } X_0 = \left( \bar{x}, \frac{2\bar{y} + m\bar{x}}{3} \right).$$

2) Una segunda cuestión a tener en cuenta es el método para resolver el problema de Weber-Fermat:  $X_{k+1} = WP_q(X_k)$ , es decir,  $X_{k+1}$  es el punto solución del problema de optimización:

$$X_{k+1} = \arg \min_{(x,y) \in \mathbb{R}^2} \left\{ \sum_{j \in S_q} w_j \sqrt{|x - c_j|^q + |y - d_j|^q} + \sum_{i \in S_p} w_i \sqrt{|x - a_i^*|^q + |y - ma_i^*|^q} \right\},$$

donde  $(a_i^*, ma_i^*) = G(X_k, P_i)$ .

Hemos considerado dos alternativas para resolver este problema: el algoritmo de Weiszfeld (con suavizado hiperbólico), que estudiamos en la sección 4.7 de este capítulo, y el algoritmo ARS, elaborando así dos rutinas diferentes, *weiszfeldq* y *arsweis*, respectivamente. Lo más destacable del algoritmo ARS es la ausencia de preocupación en la singularidad de los puntos de la sucesión  $\{X_k\}$ , ya que no se utilizan las derivadas de la función. En su contra tiene la variabilidad de la solución obtenida, en función del número máximo de iteraciones prefijadas.

3) Por último, tenemos el problema de actualización de los portones  $G(P_i, X_k)$ . En el caso  $p = 1$ , el conocimiento inmediato y exacto de los portones, usando el teorema 2.23, acelera en gran medida la convergencia del algoritmo. En otro caso, usaremos la rutina *gates\_pq*, que devuelve el portón correspondiente a dos puntos cualesquiera  $P \in \Omega_p$  y  $Q \in \Omega_q$ .

### Método 2

Es similar al método de búsqueda en la recta, con las dos diferencias sustanciales siguientes:

- 1) El punto inicial que hemos tomado es  $X_0 \in \Omega_q$  definido en la nota anterior.
- 2) Los movimientos del punto inicial  $X_{h+1,k+1} = X_k + \lambda \cdot u_{h+1}$ , se realizarán en cuatro direcciones:  $u_1 = (1, -1)$ ,  $u_2 = (-1, -1)$ ,  $u_3 = (-1, 1)$  y  $u_4 = (1, 1)$ , con un  $\lambda$  suficientemente grande.

En los casos  $\Omega_1$  y  $\Omega_\infty$ , podemos aplicar el método 2 pero no así el método 1 ya que  $X = WP_1(X)$  ó  $X = WP_\infty(X)$  no garantiza que  $X$  sea la solución, como vimos en los ejemplos del teorema 3.5, por lo que vamos a detallar los algoritmos que hemos utilizado en estos casos.

## 4.5 Algoritmos de búsqueda en $\Omega_1$

Para encontrar la posible solución en  $\Omega_1$  hemos desarrollado dos algoritmos basados en el teorema 2.23 y la proposición 3.13. La diferencia sustancial entre ambos estriba en que uno de ellos no utiliza el teorema 3.14 mientras que el expuesto en primer lugar si lo utiliza y su desarrollo es como sigue:

1. A partir del teorema 2.23 obtenemos la pendiente  $m^*$  y, para aplicar la proposición 3.13, obtenemos el punto  $(med_X, Med_Y)$ .
2. Si  $(med_X, Med_Y) \in \Omega_1$  construimos la malla  $\mathcal{M} = A^* \times B^*$ , según el teorema 3.14. En caso contrario, hacemos  $(x^*, y^*) = (med_X, Med_Y)$  y vamos al paso 7.
3. Calculamos las derivadas parciales respecto de  $x$  e  $y$  en  $A^*$  y  $B^*$ , respectivamente y, aplicando el teorema 3.14, encontramos un punto  $I = (x^*, y^*)$ , ó un intervalo unidimensional  $I = \{x_n\} \times [y_{l-1}, y_l)$  ó  $I = (x_n, x_{n+1}] \times \{y_l\}$ , ó un intervalo bidimensional  $I = (x_n, x_{n+1}] \times [y_{l-1}, y_l)$  en el

#### 4.6 Algoritmos de búsqueda en $\Omega_\infty$

que se encuentra la posible solución ya que las derivadas laterales cambian de signo.

4. Si  $I \cap \Omega_1 = \emptyset$  la solución se encuentra en la recta, ya que es la frontera de la región abierta  $\Omega_1$  (ver figura 4.3) y vamos al paso 7.
5. Si  $I \cap \Omega_1 \neq \emptyset$ , entonces la solución es  $(x^*, y^*)$  si el intervalo se reduce a un punto. En caso contrario, podemos aplicar un algoritmo estándar para encontrar la solución, ya que en dicho intervalo la función es convexa, dos veces diferenciables y no cambia de expresión según el teorema 3.14. Sea entonces  $(x^*, y^*)$  la posible solución, obtenida según el algoritmo estándar.
6. Si  $(x^*, y^*) \in \Omega_1$ , entonces  $X_1 = (x^*, y^*)$ , calculamos  $D_1(X_1)$  y el algoritmo finaliza.
7. Si  $(x^*, y^*) \notin \Omega_1$ , podemos considerar un punto de  $\Omega_1$  como solución aproximada:  $X_1 = X_{r1} + \lambda(0, -1)$ , con  $\lambda$  suficientemente pequeño, donde  $X_{r1} \in r$  es el punto solución del siguiente problema de optimización:

$$\min_{X \in r} \left\{ \sum_{i \in S_1} w_i d_1(P_i, X) + \sum_{j \in S_q} w_j d_q(X, Q_j) \right\},$$

que difiere del punto solución en la recta, en que la trayectoria de cada  $P_i$  al punto  $X$  está íntegramente en  $\Omega_1$ . Calculamos  $D_1(X_1)$  y el algoritmo finaliza.

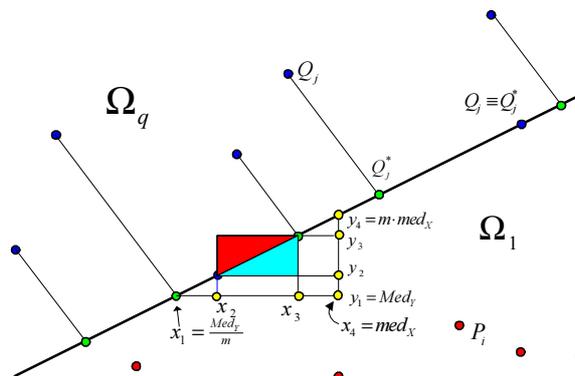


Figura 4.3: Zona roja/azul: solución no válida/válida.

Notas:

1) Respecto del último problema de optimización se tiene que la función objetivo tiene como expresión:

$$D_{r_1}(X) = D_{r_1}(x, mx) = \sum_{i \in S_1} w_i (|x - a_i| + |mx - b_i|) + \sum_{j \in S_q} w_j \sqrt[|q|]{|x - c_j|^q + |mx - d_j|^q}.$$

Dado que es una función convexa, cualquier algoritmo estándar nos proporciona la solución. Por ejemplo, se puede usar la rutina *fminbnd* de MATLAB para valores de  $x$  tales que  $\min\{a_i, c_j\}_{i \in S_1, j \in S_q} \leq x \leq \max\{a_i, c_j\}_{i \in S_1, j \in S_q}$ .

No obstante, hemos realizado una implementación *ad hoc*, adaptando el método ARS a una recta y una función convexa, de manera análoga al método de búsqueda de  $X_r$  con la ventaja de no tener que actualizar los portones en cada iteración.

2) Para el caso  $1 < q < \infty$ , las derivadas laterales se calculan según las fórmulas de la proposición 3.10 mientras que en el caso  $q = \infty$  usaremos la proposición 3.12.

En caso de no utilizar el teorema 3.14, el procedimiento sería el siguiente:

1. A partir del teorema 2.23 obtenemos la pendiente  $m^*$  y obtenemos el punto  $(med_x, Med_y)$ .
2. Si  $(med_x, Med_y) \in \Omega_1$ , aplicamos el algoritmo ARS de forma similar al método 2 del algoritmo anterior, tomando como punto inicial  $X_0 = (x_0, y_0) \in \Omega_1$ , con  $x_0 = \frac{1}{4} \left( 3 \cdot med_x + \frac{Med_y}{m} \right)$  e  $y_0 = \frac{1}{4} (3Med_y + m \cdot med_x)$ , y obtenemos la solución  $X_1$  y el valor de la función objetivo  $D_1(X_1)$ .

Los coeficientes de  $\frac{3}{4}$  y  $\frac{1}{4}$  son irrelevantes, sólo han de cumplir que el punto inicial sea un punto interior al triángulo rectángulo formado por el punto  $(med_x, Med_y)$  y la recta, ya que el movimiento del algoritmo ARS es suficientemente rápido y no influyen significativamente en la convergencia.

3. En caso contrario, hacemos  $(x^*, y^*) = (med_x, Med_y)$  y procedemos como el punto 7 anterior.

## 4.6 Algoritmo de búsqueda en $\Omega_\infty$

Cualquiera que sea  $p \in [1, \infty)$ , procedemos como sigue:

1. Proyectamos verticalmente los puntos de  $\Omega_p$  sobre la recta y realizamos un giro de amplitud  $45^\circ$  respecto del origen (ver Nota al final de este algoritmo).
2. Obtenemos el intervalo mediano  $[med_x, Med_x] \times [med_y, Med_y]$ , de los puntos transformados de  $\Omega_\infty$  y de los puntos transformados de los proyectados anteriores. En virtud de la proposición 3.13, el transformado del punto solución  $X'_s = (x'_s, y'_s)$  verifica que:  $x'_s \geq Med_x$  e  $y'_s \leq med_y$  (ver figura 4.4).
3. Deshacemos el giro y evaluamos la función objetivo en el vértice superior del triángulo, actualizando previamente los portones (ver figura 4.5).

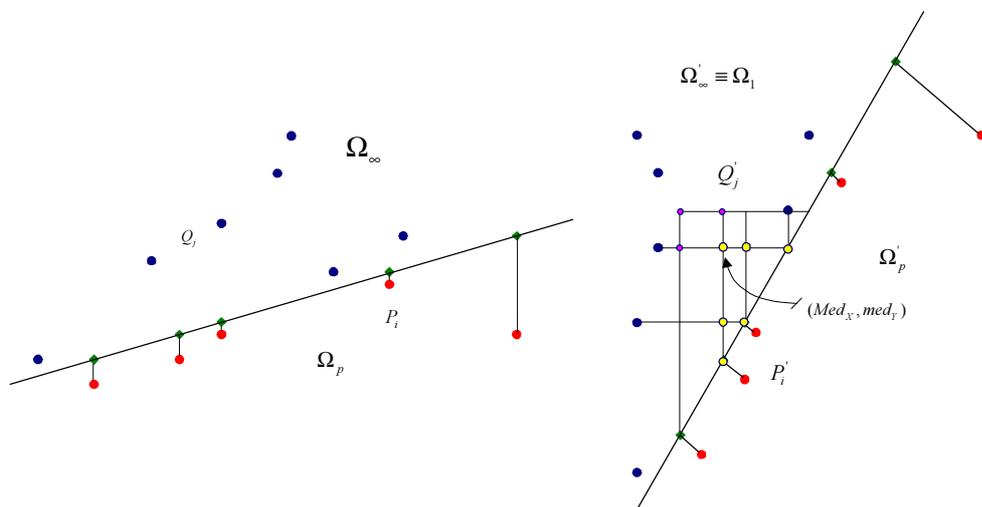


Figura 4.4: Situación inicial y situación tras giro de  $45^\circ$ .

4. La solución se encuentra en un entorno de dicho punto en virtud de la convexidad estricta de la función objetivo. Aplicamos el método ARS

modificado para desplazarse en cuatro direcciones  $u_1 = (1, -1)$ ,  $u_2 = (-1, -1)$ ,  $u_3 = (-1, 1)$  y  $u_4 = (1, 1)$ , y obtenemos la solución en  $\Omega_\infty$  y  $D_\infty$ .

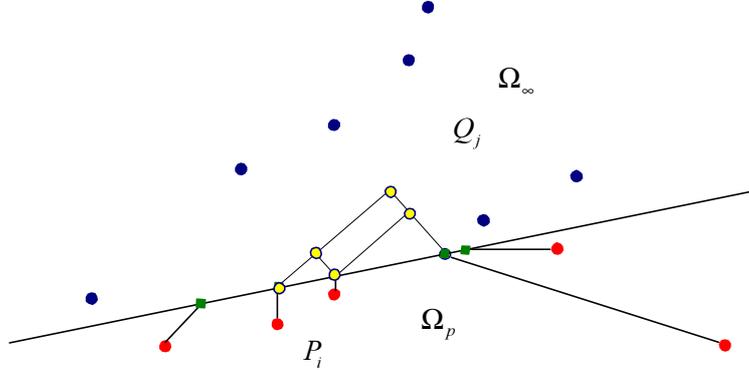


Figura 4.5: Situación final tras deshacer giro.

Nota:

Para obtener el punto  $Y = WP_\infty(X)$  hemos de resolver el siguiente problema de optimización:

$$\arg \min_{Y \in \Omega_\infty} D_\infty(Y) = \arg \min_{Y \in \Omega_\infty} \left\{ \sum_{i \in S_p} w_i d_\infty(G(P_i, X), Y) + \sum_{j \in S_\infty} w_j d_\infty(Y, Q_j) \right\},$$

ó bien,

$$\min_{(x,y) \in \mathbb{R}^2} D_\infty(x,y) = \min_{(x,y) \in \mathbb{R}^2} \left( \sum_{i \in S_p} w_i \max\{|g_i^x - x|, |g_i^y - y|\} + \sum_{j \in S_\infty} w_j \max\{|x - c_j|, |y - d_j|\} \right)$$

donde  $(g_i^x, g_i^y) = G(P_i, X)$ .

Aplicando un giro respecto del origen  $T$ , de amplitud  $\frac{\pi}{4}$ , las ecuaciones de la

$$\text{transformación son: } \begin{cases} x' = \frac{\sqrt{2}}{2}(x - y) \\ y' = \frac{\sqrt{2}}{2}(x + y) \end{cases}. \text{ Dado que } \max\{|a|, |b|\} = \frac{1}{2}(|a+b| + |a-b|),$$

tenemos que:

$$\max\{|x - c_j|, |y - d_j|\} = \frac{1}{2}(|(x+y-(c_j+d_j))| + |(x-y-(c_j-d_j))|) = \frac{1}{\sqrt{2}}(|y' - d'_j| + |x' - c'_j|),$$

$$\max\{|x - g_i^x|, |y - g_i^y|\} = \frac{1}{2}(|(x+y-(g_i^x+g_i^y))| + |(x-y-(g_i^x-g_i^y))|) = \frac{1}{\sqrt{2}}(|y' - g_i^{y'}| + |x' - g_i^{x'}|),$$

4.7 El problema del punto fijo  $X = WP_p(X)$ . El algoritmo de Weiszfeld

---

donde  $(x', y')$ ,  $(g_i^{x'} g_i^{y'})$  y  $(c_j', d_j')$  son los transformados por el giro de los puntos  $Y$ ,  $G(P_i, X)$ , y  $Q_j$ , respectivamente. Ahora, la función objetivo se puede expresar de la forma:

$$D_\infty(Y) = \frac{1}{\sqrt{2}} \min_{Y \in \Omega_1} \left\{ \sum_{i \in S_p} w_i d_1(G_i', Y') + \sum_{j \in S_q} w_j d_1(Y', Q_j') \right\} = \frac{1}{\sqrt{2}} D_1(Y').$$

Por tanto, la solución del problema de optimización en  $\Omega_\infty$  es  $T^{-1}(Y')$ , es decir, la transformación inversa de la solución en  $\Omega_1$  (medianas), para los puntos transformados según el giro  $T$ .

## 4.7 El problema del punto fijo $X = WP_p(X)$ : El algoritmo de Weiszfeld

Algunos algoritmos expuestos en las secciones anteriores construyen sucesiones con objeto de encontrar el punto fijo  $X = WP_p(X)$ , que es, según el teorema 3.5, el punto solución en  $\Omega_p$  del problema (1).

Con  $X_{k+1} = WP_p(X_k)$  denotamos en dicho teorema al punto solución del problema de localización simple de Fermat-Weber:

$$X_{k+1} = \arg \min_{(x, y) \in \Omega_p} \left\{ \sum_{i \in S_p} w_i \sqrt{|x - a_i|^p + |y - b_i|^p} + \sum_{j \in S_q} w_j \sqrt{|x - c_j^*|^p + |y - mc_j^*|^p} \right\},$$

donde  $(c_j^*, mc_j^*) = G(X_k, Q_j)$ .

El método más común y sencillo para resolver el problema anterior es el denominado algoritmo de Weiszfeld, que exponemos a continuación, primero para  $p = 2$  y después generalizamos.

Sea  $\mathcal{P} = \{P_i(a_i, b_i)\}_{i \in S}$  con  $S = \{1, 2, \dots, M\}$ , un conjunto de puntos de demanda (vértices) de  $\mathbb{R}^2$  con la métrica euclídea. Se pretende minimizar la función:

$$D(x, y) = \sum_{i \in S} w_i d_2(X, P_i) = \sum_{i \in S} w_i \sqrt{(x - a_i)^2 + (y - b_i)^2}, \text{ donde } w_i \in \mathbb{R}^+ \quad (13)$$

Dado que esta función es estrictamente convexa (Love, 1988), tiene un mínimo absoluto en el punto que satisfaga la condición necesaria de optimalidad:

$$\frac{\partial}{\partial x} D(x, y) = \sum_{i \in S} w_i \frac{x - a_i}{\sqrt{(x - a_i)^2 + (y - b_i)^2}} = \sum_{i \in S} \frac{w_i (x - a_i)}{d_2(X, P_i)} = 0.$$

$$\frac{\partial}{\partial y} D(x, y) = \sum_{i \in S} w_i \frac{y - b_i}{\sqrt{(x - a_i)^2 + (y - b_i)^2}} = \sum_{i \in S} \frac{w_i (y - b_i)}{d_2(X, P_i)} = 0.$$

Este sistema de ecuaciones no lineales presenta dos importantes problemas: en primer lugar, no puede ser resuelto explícitamente, y de forma general, para más de tres puntos y en segundo lugar, la función objetivo  $D$  no es derivable en los puntos  $P_i$  puesto que el correspondiente denominador sería nulo.

Para resolver el primer problema, Weiszfeld despeja en las ecuaciones anteriores:

$$\sum_{i \in S} w_i \frac{x}{d_2(X, P_i)} = \sum_{i \in S} w_i \frac{a_i}{d_2(X, P_i)} \Rightarrow x = \frac{\sum_{i \in S} \frac{w_i a_i}{\sqrt{(x - a_i)^2 + (y - b_i)^2}}}{\sum_{i \in S} \frac{w_i}{\sqrt{(x - a_i)^2 + (y - b_i)^2}}}$$

$$\sum_{i \in S} w_i \frac{y}{d_2(X, P_i)} = \sum_{i \in S} w_i \frac{b_i}{d_2(X, P_i)} \Rightarrow y = \frac{\sum_{i \in S} \frac{w_i b_i}{\sqrt{(x - a_i)^2 + (y - b_i)^2}}}{\sum_{i \in S} \frac{w_i}{\sqrt{(x - a_i)^2 + (y - b_i)^2}}$$

y le queda un problema de punto fijo i.e. resolver un problema tipo  $X = F(X)$ , para cierta función  $F(X)$ . Para la búsqueda del punto fijo, se toma un punto arbitrario  $X_0(x_0, y_0) \in \mathbb{R}^2$  y se construyen las sucesiones:

$$x_{k+1} = \frac{\sum_{i \in S} g_i(x_k, y_k) a_i}{\sum_{i \in S} g_i(x_k, y_k)}, \quad y_{k+1} = \frac{\sum_{i \in S} g_i(x_k, y_k) b_i}{\sum_{i \in S} g_i(x_k, y_k)}, \quad k = 0, 1, \dots$$

$$\text{con } g_i(x_k, y_k) = \frac{w_i}{\sqrt{(x_k - a_i)^2 + (y_k - b_i)^2}}.$$

4.7 El problema del punto fijo  $X = WP_p(X)$ . El algoritmo de Weiszfeld

El límite es el punto fijo  $y$ , por tanto, el punto que minimiza la función objetivo. Además, se encuentra en la envolvente convexa de los puntos de demanda, ya que los puntos  $X_k = (x_k, y_k)$  son combinaciones lineales convexas de los puntos  $\{P_i\}_{i \in S}$ . Con las limitaciones que expondremos a continuación, el método es convergente para cualquier punto inicial  $X_0$ , si bien la convergencia es más rápida si éste se encuentra cerca del punto fijo, por ello es habitual tomar como punto inicial el centro de gravedad de los puntos de demanda, si éste no coincide con ninguno de ellos, es decir:

$$x_0 = \bar{a} = \frac{\sum_{i \in S} w_i a_i}{\sum_{i \in S} w_i}, \quad y_0 = \bar{b} = \frac{\sum_{i \in S} w_i b_i}{\sum_{i \in S} w_i},$$

que es el punto que minimiza la siguiente función (White,1971):

$$D^*(x, y) = \sum_{i \in S} w_i d_2(X, P_i)^2 = \sum_{i \in S} w_i \left( (x - a_i)^2 + (y - b_i)^2 \right).$$

No obstante lo anterior, incluso eligiendo un punto inicial muy cercano a la solución, el método de Weiszfeld puede tener una convergencia excesivamente lenta (Guerrero 2007). Así, si los vértices fuesen  $P_1 = (1, -1)$ ,  $P_2 = (1, 1)$ ,  $P_3 = (-1, 1)$ ,  $P_4 = (-1, -1)$  y  $P_5 = (100, 0)$ , con pesos respectivos  $w_1 = w_2 = w_3 = w_4 = 1$ , y  $w_5 = 4$ , y elegimos como punto inicial  $X_0 = (99, 0)$ , se necesitan más de 800.000 iteraciones para alcanzar el punto  $(99.9998028, 0)$  cuya distancia al punto solución es superior a  $10^{-4}$ . En este ejemplo, la lentitud se debe al hecho de que el punto solución del problema de Weber coincide con unos de los vértices  $(100, 0)$ , donde la función objetivo no es derivable.

Para evitar la no diferenciabilidad de la función objetivo en los vértices se puede comprobar, como paso previo a la aplicación del algoritmo, si el punto fijo se encuentra en alguno de estos puntos usando la siguiente proposición (Juel y Love, 1981):

*La función  $D$  definida en (13) alcanza el mínimo en el punto de demanda*

$$P_h \text{ si y solo si } CR_h = \left[ \left( \sum_{i \in S, i \neq h} \frac{w_i (a_h - a_i)}{d_2(P_h, P_i)} \right)^2 + \left( \sum_{i \in S, i \neq h} \frac{w_i (b_h - b_i)}{d_2(P_h, P_i)} \right)^2 \right]^{\frac{1}{2}} \leq w_h.$$

Sin embargo, a pesar de superar esta condición inicial, alguna iteración podría coincidir con algún vértice y el método no sería convergente, como se pone de manifiesto en

(Kuhn, 1973) y así, si consideramos los vértices  $P_1 = (-2, 0)$ ,  $P_2 = (-1, 0)$ ,  $P_3 = (0, 1)$ ,  $P_4 = (0, -1)$ ,  $P_5 = (1, 0)$  y  $P_6 = (2, 0)$ , con pesos unitarios, y elegimos un punto inicial con coordenadas aproximadas  $X_0 = (1.62139137, 0)$ , obtenemos  $X_1 = (1, 0)$  y la sucesión permanece en el punto  $X_1$ , siendo la solución el punto  $(0, 0)$ .

Sobre el conjunto de puntos iniciales, que en un número finito de iteraciones del algoritmo de Weiszfeld coincide con algún vértice, se pueden consultar los trabajos de Kuhn (1973), Chandrasekaran y Tamir (1989), Brimberg (1995), Brimberg y Chen, (1998) y Cánovas et al. (2002).

Dado que no tenemos garantías de que en una iteración el punto obtenido no coincida con algún vértice podemos utilizar una aproximación hiperbólica de la distancia:

$$DH(x, y) = \sum_{i \in S} w_i \sqrt{(x - a_i)^2 + (y - b_i)^2 + \varepsilon}, \text{ con } \varepsilon > 0,$$

que es diferenciable en  $\mathbb{R}^2$ , ya que el radicando no se anula, aunque tiene el inconveniente de que es mayor que la distancia original, aproximándose a ella si  $\varepsilon \rightarrow 0$ .

Como indican Love et al. (1988) un valor de  $\varepsilon > 0$  nos evita la engorrosa situación de pasar por uno de los vértices, aunque se pierda en precisión de la solución. Un valor de  $\varepsilon = 0,01$  nos acerca rápidamente a la solución pero la precisión será menor que si usamos el valor  $\varepsilon = 0,0001$ , que es el que hemos usado en todo el trabajo, aunque la convergencia sea más lenta. Desgraciadamente, podemos conseguir que la función objetivo converja rápidamente a su límite, pero el punto puede no converger al punto solución en aquellos casos en los que esta función sea muy aplanada.

En realidad, el método de Weiszfeld es el método del gradiente (Ostresh 1978 y 1978a)

ya que, si  $x_{k+1} = \frac{\sum_{i \in S} g_i(x_k, y_k) a_i}{\sum_{i \in S} g_i(x_k, y_k)}$ , sumando y restando  $x_k$  obtenemos:

$$x_{k+1} = x_k - \lambda_k \frac{\nabla_x D(x_k, y_k)}{\sum_{i \in S} g_i(x_k, y_k)}, \text{ con factor multiplicador } \lambda_k = 1 \text{ (análogamente para } y_k \text{)}.$$

4.7 El problema del punto fijo  $X = WP_p(X)$ . El algoritmo de Weiszfeld

Además, si las sucesivas iteraciones no pasan por ningún vértice, Ostresh demuestra la convergencia de la sucesión  $(x_{k+1}, y_{k+1}) = (x_k, y_k) - \lambda_k \frac{\nabla D(x_k, y_k)}{\sum_{i \in S} g_i(x_k, y_k)}$ , para  $1 \leq \lambda_k \leq 2$ .

Sumando y restando  $\lambda_k(x_k, y_k)$  en el segundo miembro de la fórmula anterior, tenemos:

$$(x_{k+1}, y_{k+1}) = \lambda_k(x_{k+1}^*, y_{k+1}^*) + (1 - \lambda_k)(x_k, y_k), \quad (14)$$

$$\text{donde } (x_{k+1}^*, y_{k+1}^*) = (x_k, y_k) - \frac{\nabla D(x_k, y_k)}{\sum_{i \in S} g_i(x_k, y_k)} \quad (15)$$

Obsérvese que (14) coincide con la fórmula (12) que planteamos en el algoritmo MFP y que (15) es la fórmula del método de Weiszfeld.

Por otra parte, si  $\lambda_k \notin [1, 2]$  algún  $X_k$  puede sobrepasar la envolvente convexa de los vértices y la sucesión sería divergente.

Por todo lo anterior, en cada iteración se puede tomar un factor óptimo (Rodríguez-Chía y Valero-Franco, 2011) o bien considerar un factor constante  $\lambda$  dependiendo del valor de  $p$  en la norma  $\ell_p$ , según veremos a continuación.

Estudiamos ahora el problema de Fermat-Weber para  $p$ -normas, el cual podemos enunciarlo como sigue: dado  $\mathcal{P} = \{P_i(a_i, b_i)\}_{i \in S}$ , conjunto finito de puntos de demanda (vértices) de  $\mathbb{R}^2$  con la norma  $\ell_p$ , se pretende minimizar la función:

$$D(x, y) = \sum_{i \in S} w_i d_p(X, P_i) = \sum_{i \in S} w_i \sqrt[p]{|x - a_i|^p + |y - b_i|^p}, \text{ donde } w_i \in \mathbb{R}^+ \quad (16)$$

Dado que esta función es convexa (Love, 1988), tiene al menos un mínimo absoluto. Imponiendo la condición necesaria de optimalidad:

$$\frac{\partial}{\partial x} D(x, y) = \sum_{i \in S} w_i \frac{\text{signo}(x - a_i) |x - a_i|^{p-1}}{\left(\sqrt[p]{|x - a_i|^p + |y - b_i|^p}\right)^{p-1}} = \sum_{i \in S} \frac{w_i \text{signo}(x - a_i) |x - a_i|^{p-1}}{d_p(X, P_i)^{p-1}} = 0.$$

$$\frac{\partial}{\partial y} D(x, y) = \sum_{i \in S} w_i \frac{\text{signo}(y - b_i) |y - b_i|^{p-1}}{\left( \sqrt[p]{|x - a_i|^p + |y - b_i|^p} \right)^{p-1}} = \sum_{i \in S} \frac{w_i \text{signo}(y - b_i) |y - b_i|^{p-1}}{d_p(X, P_i)^{p-1}} = 0.$$

Reescribimos las ecuaciones anteriores como:

$$\sum_{i \in S} \frac{w_i (x - a_i) |x - a_i|^{p-2}}{d_p(X, P_i)^{p-1}} = 0 \quad \text{y} \quad \sum_{i \in S} \frac{w_i (y - b_i) |y - b_i|^{p-2}}{d_p(X, P_i)^{p-1}} = 0,$$

luego:

$$\begin{aligned} x \sum_{i \in S} \frac{w_i |x - a_i|^{p-2}}{d_p(X, P_i)^{p-1}} &= \sum_{i \in S} \frac{w_i |x - a_i|^{p-2} a_i}{d_p(X, P_i)^{p-1}} \Rightarrow x = \frac{\sum_{i \in S} g_i(x, y) a_i}{\sum_{i \in S} g_i(x, y)} \\ y \sum_{i \in S} \frac{w_i |y - b_i|^{p-2}}{d_p(X, P_i)^{p-1}} &= \sum_{i \in S} \frac{w_i |y - b_i|^{p-2} b_i}{d_p(X, P_i)^{p-1}} \Rightarrow y = \frac{\sum_{i \in S} h_i(x, y) b_i}{\sum_{i \in S} h_i(x, y)}, \end{aligned} \quad (17)$$

con  $g_i(x, y) = \frac{w_i |x - a_i|^{p-2}}{d_p(X, P_i)^{p-1}}$  y  $h_i(x, y) = \frac{w_i |y - b_i|^{p-2}}{d_p(X, P_i)^{p-1}}$ , y tenemos nuevamente un

problema de punto fijo. Dado  $X_0(x_0, y_0) \in \mathbb{R}^2$  se construyen las sucesiones:

$$x_{k+1} = \frac{\sum_{i \in S} g_i(x_k, y_k) a_i}{\sum_{i \in S} g_i(x_k, y_k)}, \quad y_{k+1} = \frac{\sum_{i \in S} h_i(x_k, y_k) b_i}{\sum_{i \in S} h_i(x_k, y_k)}, \quad k = 0, 1, \dots$$

Nótese que si  $p = 2$ , las funciones  $g_i$  y  $h_i$  coinciden, y los sucesivos puntos  $X_k$  son combinaciones lineales convexas de los puntos  $P_i$  y, por tanto, todos ellos y su límite se encuentran en la envolvente convexa de los vértices.

El problema de la no diferenciabilidad de la función objetivo en los vértices puede chequearse usando la condición necesaria y suficiente de Juel y Love (1981):

*La función  $D$  definida en (16) alcanza el mínimo en el punto de demanda*

$$P_h \text{ si y solo si: } \begin{cases} CRP_h = \left( |R_{hx}|^{\frac{p}{p-1}} + |R_{hy}|^{\frac{p}{p-1}} \right)^{\frac{p-1}{p}} \leq w_h & \text{si } p > 1 \\ \max(|R_{hx}|, |R_{hy}|) \leq w_h & \text{si } p = 1, \end{cases}$$

4.7 El problema del punto fijo  $X = WP_p(X)$ . El algoritmo de Weiszfeld

---

$$\text{donde } R_{hx} = \sum_{i \in S, i \neq h} \frac{w_i \text{signo}(a_h - a_i) |a_h - a_i|^{p-1}}{d_p(P_h, P_i)^{p-1}},$$

$$y \quad R_{hy} = \sum_{i \in S, i \neq h} \frac{w_i \text{signo}(b_h - b_i) |b_h - b_i|^{p-1}}{d_p(P_h, P_i)^{p-1}}.$$

No obstante, a veces la convergencia es muy lenta y es conveniente la aplicación de técnicas de aceleración basadas en el proceso  $\Delta^2$  de Aitken, cuya aplicación a un problema de punto fijo es conocida como método de Steffensen (Katz, 1974, Brimberg y Love 1992, Drezner 1996).

Además, puede ocurrir que las funciones  $g_i$  y  $h_i$  no estén bien definidas y, por tanto, la convergencia de la sucesión ha de ser estudiada según los valores de  $p$ :

- Si  $p \in (1, 2)$  las derivadas no están bien definidas si en alguna iteración  $x_k = a_i$  o/y  $y_k = b_i$  para algún  $i$ , ya que  $p - 2 < 0$ , y diremos que el punto  $(x_k, y_k)$  es singular. Si la sucesión no contiene puntos singulares (es regular), entonces, dicha sucesión es localmente y globalmente convergente (Brimberg y Love, 1992 y 1993).

Para los problemas de singularidad, Ortega, Mesa y Sánchez (2000) han propuesto un método para resolver problemas de localización con  $p$ -normas,  $1 < p < 2$ . Su enfoque se basa en una aproximación de las normas  $\ell_p$  por normas blocks, que utilizan para desarrollar un método iterativo de programación lineal.

- Si  $p = 1$ , la función objetivo es separable  $D(x, y) = \sum_{i \in S} w_i |x - a_i| + \sum_{i \in S} w_i |y - b_i|$ , por lo que hemos de estudiar dos funciones de una variable cuyos mínimos se alcanzan en las medianas de sus respectivas variables. Si denotamos con  $med_x$ ,  $Med_x$ ,  $med_y$  y  $Med_y$  a la menor y la mayor de las medianas en  $x$  e  $y$ , respectivamente, cualquier punto del intervalo bidimensional  $[med_x, Med_x] \times [med_y, Med_y]$  minimiza la función objetivo, que es convexa, pero no en sentido estricto.
- Si  $p = \infty$ , la función objetivo es  $D(x, y) = \sum_{i \in S} w_i \cdot \max\{|x - a_i|, |y - b_i|\}$ .

Aplicamos un giro de amplitud  $\frac{\pi}{4}$ :  $(x, y) \rightarrow (x^*, y^*)$ , (ver algoritmo de búsqueda en  $\Omega_\infty$ ), y el problema de minimizar  $D$  es equivalente al problema de minimizar:

$$D^*(x^*, y^*) = \sum_{i \in S} w_i |x^* - a_i| + \sum_{i \in S} w_i |y^* - b_i|,$$

que es el problema de localización simple, con la norma  $\ell_1$ , en el espacio transformado.

- Si  $p > 2$ , las funciones  $g_i$  y  $h_i$  definidas en (17), están bien definidas salvo que, en alguna iteración, el punto  $(x_k, y_k)$  coincida con uno de los vértices. Lamentablemente, Morris (1981), Brimberg y Love (1992) y Brimberg y Love (1993) muestran contraejemplos con ausencia de convergencia local y global para estos valores de  $p$ , aunque afirman que, en la práctica, esta situación no es una limitación para los problemas de localización en el plano, ya que valores de  $p > 2$  rara vez son necesarios si se toman los ejes adecuadamente orientados.

Realizar la comprobación de Juel y Love al principio del algoritmo no solo ralentiza el proceso, sino que, además, no tenemos garantías de que en una determinada iteración no obtengamos un punto singular. Por ello, utilizamos la aproximación hiperbólica de la función objetivo:

$$DH(x, y) = \sum_{i \in S} w_i \sqrt[2]{\left((x - a_i)^2 + \varepsilon\right)^{\frac{p}{2}} + \left((y - b_i)^2 + \varepsilon\right)^{\frac{p}{2}}}, \text{ con } w_i, \varepsilon \in \mathbb{R}^+.$$

Las derivadas parciales existen ahora en todo el plano e igualando a cero, obtenemos las ecuaciones de recurrencia:

$$x_{k+1} = \frac{\sum_{i \in S} w_i \frac{\left((x_k - a_i)^2 + \varepsilon\right)^{\frac{p-2}{2}} a_i}{DH(X_k)^{p-1}}}{\sum_{i \in S} w_i \frac{\left((x_k - a_i)^2 + \varepsilon\right)^{\frac{p-2}{2}}}{DH(X_k)^{p-1}}}, \quad y_{k+1} = \frac{\sum_{i \in S} w_i \frac{\left((y_k - b_i)^2 + \varepsilon\right)^{\frac{p-2}{2}} b_i}{DH(X_k)^{p-1}}}{\sum_{i \in S} w_i \frac{\left((y_k - b_i)^2 + \varepsilon\right)^{\frac{p-2}{2}}}{DH(X_k)^{p-1}}},$$

$$\text{similares a las ecuaciones (16): } x_{k+1} = \frac{\sum_{i \in S} g_i(x_k, y_k) a_i}{\sum_{i \in S} g_i(x_k, y_k)} \text{ e } y_{k+1} = \frac{\sum_{i \in S} h_i(x_k, y_k) b_i}{\sum_{i \in S} h_i(x_k, y_k)},$$

pero ahora, las funciones

4.7 El problema del punto fijo  $X = WP_p(X)$ . El algoritmo de Weiszfeld

---

$$g_i(x_k, y_k) = \frac{w_i \left( (x_k - a_i)^2 + \varepsilon \right)^{\frac{p-2}{2}}}{DH(X_k)^{p-1}}, \quad h_i(x_k, y_k) = \frac{w_i \left( (y_k - b_i)^2 + \varepsilon \right)^{\frac{p-2}{2}}}{DH(X_k)^{p-1}},$$

están bien definidas y son indefinidamente diferenciables en todo el plano.

Si incluimos un factor  $\lambda$ , obtenemos:

$$x_{k+1} = x_k - \frac{\lambda}{\sum_{i \in S} g_i(x_k, y_k)} \nabla_x DH(x_k, y_k), \quad y_{k+1} = y_k - \frac{\lambda}{\sum_{i \in S} h_i(x_k, y_k)} \nabla_y DH(x_k, y_k).$$

Brimberg et al. (1998a), usando la matriz Hessiana de la función objetivo, obtienen que

si  $\lambda \leq \frac{2}{(p-1)(2-Q^*)}$ , la convergencia local está garantizada,

$$\text{donde } Q^* = \frac{1}{\sum_{i \in S} g_i(X^*)} \sum_{i \in S} \frac{g_i(X^*) |x^* - a_i|^p}{d_p(X^*, P_i)^p} + \frac{1}{\sum_{i \in S} h_i(X^*)} \sum_{i \in S} \frac{h_i(X^*) |y^* - b_i|^p}{d_p(X^*, P_i)^p},$$

siendo  $X^*$  el punto solución y se cumple que  $0 < Q^* < 2$ .

Si  $p = 2$ , con un simple cálculo obtenemos  $Q^* = 1$  con lo que  $\lambda \leq 2$  garantiza la convergencia, como ya demostró Ostresh (1978).

Üster y Love (2000) comprueban que la solución de compromiso  $Q^* \approx 1$  es adecuada cuando los valores se distribuyen de manera uniforme en un rectángulo del plano, concluyendo que el factor  $\lambda = 2/(p-1)$  garantiza y acelera la convergencia. Dado que  $Q^*$  es desconocido, también podemos tomar un valor inferior  $\lambda \leq 1/(p-1)$ , que garantiza la convergencia local si  $p > 2$ .

Basados en su experiencia computacional, Morris y Verdini (1979) concluyen que la convergencia puede ser esperada para  $p > 2$ . En los trabajos de campo que hemos realizado en esta tesis también se ha producido la convergencia. No obstante, en este trabajo hemos acelerado la convergencia para cualquier valor de  $p$  usando un factor multiplicador  $\lambda = \min\{1, 2/(p-1)\}$ .



---

### RESULTADOS COMPUTACIONALES

---

#### 5.1 Consideraciones generales

En este capítulo vamos a analizar el rendimiento de los diferentes algoritmos presentados en el capítulo anterior. Nos centraremos básicamente en contrastar la velocidad de ejecución así como la precisión de la solución obtenida.

Comenzamos señalando que todos los algoritmos expuestos anteriormente, así como el algoritmo MWP (Parlar, 1994), han sido implementados en MATLAB y los programas resultantes se han ejecutado en un ordenador Intel Core2 Duo CPU P8400 @2.26 GHz 1.58 GHz, 293 GB de RAM.

En todos los casos, los programas finalizan al alcanzar un nivel de tolerancia de 0,000001 o bien al superar un prefijado número de iteraciones. En los procedimientos basados en el algoritmo de Weiszfeld el número de iteraciones prefijado ha sido de 10.000 iteraciones y en el suavizado hiperbólico se ha tomado el valor  $\varepsilon = 0,0001$ , mientras que en los procedimientos basados en el algoritmo ARS se han dado 5 vueltas, con 200 iteraciones en cada vuelta, escogiendo la solución óptima entre las 5 soluciones obtenidas.

Es muy importante señalar que en nuestro trabajo siempre hemos considerado que si  $q > p$ , entonces  $\Omega_q$  se encuentra en el semiplano superior definido por la recta. En todo el trabajo y en todas las tablas del presente capítulo se considera esta posición relativa, a excepción de las tablas I y II, en las que intervienen los citados conjuntos P-18, Z-30 y Z-50, ya que los autores (Parlar, 1994 y Zaferanieh et al., 2008) consideran la posición relativa inversa y la hemos respetado al objeto de poder comparar los resultados.

## 5.2 Programas y resultados

Los programas resultantes de los algoritmos expuestos en el capítulo IV pueden resumirse en el siguiente cuadro:

		$q = \infty$	$1 < q < \infty$					
		<b>Gate(1,∞)</b>		<b>GMFP</b>	<b>Gate(1,q)</b>	<b>Gdaq</b>	<b>Garsq</b>	
$p = 1$	$\Omega_1$	Derivadas	$\Omega_1$	Weiszfeld	Derivadas	Derivadas	ARS	Usa portones
	$\Omega_\infty$	Mediana ARS	$\Omega_q$	Weiszfeld	Weiszfeld	Weber-ARS	Mediana ARS	
		<b>Gate(p,∞)</b>		<b>MFP</b>	<b>Gate(p,q)</b>			
$1 < p < \infty$	$\Omega_p$	Weiszfeld	$\Omega_p$	Weiszfeld	ARS			Busca y usa portones
	$\Omega_\infty$	Mediana ARS	$\Omega_q$	Weiszfeld	ARS			
		Usa portones		Obtiene portones finalistas				

Los programas situados en la primera fila o/y columna usan el conocimiento exacto de los portones, según el estudio realizado en el capítulo 2. Sin embargo, el programa MFP no usa portones pero obtiene puntos cercanos a ellos al finalizar el programa, mientras que el programa  $\text{Gate}(p, q)$  los busca y usa en cada iteración.

Los programas MFP y GMFP usan en ambas regiones el método de Weiszfeld para obtener el punto solución.

En la primera columna figuran los programas que se basan en el algoritmo de búsqueda en  $\Omega_\infty$  donde, según vimos en el capítulo anterior, se usa el método ARS partiendo de

### 5.2.1 Regiones con normas $\ell_1$ y $\ell_2$

---

un punto cercano a la mediana. En  $\Omega_1(\Omega_p)$  se usa el método de las derivadas (Weiszfeld).

Los programas Gate(1,q) y Gdaq usan las derivadas en  $\Omega_1$ , pero en  $\Omega_q$  se diferencian en el método de resolución del problema de Weber, mientras que el primero usa el algoritmo de Weiszfeld, el segundo usa el método ARS.

El programa Garsq usa en ambas regiones el método ARS pero en  $\Omega_1$ , utiliza el hecho de que la posible solución tiene que estar en el triángulo formado por  $(med_x, Med_y)$  y la recta.

Por último, el programa Gate(p,q) usa en ambos casos el método ARS, calculando en cada iteración los portones correspondientes.

### 5.2.1 Regiones con normas $\ell_1$ y $\ell_2$

Los resultados del primer test que hemos realizado se encuentran en la Tabla I. Se han considerado un conjunto de 18 puntos (P-18), otro de 30 puntos (Z-30) y un tercero de 50 puntos (Z-50) propuestos por Parlar (1994) y Zaferanieh et al. (2008), con objeto de contrastar los resultados obtenidos en nuestros algoritmos con los obtenidos por estos autores. Además, se han considerado tres valores de la pendiente  $m$  y los casos  $p=1$  y  $q=2$ , ya que el algoritmo MWP sólo está diseñado para estos valores. Nuestros resultados coinciden con los expuestos en los artículos anteriores e incluso los mejoran en tiempo y precisión.

En la Tabla I se muestra el punto solución  $(x_s, y_s)$ , el valor de la función objetivo  $D$ , el tiempo  $T$  de CPU, promedio temporal de 10 vueltas del programa y la región  $\Omega$  en la que se encuentra la solución. Podemos ver como el menor tiempo de CPU se alcanza por el algoritmo MWP, sin embargo, la solución proporcionada es la peor de todas ya que el valor de  $D$  proporcionado es mayor que los obtenidos por el resto de algoritmos, especialmente en el caso  $m=0,5$ , donde el punto solución proporcionado llega a estar en una región diferente, y por ello sombreada, de los puntos hallados por los otros algoritmos.

También se observa como el algoritmo GMFP es más rápido y, prácticamente, más preciso que el algoritmo MFP. Además, sobresale por su precisión el algoritmo Gate(1,2) que, salvo en  $m=0,5$ , es también el algoritmo más rápido de todos los propuestos en este trabajo doctoral.

Para confirmar estas observaciones iniciales vamos a considerar cuatro conjuntos de 100, 200, 500 y 1000 puntos, generados aleatoriamente y distribuidos uniformemente sobre el cuadrado  $[0,100] \times [0,100]$  con pesos unitarios (ver anexo I).

Tabla I

Resultados para los puntos de Parlar y Zaferanieh, con normas  $\ell_1$  y  $\ell_2$ , variando la pendiente  $m$ .

m	Conjuntos	MWP				$\Omega$	MFP				$\Omega$	Gate(1,2)			
		$x_s$	$y_s$	D	T		$x_s$	$y_s$	D	T		$x_s$	$y_s$	D	T
0,5	P-18	8,98	6,10	138,70	0,000	$\Omega_1$	9,03	4,51	127,06	0,339	$\Omega_2$	9,00	4,50	127,02	0,205
	Z-30	5,47	4,41	316,50	0,002	$\Omega_1$	6,00	4,01	299,94	0,058	$\Omega_1$	6,00	4,00	299,85	0,323
	Z-50	10,36	8,44	1.179,02	0,003	$\Omega_1$	11,01	8,02	1125,36	0,259	$\Omega_1$	11,00	8,00	1.125,25	0,119
1	P-18	9,07	6,12	114,96	0,000	$\Omega_2$	9,02	6,58	114,82	0,269	$\Omega_2$	9,02	6,58	114,82	0,009
	Z-30	5,70	4,34	266,43	0,000	$\Omega_2$	5,65	4,58	266,02	0,214	$\Omega_2$	5,66	4,59	265,97	0,009
	Z-50	10,70	8,39	966,89	0,003	$\Omega_2$	10,73	8,66	966,46	1,478	$\Omega_2$	10,73	8,66	966,38	0,014
1,5	P-18	8,99	6,23	112,40	0,000	$\Omega_2$	8,93	6,47	112,36	1,091	$\Omega_2$	8,93	6,47	112,35	0,005
	Z-30	5,60	4,42	257,96	0,000	$\Omega_2$	5,51	4,56	257,83	0,088	$\Omega_2$	5,51	4,56	257,81	0,006
	Z-50	10,58	8,46	939,70	0,000	$\Omega_2$	10,53	8,60	939,53	1,461	$\Omega_2$	10,53	8,60	939,49	0,006
m	Conjuntos	Garsq				$\Omega$	GMFP				$\Omega$	Gdaq			
		$x_s$	$y_s$	D	T		$x_s$	$y_s$	D	T		$x_s$	$y_s$	D	T
0,5	P-18	9,00	4,50	127,02	0,031	$\Omega_2$	10,00	4,99	127,54	0,064	$\Omega_2$	9,00	4,50	127,02	0,009
	Z-30	6,00	4,00	299,85	0,116	$\Omega_1$	6,00	4,01	299,89	0,102	$\Omega_1$	6,00	4,00	299,85	0,013
	Z-50	11,00	8,00	1.125,25	0,081	$\Omega_1$	11,00	8,11	1125,39	0,075	$\Omega_1$	11,00	8,00	1.125,25	0,019
1	P-18	9,02	6,58	114,82	0,019	$\Omega_2$	9,02	6,58	114,82	0,384	$\Omega_2$	9,02	6,59	114,82	0,028
	Z-30	5,66	4,59	265,97	0,028	$\Omega_2$	5,66	4,59	265,97	0,114	$\Omega_2$	5,66	4,59	265,97	0,019
	Z-50	10,73	8,66	966,38	0,034	$\Omega_2$	10,73	8,66	966,38	0,181	$\Omega_2$	10,73	8,66	966,38	0,069
1,5	P-18	8,93	6,47	112,35	0,022	$\Omega_2$	8,93	6,47	112,35	0,295	$\Omega_2$	8,93	6,47	112,35	0,016
	Z-30	5,51	4,56	257,81	0,025	$\Omega_2$	5,51	4,56	257,81	0,033	$\Omega_2$	5,51	4,56	257,81	0,025
	Z-50	10,53	8,60	939,49	0,028	$\Omega_2$	10,53	8,60	939,49	0,966	$\Omega_2$	10,53	8,60	939,49	0,028

En la Tabla II se confirman nuestras observaciones anteriores, pero hemos de tener en cuenta que ahora la posición relativa de las normas es la inversa que en la Tabla I, y por ello, MWP es impreciso si la pendiente es alta. Asimismo, la rapidez del método Gate(1,2) es la inversa, es decir, es alta para pendientes bajas y decrece a medida que la

### 5.2.2 Regiones con normas $\ell_1$ y $\ell_q$

pendiente crece. Ello se debe al aumento del número de puntos en  $\Omega_1$  a medida que la pendiente crece con lo que el número de iteraciones en el procedimiento de Weiszfeld crece en mayor medida que en los procedimientos Gars y Gda.

Tabla II

Resultados para conjuntos de puntos aleatorios (pesos unitarios), con normas  $\ell_1$  y  $\ell_2$ , variando  $m$ .

m	Ptos.	MWP				$\Omega$	MFP				$\Omega$	Gate(1,2)			
		$x_s$	$y_s$	D	T		$x_s$	$y_s$	D	T		$x_s$	$y_s$	D	T
0,5	100	44,52	52,03	3.639,38	0,009	$\Omega_2$	45,38	51,81	3.638,64	4,366	$\Omega_2$	45,38	51,81	3.638,62	0,011
	200	47,59	49,89	7.639,14	0,002	$\Omega_2$	48,84	50,04	7.636,43	7,313	$\Omega_2$	48,84	50,04	7.636,36	0,016
	500	53,68	49,48	19.658,83	0,005	$\Omega_2$	55,55	49,48	19.645,49	16,572	$\Omega_2$	55,55	49,48	19.645,32	0,048
	1000	49,82	51,24	39.714,96	0,005	$\Omega_2$	51,22	51,24	39.698,61	33,225	$\Omega_2$	51,22	51,24	39.698,29	0,091
1	100	43,88	52,96	3.821,06	0,002	$\Omega_2$	43,90	50,86	3.816,56	3,981	$\Omega_2$	43,90	50,86	3.816,50	0,036
	200	46,49	51,11	8.045,83	0,002	$\Omega_2$	47,32	51,22	8.044,79	6,588	$\Omega_2$	47,32	51,22	8.044,66	0,178
	500	53,58	49,85	23.260,46	0,003	$\Omega_1$	51,96	52,01	21.103,14	16,319	$\Omega_2$	52,00	52,00	21.102,46	2,509
	1000	48,34	52,65	42.001,40	0,005	$\Omega_2$	49,04	51,46	41.977,83	28,047	$\Omega_2$	49,04	51,46	41.977,25	1,658
1,5	100	44,14	51,53	4.463,80	0,002	$\Omega_1$	35,80	53,73	4.132,06	5,756	$\Omega_2$	35,87	53,80	4.131,85	0,713
	200	47,34	49,24	9.580,53	0,000	$\Omega_1$	37,33	56,01	8.809,39	8,819	$\Omega_2$	37,33	56,00	8.808,92	1,167
	500	53,51	48,67	24.195,20	0,003	$\Omega_1$	55,01	49,28	23.373,89	25,256	$\Omega_1$	55,00	49,32	23.373,70	2,597
	1000	49,70	50,34	49.551,25	0,006	$\Omega_1$	37,91	56,88	46.268,83	38,797	$\Omega_2$	38,00	57,00	46.265,69	5,003
0,5	100	45,38	51,81	3.638,62	0,031	$\Omega_2$	45,38	51,81	3.638,62	2,241	$\Omega_2$	45,38	51,81	3.638,62	0,033
	200	48,84	50,04	7.636,36	0,050	$\Omega_2$	48,84	50,04	7.636,36	3,719	$\Omega_2$	48,84	50,04	7.636,36	0,066
	500	55,55	49,48	19.645,32	0,114	$\Omega_2$	55,55	49,48	19.645,32	8,313	$\Omega_2$	55,55	49,48	19.645,32	0,219
	1000	51,22	51,24	39.698,29	0,194	$\Omega_2$	51,22	51,24	39.698,29	15,931	$\Omega_2$	51,22	51,24	39.698,29	0,375
1	100	43,90	50,85	3.816,50	0,053	$\Omega_2$	43,90	50,86	3.816,50	1,484	$\Omega_2$	43,90	50,85	3.816,50	0,109
	200	47,32	51,22	8.044,66	0,078	$\Omega_2$	47,32	51,22	8.044,66	2,275	$\Omega_2$	47,32	51,21	8.044,66	0,500
	500	52,00	52,00	21.102,46	0,252	$\Omega_2$	51,96	52,01	21.102,83	8,294	$\Omega_2$	52,00	52,00	21.102,46	0,163
	1000	49,04	51,46	41.977,25	0,272	$\Omega_2$	49,04	51,46	41.977,25	9,466	$\Omega_2$	49,04	51,46	41.977,25	3,289
1,5	100	35,87	53,80	4.131,85	0,083	$\Omega_2$	35,86	53,81	4.131,97	2,394	$\Omega_2$	35,87	53,80	4.131,85	0,028
	200	37,33	56,00	8.808,92	0,102	$\Omega_2$	38,65	58,00	8.817,12	5,716	$\Omega_2$	37,33	56,00	8.808,92	0,055
	500	55,00	49,40	23.373,71	0,323	$\Omega_1$	55,94	49,44	23.376,34	11,353	$\Omega_1$	55,00	49,32	23.373,70	0,147
	1000	38,00	57,00	46.265,69	0,470	$\Omega_2$	39,27	58,92	46.326,74	23,241	$\Omega_2$	38,00	57,00	46.265,69	0,341

### 5.2.2 Regiones con normas $\ell_1$ y $\ell_q$

En la Tabla III que mostramos a continuación, vamos a considerar las normas  $\ell_1$  y  $\ell_q$ , con  $q$  variable, para los conjuntos P-18, Z-30 y Z-50. En este caso, como se comentó anteriormente, no figura el algoritmo MWP pues sólo es válido para  $p = 1$  y  $q = 2$ .

Tabla III: Resultados para los puntos de Parlar y Zaferanieh, con normas  $\ell_1$  y  $\ell_q$ , variando  $q$  y la pendiente  $m$ .

q	m	Ptos.	MFP				GMFP				Gates(1,q)				Gdaq				Garsq			
			$x_s$	$y_s$	D	T	$x_s$	$y_s$	D	T	$x_s$	$y_s$	D	T	$x_s$	$y_s$	D	T	$x_s$	$y_s$	D	T
0,5		P-18	9,00	5,48	140,20	0,658	9,01	4,50	140,25	0,019	9,00	5,40	140,19	0,625	9,00	5,40	140,19	0,041	9,00	5,40	140,19	0,053
		Z-30	6,00	4,01	311,82	0,219	6,00	4,01	311,77	0,227	6,00	4,00	311,73	0,859	6,00	4,00	311,73	0,038	6,00	4,00	311,73	0,097
		Z-50	11,01	8,01	1.161,28	0,339	11,00	8,02	1.161,22	0,117	11,00	8,00	1.161,18	0,313	11,00	8,00	1.161,18	0,100	11,00	8,00	1.161,18	0,094
1,1	1	P-18	8,99	6,45	136,06	0,261	8,99	6,45	136,05	0,427	8,99	6,45	136,05	0,031	9,00	5,83	136,05	0,081	9,00	6,40	136,05	0,038
		Z-30	5,99	4,14	304,80	0,250	5,99	4,14	304,76	0,131	5,99	4,15	304,76	0,031	6,00	4,15	304,75	0,091	6,00	4,14	304,75	0,059
		Z-50	11,01	8,26	1.127,09	0,361	11,00	8,26	1.127,00	0,220	11,00	8,26	1.127,00	0,078	11,00	8,34	1.126,99	0,241	11,00	8,26	1.126,99	0,063
1,5		P-18	8,99	6,42	135,49	1,463	8,99	6,42	135,48	0,305	8,99	6,42	135,48	0,031	9,00	6,86	135,48	0,047	9,00	6,19	135,48	0,041
		Z-30	5,99	4,14	302,86	0,155	5,99	4,14	302,83	0,048	5,99	4,14	302,83	0,031	6,00	4,14	302,82	0,084	6,00	4,14	302,82	0,063
		Z-50	11,00	8,31	1.120,80	0,586	11,00	8,31	1.120,75	1,359	11,00	8,31	1.120,75	0,031	11,00	8,31	1.120,74	0,131	11,00	8,32	1.120,74	0,047
0,5		P-18	9,01	4,50	137,35	0,528	9,25	4,62	137,63	0,977	9,00	4,50	137,30	0,641	9,53	4,77	137,30	0,019	9,00	4,50	137,30	0,041
		Z-30	6,00	4,01	309,28	0,061	6,00	4,01	309,22	0,200	6,00	4,00	309,19	0,859	6,00	4,00	309,19	0,031	6,00	4,00	309,19	0,084
		Z-50	11,01	8,02	1.153,63	0,311	11,00	8,02	1.153,57	0,120	11,00	8,00	1.153,52	0,313	11,00	8,00	1.153,52	0,084	11,00	8,00	1.153,52	0,109
1,2	1	P-18	8,99	6,47	131,44	0,253	8,99	6,47	131,43	0,419	8,99	6,47	131,43	0,016	9,00	6,50	131,43	0,069	9,00	6,48	131,43	0,056
		Z-30	5,98	4,32	296,55	0,245	5,98	4,33	296,50	0,131	5,98	4,32	296,50	0,016	5,99	4,33	296,50	0,044	5,99	4,33	296,50	0,041
		Z-50	11,00	8,50	1.092,42	0,352	11,00	8,50	1.092,32	0,216	11,00	8,50	1.092,32	0,078	11,00	8,50	1.092,31	0,200	11,00	8,48	1.092,31	0,075
1,5		P-18	8,99	6,42	130,43	1,684	8,99	6,43	130,43	0,305	8,99	6,43	130,43	0,016	9,00	6,43	130,43	0,044	9,00	6,44	130,43	0,072
		Z-30	5,98	4,31	293,16	0,134	5,98	4,31	293,13	0,039	5,98	4,31	293,13	0,016	5,98	4,31	293,13	0,031	5,98	4,31	293,13	0,034
		Z-50	11,00	8,52	1.081,37	0,652	11,00	8,52	1.081,32	1,314	11,00	8,52	1.081,32	0,016	11,00	8,47	1.081,32	0,084	11,00	8,50	1.081,32	0,059
0,5		P-18	9,02	4,50	131,68	0,400	9,99	4,99	132,56	0,181	9,00	4,50	131,63	0,672	9,58	4,79	131,63	0,019	9,00	4,50	131,63	0,038
		Z-30	6,00	4,01	304,23	0,231	6,00	4,01	304,17	0,173	6,00	4,00	304,14	0,813	6,00	4,00	304,14	0,028	6,00	4,00	304,14	0,075
		Z-50	11,01	8,02	1.138,42	0,328	11,00	8,05	1.138,38	0,119	11,00	8,00	1.138,30	0,313	11,00	8,00	1.138,30	0,053	11,00	8,04	1.138,30	0,103
1,5	1	P-18	8,97	6,51	122,35	0,269	8,97	6,51	122,35	0,413	8,97	6,51	122,35	0,016	8,97	6,52	122,35	0,050	8,97	6,52	122,35	0,031
		Z-30	5,84	4,49	279,99	0,250	5,84	4,50	279,93	0,127	5,84	4,50	279,93	0,016	5,84	4,50	279,93	0,059	5,84	4,50	279,93	0,034
		Z-50	10,98	8,67	1.024,00	1,127	10,98	8,67	1.023,89	0,208	10,98	8,67	1.023,89	0,047	10,98	8,67	1.023,89	0,122	10,98	8,67	1.023,89	0,047
1,5		P-18	8,93	6,44	120,52	1,283	8,93	6,44	120,52	0,309	8,93	6,44	120,52	0,000	8,93	6,44	120,52	0,031	8,93	6,44	120,52	0,031
		Z-30	5,76	4,47	273,85	0,116	5,76	4,47	273,82	0,038	5,76	4,47	273,82	0,016	5,76	4,47	273,82	0,047	5,76	4,47	273,82	0,031
		Z-50	10,89	8,65	1.004,17	1,275	10,89	8,65	1.004,12	1,233	10,89	8,64	1.004,12	0,016	10,89	8,65	1.004,12	0,063	10,89	8,65	1.004,12	0,044

Tabla III: Resultados para los puntos de Parlar y Zaferanieh, con normas  $\ell_1$  y  $\ell_q$ , variando  $q$  y la pendiente  $m$  (Continuación)

<b>q</b>	<b>m</b>	<b>Ptos.</b>	<b>MFP</b>				<b>GMFP</b>				<b>Gates(1,q)</b>				<b>Gdaq</b>				<b>Garsq</b>			
			<b>x<sub>s</sub></b>	<b>y<sub>s</sub></b>	<b>D</b>	<b>T</b>																
<b>1,8</b>	<b>0,5</b>	P-18	9,02	4,50	128,48	0,359	10,00	4,99	129,08	0,098	9,00	4,50	128,43	0,406	9,65	4,83	128,43	0,019	9,00	4,50	128,43	0,034
		Z-30	6,00	4,01	301,28	0,219	6,00	4,01	301,23	0,156	6,00	4,00	301,19	0,688	6,00	4,00	301,19	0,025	6,00	4,00	301,19	0,100
		Z-50	11,01	8,02	1.129,48	0,366	11,00	8,08	1.129,49	0,114	11,00	8,00	1.129,36	0,281	11,00	8,00	1.129,36	0,047	11,00	8,08	1.129,36	0,100
	<b>1</b>	P-18	8,97	6,56	117,15	0,297	8,97	6,56	117,15	0,408	8,98	6,56	117,15	0,016	8,98	6,56	117,15	0,059	8,98	6,56	117,15	0,025
		Z-30	5,72	4,55	270,36	0,253	5,72	4,56	270,31	0,127	5,72	4,56	270,31	0,016	5,72	4,56	270,31	0,056	5,72	4,56	270,31	0,038
		Z-50	10,83	8,68	984,46	1,641	10,83	8,68	984,36	0,205	10,83	8,68	984,36	0,031	10,83	8,68	984,36	0,184	10,83	8,68	984,36	0,044
	<b>1,5</b>	P-18	8,91	6,46	114,87	1,228	8,91	6,46	114,87	0,313	8,91	6,46	114,87	0,000	8,91	6,46	114,87	0,031	8,91	6,46	114,87	0,025
		Z-30	5,59	4,53	262,77	0,108	5,59	4,53	262,75	0,039	5,59	4,53	262,75	0,016	5,59	4,53	262,75	0,056	5,59	4,53	262,75	0,031
		Z-50	10,65	8,63	959,69	1,645	10,65	8,63	959,64	1,148	10,65	8,63	959,64	0,016	10,65	8,63	959,64	0,081	10,65	8,63	959,64	0,041
<b>0,5</b>	P-18	9,04	4,51	124,85	0,397	10,01	5,00	125,14	0,088	9,00	4,50	124,81	0,484	10,06	5,03	124,81	0,019	9,00	4,50	124,81	0,038	
	Z-30	6,00	4,01	297,75	0,083	6,00	4,01	297,71	0,184	6,00	4,00	297,67	0,844	6,00	4,00	297,67	0,025	6,00	4,00	297,67	0,066	
	Z-50	11,01	8,02	1.118,54	0,572	11,00	8,15	1.118,60	0,145	11,00	8,00	1.118,43	0,500	11,00	8,00	1.118,43	0,047	11,00	8,01	1.118,59	0,103	
<b>2,5</b>	<b>1</b>	P-18	9,21	6,63	111,07	0,319	9,21	6,63	111,07	0,408	9,21	6,64	111,07	0,031	9,21	6,64	111,07	0,069	9,21	6,64	111,07	0,025
		Z-30	5,54	4,64	259,04	0,256	5,54	4,64	258,99	0,128	5,54	4,64	258,99	0,031	5,54	4,64	258,99	0,097	5,54	4,64	258,99	0,034
		Z-50	10,52	8,61	937,09	1,636	10,52	8,61	937,03	0,205	10,52	8,61	937,03	0,063	10,52	8,61	937,03	0,309	10,52	8,61	937,03	0,044
<b>1,5</b>	P-18	9,08	6,48	108,36	1,272	9,08	6,48	108,36	0,320	9,08	6,48	108,36	0,016	9,08	6,48	108,36	0,041	9,08	6,48	108,36	0,025	
	Z-30	5,35	4,64	249,97	0,088	5,35	4,64	249,96	0,042	5,35	4,64	249,96	0,016	5,35	4,64	249,96	0,088	5,35	4,64	249,96	0,031	
	Z-50	10,32	8,55	906,90	1,858	10,32	8,55	906,87	1,052	10,32	8,55	906,87	0,016	10,32	8,55	906,87	0,103	10,32	8,55	906,87	0,044	
<b>0,5</b>	P-18	9,24	4,61	122,48	1,516	9,41	4,69	122,46	0,030	9,40	4,70	122,44	0,438	10,23	5,11	122,44	0,019	9,39	4,70	122,44	0,028	
	Z-30	6,00	4,01	295,10	0,088	6,00	4,01	295,07	0,244	6,00	4,00	295,03	0,563	6,00	4,00	295,03	0,028	6,00	4,00	295,03	0,050	
	Z-50	11,98	5,98	1.105,48	0,764	12,01	5,99	1.105,45	0,173	12,00	6,00	1.105,18	0,719	11,84	5,92	1.105,18	0,069	12,00	6,00	1.105,18	0,088	
<b>4</b>	<b>1</b>	P-18	9,55	6,58	106,61	0,359	9,56	6,59	106,61	0,416	9,56	6,59	106,61	0,016	9,56	6,59	106,61	0,053	9,56	6,59	106,61	0,025
		Z-30	5,34	4,75	250,58	0,267	5,34	4,75	250,54	0,131	5,34	4,75	250,54	0,063	5,34	4,75	250,54	0,156	5,34	4,75	250,54	0,038
		Z-50	10,21	8,50	900,74	1,258	10,21	8,50	900,71	0,211	10,21	8,50	900,71	0,109	10,21	8,50	900,71	0,506	10,21	8,50	900,71	0,044
<b>1,5</b>	P-18	9,48	6,46	103,69	1,244	9,48	6,46	103,69	0,330	9,48	6,46	103,69	0,016	9,48	6,46	103,69	0,044	9,48	6,46	103,69	0,022	
	Z-30	5,08	4,83	240,50	0,198	5,08	4,83	240,50	0,052	5,08	4,83	240,50	0,047	5,08	4,83	240,50	0,156	5,08	4,83	240,50	0,031	
	Z-50	9,96	8,52	867,42	1,581	9,96	8,52	867,41	0,931	9,96	8,52	867,41	0,047	9,96	8,52	867,41	0,125	9,96	8,52	867,41	0,044	

En la tabla III se aprecia la mayor velocidad del algoritmo GMFP respecto del algoritmo MFP, si bien no siempre tiene mayor precisión. Sin embargo, el algoritmo Gate(1,q) supera en precisión a ambos pero, en pendientes bajas, el algoritmo MFP consigue mayor velocidad, ello puede deberse a la distribución geométrica de los puntos y es estudiado en las tablas IV y V. Por otra parte, se repite el hecho de que el algoritmo Gate(1,q), con igual precisión que los algoritmos Gars y Gda, aumenta su velocidad cuando aumenta la pendiente, mientras que los otros dos disminuyen la suya.

Consideremos ahora 1000 puntos generados aleatoriamente y uniformemente distribuidos en el cuadrado  $[0,100] \times [0,100]$  con pesos unitarios. A continuación hemos tomado 10, 10, 10, 10, 10, 5, 2 y 1 subconjuntos de 10, 20, 40, 50, 100, 200, 500 y 1000 puntos, respectivamente, y mostramos en las Tablas IV y V los promedios de los resultados obtenidos en cuanto a distancias y tiempos.

Tabla IV: Promedio de distancias en conjuntos de puntos aleatorios (pesos unitarios).

q=2						
m	Puntos	MFP	GMFP	Gate(1,q)	Gdaq	Garsq
0,5	10	342,71	342,71	342,71	342,71	342,71
	20	726,13	726,13	726,13	726,13	726,13
	40	1.522,89	1.522,88	1.522,88	1.522,88	1.522,88
	50	1.948,15	1.948,14	1.948,14	1.948,14	1.948,14
	100	3.947,23	3.947,20	3.947,20	3.947,20	3.947,20
	200	7.915,60	7.915,53	7.915,53	7.915,53	7.915,53
	500	19.826,92	19.826,76	19.826,76	19.826,76	19.826,76
1000	39.698,61	39.698,29	39.698,29	39.698,29	39.698,29	
1	10	370,88	370,88	370,87	370,87	370,87
	20	773,69	773,68	773,67	773,67	773,67
	40	1.612,25	1.612,23	1.612,26	1.612,26	1.612,22
	50	2.062,76	2.062,73	2.062,72	2.062,72	2.062,72
	100	4.176,93	4.176,87	4.176,85	4.176,85	4.176,85
	200	8.372,50	8.372,39	8.372,42	8.372,42	8.372,39
	500	20.969,91	20.969,62	20.969,62	20.969,62	20.969,62
1000	41.977,83	41.977,25	41.977,25	41.977,25	41.977,25	
1,5	10	395,62	395,65	395,62	395,62	395,62
	20	833,70	833,97	833,68	833,68	833,68
	40	1.753,95	1.754,35	1.753,87	1.753,87	1.753,87
	50	2.253,83	2.254,04	2.253,47	2.253,47	2.253,47
	100	4.606,79	4.611,88	4.606,48	4.606,48	4.606,48
	200	9.233,45	9.243,09	9.232,90	9.232,90	9.232,90
	500	23.111,95	23.131,03	23.110,33	23.110,33	23.110,33
1000	46.268,83	46.326,74	46.265,69	46.265,69	46.265,69	

5.2.4 Regiones con normas  $\ell_1$  y  $\ell_q$

Tabla IV: (Continuación)

<b>q=1,2</b>		<b>MFP</b>	<b>GMFP</b>	<b>Gate(1,q)</b>	<b>Gdaq</b>	<b>Garsq</b>
<b>m</b>	<b>Puntos</b>					
	10	396,55	396,54	396,54	396,54	396,54
	20	843,17	843,16	843,16	843,16	843,16
	40	1.775,65	1.775,63	1.775,63	1.775,63	1.775,63
<b>0,5</b>	50	2.264,97	2.264,96	2.264,96	2.264,96	2.264,96
	100	4.602,95	4.602,91	4.602,91	4.602,91	4.602,91
	200	9.233,50	9.233,43	9.233,43	9.233,43	9.233,43
	500	23.130,80	23.130,62	23.130,62	23.130,62	23.130,62
	1000	46.310,78	46.310,44	46.310,44	46.310,44	46.310,44
	10	408,10	408,09	408,09	408,09	408,09
	20	864,59	864,80	864,52	864,52	864,53
	40	1.815,14	1.815,50	1.815,12	1.815,12	1.815,12
<b>1</b>	50	2.311,87	2.313,47	2.311,83	2.311,83	2.311,83
	100	4.704,18	4.704,09	4.704,03	4.704,03	4.704,03
	200	9.428,01	9.427,89	9.427,76	9.427,76	9.427,76
	500	23.610,36	23.610,04	23.609,83	23.609,83	23.609,83
	1000	47.249,84	47.249,19	47.249,19	47.249,19	47.249,19
	10	417,48	417,50	417,48	417,48	418,16
	20	887,26	887,52	887,25	887,25	887,27
	40	1.875,52	1.876,36	1.875,47	1.875,47	1.875,53
<b>1,5</b>	50	2.395,54	2.396,96	2.395,48	2.395,48	2.395,48
	100	4.900,20	4.900,97	4.900,05	4.900,05	4.900,05
	200	9.841,23	9.842,34	9.841,04	9.841,04	9.841,05
	500	24.665,08	24.664,89	24.664,82	24.664,82	24.664,82
	1000	49.378,14	49.378,88	49.377,60	49.377,60	49.377,60
<b>q=1,4</b>		<b>MFP</b>	<b>GMFP</b>	<b>Gate(1,q)</b>	<b>Gdaq</b>	<b>Garsq</b>
<b>m</b>	<b>Puntos</b>					
	10	374,98	374,98	374,98	374,97	374,97
	20	796,64	796,63	796,63	796,63	796,63
	40	1.674,95	1.674,93	1.674,93	1.674,93	1.674,93
<b>0,5</b>	50	2.139,04	2.139,02	2.139,02	2.139,02	2.139,02
	100	4.342,23	4.342,19	4.342,19	4.342,19	4.342,19
	200	8.708,87	8.708,80	8.708,80	8.708,80	8.708,80
	500	21.814,23	21.814,04	21.814,04	21.814,04	21.814,04
	1000	43.676,09	43.675,71	43.675,71	43.675,71	43.675,71
	10	393,39	393,38	393,38	393,38	393,38
	20	829,51	829,50	829,49	829,49	829,49
	40	1.735,75	1.735,72	1.735,73	1.735,73	1.735,71
<b>1</b>	50	2.214,32	2.215,17	2.214,28	2.214,28	2.214,28
	100	4.496,74	4.496,67	4.496,64	4.496,63	4.496,63
	200	9.012,11	9.011,97	9.011,89	9.011,89	9.011,89
	500	22.571,08	22.570,74	22.570,64	22.570,64	22.570,64
	1000	45.175,44	45.174,76	45.174,76	45.174,76	45.174,76
	10	408,84	408,86	408,84	408,84	408,83
	20	866,93	867,17	866,88	866,90	866,88
	40	1.831,18	1.831,80	1.831,13	1.831,13	1.831,17
<b>1,5</b>	50	2.346,49	2.347,09	2.346,39	2.346,39	2.346,39
	100	4.796,48	4.797,17	4.796,34	4.796,34	4.796,36
	200	9.658,15	9.658,91	9.657,87	9.657,87	9.657,87
	500	24.294,12	24.299,04	24.293,28	24.293,28	24.293,28
	1000	48.651,29	48.653,04	48.650,69	48.650,69	48.650,70

Tabla IV: (Continuación)

<b>q=1,6</b>		<b>MFP</b>	<b>GMFP</b>	<b>Gate(1,q)</b>	<b>Gdaq</b>	<b>Garsq</b>
<b>m</b>	<b>Puntos</b>					
	10	360,52	360,52	360,52	360,52	360,52
	20	765,17	765,16	765,16	765,16	765,16
	40	1.606,99	1.606,97	1.606,97	1.606,97	1.606,97
<b>0,5</b>	50	2.053,81	2.053,80	2.053,80	2.053,80	2.053,80
	100	4.165,97	4.165,93	4.165,93	4.165,93	4.165,93
	200	8.354,63	8.354,55	8.354,55	8.354,55	8.354,55
	500	20.926,19	20.926,01	20.926,01	20.926,01	20.926,01
	1000	41.898,85	41.898,49	41.898,49	41.898,49	41.898,49
	10	383,47	383,46	383,46	383,46	383,46
	20	804,88	804,87	804,86	804,86	804,86
	40	1.680,96	1.680,93	1.680,96	1.680,96	1.680,92
<b>1</b>	50	2.147,61	2.147,90	2.147,54	2.147,54	2.147,56
	100	4.355,14	4.355,07	4.355,04	4.355,04	4.355,04
	200	8.728,54	8.728,41	8.728,34	8.728,34	8.728,34
	500	21.862,31	21.861,97	21.862,21	21.862,04	21.861,97
	1000	43.760,25	43.759,56	43.759,56	43.759,56	43.759,56
	10	402,97	403,00	402,97	402,97	402,97
	20	852,19	852,45	852,15	852,15	852,15
	40	1.799,38	1.799,85	1.799,32	1.799,32	1.799,32
<b>1,5</b>	50	2.308,49	2.308,42	2.308,09	2.308,09	2.308,09
	100	4.719,86	4.720,63	4.719,70	4.719,70	4.719,71
	200	9.487,88	9.493,41	9.487,40	9.487,40	9.487,40
	500	23.779,94	23.796,62	23.778,24	23.778,24	23.778,24
	1000	47.601,14	47.654,21	47.598,57	47.598,57	47.598,57
<b>q=1,8</b>		<b>MFP</b>	<b>GMFP</b>	<b>Gate(1,q)</b>	<b>Gdaq</b>	<b>Garsq</b>
<b>m</b>	<b>Puntos</b>					
	10	350,28	350,27	350,27	350,27	350,27
	20	742,74	742,73	742,73	742,73	742,73
	40	1.558,65	1.558,64	1.558,64	1.558,64	1.558,64
<b>0,5</b>	50	1.993,12	1.993,10	1.993,10	1.993,10	1.993,10
	100	4.040,36	4.040,32	4.040,32	4.040,32	4.040,32
	200	8.102,42	8.102,35	8.102,35	8.102,35	8.102,35
	500	20.294,52	20.294,35	20.294,35	20.294,35	20.294,35
	1000	40.634,63	40.634,29	40.634,29	40.634,29	40.634,29
	10	376,24	376,24	376,23	376,23	376,23
	20	787,06	787,04	787,06	787,03	787,03
	40	1.641,62	1.641,60	1.641,62	1.641,62	1.641,59
<b>1</b>	50	2.099,48	2.099,50	2.099,45	2.099,45	2.099,45
	100	4.253,22	4.253,16	4.253,13	4.253,13	4.253,13
	200	8.524,83	8.524,71	8.524,68	8.524,68	8.524,68
	500	21.352,10	21.351,79	21.351,79	21.351,79	21.351,79
	1000	42.741,03	42.740,40	42.740,40	42.740,40	42.740,40
	10	398,77	398,80	398,77	398,77	399,15
	20	841,61	841,86	841,58	841,58	841,58
	40	1.773,61	1.774,59	1.773,54	1.773,54	1.773,54
<b>1,5</b>	50	2.277,65	2.277,90	2.277,24	2.277,24	2.277,24
	100	4.658,92	4.659,61	4.658,70	4.661,85	4.658,70
	200	9.347,23	9.353,05	9.346,67	9.346,67	9.346,67
	500	23.397,49	23.415,37	23.395,71	23.395,71	23.395,71
	1000	46.838,16	46.894,76	46.835,14	46.835,14	46.835,14

5.2.4 Regiones con normas  $\ell_1$  y  $\ell_q$

Tabla IV: (Continuación)

<b>q=3</b>		<b>MFP</b>	<b>GMFP</b>	<b>Gate(1,q)</b>	<b>Gdaq</b>	<b>Garsq</b>
<b>m</b>	<b>Puntos</b>					
	10	323,47	323,47	323,47	323,47	323,47
	20	684,12	684,12	684,12	684,12	684,12
	40	1.431,90	1.431,89	1.431,89	1.431,89	1.431,89
<b>0,5</b>	50	1.833,49	1.833,48	1.833,48	1.833,48	1.833,48
	100	3.709,07	3.709,05	3.709,05	3.709,05	3.709,05
	200	7.438,62	7.438,58	7.438,58	7.438,58	7.438,58
	500	18.634,09	18.633,98	18.633,98	18.633,98	18.633,98
	1000	37.309,87	37.309,65	37.309,65	37.309,65	37.309,65
	10	357,65	357,14	357,14	357,14	357,14
	20	739,12	739,07	739,06	739,06	739,06
	40	1.535,86	1.535,84	1.535,84	1.535,84	1.535,84
<b>1</b>	50	1.966,82	1.966,72	1.966,71	1.966,71	1.966,71
	100	3.977,39	3.977,35	3.977,37	3.977,37	3.977,35
	200	7.973,47	7.973,40	7.973,40	7.973,41	7.973,40
	500	19.968,99	19.968,79	19.968,80	19.968,80	19.968,79
	1000	39.978,52	39.978,13	39.978,13	39.978,13	39.978,13
	10	387,39	387,41	387,39	387,39	387,58
	20	813,21	813,55	813,16	813,16	813,16
	40	1.703,24	1.703,20	1.702,69	1.702,69	1.702,69
<b>1,5</b>	50	2.192,02	2.191,30	2.190,83	2.190,83	2.190,83
	100	4.462,03	4.463,83	4.460,78	4.460,78	4.460,78
	200	8.940,82	8.944,11	8.939,38	8.939,38	8.939,38
	500	22.377,51	22.380,29	22.374,31	22.374,31	22.374,31
	1000	44.808,02	44.844,63	44.798,98	44.798,98	44.798,98
<b>q=5</b>		<b>MFP</b>	<b>GMFP</b>	<b>Gate(1,q)</b>	<b>Gdaq</b>	<b>Garsq</b>
<b>m</b>	<b>Puntos</b>					
	10	311,88	311,88	311,88	311,88	311,88
	20	659,86	659,86	659,86	659,86	659,86
	40	1.378,60	1.378,60	1.378,60	1.378,60	1.378,60
<b>0,5</b>	50	1.766,03	1.766,03	1.766,03	1.766,03	1.766,03
	100	3.567,67	3.567,66	3.567,66	3.567,66	3.567,66
	200	7.156,14	7.156,11	7.156,11	7.156,11	7.156,11
	500	17.928,59	17.928,53	17.928,53	17.928,53	17.928,53
	1000	35.894,77	35.894,65	35.894,65	35.894,65	35.894,65
	10	349,60	348,69	348,69	348,71	348,70
	20	719,20	718,19	718,19	718,19	718,19
	40	1.488,73	1.488,72	1.488,73	1.488,73	1.488,72
<b>1</b>	50	1.907,93	1.907,85	1.907,85	1.907,85	1.907,85
	100	3.853,00	3.852,97	3.853,13	3.853,14	3.852,97
	200	7.724,40	7.724,36	7.724,36	7.724,41	7.724,36
	500	19.344,96	19.344,86	19.344,86	19.344,92	19.344,86
	1000	38.729,90	38.729,68	38.729,68	38.729,68	38.729,68
	10	382,25	382,20	382,19	382,21	382,19
	20	800,80	800,76	800,54	800,54	800,52
	40	1.674,22	1.672,34	1.671,93	1.671,94	1.671,92
<b>1,5</b>	50	2.157,07	2.153,92	2.153,74	2.153,74	2.153,74
	100	4.377,74	4.376,40	4.375,96	4.375,96	4.375,95
	200	8.773,32	8.772,27	8.769,87	8.769,87	8.769,87
	500	21.954,33	21.949,92	21.949,50	21.949,50	21.949,50
	1000	43.972,67	43.974,54	43.955,63	43.955,63	43.955,63

En las tablas IV y V podemos observar como el algoritmo GMFP es más rápido y preciso que el algoritmo MFP para pendientes no superiores a la unidad. A medida que aumenta la pendiente se mantiene la ganancia en tiempo de GMFP respecto de MFP pero pierde en precisión. Esto se debe, fundamentalmente, a que los programas finalizan al alcanzar las 10000 iteraciones, cuando todavía están relativamente lejos de la solución óptima.

Sin embargo, los otros tres programas ganan siempre en precisión y velocidad, destacando la velocidad del algoritmo Gate(1,q) en pendientes bajas, del algoritmo Garsq en pendientes cercanas a 1 y del algoritmo Gdaq en pendientes altas.

Se podría pensar que el algoritmo MFP no es preciso a causa del número limitado de iteraciones pero, en la tabla VI, se muestra que apenas hay ganancia de precisión si aumentamos el número de iteraciones hasta 30.000 (valores en negrita) y sin embargo los tiempos aumentan considerablemente.

Tabla V: Promedio de tiempos de CPU en conjuntos de puntos aleatorios (pesos unitarios).

<b>q=2</b>						
<b>m</b>	<b>Puntos</b>	<b>MFP</b>	<b>GMFP</b>	<b>Gate(1,q)</b>	<b>Gdaq</b>	<b>Garsq</b>
	10	1,088	0,639	0,011	0,012	0,034
	20	1,550	0,941	0,009	0,013	0,030
	40	1,936	1,261	0,011	0,021	0,033
<b>0,5</b>	50	2,259	1,388	0,009	0,028	0,036
	100	3,375	2,213	0,014	0,044	0,042
	200	5,569	3,713	0,016	0,077	0,075
	500	12,164	8,289	0,047	0,198	0,102
	1000	23,406	16,016	0,078	0,391	0,203
	10	1,277	0,641	0,131	0,021	0,055
	20	1,456	0,647	0,125	0,055	0,080
	40	1,650	0,847	0,197	0,095	0,058
<b>1</b>	50	1,975	0,917	0,100	0,107	0,070
	100	3,008	1,634	0,300	0,188	0,086
	200	3,747	3,181	0,622	0,851	0,128
	500	8,992	4,359	1,531	2,134	0,195
	1000	19,953	9,438	1,828	3,447	0,313
	10	1,041	0,670	0,191	0,050	0,089
	20	1,331	0,852	0,314	0,040	0,083
	40	2,011	1,141	0,383	0,052	0,084
<b>1,5</b>	50	2,248	1,181	0,494	0,072	0,105
	100	3,609	2,972	0,733	0,031	0,116
	200	4,894	3,881	1,244	0,056	0,153
	500	13,523	11,789	2,625	0,148	0,320
	1000	24,797	22,781	5,219	0,372	0,500

5.2.4 Regiones con normas  $\ell_1$  y  $\ell_q$

Tabla V: (Continuación)

<b>q=1,2</b>		<b>MFP</b>	<b>GMFP</b>	<b>Gate(1,q)</b>	<b>Gdaq</b>	<b>Garsq</b>
<b>m</b>	<b>Puntos</b>					
<b>0,5</b>	10	1,366	0,625	0,017	0,019	0,041
	20	1,920	0,891	0,019	0,032	0,039
	40	2,778	1,327	0,030	0,059	0,056
	50	3,189	1,477	0,031	0,065	0,063
	100	5,191	2,555	0,053	0,127	0,097
	200	9,269	4,453	0,100	0,320	0,181
	500	21,414	10,141	0,313	0,644	0,453
	1000	41,422	19,688	0,672	1,434	0,656
<b>1</b>	10	1,672	0,681	0,194	0,033	0,066
	20	1,678	0,730	0,211	0,060	0,097
	40	2,189	0,892	0,466	0,118	0,102
	50	2,445	0,983	0,334	0,139	0,116
	100	4,827	2,369	1,225	0,473	0,192
	200	8,009	4,384	2,694	0,403	0,313
	500	17,289	6,828	5,789	1,377	0,609
	1000	34,469	11,813	16,484	24,053	1,063
<b>1,5</b>	10	1,452	0,495	0,255	0,041	0,088
	20	2,061	0,838	0,514	0,130	0,098
	40	3,370	1,633	0,941	0,123	0,120
	50	3,441	1,841	1,128	0,227	0,155
	100	6,550	4,170	2,233	0,181	0,208
	200	11,800	7,703	4,213	0,448	0,375
	500	31,211	12,867	10,203	0,980	0,836
	1000	63,484	56,250	19,953	1,913	1,000
<b>q=1,4</b>		<b>MFP</b>	<b>GMFP</b>	<b>Gate(1,q)</b>	<b>Gdaq</b>	<b>Garsq</b>
<b>m</b>	<b>Puntos</b>					
<b>0,5</b>	10	1,294	0,719	0,013	0,020	0,039
	20	1,884	0,945	0,013	0,028	0,031
	40	2,741	1,338	0,019	0,055	0,045
	50	3,158	1,527	0,023	0,067	0,059
	100	5,153	2,558	0,042	0,124	0,075
	200	9,091	4,447	0,091	0,261	0,150
	500	21,156	10,133	0,219	0,789	0,359
	1000	41,625	19,625	0,391	1,528	0,578
<b>1</b>	10	1,575	0,694	0,188	0,030	0,059
	20	1,708	0,686	0,198	0,060	0,077
	40	2,202	0,878	0,452	0,111	0,088
	50	2,448	1,038	0,334	0,354	0,105
	100	4,898	2,133	1,209	0,831	0,178
	200	7,972	3,731	2,628	1,251	0,288
	500	21,430	8,406	5,688	8,788	0,563
	1000	26,359	11,109	7,016	8,288	1,344
<b>1,5</b>	10	1,227	0,698	0,259	0,066	0,098
	20	1,894	1,109	0,544	0,081	0,111
	40	3,280	1,622	0,872	0,098	0,105
	50	3,373	1,478	1,152	0,238	0,139
	100	7,006	3,914	2,233	0,100	0,197
	200	10,363	6,741	4,272	0,203	0,338
	500	29,969	15,016	10,250	0,550	0,695
	1000	51,406	41,953	20,594	1,231	1,516

Tabla V: (Continuación)

<b>q=1,6</b>		<b>MFP</b>	<b>GMFP</b>	<b>Gate(1,q)</b>	<b>Gdaq</b>	<b>Garsq</b>
<b>m</b>	<b>Puntos</b>					
	10	1,263	0,719	0,009	0,021	0,036
	20	1,880	0,959	0,019	0,028	0,034
	40	2,742	1,395	0,020	0,058	0,052
<b>0,5</b>	50	3,152	1,539	0,022	0,068	0,055
	100	5,119	2,577	0,039	0,146	0,083
	200	9,034	4,466	0,075	0,298	0,141
	500	21,023	10,242	0,180	0,806	0,352
	1000	41,141	19,766	0,344	1,591	0,547
	10	1,556	0,755	0,186	0,031	0,070
	20	1,747	0,727	0,197	0,127	0,073
	40	2,370	0,967	0,448	0,247	0,088
<b>1</b>	50	2,842	0,908	0,334	0,403	0,097
	100	4,659	2,183	0,850	0,630	0,145
	200	7,363	3,766	2,275	1,226	0,284
	500	16,102	5,539	5,805	9,286	0,523
	1000	28,219	11,250	9,984	14,828	0,922
	10	1,253	0,620	0,264	0,074	0,103
	20	1,814	1,078	0,548	0,080	0,100
	40	3,200	1,359	0,852	0,120	0,095
<b>1,5</b>	50	3,775	1,609	1,167	0,219	0,131
	100	6,295	3,609	2,194	0,097	0,184
	200	11,588	7,800	4,259	0,193	0,300
	500	30,320	12,984	10,242	0,538	0,719
	1000	52,891	50,906	20,516	1,113	1,531
<b>q=1,8</b>		<b>MFP</b>	<b>GMFP</b>	<b>Gate(1,q)</b>	<b>Gdaq</b>	<b>Garsq</b>
<b>m</b>	<b>Puntos</b>					
	10	1,236	0,728	0,009	0,020	0,039
	20	1,875	0,992	0,019	0,030	0,036
	40	2,716	1,406	0,017	0,059	0,044
<b>0,5</b>	50	3,138	1,539	0,020	0,073	0,055
	100	5,102	2,573	0,034	0,148	0,084
	200	9,000	4,478	0,069	0,294	0,134
	500	20,922	10,148	0,164	0,789	0,313
	1000	40,797	19,781	0,313	1,738	0,734
	10	1,523	0,669	0,186	0,032	0,067
	20	1,861	0,725	0,213	0,134	0,073
	40	2,252	0,936	0,448	0,251	0,083
<b>1</b>	50	2,478	1,189	0,281	0,350	0,095
	100	4,964	2,155	0,914	0,617	0,152
	200	6,950	3,716	2,188	2,310	0,259
	500	15,766	5,219	5,797	9,283	0,492
	1000	29,719	11,141	7,672	14,766	0,953
	10	1,213	0,713	0,270	0,079	0,091
	20	1,795	1,044	0,539	0,082	0,106
	40	3,161	1,495	0,852	0,139	0,105
<b>1,5</b>	50	3,817	1,798	1,169	0,212	0,148
	100	6,475	3,145	1,994	0,099	0,186
	200	11,300	6,797	4,250	0,191	0,288
	500	29,188	16,656	10,500	0,513	0,727
	1000	59,391	45,875	22,016	1,144	1,219

5.2.4 Regiones con normas  $\ell_1$  y  $\ell_q$

Tabla V: (Continuación)

<b>q=3</b>		<b>MFP</b>	<b>GMFP</b>	<b>Gate(1,q)</b>	<b>Gdaq</b>	<b>Garsq</b>
<b>m</b>	<b>Puntos</b>					
<b>0,5</b>	10	1,475	0,663	0,011	0,026	0,038
	20	1,845	0,980	0,006	0,036	0,044
	40	2,313	1,459	0,013	0,071	0,044
	50	2,850	1,608	0,017	0,088	0,050
	100	4,520	2,503	0,028	0,194	0,072
	200	7,897	4,506	0,066	0,379	0,128
	500	18,078	10,320	0,148	0,906	0,297
1000	35,031	20,047	0,328	2,063	0,531	
<b>1</b>	10	1,467	0,745	0,175	0,050	0,061
	20	1,750	0,820	0,216	0,150	0,072
	40	2,244	0,995	0,409	0,665	0,091
	50	2,433	1,075	0,247	0,478	0,092
	100	3,469	1,464	0,903	1,519	0,153
	200	4,697	2,859	1,794	3,633	0,247
	500	9,453	4,617	3,938	9,858	0,477
1000	28,250	11,188	6,172	20,759	1,078	
<b>1,5</b>	10	1,302	0,798	0,245	0,088	0,098
	20	2,027	1,363	0,542	0,098	0,091
	40	2,831	1,514	0,863	0,447	0,103
	50	3,219	2,106	1,183	0,512	0,130
	100	5,856	3,617	2,234	0,384	0,183
	200	9,269	7,034	4,209	0,203	0,263
	500	25,430	19,875	10,250	0,502	0,469
1000	49,531	39,469	20,359	1,134	1,391	
<b>q=5</b>		<b>MFP</b>	<b>GMFP</b>	<b>Gate(1,q)</b>	<b>Gdaq</b>	<b>Garsq</b>
<b>m</b>	<b>Puntos</b>					
<b>0,5</b>	10	1,539	0,644	0,017	0,042	0,039
	20	1,983	1,016	0,019	0,050	0,034
	40	2,513	1,456	0,033	0,094	0,045
	50	3,145	1,542	0,034	0,116	0,045
	100	5,111	2,500	0,052	0,255	0,078
	200	9,028	4,725	0,103	0,489	0,122
	500	20,984	10,633	0,273	1,273	0,297
1000	41,281	20,531	0,500	2,369	0,547	
<b>1</b>	10	1,581	0,761	0,214	0,167	0,070
	20	1,728	0,925	0,231	0,186	0,078
	40	2,214	0,967	0,497	0,774	0,086
	50	2,633	1,134	0,473	0,696	0,100
	100	3,688	1,486	1,231	2,168	0,148
	200	7,072	3,781	2,828	4,209	0,213
	500	16,086	4,594	7,141	11,445	0,414
1000	33,125	11,656	15,469	28,169	1,156	
<b>1,5</b>	10	1,786	0,817	0,291	0,110	0,083
	20	2,311	1,169	0,530	0,300	0,078
	40	3,375	1,906	0,831	0,559	0,109
	50	3,995	2,656	1,256	0,834	0,138
	100	8,006	3,342	2,320	0,722	0,177
	200	16,422	7,913	4,334	1,418	0,256
	500	37,406	14,391	10,094	1,059	0,445
1000	72,781	43,938	20,469	1,772	1,016	

Tabla VI: Comparativa de distancias al aumentar el número máximo de iteraciones.

q= 1.2		10.000 iter.		30.000 iter.		q=1,4		10.000 iter.		30.000 iter.	
m	Ptos.	MFP	Gate(1,q)	MFP	Gate(1,q)	MFP	Gate(1,q)	MFP	Gate(1,q)	MFP	Gate(1,q)
0,5	10	396,55	396,54	396,55	396,54	374,98	374,98	374,98	374,98	374,98	374,98
	20	843,17	843,16	843,17	843,16	796,64	796,63	796,64	796,63	796,64	796,63
	40	1.775,65	1.775,63	1.775,65	1.775,63	1.674,95	1.674,93	1.674,95	1.674,93	1.674,95	1.674,93
	50	2.264,97	2.264,96	2.264,97	2.264,96	2.139,04	2.139,02	2.139,04	2.139,02	2.139,04	2.139,02
	100	4.602,95	4.602,91	4.602,95	4.602,91	4.342,23	4.342,19	4.342,23	4.342,19	4.342,23	4.342,19
	200	9.233,50	9.233,43	9.233,50	9.233,43	8.708,87	8.708,80	8.708,87	8.708,80	8.708,87	8.708,80
	500	23.130,80	23.130,62	23.130,80	23.130,62	21.814,23	21.814,04	21.814,23	21.814,04	21.814,23	21.814,04
	1000	46.310,78	46.310,44	46.310,78	46.310,44	43.676,09	43.675,71	43.676,09	43.675,71	43.676,09	43.675,71
1	10	408,10	408,09	408,10	408,09	393,39	393,38	393,39	393,38	393,39	393,38
	20	864,59	864,52	864,59	864,52	829,51	829,49	829,51	829,49	829,51	829,49
	40	1.815,14	1.815,12	1.815,14	1.815,11	1.735,75	1.735,73	1.735,75	1.735,73	1.735,75	1.735,71
	50	2.311,87	2.311,83	2.311,87	2.311,83	2.214,32	2.214,28	2.214,32	2.214,28	2.214,32	2.214,28
	100	4.704,18	4.704,03	4.704,16	4.704,03	4.496,74	4.496,64	4.496,74	4.496,64	4.496,74	<b>4.496,63</b>
	200	9.428,01	9.427,76	9.428,01	9.427,76	9.012,11	9.011,89	9.012,11	9.011,89	9.012,11	9.011,89
	500	23.610,36	23.609,83	23.610,36	23.609,83	22.571,08	22.570,64	22.571,08	22.570,64	22.571,08	22.570,64
	1000	47.249,84	47.249,19	47.249,84	47.249,19	45.175,44	45.174,76	45.175,44	45.174,76	45.175,44	45.174,76
1,5	10	417,48	417,48	417,48	417,48	408,84	408,84	408,84	408,84	408,84	408,84
	20	887,26	887,25	887,26	887,25	866,93	866,88	866,91	866,88	866,91	866,88
	40	1.875,52	1.875,47	1.875,52	1.875,47	1.831,18	1.831,13	1.831,18	1.831,13	1.831,18	1.831,13
	50	2.395,54	2.395,48	<b>2.395,53</b>	2.395,48	2.346,49	2.346,39	2.346,46	2.346,39	2.346,49	2.346,39
	100	4.900,20	4.900,05	<b>4.900,16</b>	4.900,05	4.796,48	4.796,34	4.796,47	4.796,34	4.796,48	4.796,34
	200	9.841,23	9.841,04	9.841,23	9.841,04	9.658,15	9.657,87	9.658,14	9.657,87	9.658,15	9.657,87
	500	24.665,08	24.664,82	24.665,08	24.664,82	24.294,12	24.293,28	24.294,06	24.293,28	24.294,12	24.293,28
	1000	49.378,14	49.377,60	49.378,14	49.377,60	48.651,29	48.650,69	48.651,29	48.650,69	48.651,29	48.650,69
q= 1.6		10.000 iter.		30.000 iter.		q=1,8		10.000 iter.		30.000 iter.	
m	Ptos.	MFP	Gate(1,q)	MFP	Gate(1,q)	MFP	Gate(1,q)	MFP	Gate(1,q)	MFP	Gate(1,q)
0,5	10	360,52	360,52	360,52	360,52	350,28	350,27	350,28	350,27	350,28	350,27
	20	765,17	765,16	765,17	765,16	742,74	742,73	742,74	742,73	742,74	742,73
	40	1.606,99	1.606,97	1.606,99	1.606,97	1.558,65	1.558,64	1.558,65	1.558,64	1.558,65	1.558,64
	50	2.053,81	2.053,80	2.053,81	2.053,80	1.993,12	1.993,10	1.993,12	1.993,10	1.993,12	1.993,10
	100	4.165,97	4.165,93	4.165,97	4.165,93	4.040,36	4.040,32	4.040,36	4.040,32	4.040,36	4.040,32
	200	8.354,63	8.354,55	8.354,63	8.354,55	8.102,42	8.102,35	8.102,42	8.102,35	8.102,42	8.102,35
	500	20.926,19	20.926,01	20.926,19	20.926,01	20.294,52	20.294,35	20.294,52	20.294,35	20.294,52	20.294,35
	1000	41.898,85	41.898,49	41.898,85	41.898,49	40.634,63	40.634,29	40.634,63	40.634,29	40.634,63	40.634,29
1	10	383,47	383,46	383,47	383,46	376,24	376,23	376,24	376,23	376,24	376,23
	20	804,88	804,86	804,88	804,86	787,06	787,06	<b>787,05</b>	<b>787,04</b>	787,06	<b>787,04</b>
	40	1.680,96	1.680,96	1.680,96	<b>1.680,92</b>	1.641,62	1.641,62	1.641,62	<b>1.641,59</b>	1.641,62	<b>1.641,59</b>
	50	2.147,61	2.147,54	2.147,61	2.147,54	2.099,48	2.099,45	2.099,48	2.099,45	2.099,48	2.099,45
	100	4.355,14	4.355,04	4.355,14	4.355,04	4.253,22	4.253,13	4.253,22	4.253,13	4.253,22	4.253,13
	200	8.728,54	8.728,34	8.728,54	8.728,34	8.524,83	8.524,68	8.524,83	8.524,68	8.524,83	8.524,68
	500	21.862,31	21.862,21	21.862,31	<b>21.861,97</b>	21.352,10	21.351,79	21.352,10	21.351,79	21.352,10	21.351,79
	1000	43.760,25	43.759,56	43.760,25	43.759,56	42.741,03	42.740,40	42.741,03	42.740,40	42.741,03	42.740,40
1,5	10	402,97	402,97	402,97	402,97	398,77	398,77	398,77	398,77	398,77	398,77
	20	852,19	852,15	852,18	852,15	841,61	841,58	<b>841,60</b>	841,58	841,61	841,58
	40	1.799,38	1.799,32	1.799,37	1.799,32	1.773,61	1.773,54	1.773,61	1.773,54	1.773,61	1.773,54
	50	2.308,49	2.308,09	2.308,20	2.308,09	2.277,65	2.277,24	<b>2.277,33</b>	2.277,24	2.277,65	2.277,24
	100	4.719,86	4.719,70	4.719,85	4.719,70	4.658,92	4.658,70	<b>4.658,89</b>	4.658,70	4.658,92	4.658,70
	200	9.487,88	9.487,40	9.487,82	9.487,40	9.347,23	9.346,67	<b>9.347,16</b>	9.346,67	9.347,23	9.346,67
	500	23.779,94	23.778,24	23.779,51	23.778,24	23.397,49	23.395,71	<b>23.396,93</b>	23.395,71	23.397,49	23.395,71
	1000	47.601,14	47.598,57	47.601,12	47.598,57	46.838,16	46.835,14	<b>46.837,61</b>	46.835,14	46.838,16	46.835,14

5.2.4 Regiones con normas  $\ell_1$  y  $\ell_q$

Tabla VI (Continuación)

<b>q=2</b>		<b>10.000 iter.</b>		<b>30.000 iter.</b>					
<b>m</b>	<b>Ptos.</b>	<b>MFP</b>	<b>Gate(1,q)</b>	<b>MFP</b>	<b>Gate(1,q)</b>				
<b>0,5</b>	10	342,71	342,71	342,71	342,71				
	20	726,13	726,13	726,13	726,13				
	40	1.522,89	1.522,88	1.522,89	1.522,88				
	50	1.948,15	1.948,14	1.948,15	1.948,14				
	100	3.947,23	3.947,20	3.947,23	3.947,20				
	200	7.915,60	7.915,53	7.915,60	7.915,53				
	500	19.826,92	19.826,76	19.826,92	19.826,76				
	1000	39.698,61	39.698,29	39.698,61	39.698,29				
<b>1</b>	10	370,88	370,87	370,88	370,87				
	20	773,69	773,67	773,69	773,67				
	40	1.612,25	1.612,26	1.612,25	<b>1.612,22</b>				
	50	2.062,76	2.062,72	2.062,76	2.062,72				
	100	4.176,93	4.176,85	4.176,93	4.176,85				
	200	8.372,50	8.372,42	8.372,50	8.372,42				
	500	20.969,91	20.969,62	20.969,91	20.969,62				
	1000	41.977,83	41.977,25	41.977,83	41.977,25				
<b>1,5</b>	10	395,62	395,62	395,62	395,62				
	20	833,70	833,68	833,70	833,68				
	40	1.753,95	1.753,87	<b>1.753,94</b>	1.753,87				
	50	2.253,83	2.253,47	<b>2.253,57</b>	2.253,47				
	100	4.606,79	4.606,48	<b>4.606,69</b>	4.606,48				
	200	9.233,45	9.232,90	<b>9.233,37</b>	9.232,90				
	500	23.111,95	23.110,33	<b>23.111,51</b>	23.110,33				
	1000	46.268,83	46.265,69	<b>46.268,08</b>	46.265,69				
<b>q=3</b>		<b>10.000 iter.</b>		<b>30.000 iter.</b>		<b>q=5 10.000 iter.</b>		<b>30.000 iter.</b>	
<b>m</b>	<b>Ptos.</b>	<b>MFP</b>	<b>Gate(1,q)</b>	<b>MFP</b>	<b>Gate(1,q)</b>	<b>MFP</b>	<b>Gate(1,q)</b>	<b>MFP</b>	<b>Gate(1,q)</b>
<b>0,5</b>	10	323,47	323,47	323,47	323,47	311,88	311,88	311,88	311,88
	20	684,12	684,12	684,12	684,12	659,86	659,86	659,86	659,86
	40	1.431,90	1.431,89	1.431,90	1.431,89	1.378,60	1.378,60	1.378,60	1.378,60
	50	1.833,49	1.833,48	1.833,49	1.833,48	1.766,03	1.766,03	1.766,03	1.766,03
	100	3.709,07	3.709,05	3.709,07	3.709,05	3.567,67	3.567,66	3.567,67	3.567,66
	200	7.438,62	7.438,58	7.438,62	7.438,58	7.156,14	7.156,11	7.156,14	7.156,11
	500	18.634,09	18.633,98	18.634,09	18.633,98	17.928,59	17.928,53	17.928,59	17.928,53
	1000	37.309,87	37.309,65	37.309,87	37.309,65	35.894,77	35.894,65	35.894,77	35.894,65
<b>1</b>	10	357,65	357,14	<b>357,14</b>	357,14	349,60	348,69	<b>348,80</b>	348,69
	20	739,12	739,06	<b>739,08</b>	739,06	719,20	718,19	718,20	718,19
	40	1.535,86	1.535,84	1.535,86	1.535,84	1.488,73	1.488,73	1.488,73	1.488,73
	50	1.966,82	1.966,71	1.966,73	1.966,71	1.907,93	1.907,85	<b>1.907,86</b>	1.907,85
	100	3.977,39	3.977,37	3.977,39	<b>3.977,36</b>	3.853,00	3.853,13	3.853,00	<b>3.853,07</b>
	200	7.973,47	7.973,40	7.973,47	7.973,40	7.724,40	7.724,36	7.724,40	7.724,36
	500	19.968,99	19.968,80	19.968,99	19.968,80	19.344,96	19.344,86	19.344,96	19.344,86
	1000	39.978,52	39.978,13	39.978,52	39.978,13	38.729,90	38.729,68	38.729,90	38.729,68
<b>1,5</b>	10	387,39	387,39	387,39	387,39	382,25	382,19	382,25	382,19
	20	813,21	813,16	<b>813,18</b>	813,16	800,80	800,54	<b>800,53</b>	<b>800,53</b>
	40	1.703,24	1.702,69	<b>1.702,74</b>	1.702,69	1.674,22	1.671,93	<b>1.672,11</b>	1.671,93
	50	2.192,02	2.190,83	<b>2.190,91</b>	2.190,83	2.157,07	2.153,74	<b>2.154,01</b>	2.153,74
	100	4.462,03	4.460,78	<b>4.460,98</b>	4.460,78	4.377,74	4.375,96	<b>4.376,19</b>	<b>4.376,77</b>
	200	8.940,82	8.939,38	<b>8.939,78</b>	8.939,38	8.773,32	8.769,87	<b>8.770,23</b>	8.769,87
	500	22.377,51	22.374,31	<b>22.375,31</b>	22.374,31	21.954,33	21.949,50	<b>21.950,27</b>	21.949,50
	1000	44.808,02	44.798,98	<b>44.801,00</b>	44.798,98	43.972,67	43.955,63	<b>43.957,24</b>	43.955,63

### 5.2.3 Regiones con normas $\ell_1$ y $\ell_\infty$

En las tablas VII y VIII tenemos los resultados obtenidos para la norma  $\ell_\infty$  con las normas  $\ell_1$  y  $\ell_p$ , respectivamente. Dado que la literatura científica al respecto prácticamente no estudia el caso  $\ell_\infty$ , hemos contrastados los resultados con los programas Gate( $p,128$ ) y MFP( $p,128$ ), ya que un mayor valor de  $q$  da lugar a “0/0” ó “inf” en el software MATLAB, y obtenemos, como era de esperar, mayor precisión y, sobre todo, mayor velocidad en la solución del problema (1).

Nótese que, para una comparación más precisa de la función objetivo, se pueden utilizar las desigualdades:  $\| \cdot \|_\infty \leq \| \cdot \|_q \leq 2^{1/q} \cdot \| \cdot \|_\infty$

Tabla VII: Promedio de distancias y tiempos para  $\ell_1$  y  $\ell_\infty$  (pesos unitarios)

Pesos unitarios		Gate(1, $\infty$ )		Gate(1,128)		MFP(1,128)	
m	Ptos.	D	T	D	T	D	T
0,5	10	302,18	0,078	302,81	2,548	308,00	0,067
	20	641,33	0,078	642,16	3,455	651,72	0,081
	40	1.338,90	0,078	1.340,95	5,652	1.370,04	0,152
	50	1.714,13	0,078	1.716,87	7,138	1.757,58	0,186
	100	3.460,62	0,078	3.466,64	11,833	3.548,47	0,573
	200	6.942,73	0,078	6.952,73	21,134	7.124,97	0,516
	500	17.393,50	0,078	17.419,76	48,938	17.851,42	1,367
	1000	34.825,00	0,078	34.892,22	96,250	35.732,22	3,234
1	10	340,75	0,438	342,81	2,792	355,07	0,288
	20	700,05	0,438	703,53	3,080	713,82	0,336
	40	1.448,80	0,438	1.459,16	5,530	1.464,36	0,975
	50	1.861,00	0,438	1.881,07	6,170	1.870,08	1,242
	100	3.749,70	0,438	3.795,47	10,828	3.756,55	1,947
	200	7.520,00	0,438	7.605,51	19,047	7.530,90	3,494
	500	18.831,50	0,438	19.057,06	38,820	18.855,01	10,828
	1000	37.716,00	0,438	38.127,34	89,016	37.741,70	25,531
1,5	10	377,61	0,422	378,95	2,883	384,91	0,295
	20	790,57	0,422	796,50	3,731	801,16	0,516
	40	1.652,45	0,422	1.662,27	5,461	1.664,37	0,830
	50	2.130,10	0,422	2.142,45	6,656	2.136,77	0,922
	100	4.333,54	0,422	4.366,46	10,634	4.333,81	2,419
	200	8.684,95	0,422	8.763,97	19,522	8.685,34	3,813
	500	21.738,00	0,422	21.945,93	45,133	21.738,76	10,180
	1000	43.550,00	0,422	43.988,16	88,938	43.550,98	19,188

5.2.3 Regiones con normas  $\ell_1$  y  $\ell_\infty$

Tabla VII (cont.): Promedio de distancias y tiempos para  $\ell_1$  y  $\ell_\infty$  (pesos no unitarios)

Pesos no unitarios		Gate(1, $\infty$ )		Gate(1,128)		MFP(1,128)	
m	Ptos.	D	T	D	T	D	T
0,5	10	1.704,14	0,031	1.716,53	2,981	1.733,39	0,100
	20	3.717,87	0,031	3.733,29	3,592	3.788,38	0,063
	40	7.485,28	0,031	7.500,97	5,606	7.659,39	0,145
	50	9.366,41	0,031	9.380,05	6,623	9.534,84	0,281
	100	18.720,89	0,031	18.748,36	11,498	19.151,59	0,327
	200	37.591,65	0,031	37.647,85	20,713	38.457,09	0,797
	500	94.131,50	0,031	94.286,53	47,836	96.352,92	3,367
	1000	188.539,50	0,031	188.943,01	93,641	193.026,35	5,797
1	10	1.860,60	0,031	1.876,50	2,600	1.888,28	0,294
	20	4.067,10	0,031	4.103,85	3,786	4.083,95	0,516
	40	8.227,90	0,031	8.361,11	5,638	8.247,67	0,964
	50	10.256,00	0,031	10.346,79	6,427	10.267,76	1,183
	100	20.438,65	0,031	20.665,20	10,981	20.482,70	2,306
	200	40.967,60	0,031	41.519,85	19,563	40.992,53	4,791
	500	102.500,00	0,031	103.830,12	46,813	102.561,33	10,383
	1000	205.044,00	0,031	207.730,82	90,734	205.171,17	23,625
1,5	10	2.104,55	0,281	2.130,08	2,683	2.140,70	0,316
	20	4.600,94	0,281	4.625,30	4,047	4.661,45	0,473
	40	9.508,57	0,281	9.562,61	5,798	9.576,11	0,936
	50	11.945,93	0,281	12.018,01	6,691	12.047,98	1,239
	100	23.778,89	0,281	23.925,00	11,100	23.832,55	2,061
	200	47.708,07	0,281	48.159,11	19,828	47.734,20	4,309
	500	119.319,90	0,281	120.349,18	45,828	119.320,35	11,711
	1000	238.640,33	0,281	240.945,07	89,422	238.640,61	22,297

### 5.2.4 Regiones con normas $l_p$ y $l_\infty$

Tabla VIII: Promedio de distancias y tiempos para  $l_p$  y  $l_\infty$

<b>p= 1.2</b>		<b>Gate(p,∞)</b>		<b>Gate(p,128)</b>		<b>MFP(p,128)</b>	
<b>m</b>	<b>Ptos.</b>	<b>D</b>	<b>T</b>	<b>D</b>	<b>T</b>	<b>D</b>	<b>T</b>
<b>0,5</b>	10	298,31	0,586	298,75	0,453	300,13	3,044
	20	636,15	0,825	636,75	0,892	637,70	4,014
	40	1.326,71	1,288	1.327,60	1,731	1.329,01	6,847
	50	1.697,02	1,498	1.698,04	1,834	1.700,12	8,352
	100	3.427,56	2,614	3.429,19	3,473	3.433,45	14,384
	200	6.877,08	4,838	6.879,62	7,984	6.889,25	26,431
	500	17.228,28	11,750	17.234,42	21,227	17.256,41	62,461
1000	34.499,92	23,094	34.511,62	49,500	34.547,63	122,844	
<b>1</b>	10	327,38	0,394	329,48	0,955	329,68	3,238
	20	680,94	0,738	683,77	1,789	683,61	4,150
	40	1.406,66	1,216	1.413,08	3,539	1.415,37	6,403
	50	1.803,69	1,405	1.813,36	4,927	1.818,88	7,655
	100	3.636,58	2,489	3.653,00	6,894	3.661,64	12,670
	200	7.293,64	4,853	7.321,49	15,469	7.347,23	23,800
	500	18.267,60	11,703	18.333,09	36,969	18.416,73	62,719
1000	36.589,01	22,922	36.730,78	94,813	36.805,97	104,547	
<b>1,5</b>	10	354,09	0,336	354,55	0,677	355,82	3,330
	20	748,70	0,553	754,50	1,422	755,09	4,159
	40	1.573,58	1,014	1.582,04	3,864	1.583,28	6,972
	50	2.026,66	0,925	2.044,87	4,977	2.038,74	8,358
	100	4.138,93	2,278	4.176,18	8,888	4.168,70	14,491
	200	8.302,90	4,309	8.390,96	18,800	8.358,55	26,697
	500	20.783,05	11,969	21.057,59	45,516	20.996,98	62,969
1000	41.624,90	24,047	42.179,62	91,734	41.934,06	123,656	
<b>p= 1.4</b>		<b>Gate(p,∞)</b>		<b>Gate(p,128)</b>		<b>MFP(p,128)</b>	
<b>m</b>	<b>Ptos.</b>	<b>D</b>	<b>T</b>	<b>D</b>	<b>T</b>	<b>D</b>	<b>T</b>
<b>0,5</b>	10	295,79	0,566	296,10	0,398	297,75	2,989
	20	632,74	0,813	633,16	0,620	633,69	4,130
	40	1.318,99	1,270	1.319,86	1,550	1.322,02	6,433
	50	1.686,16	1,488	1.687,04	1,588	1.689,89	8,328
	100	3.406,71	2,489	3.408,12	3,189	3.413,12	14,300
	200	6.835,65	4,875	6.838,08	7,291	6.846,36	26,447
	500	17.124,82	11,648	17.130,76	18,156	17.155,72	62,469
1000	34.294,52	23,094	34.306,09	46,016	34.343,92	122,906	
<b>1</b>	10	318,77	0,428	320,32	0,858	320,84	3,244
	20	669,05	0,577	671,20	1,430	670,69	4,209
	40	1.380,38	0,947	1.386,63	3,378	1.385,54	6,252
	50	1.767,76	1,194	1.777,54	4,853	1.778,93	6,758
	100	3.565,99	2,294	3.582,86	8,803	3.582,99	11,341
	200	7.152,43	4,647	7.181,80	16,469	7.180,70	22,463
	500	17.915,16	6,945	17.983,95	37,766	18.002,27	52,563
1000	35.884,33	23,094	36.034,91	95,969	36.024,39	104,500	
<b>1,5</b>	10	339,18	0,256	339,74	0,536	341,33	3,223
	20	721,92	0,425	723,85	1,175	726,05	3,902
	40	1.516,93	0,563	1.518,83	2,233	1.527,09	6,770
	50	1.954,18	0,480	1.957,73	2,516	1.971,11	8,359
	100	3.988,80	0,811	3.997,64	4,727	4.016,32	14,470
	200	8.035,04	1,231	8.047,05	9,269	8.086,27	26,575
	500	20.166,98	4,352	20.232,52	21,391	20.317,97	62,953
1000	40.383,10	9,984	40.522,32	60,797	40.672,40	123,609	

5.2.4 Regiones con normas  $\ell_p$  y  $\ell_\infty$

Tabla VIII: (Continuación)

<b>p= 1.6</b>		<b>Gate(p,∞)</b>		<b>Gate(p,128)</b>		<b>MFP(p,128)</b>	
<b>m</b>	<b>Ptos.</b>	<b>D</b>	<b>T</b>	<b>D</b>	<b>T</b>	<b>D</b>	<b>T</b>
<b>0,5</b>	10	294,01	0,578	294,28	0,344	295,19	2,864
	20	630,27	0,814	630,62	0,616	631,43	4,108
	40	1.313,70	1,280	1.314,56	1,513	1.315,68	6,792
	50	1.678,69	1,484	1.679,50	1,398	1.682,46	8,142
	100	3.392,61	2,442	3.393,90	2,786	3.398,39	14,180
	200	6.807,61	4,950	6.809,96	7,300	6.818,58	26,416
	500	17.054,95	11,688	17.060,51	16,195	17.082,83	62,484
	1000	34.155,94	23,328	34.167,09	42,328	34.204,59	123,281
<b>1</b>	10	312,63	0,373	313,94	0,644	313,93	2,786
	20	660,11	0,545	661,75	1,186	661,44	4,109
	40	1.362,74	0,956	1.368,86	2,328	1.367,07	6,214
	50	1.743,55	1,109	1.753,17	3,398	1.751,39	6,309
	100	3.518,63	2,364	3.535,73	6,484	3.529,03	11,073
	200	7.057,51	4,200	7.087,19	13,178	7.078,36	19,869
	500	17.678,40	8,516	17.749,48	29,734	17.734,05	46,852
	1000	35.410,61	22,891	35.567,19	81,422	35.506,64	102,719
<b>1,5</b>	10	329,02	0,233	329,66	0,398	330,69	2,980
	20	702,52	0,264	703,50	0,786	705,89	4,114
	40	1.475,51	0,331	1.476,49	0,906	1.484,04	6,894
	50	1.897,18	0,239	1.899,06	1,314	1.913,44	8,355
	100	3.874,83	0,375	3.878,83	2,192	3.902,73	14,481
	200	7.787,32	0,725	7.791,25	5,194	7.847,09	26,597
	500	19.528,97	1,852	19.534,69	13,867	19.690,43	62,945
	1000	39.114,66	3,891	39.125,87	31,938	39.384,98	123,641
<b>p= 1.8</b>		<b>Gate(p,∞)</b>		<b>Gate(p,128)</b>		<b>MFP(p,128)</b>	
<b>m.</b>	<b>Ptos.</b>	<b>D</b>	<b>T</b>	<b>D</b>	<b>T</b>	<b>D</b>	<b>T</b>
<b>0,5</b>	10	292,71	0,573	292,92	0,236	294,49	2,614
	20	628,44	0,822	628,73	0,569	629,76	3,989
	40	1.309,87	1,270	1.310,69	1,289	1.312,01	6,738
	50	1.673,28	1,489	1.674,06	1,233	1.676,94	7,766
	100	3.382,58	2,403	3.383,77	2,452	3.388,39	13,880
	200	6.787,66	4,894	6.789,89	6,478	6.798,92	26,438
	500	17.005,03	11,664	17.010,38	13,336	17.033,16	62,516
	1000	34.057,27	23,328	34.067,88	34,016	34.108,93	122,859
<b>1</b>	10	308,18	0,313	309,32	0,433	309,58	2,894
	20	653,63	0,491	655,28	0,966	655,38	3,656
	40	1.350,25	0,833	1.356,04	1,898	1.353,51	6,081
	50	1.726,40	1,008	1.735,91	2,805	1.730,28	6,028
	100	3.485,13	2,195	3.502,33	5,069	3.492,81	10,578
	200	6.990,28	4,563	7.020,24	9,966	7.003,77	19,466
	500	17.510,82	11,914	17.584,36	29,547	17.541,64	45,266
	1000	35.075,07	24,266	35.235,75	62,219	35.152,78	101,094
<b>1,5</b>	10	321,68	0,236	322,50	0,344	324,00	3,011
	20	688,11	0,205	688,40	0,630	691,88	4,317
	40	1.444,20	0,281	1.444,61	0,644	1.454,48	6,638
	50	1.856,02	0,169	1.856,51	1,119	1.871,66	8,361
	100	3.785,04	0,288	3.786,05	1,080	3.815,71	14,463
	200	7.594,89	0,528	7.597,01	3,097	7.650,82	26,597
	500	19.031,39	1,320	19.036,75	12,117	19.174,44	62,898
	1000	38.118,03	2,750	38.128,60	7,438	38.373,62	123,813

Tabla VIII: (Continuación)

<b>p= 3</b>		<b>Gate(p,∞)</b>		<b>Gate(p,128)</b>		<b>MFP(p,128)</b>	
<b>m</b>	<b>Ptos.</b>	<b>D</b>	<b>T</b>	<b>D</b>	<b>T</b>	<b>D</b>	<b>T</b>
<b>0,5</b>	10	289,14	0,508	289,29	0,228	290,80	2,961
	20	563,52	0,817	623,56	0,442	624,98	3,927
	40	1.299,49	1,266	1.300,03	1,159	1.302,15	6,416
	50	1.658,70	1,486	1.659,28	1,044	1.661,77	7,778
	100	3.356,44	2,597	3.357,34	1,900	3.362,09	13,253
	200	6.735,11	4,231	6.736,86	4,888	6.745,03	24,134
	500	16.873,68	11,508	16.877,94	9,094	16.901,31	56,727
	1000	33.795,85	23,953	33.803,78	19,656	33.845,35	111,219
<b>1</b>	10	296,13	0,253	296,56	0,397	297,99	2,588
	20	636,06	0,284	636,39	0,619	637,75	3,461
	40	1.317,74	0,716	1.321,93	1,395	1.319,56	5,013
	50	1.681,65	1,081	1.689,31	2,245	1.683,48	6,123
	100	3.398,55	2,100	3.413,61	3,939	3.400,83	8,703
	200	6.816,22	4,719	6.843,64	7,559	6.820,39	18,775
	500	17.077,02	6,023	17.150,08	21,938	17.087,92	44,406
	1000	34.204,76	23,797	34.378,25	43,109	34.226,44	98,078
<b>1,5</b>	10	301,69	0,205	302,93	0,266	302,92	2,902
	20	649,17	0,156	649,39	0,502	651,15	3,455
	40	1.356,12	0,195	1.356,69	0,408	1.362,49	5,416
	50	1.736,89	0,159	1.737,33	0,605	1.746,34	7,133
	100	3.524,49	0,241	3.525,30	0,383	3.542,55	12,136
	200	7.068,07	0,431	7.069,70	0,778	7.108,27	24,416
	500	17.713,74	0,992	17.717,86	2,422	17.812,15	57,336
	1000	35.473,71	2,109	35.482,11	3,969	35.662,87	112,594
<b>p= 5</b>		<b>Gate(p,∞)</b>		<b>Gate(p,128)</b>		<b>MFP(p,128)</b>	
<b>m</b>	<b>Ptos.</b>	<b>D</b>	<b>T</b>	<b>D</b>	<b>T</b>	<b>D</b>	<b>T</b>
<b>0,5</b>	10	287,58	0,477	287,70	0,136	289,14	2,759
	20	561,22	0,747	621,19	0,334	622,31	4,113
	40	1.294,76	1,163	1.295,12	0,888	1.296,70	6,800
	50	1.652,19	1,373	1.652,67	0,791	1.655,36	8,289
	100	3.345,82	2,433	3.346,53	1,108	3.350,35	14,288
	200	6.712,84	4,584	6.714,24	3,413	6.721,11	26,313
	500	16.817,91	11,016	16.821,33	7,055	16.843,85	62,086
	1000	33.681,70	21,984	33.688,08	15,000	33.729,62	122,109
<b>1</b>	10	290,78	0,239	290,98	0,316	291,53	2,517
	20	626,69	0,361	626,85	0,394	628,24	2,984
	40	1.304,12	0,667	1.306,38	1,263	1.304,79	5,105
	50	1.662,71	1,055	1.666,71	2,228	1.663,79	6,163
	100	3.363,28	1,820	3.372,16	4,105	3.364,86	8,583
	200	6.745,28	3,681	6.762,67	7,050	6.748,05	20,209
	500	16.899,54	11,016	16.963,25	20,844	16.906,73	47,516
	1000	33.847,00	7,094	33.979,34	41,906	33.852,37	89,875
<b>1,5</b>	10	292,84	0,255	293,88	0,216	293,36	2,606
	20	632,09	0,156	632,21	0,586	632,95	3,183
	40	1.318,22	0,233	1.318,63	0,508	1.319,70	6,359
	50	1.684,58	0,153	1.684,91	0,456	1.687,32	7,786
	100	3.413,52	0,277	3.414,10	0,423	3.417,61	13,519
	200	6.846,39	0,453	6.847,48	0,728	6.853,46	26,472
	500	17.158,86	1,320	17.161,81	2,023	17.181,51	62,625
	1000	34.358,02	2,313	34.364,09	3,500	34.402,02	123,297

5.2.4 Regiones con normas  $\ell_p$  y  $\ell_\infty$

Tabla VIII: (Continuación)

$p=2$		Gate( $p,\infty$ )		Gate( $p,128$ )		MFP( $p,128$ )	
$m$	Ptos.	D	T	D	T	D	T
0,5	10	291,72	0,414	291,92	0,202	293,45	2,363
	20	627,05	0,466	627,32	0,531	629,28	2,931
	40	1.307,00	0,572	1.307,77	1,231	1.309,43	4,905
	50	1.669,24	0,605	1.669,95	1,138	1.672,65	5,844
	100	3.375,16	0,875	3.376,26	2,053	3.381,34	9,794
	200	6.772,88	1,381	6.775,00	5,769	6.784,89	17,550
	500	16.968,05	2,953	16.973,19	11,000	17.001,49	40,117
	1000	33.984,10	5,641	33.994,13	27,875	34.035,06	78,359
1	10	304,84	0,213	305,82	0,473	306,54	2,466
	20	648,81	0,247	650,51	0,706	650,56	2,989
	40	1.341,04	0,370	1.346,52	1,425	1.344,31	4,077
	50	1.713,75	0,463	1.723,11	2,209	1.717,80	4,463
	100	3.460,48	0,684	3.477,47	3,477	3.465,58	7,248
	200	6.940,75	1,078	6.970,45	6,019	6.950,23	13,334
	500	17.387,41	1,211	17.460,45	22,469	17.411,47	29,875
	1000	34.827,80	6,359	34.993,08	42,391	34.890,61	62,141
1,5	10	316,14	0,169	317,14	0,281	318,20	2,548
	20	677,29	0,117	677,55	0,500	681,43	3,102
	40	1.419,91	0,142	1.420,40	0,522	1.434,00	4,895
	50	1.824,57	0,105	1.825,01	0,908	1.848,92	5,677
	100	3.712,83	0,159	3.713,85	0,648	3.754,89	9,225
	200	7.447,27	0,266	7.449,29	0,909	7.530,40	16,594
	500	18.661,81	0,586	18.666,87	1,992	18.885,61	37,852
	1000	37.377,14	1,281	37.387,16	4,141	37.800,19	73,641

5.2.5 Regiones con normas  $\ell_p$  y  $\ell_q$

En la Tabla IX se hace un estudio comparativo de los procedimientos MFP y Gate( $p,q$ ), para los mismos conjuntos de puntos anteriores, variando la pendiente  $m$ , y tomando  $p < q$ , pertenecientes al conjunto de valores  $\{1.2, 1.5, 1.8, 2, 2.5, 3\}$ . En cuanto a precisión, el algoritmo MFP gana en el 45% de los casos estudiados, pero el promedio de mejora se sitúa en torno el 0,02%. Por el contrario, tarda en promedio 2,4 veces lo que tarda el procedimiento Gate( $p,q$ ) perdiendo en velocidad en el 75% de las ocasiones. En general, Gate( $p,q$ ) es más eficiente que el procedimiento MFP cuanto más lejana la unidad sea la pendiente de la recta de separación

Tabla IX: MFP vs Gate(p,q) variando  $p, q$  y  $m$ .

p= 1,2			MFP		Gate(p,q)		MFP			Gate(p,q)	
q	m	Ptos.	Ds	Time	Ds	Time	q	Ds	Time	Ds	Time
		P-18	120,50	0,406	120,50	0,234		115,30	0,297	115,30	0,188
		Z-30	272,05	0,094	272,05	0,141		261,39	0,281	261,39	0,188
		Z-50	993,45	0,797	993,45	0,250		949,41	0,969	949,41	0,375
	<b>0,5</b>	100	3.867,67	9,000	3.867,67	0,344		3.692,10	8,891	3.692,10	0,375
		200	8.160,13	16,109	8.160,13	0,750		7.758,46	16,031	7.758,45	0,828
		500	20.776,22	37,672	20.776,22	2,281		19.852,65	37,609	19.852,65	2,203
		1000	42.348,06	75,922	42.348,06	3,891		40.291,23	75,859	40.291,22	4,313
		P-18	126,60	0,172	126,60	0,297		123,56	0,250	123,68	0,484
		Z-30	281,34	0,297	281,34	1,266		274,87	0,375	274,75	1,438
		Z-50	1.029,79	0,688	1.029,81	1,766		1.004,31	0,422	1.004,45	2,328
<b>1,5</b>	<b>1</b>	100	3.936,36	7,625	3.936,36	1,891	<b>1,8</b>	3.793,71	7,625	3.793,71	3,109
		200	8.328,25	13,797	8.328,26	6,344		8.000,33	13,813	8.000,35	9,391
		500	21.414,82	20,984	21.414,24	11,391		20.729,99	23,172	20.781,66	22,016
		1000	43.247,40	65,688	43.247,53	48,953		41.601,29	65,063	41.601,41	49,328
		P-18	128,11	1,094	128,11	0,109		127,11	1,047	127,11	0,156
		Z-30	286,78	0,984	286,78	0,234		284,40	0,906	284,40	0,359
		Z-50	1.053,58	1,563	1.053,59	0,438		1.041,97	1,516	1.041,97	0,422
<b>1,5</b>		100	4.094,83	7,188	4.094,90	1,859		4.020,81	7,734	4.020,66	2,359
		200	8.707,31	15,828	8.707,37	3,516		8.579,75	15,688	8.579,44	5,281
		500	22.050,84	37,891	22.050,92	6,484		21.833,99	38,188	21.834,25	9,547
		1000	45.030,62	73,266	45.030,77	10,031		44.468,43	73,672	44.468,70	15,969
p= 1,2			MFP		Gate(p,q)		MFP			Gate(p,q)	
q	m	Ptos.	Ds	Time	Ds	Time	q	Ds	Time	Ds	Time
		P-18	112,96	0,266	112,96	0,203		109,26	0,188	109,26	0,234
		Z-30	256,61	0,219	256,60	0,172		248,95	0,109	248,94	0,234
		Z-50	929,48	0,828	929,48	0,234		897,31	1,156	897,30	0,328
	<b>0,5</b>	100	3.613,46	6,859	3.613,46	0,266		3.488,15	8,922	3.488,15	0,391
		200	7.576,88	12,266	7.576,88	0,609		7.284,12	16,172	7.284,11	0,906
		500	19.437,44	28,438	19.437,44	1,578		18.771,85	37,813	18.771,84	2,734
		1000	39.360,31	56,563	39.360,30	3,219		37.855,57	76,047	37.855,56	4,797
		P-18	121,94	0,219	122,05	0,813		119,19	0,406	119,30	0,875
		Z-30	271,22	0,328	271,10	1,281		265,21	0,563	265,11	1,469
		Z-50	991,60	0,266	992,76	2,125		966,65	0,922	968,56	0,203
<b>2</b>	<b>1</b>	100	3.728,99	6,219	3.728,99	3,516	<b>2,5</b>	3.624,58	7,688	3.624,58	4,641
		200	7.849,36	11,172	7.849,38	8,297		7.601,72	13,859	7.601,74	10,219
		500	20.416,09	12,125	20.457,98	24,000		19.903,22	11,656	19.912,04	40,109
		1000	40.841,50	51,250	40.841,63	41,188		39.590,75	46,953	39.590,90	53,719
		P-18	126,62	0,766	126,62	0,172		125,78	1,422	125,78	0,281
		Z-30	283,26	0,594	283,26	0,531		281,35	1,203	281,39	1,391
		Z-50	1.036,37	0,906	1.036,37	0,391		1.026,93	2,094	1.026,92	0,766
<b>1,5</b>		100	3.976,27	3,938	3.976,13	3,203		3.904,05	6,984	3.932,75	8,078
		200	8.474,21	8,547	8.473,91	7,125		8.300,87	16,578	8.429,35	8,328
		500	21.731,37	19,656	21.731,83	13,844		21.559,06	37,922	21.560,56	18,938
		1000	44.205,09	36,750	44.205,32	24,938		43.427,33	75,469	43.770,77	40,250

5.2.5 Regiones con normas  $\ell_p$  y  $\ell_q$

Tabla IX: (Continuación)

$p=1,2$		MFP		Gate(p,q)		
q	m	Ptos.	Ds	Time	Ds	Time
		P-18	107,17	0,234	107,17	0,250
		Z-30	244,48	0,172	244,48	0,297
		Z-50	878,67	1,172	878,67	0,328
	<b>0,5</b>	100	3.417,21	8,453	3.417,21	0,391
		200	7.114,76	15,219	7.114,76	0,891
		500	18.389,33	36,500	18.389,32	2,813
		1000	36.980,57	72,609	36.980,55	4,750
		P-18	117,50	0,453	117,61	0,969
		Z-30	261,64	0,672	261,54	1,531
		Z-50	951,78	1,281	953,98	0,188
<b>3</b>	<b>1</b>	100	3.564,27	7,344	3.564,27	4,766
		200	7.455,01	13,281	7.455,04	10,656
		500	19.599,00	13,875	19.602,84	38,000
		1000	38.846,27	55,469	38.846,42	56,313
		P-18	125,26	1,672	125,26	0,344
		Z-30	280,20	1,391	280,36	1,406
		Z-50	1.021,17	2,484	1.021,16	1,203
	<b>1,5</b>	100	3.862,39	6,281	3.880,55	7,891
		200	8.199,29	14,938	8.323,36	9,266
		500	21.453,93	34,359	21.455,87	16,922
		1000	42.892,30	66,094	43.444,98	40,234

$p=1,5$		MFP		Gate(p,q)		MFP		Gate(p,q)			
q	m	Ptos.	Ds	Time	Ds	Time	q	Ds	Time	Ds	Time
		P-18	113,72	0,344	113,72	0,125		111,44	0,297	111,44	0,094
		Z-30	258,12	0,156	258,12	0,141		253,42	0,063	253,42	0,156
		Z-50	938,75	0,938	938,75	0,297		919,07	0,828	919,07	0,234
	<b>0,5</b>	100	3.665,50	8,938	3.665,50	0,313		3.587,77	7,016	3.587,77	0,250
		200	7.692,45	16,125	7.692,45	0,766		7.513,74	12,859	7.513,74	0,578
		500	19.635,78	37,703	19.635,78	2,281		19.228,41	29,141	19.228,41	1,688
		1000	39.926,73	76,156	39.926,73	4,203		39.010,44	58,438	39.010,44	2,969
		P-18	117,35	0,156	117,35	0,125		116,56	0,109	116,56	0,125
		Z-30	263,54	0,266	263,54	0,328		260,96	0,422	260,96	0,281
		Z-50	961,23	1,078	961,23	0,375		950,21	0,609	950,21	0,266
<b>1,8</b>	<b>1</b>	100	3.708,41	7,625	3.708,41	1,063	<b>2</b>	3.647,02	6,359	3.647,02	0,938
		200	7.806,03	14,172	7.806,04	3,656		7.665,37	11,422	7.665,39	2,813
		500	20.060,97	28,844	20.060,46	6,281		19.774,86	10,469	19.774,51	5,906
		1000	40.502,19	52,313	40.502,35	27,313		39.789,48	48,297	39.789,60	18,625
		P-18	118,13	0,984	118,13	0,094		117,78	0,719	117,78	0,063
		Z-30	266,70	0,766	266,70	0,156		265,81	0,500	265,81	0,109
		Z-50	974,52	1,813	974,52	0,219		970,48	1,125	970,48	0,172
	<b>1,5</b>	100	3.798,29	8,188	3.798,29	0,672		3.778,43	4,422	3.778,43	0,516
		200	8.012,05	16,125	8.012,05	1,313		7.974,07	8,516	7.974,08	0,953
		500	20.367,27	38,609	20.367,27	2,734		20.281,29	19,641	20.281,30	1,656
		1000	41.491,02	74,313	41.491,02	5,953		41.261,97	37,125	41.261,98	3,859

Tabla IX: (Continuación)

<b>p= 1,5</b>		<b>MFP</b>		<b>Gate(p,q)</b>		<b>MFP</b>		<b>Gate(p,q)</b>	
<b>q</b>	<b>m</b>	<b>Ptos.</b>	<b>Ds</b>	<b>Time</b>	<b>Ds</b>	<b>Time</b>	<b>q</b>	<b>Ds</b>	<b>Time</b>
		P-18	107,81	0,266	107,81	0,125		105,73	0,281
		Z-30	245,86	0,063	245,86	0,156		241,45	0,234
		Z-50	887,22	1,219	887,22	0,281		868,75	1,422
	<b>0,5</b>	100	3.463,92	9,250	3.463,92	0,328		3.393,83	9,797
		200	7.225,01	17,078	7.225,01	0,797		7.057,83	17,484
		500	18.573,64	39,109	18.573,64	2,484		18.196,64	39,938
		1000	37.526,71	79,766	37.526,71	4,172		36.662,97	78,344
		P-18	115,14	0,156	115,14	0,203		113,62	0,234
		Z-30	255,55	0,422	255,46	0,453		252,00	0,516
		Z-50	931,18	1,328	931,18	0,344		917,89	1,453
<b>2,5</b>	<b>1</b>	100	3.546,23	7,969	3.546,23	1,469	<b>3</b>	3.487,30	8,078
		200	7.428,25	14,406	7.428,27	4,578		7.285,27	14,703
		500	19.292,65	8,047	19.298,38	28,047		18.999,36	6,594
		1000	38.587,47	67,625	38.587,60	30,125		37.860,14	70,188
		P-18	117,12	1,313	117,12	0,094		116,68	1,656
		Z-30	264,23	0,984	264,23	0,172		263,21	1,188
		Z-50	963,00	2,141	963,00	0,281		958,11	2,625
	<b>1,5</b>	100	3.745,47	8,438	3.745,48	0,813		3.718,34	8,250
		200	7.908,39	16,891	7.908,40	1,422		7.867,93	15,531
		500	20.130,81	42,453	20.130,82	3,188		20.036,10	37,359
		1000	40.865,40	83,219	40.865,42	6,250		40.619,63	71,109
<b>p= 1,8</b>		<b>MFP</b>		<b>Gate(p,q)</b>		<b>MFP</b>		<b>Gate(p,q)</b>	
<b>q</b>	<b>m</b>	<b>Ptos.</b>	<b>Ds</b>	<b>Time</b>	<b>Ds</b>	<b>Time</b>	<b>q</b>	<b>Ds</b>	<b>Time</b>
		P-18	110,47	0,328	110,47	0,078		106,90	0,328
		Z-30	251,43	0,156	251,43	0,125		243,99	0,297
		Z-50	912,41	0,859	912,41	0,250		880,86	1,203
	<b>0,5</b>	100	3.571,15	7,859	3.571,15	0,250		3.448,51	10,172
		200	7.471,14	14,141	7.471,14	0,578		7.186,41	18,469
		500	19.089,80	33,391	19.089,80	1,766		18.445,65	43,344
		1000	38.777,74	65,406	38.777,74	3,078		37.313,85	87,266
		P-18	112,12	0,125	112,12	0,094		111,00	0,250
		Z-30	253,90	0,297	253,89	0,156		249,88	0,594
		Z-50	923,32	0,734	923,32	0,203		906,34	1,484
<b>2</b>	<b>1</b>	100	3.591,69	7,016	3.591,69	0,703	<b>2,5</b>	3.496,68	8,938
		200	7.530,81	9,141	7.530,83	2,250		7.314,82	14,609
		500	19.311,54	17,563	19.311,55	2,563		18.897,18	23,797
		1000	39.063,42	29,250	39.063,64	16,922		37.950,24	57,047
		P-18	112,43	0,750	112,43	0,047		111,93	1,563
		Z-30	255,27	0,531	255,26	0,109		253,90	1,047
		Z-50	928,66	1,156	928,65	0,125		922,95	2,766
	<b>1,5</b>	100	3.631,96	5,156	3.631,96	0,344		3.603,24	9,188
		200	7.616,53	9,344	7.616,53	0,641		7.560,90	18,125
		500	19.417,71	21,844	19.417,71	1,406		19.288,87	39,625
		1000	39.488,27	40,984	39.488,27	3,453		39.143,17	73,688

5.2.5 Regiones con normas  $\ell_p$  y  $\ell_q$

Tabla IX: (Continuación)

		<b>p= 1,8</b>		<b>MFP</b>		<b>Gate(p,q)</b>			
<b>q</b>	<b>m</b>	<b>Ptos.</b>	<b>Ds</b>	<b>Time</b>	<b>Ds</b>	<b>Time</b>	<b>Ds</b>	<b>Time</b>	
		P-18	104,86	0,234	104,86	0,109			
		Z-30	239,64	0,172	239,64	0,141			
		Z-50	862,54	1,141	862,54	0,266			
	<b>0,5</b>	100	3.379,15	8,594	3.379,15	0,344			
		200	7.021,46	15,453	7.021,46	0,828			
		500	18.074,44	36,016	18.074,44	2,609			
		1000	36.461,27	72,531	36.461,28	4,484			
		P-18	110,22	0,172	110,22	0,156			
		Z-30	246,43	0,531	246,35	0,297			
		Z-50	895,33	1,359	895,33	0,281			
<b>3</b>	<b>1</b>	100	3.440,24	7,359	3.440,24	1,141			
		200	7.179,73	13,266	7.179,75	3,719			
		500	18.627,27	5,500	18.628,68	23,109			
		1000	37.257,74	52,828	37.257,89	24,438			
		P-18	111,58	1,438	111,58	0,109			
		Z-30	253,02	1,000	253,02	0,141			
		Z-50	919,07	2,594	919,07	0,219			
	<b>1,5</b>	100	3.585,84	7,156	3.585,84	0,578			
		200	7.525,13	13,750	7.525,13	1,047			
		500	19.205,73	33,141	19.205,73	2,391			
		1000	38.921,98	63,469	38.921,98	5,250			
		<b>p= 2</b>		<b>MFP</b>		<b>Gate(p,q)</b>			
<b>q</b>	<b>m</b>	<b>Ptos.</b>	<b>Ds</b>	<b>Time</b>	<b>Ds</b>	<b>Time</b>	<b>q</b>	<b>Ds</b>	<b>Time</b>
		P-18	106,48	0,219	106,48	0,063		104,45	0,219
		Z-30	243,12	0,078	243,12	0,078		238,79	0,047
		Z-50	877,82	0,609	877,82	0,156		859,59	0,625
	<b>0,5</b>	100	3.441,08	5,156	3.441,08	0,203		3.372,13	4,656
		200	7.167,47	8,703	7.167,47	0,453		7.003,85	7,750
		500	18.383,16	19,156	18.383,16	1,328		18.015,36	16,969
		1000	37.210,00	38,641	37.210,00	2,234		36.363,94	33,969
		P-18	109,03	0,172	109,03	0,078		108,36	0,172
		Z-30	246,82	0,438	246,82	0,188		243,90	0,438
		Z-50	894,41	1,125	894,41	0,234		884,03	1,141
<b>2,5</b>	<b>1</b>	100	3.472,24	3,750	3.472,24	0,563	<b>3</b>	3.417,64	3,375
		200	7.255,57	5,141	7.255,58	2,266		7.127,26	5,281
		500	18.698,97	24,500	18.698,61	4,625		18.444,56	8,313
		1000	37.631,11	18,141	37.631,26	16,156		36.966,25	19,172
		P-18	109,55	1,109	109,55	0,063		109,24	1,234
		Z-30	249,21	0,719	249,20	0,109		248,38	0,813
		Z-50	903,90	1,906	903,89	0,141		900,50	2,094
	<b>1,5</b>	100	3.537,55	6,547	3.537,55	0,406		3.521,53	5,609
		200	7.399,88	12,234	7.399,88	0,688		7.367,20	10,203
		500	18.902,78	29,031	18.902,78	1,500		18.826,07	23,828
		1000	38.345,18	56,047	38.345,18	3,703		38.139,66	46,078
								38.139,66	3,563

Tabla IX: (Continuación)

$p=2,5$			MFP		Gate(p,q)	
q	m	Ptos.	Ds	Time	Ds	Time
		P-18	103,75	0,688	103,75	0,078
		Z-30	237,39	0,188	237,39	0,141
		Z-50	854,57	1,453	854,57	0,234
	<b>0,5</b>	100	3.360,08	8,578	3.360,08	0,359
		200	6.973,25	15,625	6.973,25	0,766
		500	17.912,83	36,156	17.912,83	2,453
		1000	36.195,36	72,609	36.195,36	4,203
		P-18	105,22	0,219	105,21	0,109
		Z-30	239,48	0,469	239,48	0,266
		Z-50	864,54	1,375	864,54	0,313
<b>3</b>	<b>1</b>	100	3.378,16	7,500	3.378,16	0,922
		200	7.028,65	11,359	7.028,68	3,453
		500	18.114,26	31,516	18.114,28	5,031
		1000	36.446,45	42,438	36.446,64	23,297
		P-18	105,49	1,547	105,49	0,078
		Z-30	240,91	0,984	240,90	0,141
		Z-50	869,60	2,656	869,59	0,188
	<b>1,5</b>	100	3.415,56	7,313	3.415,56	0,516
		200	7.106,94	13,922	7.106,94	0,969
		500	18.207,70	33,172	18.207,70	2,094
		1000	36.850,34	64,984	36.850,34	5,172

### 5.3 Conclusiones

En conclusión, después del análisis detallado de cada una de las tablas podemos señalar los siguientes puntos:

- 1) El algoritmo MWP es muy rápido, pero es heurístico y comete errores considerables, según el valor de la pendiente.
- 2) El conocimiento de los portones acelera notablemente la convergencia y se pone de manifiesto en la comparación entre los algoritmos MFP y GMFP, si bien la precisión de GMFP se resiente para valores altos de la pendiente (supuesta la posición relativa " $l_q / l_1$ ").
- 3) Para paliar esta situación hemos elaborado el algoritmo Gate( $p, q$ ), que es más eficiente que los anteriores si intervienen las normas  $l_1$  o/y  $l_\infty$ .

- 4) Se han elaborado también los algoritmos Gdaq y Garsq que nos han permitido contrastar los resultados obtenidos con el algoritmo anterior.
- 5) En el caso general de normas  $\ell_p$  y  $\ell_q$  hemos comprobado que es más rápido el cálculo preciso de los portones en cada iteración, si bien la precisión final es, en promedio, ligeramente inferior (0,02%) que la obtenida en el algoritmo MFP, que pierde en el 55% de los casos, debido a la finalización del procedimiento tras 10.000 iteraciones, especialmente en valores de la pendiente cercanos a la unidad.



---

### CONCLUSIONES Y TRABAJOS FUTUROS

---

#### 6.1 Conclusiones

En esta tesis hemos trabajado con el Problema de Localización de Fermat-Weber en el plano  $\mathbb{R}^2$  dividido en dos subregiones  $\Omega_p$  y  $\Omega_q$ , separadas por una línea recta  $r$ . En cada subregión  $\Omega_p$  y  $\Omega_q$  hemos considerado las métricas  $d_p$  y  $d_q$  asociadas a la correspondientes normas  $\ell_p$  y  $\ell_q$ .

En el primer capítulo exponemos el planteamiento general de los problemas de localización y realizamos una breve introducción histórica al problema de Fermat-Weber general, así como al problema particular tratado en esta tesis. Nos ha parecido conveniente exponer también alguna aplicación real del problema que planteamos.

En el capítulo 2 nos hemos centrado en encontrar el camino más corto entre puntos situados en regiones diferentes (problema 2), empleando una metodología basada en la pendiente de la recta de separación de las regiones y las pendientes de las líneas “geodésicas” de cada región.

Hemos realizado una generalización acerca de cómo encontrar el camino más corto entre dos puntos  $P$  y  $Q$  situados en regiones con normas  $\ell_p$  y  $\ell_q$ , respectivamente, cruzando la recta  $r$  de separación de las regiones, que mejora sustancialmente lo realizado anteriormente por (Zaferanieh et al., 2008), ya que estos autores obtienen sus resultados para el caso de las normas  $\ell_1$  y  $\ell_2$ . Además el intervalo en el que se encuentra el punto de cruce o puerta de acceso (gate) entre las regiones es mucho más ajustado que el encontrado por estos autores. El intervalo obtenido es el intervalo central de un conjunto de seis puntos ordenados en la recta, llamados puntos notables. Obtuvimos también un sistema de ecuaciones que caracteriza la puerta de acceso.

Utilizando dicho sistema, y el conjunto de puntos notables, hemos encontrado las puertas de acceso que se obtienen al particularizar las normas  $\ell_p$  y  $\ell_q$ .

En el caso  $\ell_1$  y  $\ell_q$  se obtienen directamente las puertas de accesos. Así, dependiendo de la pendiente  $m$  de la recta  $r$ , se obtienen diferentes subregiones dentro de cada región  $\Omega_p$  y  $\Omega_q$ , y, respecto de ellas, obtenemos de forma inmediata la puerta de acceso. En algunas ocasiones todos los puntos de una subregión tienen la misma puerta de acceso a la que llamamos Portón (Gate).

Para el caso  $\ell_p$  y  $\ell_\infty$  se encuentran de forma análoga las puertas de acceso y los portones.

En el caso en que el punto esté situado sobre la recta se obtienen resultados análogos a los estudiados anteriormente.

En el capítulo 3 entramos a resolver el problema principal de este trabajo (problema 1) usando el concepto de portón. Encontramos una condición necesaria y suficiente para el punto solución del problema (1) en  $\Omega_q$ , para  $1 < q < \infty$  (teorema 3.5). A partir de la condición suficiente se obtiene un algoritmo para encontrar la solución al problema (1). Sin embargo, la condición sólo es necesaria para  $q = 1$  ó  $q = \infty$ , pero no es suficiente en estos casos, como se ha puesto de manifiesto en varios ejemplos.

El caso  $\ell_1$  y  $\ell_q$  nos trae diversas dificultades como son la posibilidad de encontrar puntos en  $\Omega_1$  que verifiquen la condición necesaria y suficiente del teorema 3.5 y, sin embargo, no son soluciones del problema (1). La proposición 3.8 nos indica que hay una subregión en  $\Omega_1$  que hace de cuenca de atracción para cualquier punto en  $\Omega_1$ , que tomemos como punto inicial del algoritmo obtenido, con lo que podemos obtener un “agujero negro” que no coincidirá con la solución en  $\Omega_1$ .

Para evitar este problema calculamos las derivadas de la función objetivo en una malla que será utilizada más adelante en la obtención del algoritmo correspondiente. Se realiza el mismo estudio para el caso  $\ell_1$  y  $\ell_\infty$ .

La obtención de un método para obtener la solución en  $\Omega_1$  consiste en dividir la región  $\Omega_1$  en rectángulos (regiones factibles) en los que la función objetivo tiene una expresión analítica única con variables separables y además es diferenciable, con derivadas parciales respecto de cada variable independientes de la otra variable, con lo que se aplica un algoritmo estándar para encontrar el mínimo absoluto de una función convexa.

En el cuarto capítulo se detallan los algoritmos que se presentan en la tesis y que están basados en los resultados teóricos del capítulo 3 y en la caracterización de los portones entre dos regiones con métricas diferentes estudiados en el segundo capítulo.

El algoritmo **GMFP** (Gate MultiFacility Procedure) es una modificación del propuesto por Brimberg et al. (2003), (MFP), en el que  $1 \leq p < q < \infty$ . El algoritmo GMFP, de forma similar al MFP, busca la solución óptima en cada región pero, a diferencia del anterior, en cada iteración, además de una aproximación a la solución, se calculan las puertas de acceso y se vuelve a iterar el proceso. El número de puntos a obtener en cada iteración es por tanto inferior al algoritmo MFP.

A continuación se obtiene una familia de algoritmos que dividen el problema inicial, que es no convexo, en tres problemas convexos, buscando la solución en cada región y también en la recta. Para la solución en la recta se utiliza un algoritmo ARS. Para la obtención de la solución en  $\Omega_q$  se utilizan dos métodos. En el primero, tomamos un

punto inicial (media ponderada de los puntos iniciales) y en cada iteración se hallan los portones correspondientes a dicho punto y a los puntos de la región  $\Omega_p$ , obteniendo un nuevo punto como solución del problema de localización simple correspondiente a los portones y a los puntos de  $\Omega_q$ , y así sucesivamente hasta encontrar la solución con una tolerancia determinada. Las iteraciones se puede hacer por dos caminos; el primero utilizando el algoritmo de Weiszfeld y el segundo por medio de un algoritmo ARS. Para la obtención de los portones, en el caso en que sea  $p=1$ , el conocimiento de ellos es inmediato, y en otro caso se utiliza la rutina *gates\_pq*, que devuelve el portón correspondiente a dos puntos cualesquiera  $P \in \Omega_p$  y  $Q \in \Omega_q$  (programa **Gate**( $p,q$ )). El segundo método comienza con el mismo punto inicial, si bien usamos el algoritmo ARS con cuatro posibles direcciones de búsqueda.

Para la obtención de una solución en  $\Omega_1$  se aplica el método de búsqueda basado en las derivadas y en la malla expuesto en el capítulo 3 (programas **Gate**( $1,q$ ) y **Gdaq**) o bien se aplica directamente el algoritmo ARS (programa **Garsq**)

Para obtener la solución en  $\Omega_\infty$  realizamos un giro de 45 grados y obtenemos la mediana. Tras deshacer el giro, se aplica el algoritmo ARS (programas **Gate**( $\bullet,\infty$ )).

En el capítulo 5 se hace una comparativa entre los algoritmos propuestos en este trabajo frente a los algoritmos propuestos anteriormente por otros autores. Para ello se han utilizado conjuntos de puntos utilizados previamente en las referencias y se ha tomado un conjunto de puntos generados aleatoriamente y distribuidos uniformemente en el cuadrado  $[0,100] \times [0,100]$  y con pesos unitarios para  $\ell_1$  y  $\ell_2$ .

Para los conjuntos P-18, Z-30 y Z-50, el menor tiempo de CPU se alcanza por el algoritmo MWP. No obstante la función objetivo es mayor que las dadas por el resto de algoritmos.

El algoritmo GMFP es más rápido y, prácticamente, más preciso que el algoritmo MFP. Además, sobresale por su precisión el algoritmo Gate(1,2) que, salvo en  $m=0,5$ , es también el algoritmo que menos CPU utiliza.

Se consideraron además cuatro conjuntos de 100, 200, 500 y 1000 puntos, generados aleatoriamente y distribuidos uniformemente sobre el cuadrado  $[0,100] \times [0,100]$  y con pesos unitarios y se confirmaron las observaciones anteriores.

Las mismas pruebas se realizaron para  $\ell_1$  y  $\ell_q$ , con  $q$  variable, para los conjuntos P-18, Z-30 y Z-50. Se aprecia la mayor velocidad de GMFP respecto del algoritmo MFP, si bien no siempre tiene mayor precisión. Ahora bien, el algoritmo  $\text{Gate}(1,q)$  supera en precisión a ambos pero, en pendientes bajas, el algoritmo MFP consigue mayor velocidad. Además el algoritmo  $\text{Gate}(1,q)$ , con igual precisión que los algoritmos Garsq y Gdaq, aumenta su velocidad cuando aumenta la pendiente, mientras que los otros dos disminuyen la suya.

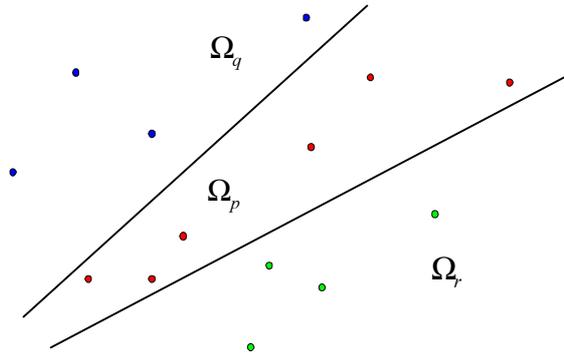
Se tomaron 1000 puntos generados aleatoriamente y uniformemente distribuidos en el cuadrado  $[0,100] \times [0,100]$  y con pesos unitarios. Se tomaron 10, 10, 10, 10, 10, 5, 2 y 1 subconjuntos de 10, 20, 40, 50, 100, 200, 500 y 1000 puntos, respectivamente. El algoritmo GMFP es más rápido y preciso que el algoritmo MFP para pendientes no superiores a la unidad. A medida que aumenta la pendiente se mantiene la ganancia en tiempo de GMFP respecto de MFP pero pierde en precisión. No obstante, los algoritmos  $\text{Gate}(1,q)$ , Garsq y Gdaq ganan siempre en precisión y velocidad, destacando la velocidad del algoritmo  $\text{Gate}(1,q)$  en pendientes bajas, del algoritmo Garsq en pendientes cercanas a 1 y del algoritmo Gdaq en pendientes altas.

Análogamente se han realizado estudios con  $\ell_1$  y  $\ell_\infty$  y con  $\ell_p$  y  $\ell_\infty$ .

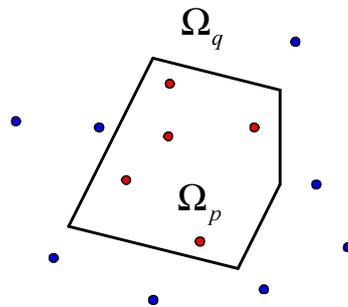
## 6.2 Trabajos futuros

Como trabajo futuro podemos pensar en diferentes situaciones más reales que puedan darse en el plano y que nos permita generalizar nuestro trabajo. Así tenemos:

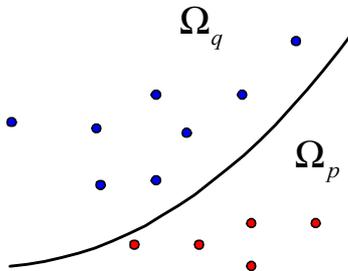
- 1) Desarrollo del problema con más de dos regiones usando puertas de acceso.



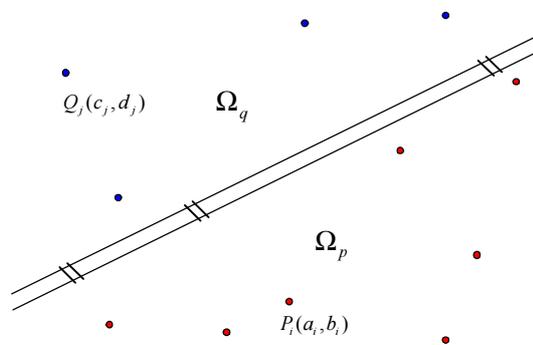
- 2) Estudio del problema para dos regiones, una de las cuales esté encerrada en un polígono convexo, usando puertas de acceso.



- 3) Desarrollo del problema con una línea convexa, usando puertas de acceso.



- 4) Desarrollo del problema con barrera en línea recta y pasarelas fijas.



---

## BIBLIOGRAFÍA

---

Apostol, T.M. (1960), *Análisis Matemático*. Ed. Reverté S.A.

Batta, R. y U.S. Palekar (1988). Mixed Planar/Network Facility Location Problems. *Computers and Operations Research*, 15(1): 61–67.

Bazaraa, M.S., H. Sherali, y C.M. Shetty (1993). *Nonlinear Programming*, second ed., John Wiley & Sons.

Beckenbach, E. y R. Bellman (1961). *Introduction to Inequalities*, tenth ed., Mathematical Association of America.

Brimberg, J. y R.F. Love (1992). Local convergence in a generalized Fermat-Weber problem. *Annals of Operations Research*, 40: 33-66.

Brimberg, J. y R.F. Love (1993). Global convergence of a generalized iterative procedure for the minisum location problem with  $\ell_p$  distances. *Operations Research*, 41: 1153-1163.

Brimberg, J. (1995). The Fermat-Weber location problem revisited. *Math. Program.* 71: 71-76.

Brimberg, J., P.D. Dowling y R.F. Love (1996). Estimating the parameters of the weighted  $\ell_p$  norm by linear regression. *IEE Transactions*, 28: 363-367.

- Brimberg, J. y H. Juel (1998). A bicriteria model for locating a semi-desirable facility in the plane. *European Journal of Operational Research*, 106: 144-151.
- Brimberg, J. y R. Chen (1998). A note on convergence in the Single Facility Minisum Location Problem. *Comput. Math. Appl.* 35: 25-31.
- Brimberg, J., R. Chen y D. Chen (1998a). Accelerating convergence in the Fermat-Weber location problem. *Operations Research Letters*, 22: 151-157.
- Brimberg, J., H. Kakhki y G.O. Wesolowsky (2003). Location among regions with varying norms. *Annals of Operations Researchs*, 122: 873–102.
- Brimberg, J., H. Kakhki y G.O. Wesolowsky (2005). Locating a single facility in the plane in the presence of bounded regions and different norms. *Journal of the Operational Research Society of Japan*, 48(2): 135-147.
- Brimberg, J., J.H. Walker y R.F. Love (2007). Estimation of travel distances with the weighted  $\ell_p$  – norm: Some empirical results. *Journal of Transport Geography*, 15: 62–72.
- Cánovas, L., R. Cañavate y A. Marín (2002). On the convergente of the Weiszfeld algorithm. *Math. Program.*, Ser. A 93: 327-330.
- Carrizosa, E. y A.M. Rodríguez-Chía (1997). Weber problems with alternative transportation systems. *European Journal of Operational Research*, 97(1): 87–93.
- Chandrasekaran, R. y A. Tamir (1989). Open questions concerning Weiszfeld’s algorithm for the Fermat-Weber location problem. *Mathematical Programming*, 44: 293-295.
- Cooper, L. (1963). Location-allocation problems. *Operations Research*, 11:37-52.
- Dax, A. (1986). A Note on Optimality Conditions for the Euclidean Multifacility Location Problem. *Mathematical Programming*, 36: 72-80.
- Dax, A. (2006). The distance between two convex sets. *Linear Algebra and its Applications*, 416: 184–213.

- Dörrie, H. (1965). *100 Great Problems of Elementary Mathematics, their History and Solution*. Dover. (traducido por David Antin).
- Drezner, Z. (1996). A Note on Accelerating the Weiszfeld Procedure. *Location Science*, 3: 275-279.
- Drezner, Z. y H.W. Hamacher (2002). *Facility Location. Applications and Theory*. Springer-Verlag - Berlín - Heidelberg.
- Drezner, Z., K. Klamroth, A. Schöbel y G.O. Wesolowsky (2002). The Weber Problem, in *Facility Location. Applications and Theory*. Springer-Verlag - Berlin - Heidelberg.
- Drezner, Z. y A. Suzuki (2004). The Big Triangle Small Triangle Method for the solution of Nonconvex Facility Location Problems. *Operations Research*, 52: 128-135.
- Fernández, J., P. Fernández y B. Pelegrin (2002). Estimating actual distances by norm functions: a comparison between the  $\ell_{k,p,\theta}$ -norm and the  $\ell_{b_1,b_2,\theta}$ -norm and a study about the selection of the data set. *Computers and Operations Research*, 29: 609-623.
- Francis, R.L., F. Leon, L.F. McGinnis, y J.A. White (1992). *Facility Layout and Location. An analytical Approach*, second ed. Prentice Hall, New York.
- Franco, L., F. Velasco y L. González-Abril (2012). Gate points in continuous location between regions with different  $\ell_p$  norms. *European Journal of Operational Research*, 218: 648-655.
- González-Abril L., F. Velasco y L. Franco (2011). Support vector machines for classification of input vectors with different metrics. *Computers and Mathematics with Applications*, 61: 2874-2878.
- Gottfried, B.S. y J. Weisman (1973). *Introduction to Optimization Theory*. Englewood Cliffs, NJ: Prentice-Hall.
- Guerrero, C. (2007). Un modelo multicriterio de localización de centros no deseados con pesos. Primera Edición. Consejo Económico y Social de Andalucía.

- Guibas, L. y J. Herschberger (1989). Optimal shortest path queries in a sample polygon. *Journal of Computers and System Sciences*, 39: 126-152.
- Hansen, P., D. Peeters, D. Richard y J. Thisse (1981). On the location of an obnoxious facility. *Sistemi Urbani*, 3: 229-317.
- Hansen, P., D. Peeters, D. Richard y J. Thisse (1985). The Minisum and Minimax Location Problems Revisited. *Operations Research*, 33 (6): 1251-1265.
- Herschberger, J. y S. Suri (1999). An optimal algorithm for euclidean shortest paths in the plane. *SIAM Journal on Computing*, 28: 2215-2256.
- Honsberger, R. (1973). *Mathematical Gems from Elementary Combinatorics, Number Theory, and Geometry I*. Mathematical Association of America.
- Juel, H. y R.F. Love (1976). An efficient computational procedure criteria for solving multi-facility rectilinear facilities location problems. *Operational Research Quarterly*, 27: 697-703.
- Juel, H. y R.F. Love (1981). Fixed point optimality criteria for the location problem with arbitrary norms. *Journal of the Operational Research Society*, 32: 891-897.
- Juel, H. y R.F. Love (1986). A Geometrical Interpretation of the Existing Facility Solution Condition for the Weber Problem. *Journal of the Operational Research Society*, 37: 1129-1131.
- Katz, I.N. (1974). Local convergence in Fermat's Problem. *Math. Programming*, 6: 89-104.
- Kuhn, H.W. y R.E Kuenne (1962). An efficient algorithm for the numerical solutions of the generalized Weber problem in spatial economics. *J. Region. Sci.*, 4: 21-34.
- Kuhn, H.W. (1967). On a pair of dual nonlinear programs, en *Methods of non linear programming*, Ed. J. Abadie (North-Holland, Amsterdam) 38-54.
- Kuhn, H.W. (1973). A note on Fermat's problem. *Math. Program.*, 4: 98-107.

- Kuhn, H.W. (1976). Nonlinear Programming: A Historical View, *American Mathematical Society*, 9: 1-26.
- Love, R.F., J.G. Morris y G.O. Wesolowsky (1988). *Facilities Location: Models and Methods*. Noth-Holland, New York.
- Love, R.F. y J.H. Walker (1994). An empirical Comparison of Block and Round Norms for Modelling Actual Distances. *Location Science*, 2(1): 21-43.
- Mangasarian, O.L. (1999). Arbitrary-norm separating plane. *Operations Research Letters*, 24: 15–23.
- Melzak (1983). *Invitation to Geometry*. Wiley.
- Miehle, W. (1958). Link-Length Minimization in Networks. *Operations Research*, 6: 232-243.
- Michelot, C. y O. Lefebvre (1987). A Primal-Dual Algorithm for the Fermat-Weber Problem Involving Mixed Gauges. *Mathematical Programming*, 39(3): 319-335.
- Mitchell, J.S.B. y C.H. Papadimitriou (1991). The Weighted Region Problem: Finding Shortest Paths Through a Weighted Planar Subdivision. *Journal of the Association for Computing Machinery*, 38(1): 18-73.
- Morris, J.G. (1978). Analisis of Generalized Empirical “Distance” Function for Use in Location Problems. International Symposium on Locational Decisions. Banff, Alberta.
- Morris, J.G. y W.A. Verdini (1979). Minisum  $\ell_p$  distance location problems solved via a perturbed problem and Weiszfeld’s algorithm. *Operations Research*, 27: 1180-1188.
- Morris, J.G. (1981). Convergence of the Weiszfeld Algorithm for Weber Problems Using a Generalized “Distance” Function. *Operations Research*, 29: 37-48.
- Ortega, F.A., J.A. Mesa y A.B. Sánchez (2000). An Iterative Method for Solving the Weber Problem in  $\mathbb{R}^2$  with  $\ell_p$  Norms,  $p \in (1, 2)$ , Based in Linear Programing. *Studies in Locational Analysis*, 14: 137-152.

## BIBLIOGRAFÍA

---

- Ostresh, L.M. (1977). The Multifacility Location Problem: Applications and Descent Theorems. *Journal of Regional Science*, 17: 409-419.
- Ostresh, L.M. (1978). On the Convergence of a Class of Iterative Methods for Solving the Weber Location Problem. *Operations Research*, 26: 597-609.
- Ostresh, L.M. (1978a). Convergence and Descent in the Fermat Location Problem. *Transportation Science*, 12: 153-164.
- Parlar, M. (1994). Single facility location problem with region-dependent distance metrics. *International Journal of System Sciences*, 25(3): 513–525.
- Planchart, A. y P.H. Hurter (1975). An Efficient Algorithm for the Solution of the Weber Problem with Mixed Norms. *Siam Journal On Control*, 13(3): 650–665.
- Pick, G. (1929). Mathematical Appendix, en *Theory of the Location of Industries*, A. Weber (ed.). Versión inglesa de C.J. Friedrich, 227-252. Russell & Russell, New York.
- Plastria, F. (1992). On destination optimality in asymmetric distance Fermat-Weber problems. *Annals of Operations Research*, 40: 355-369.
- Plastria, F. (1992a). GBSSS, the generalized Big Square Small Square Method for planar single facility location. *European Journal of Operational Research*. 123: 173-188.
- Plastria, F. y E. Carrizosa (1999). Undesirable facility location with minimal covering objectives. *European Journal of Operational Research*, 119: 158-180.
- Plastria, F. y E. Carrizosa (2001). Gauge Distances and Median Hiperplanes. *Journal of Optimization Theory and Applications*, 110(1): 173-182.
- Plastria, F. (2002). Continuous Covering Locations Problems, en *Facility Location. Applications and Theory*. Springer-Verlag, Berlín - Heidelberg.
- Rochat M., Vecten, Fauquier y Pillate (1811). Questions Resolues: Solutions des deux problemes proposes a la page 384 du premier volume des annals. *Ann. Math. Pures et Appl.*, 2: 88-96.

RocKafellar, R.T. (1970). *Convex Analysis*. Princeton University Press, Princeton. New York.

Rodríguez-Chía, A.M. y C. Valero-Franco (2011). On the global convergence of a generalized iterative procedure for the minisum location problema with  $\ell_p$  distances for  $p > 2$ . *Math. Program., Ser. A*. DOI 10.1007/s10107-011-0501-z. (26 págs.)

Sinha, S.M. (2006), *Mathematical Programming*. Elsevier Inc.

Spiekerman, K. y Wegener, M. (1994). The shrinking continent. New time space maps of Europe. *Environment and Planning: Planning and Design*, 21: 653-673.

Thisse J-F. y H.G. Zoller (1983). *Locational Analysis of Public Facilities*. North-Holland.

Üster, H. y R.F. Love (2000). The Convergence of the Weiszfeld Algorithm. *Computers and Mathematics with Applications*, 40: 443-451.

Velasco, F. (1991). Localización de centros atractivos y repulsivos. Tesis doctoral. Universidad de Sevilla.

Weber, A. (1909). *Über den Standort der Industrien*. Tübingen, Germany. (Versión inglesa de C.J. Friedrich, 1929. *Theory of the Location of Industries*. University of Chicago Press).

Weiszfeld, E.V. (1937). Sur le point pour lequel la somme des distances de n points donnés est minisum. *Tohoku Math. J.*, 43: 335-386.

Wesolowsky, G.O. y R.F. Love (1971). The Optimal Location of New Facilities Using Rectangular Distances. *Operations Research*, 19: 124-130.

Wesolowsky, G.O. (1993). The Weber problem: History and Perspectives. *Location Science*, 1: 15-23.

White, J.A. (1971). A Quadratic Facility Location Problem. *American Institute of Industrial Engineers Transactions*, 3: 156-157.

## BIBLIOGRAFÍA

---

Zaferanieh, M., H. Kakhki, J. Brimberg, y G.O. Wesolowsky, (2008). A BSSS algorithm for the single facility location problem in two regions with different norms. *European Journal of Operational Research*, 190: 79-89.

Zacharias, M. (1913). Elementargeometrie und elementare nicht-euklidische Geometrie in Synthetischer Behandlung. En *Encyclopadie der mathematischen Winssenschaften (Geometrie)*. Leipzig, Germany.

---

## ANEXOS

---



## Anexo I: Conjuntos de puntos

Los tres primeros conjuntos de puntos han sido utilizados por otros autores citados en el presente trabajo y los incluimos para contrastar los resultados obtenidos.

P-18: 18 puntos del ejemplo del Parlar (1994)

$x$	1	2	3	5	6	6	7	8	9	9	9	10	14	14	16	17	17	19
$y$	2	8	12	5	1	11	4	8	1	5	10	12	2	4	8	4	10	13

Z-30: 30 puntos del ejemplo de Zaferanieh et al. (2008)

$x$	$y$	$w$												
1	2	1	3	6	1	5	7	1	7	5	2	9	8	1
1	5	4	3	7	3	6	2	2	7	8	4	9	10	2
1	9	1	3	8	4	6	3	4	8	4	3	10	1	3
2	3	2	4	4	4	6	9	3	8	6	1	10	4	1
2	4	3	4	7	1	7	0	1	9	2	1	10	5	2
3	5	2	5	1	4	7	1	2	9	3	4	10	9	1

Z-50: 50 puntos del ejemplo de Zaferanieh et al. (2008)

$x$	$y$	$w$												
1	5	1	4	11	2	10	1	4	12	12	1	16	3	2
1	10	2	5	3	1	10	5	3	12	14	3	16	10	5
1	15	1	5	6	4	10	8	4	13	5	3	16	14	3
2	3	3	5	14	5	11	2	2	13	16	1	17	7	4
2	8	3	5	18	1	11	9	2	14	2	4	17	17	1
3	5	5	6	9	3	11	12	5	14	20	1	18	2	3
3	14	1	7	10	2	11	13	1	15	4	5	18	19	5
3	19	4	8	10	5	12	4	5	15	10	3	19	6	2
4	7	1	8	19	1	12	6	4	15	11	2	19	7	1
4	8	4	9	20	1	12	11	4	16	1	1	20	14	2

A-100: conjunto de 100 puntos con pesos unitarios

$x$	1	2	3	4	5	6	6	6	7	7	8	9	9	9	12	13	15	18	19	19
$y$	60	33	56	18	74	40	40	68	32	52	13	11	53	58	59	100	66	42	14	43
	22	22	24	24	24	26	26	26	27	27	29	29	32	32	33	34	34	34	35	36
	11	67	12	44	92	4	59	61	15	20	43	62	12	78	11	58	78	95	15	74
	37	37	38	39	41	42	42	42	42	42	43	43	44	45	45	46	47	50	54	54
	20	76	58	83	75	46	55	66	84	98	31	89	53	43	46	98	15	15	65	87
	55	59	60	61	64	65	65	65	67	67	67	68	68	68	70	70	70	70	72	72
	43	15	47	19	96	41	48	68	19	44	74	29	50	70	7	64	67	67	52	97
	76	77	77	77	77	77	78	79	79	82	82	86	86	88	91	92	95	97	98	99
	38	17	19	35	40	93	42	32	95	82	98	48	64	52	88	0	21	65	17	51

A-200: conjunto de 200 puntos con pesos unitarios

$x$	1	2	3	3	5	5	6	6	6	7	8	8	9	9	10	10	11	11	11	12
$y$	5	68	42	61	7	49	32	78	86	24	63	78	1	41	39	49	14	63	78	2
	12	12	12	13	13	13	13	13	14	14	15	16	17	18	18	18	18	18	18	18
	21	49	64	3	10	21	55	72	39	84	78	76	20	4	21	34	36	60	66	83
	19	19	20	21	21	21	21	21	22	22	23	23	23	25	26	27	28	29	29	30
	4	27	9	9	39	55	91	95	27	37	6	50	64	45	2	4	54	24	41	5
	30	30	31	31	32	32	32	32	33	33	34	34	34	34	35	35	35	36	37	38
	30	68	17	73	22	55	65	81	30	84	29	30	66	94	45	69	89	76	37	63
	38	40	40	40	40	41	41	41	42	42	43	44	44	45	46	46	47	47	48	49
	70	7	42	75	98	22	60	98	36	39	45	1	26	24	10	21	65	91	62	69
	49	50	50	51	51	51	52	53	53	53	53	53	54	54	54	55	56	56	57	58
	77	43	54	8	18	91	34	11	33	48	83	89	9	68	71	33	64	77	85	2
	58	58	59	59	60	60	60	60	61	62	62	62	63	64	65	65	66	66	66	66
	53	55	25	37	39	53	79	96	74	15	57	94	14	18	75	93	5	11	28	72
	67	67	68	69	69	69	70	70	70	71	71	72	73	73	73	73	74	74	74	75
	72	89	99	1	56	76	13	15	58	62	78	35	37	57	65	89	11	23	42	74
	76	76	76	77	80	80	81	81	81	82	84	85	85	86	86	87	87	87	88	89
	63	76	82	58	94	99	38	49	75	18	56	51	56	28	48	27	61	63	99	70
	89	90	90	91	91	92	92	92	93	93	94	95	95	96	97	98	99	99	100	100
	80	10	12	71	80	71	77	79	11	73	46	3	44	27	87	86	7	17	22	35

A-500: conjunto de 500 puntos con pesos unitarios

<i>x</i>	<i>y</i>														
0	32	14	60	30	48	43	54	55	94	67	66	80	5	88	78
1	62	14	79	30	94	43	69	56	0	67	84	80	11	88	91
2	8	15	2	31	7	43	88	56	16	68	14	80	23	89	20
2	49	15	14	31	58	44	5	56	18	68	23	80	23	89	23
2	61	15	40	31	69	44	9	56	27	68	71	80	42	89	24
2	80	15	59	32	13	44	83	57	6	68	90	80	50	89	25
2	86	15	62	32	77	44	90	57	12	68	96	80	92	89	25
2	96	16	0	32	83	44	94	57	17	69	26	80	99	89	33
3	5	16	1	33	10	45	21	57	57	69	41	81	7	89	36
3	21	16	33	33	15	45	25	57	70	69	43	81	14	90	13
3	41	16	72	33	23	45	33	57	96	69	72	81	40	90	19
3	45	16	84	33	43	45	38	58	29	69	91	81	56	90	50
3	75	17	9	33	46	45	47	58	37	70	38	81	60	90	52
3	90	17	19	33	60	45	84	58	42	70	52	81	90	90	58
3	95	17	94	33	80	45	84	58	61	70	63	82	0	90	59
4	33	18	19	33	84	46	64	58	64	70	80	82	10	90	73
4	51	18	36	33	98	46	96	58	70	71	24	82	19	90	100
4	95	18	63	34	19	47	50	58	94	71	31	82	62	91	1
4	97	18	74	34	29	47	95	58	95	71	68	82	64	91	39
5	2	18	93	34	74	48	26	59	17	71	83	82	72	92	14
5	21	18	97	34	85	48	40	59	20	71	87	83	13	92	50
5	23	19	19	35	9	48	91	59	33	72	35	83	16	92	64
5	30	19	43	35	42	49	22	59	43	72	48	83	18	92	80
5	30	19	44	35	45	49	65	59	67	73	23	83	29	93	5
5	47	20	20	35	62	50	40	59	87	73	36	83	55	93	7
6	27	20	45	35	78	50	49	60	6	73	37	83	61	93	19
6	45	20	61	35	95	50	65	60	11	73	44	83	62	93	30
6	73	20	67	36	53	50	76	60	15	74	7	83	63	93	39
7	39	20	72	36	62	50	81	60	60	74	51	83	68	93	40
7	41	21	68	37	21	50	100	60	86	74	52	83	68	93	41
7	44	22	0	37	31	51	27	60	88	74	69	83	85	93	48
7	59	22	52	37	94	51	59	61	1	75	11	83	99	93	74
7	89	23	9	38	3	51	86	61	76	75	13	84	35	93	81
8	16	23	35	38	8	52	9	61	82	75	26	84	43	94	1
8	51	23	37	38	52	52	12	61	82	75	47	84	62	94	22
8	66	24	15	38	54	52	30	61	86	75	54	84	84	94	26
8	97	24	24	38	57	52	33	61	91	75	57	84	86	94	82
9	26	24	58	39	50	52	37	61	99	76	39	84	90	95	7
9	32	24	65	39	59	52	99	62	7	76	42	85	3	95	27
9	46	24	70	39	78	52	99	62	93	76	42	85	41	95	54
9	59	25	32	40	18	53	3	63	23	76	58	85	56	95	72
10	15	25	34	40	30	53	5	63	54	76	60	85	91	96	9
10	35	25	43	40	39	53	25	63	80	76	83	85	92	96	10
11	4	26	9	40	52	53	28	64	22	76	88	86	4	96	12
11	31	26	89	40	55	53	69	64	34	76	89	86	6	96	54
11	46	26	93	40	70	53	72	64	90	76	96	86	10	96	57
11	52	27	52	40	71	53	72	65	7	76	98	86	34	96	62
11	52	27	56	41	4	53	83	65	12	76	100	86	40	96	68
11	59	28	45	41	11	53	90	65	23	76	100	86	40	96	76

A-500: conjunto de 500 puntos con pesos unitarios (continuación)

<i>x</i>	<i>y</i>														
11	98	28	59	41	28	54	21	65	38	77	48	86	70	97	8
12	6	28	62	41	31	54	28	65	49	78	20	86	91	97	20
12	27	28	75	41	39	54	69	65	52	78	27	86	97	97	62
12	47	28	88	41	71	54	70	65	73	78	37	87	5	97	82
12	81	28	90	41	88	54	76	65	92	78	50	87	8	97	85
12	88	29	40	42	12	54	96	66	12	78	61	87	33	97	97
13	1	29	50	42	34	54	98	66	17	78	71	88	1	98	13
13	2	29	75	42	36	55	22	66	52	78	91	88	4	98	28
13	12	29	80	42	38	55	27	67	21	79	24	88	44	98	40
13	44	29	80	43	1	55	56	67	44	79	32	88	47	99	40
14	26	30	1	43	14	55	68	67	44	79	33	88	47	99	56
14	27	30	15	43	17	55	73	67	50	79	72	88	58	99	86
14	39	30	40	43	26	55	81	67	59	79	78	88	72	100	55
14	48	30	40	43	27	55	87								

A-1000: conjunto de 1000 puntos con pesos unitarios

<i>x</i>	<i>y</i>														
0	1	12	88	24	82	37	62	51	62	64	13	76	92	87	94
0	15	12	89	24	84	37	92	51	68	64	21	76	96	88	6
0	28	12	93	24	87	38	16	51	72	64	38	77	8	88	39
0	38	12	98	24	94	38	27	51	91	64	57	77	17	88	40
0	64	13	4	25	6	38	32	52	5	64	59	77	23	88	42
0	71	13	5	25	19	38	48	52	40	64	66	77	28	88	44
0	82	13	13	25	20	38	50	52	54	64	71	77	39	88	46
0	86	13	16	25	25	38	56	52	65	64	79	77	57	88	61
0	95	13	17	25	39	38	57	52	77	64	88	77	65	88	66
0	98	13	19	25	58	38	73	52	82	65	3	77	82	88	67
0	99	13	25	25	65	38	81	52	90	65	11	77	85	88	71
1	11	13	30	25	66	38	86	52	91	65	22	77	97	88	73
1	19	13	32	25	81	38	88	52	94	65	38	77	100	89	2
1	23	13	36	26	9	38	89	52	99	65	51	78	7	89	6
1	32	13	65	26	20	39	3	53	23	65	53	78	25	89	15
1	38	13	67	26	23	39	36	53	25	65	66	78	38	89	21
1	60	13	70	26	55	39	43	53	26	65	68	78	40	89	26
1	61	13	71	26	65	39	47	53	33	65	76	78	50	89	47
1	64	13	76	26	76	39	74	53	35	65	80	78	53	89	49
1	75	13	87	26	79	39	78	53	37	65	82	78	62	89	66
1	80	14	17	26	81	40	11	53	44	65	83	78	73	90	6
2	5	14	44	26	88	40	48	53	48	65	88	78	78	90	9
2	7	14	77	26	92	40	53	53	52	66	1	78	86	90	11
2	39	14	86	26	99	40	58	53	57	66	5	78	95	90	14
2	81	14	88	27	4	40	70	53	63	66	39	79	15	90	22
2	88	14	90	27	26	40	88	53	80	66	55	79	20	90	32
2	96	15	0	27	40	40	99	53	87	66	56	79	29	90	35
3	1	15	14	27	55	41	12	53	93	66	60	79	33	90	48
3	14	15	15	27	70	41	22	53	96	66	89	79	35	90	55
3	43	15	16	27	76	41	24	54	10	66	94	79	37	90	68

## A-1000: conjunto de 1000 puntos con pesos unitarios (continuación)

<i>x</i>	<i>y</i>														
3	45	15	19	27	84	41	31	54	13	67	2	79	42	90	70
3	47	15	20	27	86	41	36	54	28	67	4	79	43	90	72
3	56	15	25	27	87	41	37	54	31	67	18	79	47	90	79
3	96	15	36	27	94	41	50	54	40	67	55	79	50	90	85
4	14	15	38	28	1	41	63	54	42	67	84	79	52	90	93
4	23	15	50	28	5	41	95	54	46	67	86	79	57	91	8
4	41	15	54	28	8	42	41	54	86	67	90	79	59	91	48
4	47	15	66	28	44	42	49	54	87	67	92	79	60	91	57
4	74	15	70	28	52	42	51	54	90	67	96	79	71	91	64
4	95	16	6	28	74	42	91	54	95	68	6	79	82	91	68
4	99	16	43	28	83	43	21	54	100	68	21	79	87	91	89
5	7	16	50	29	5	43	26	55	3	68	37	80	55	91	91
5	24	16	63	29	8	43	42	55	11	68	38	80	80	91	97
5	26	16	67	29	25	43	48	55	14	68	44	80	85	91	99
5	31	16	80	29	27	43	63	55	21	68	47	80	93	92	8
5	38	16	81	29	38	43	83	55	40	68	51	81	8	92	26
5	39	16	97	29	56	44	9	55	48	68	57	81	19	92	30
5	42	17	7	29	69	44	13	55	51	68	60	81	22	92	32
5	53	17	12	29	71	44	15	55	55	68	61	81	33	92	33
5	57	17	18	29	78	44	28	55	69	68	92	81	61	92	49
5	61	17	23	29	80	44	30	55	73	68	98	81	71	92	56
5	69	17	24	29	82	44	45	55	75	69	3	81	75	92	60
5	73	17	32	29	97	44	46	55	83	69	47	81	79	92	64
5	80	17	33	30	5	44	55	55	88	69	85	81	85	93	1
5	84	17	40	30	9	44	57	56	46	69	95	81	93	93	9
6	13	17	44	30	13	44	66	56	61	69	99	82	13	93	12
6	14	17	50	30	15	45	7	56	68	70	1	82	30	93	18
6	36	17	65	30	30	45	12	56	97	70	9	82	35	93	28
6	38	17	66	30	46	45	24	57	19	70	11	82	38	93	51
6	39	17	71	30	50	45	27	57	25	70	46	82	45	93	53
6	43	17	91	30	53	45	34	57	38	70	51	82	60	93	58
6	66	17	95	30	56	45	51	57	41	70	57	82	77	93	74
7	2	18	5	30	61	45	65	57	43	70	92	82	81	93	92
7	5	18	7	30	65	45	67	57	56	70	93	82	83	93	97
7	8	18	8	30	69	45	71	57	79	70	96	82	85	93	98
7	17	18	10	30	78	45	85	57	81	71	1	82	98	94	0
7	18	18	57	30	82	46	5	57	86	71	4	83	19	94	11
7	34	18	63	30	83	46	9	57	88	71	18	83	20	94	26
7	41	18	80	30	92	46	11	57	93	71	31	83	21	94	39
7	44	18	86	31	2	46	13	57	98	71	40	83	26	94	43
7	58	19	2	31	6	46	58	58	26	71	49	83	27	94	48
7	59	19	3	31	16	46	72	58	40	71	69	83	30	94	51
7	63	19	16	31	34	46	81	58	47	71	74	83	41	94	62
7	65	19	29	31	38	46	86	58	48	71	81	83	51	94	64
7	75	19	32	31	73	47	3	58	84	71	86	83	69	94	87
7	92	19	43	31	79	47	14	58	91	71	96	83	71	94	95
7	95	19	53	31	90	47	50	58	93	72	2	83	75	95	8
7	97	19	55	32	16	47	54	59	14	72	7	83	81	95	9
8	5	19	61	32	49	47	75	59	33	72	14	83	93	95	12
8	21	19	69	32	67	47	84	59	78	72	30	83	94	95	22

A-1000: conjunto de 1000 puntos con pesos unitarios (continuación)

<i>x</i>	<i>y</i>														
8	27	19	75	32	78	47	89	59	91	72	40	83	97	95	30
8	34	20	12	32	98	47	94	60	3	72	41	83	100	95	35
8	41	20	66	33	13	47	95	60	7	72	51	84	3	95	39
8	51	20	81	33	20	48	7	60	8	72	56	84	12	95	46
8	56	20	96	33	33	48	20	60	15	72	58	84	14	95	51
8	58	20	99	33	37	48	24	60	38	72	61	84	26	95	69
8	85	21	0	33	41	48	31	60	43	72	68	84	29	95	89
8	95	21	19	33	45	48	32	60	52	72	92	84	47	95	95
9	3	21	30	33	55	48	37	60	60	73	4	84	66	95	98
9	25	21	34	33	66	48	59	60	64	73	10	84	67	96	9
9	53	21	44	33	75	48	73	60	81	73	12	84	68	96	43
9	63	21	54	33	88	48	88	60	94	73	16	84	74	96	44
9	73	21	55	34	20	48	89	61	25	73	18	84	86	96	53
9	80	21	65	34	46	49	7	61	34	73	20	85	19	96	60
9	88	21	67	34	49	49	26	61	58	73	34	85	28	96	77
10	11	21	76	34	55	49	27	61	71	73	50	85	34	96	85
10	38	21	77	34	67	49	35	61	92	73	53	85	37	97	11
10	46	22	0	34	72	49	38	61	99	73	64	85	40	97	18
10	47	22	3	34	77	49	40	62	3	73	72	85	58	97	22
10	48	22	8	34	78	49	44	62	6	73	75	85	88	97	30
10	51	22	10	34	82	49	49	62	12	73	86	85	91	97	31
10	54	22	21	34	92	49	54	62	17	73	97	85	95	97	43
10	67	22	87	35	5	49	58	62	30	74	39	85	100	97	63
10	70	22	88	35	27	49	86	62	38	74	44	86	5	97	73
10	91	23	1	35	32	49	97	62	45	74	46	86	17	97	84
10	95	23	2	35	33	50	15	62	72	74	84	86	20	97	90
10	96	23	5	35	34	50	17	62	73	74	98	86	21	98	6
10	100	23	18	35	35	50	24	63	6	75	14	86	28	98	11
11	1	23	36	35	49	50	31	63	10	75	41	86	38	98	25
11	16	23	40	35	56	50	33	63	12	75	42	86	45	98	52
11	18	23	46	35	96	50	43	63	28	75	48	86	46	98	59
11	22	23	53	36	3	50	49	63	32	75	50	86	68	98	63
11	24	23	70	36	13	50	61	63	35	75	52	86	76	98	71
11	28	24	3	36	17	50	65	63	36	75	83	86	90	98	89
11	35	24	5	36	41	50	74	63	39	75	84	86	91	98	90
11	38	24	7	36	48	50	77	63	57	75	87	86	92	98	98
11	91	24	24	36	70	50	78	63	58	75	89	86	94	99	3
12	4	24	43	36	72	50	91	63	69	75	98	87	7	99	10
12	8	24	52	36	74	51	3	63	71	75	100	87	8	99	16
12	38	24	57	36	75	51	15	63	84	76	3	87	12	99	33
12	51	24	60	36	99	51	22	63	86	76	29	87	17	99	43
12	57	24	62	37	1	51	38	63	91	76	80	87	37	99	46
12	58	24	68	37	19	51	39	63	98	76	81	87	41	99	49
12	65	24	75	37	24	51	55	63	99	76	84	87	60	99	60
12	74	24	76	37	41	51	61	64	7	76	89	87	75	99	63

A-1000\_2: conjunto de 1000 puntos con pesos no unitarios

$x$	$y$	$w$															
0	32	7	18	26	2	35	78	5	52	34	2	68	23	5	83	55	10
0	35	10	18	34	3	35	82	1	52	37	10	68	29	7	83	59	6
0	38	10	18	36	1	35	89	5	52	49	5	68	38	7	83	61	6
1	5	7	18	42	1	35	94	9	52	69	3	68	50	2	83	62	6
1	40	7	18	60	3	35	95	6	52	99	9	68	61	3	83	63	6
1	60	4	18	63	7	36	53	8	53	3	9	68	66	2	83	67	3
1	62	3	18	66	4	36	62	9	53	5	8	68	70	5	83	68	3
2	8	10	18	74	7	36	74	4	53	11	9	68	71	7	83	81	7
2	33	5	18	83	8	36	76	5	53	25	10	68	76	8	83	85	4
2	49	3	18	93	1	36	99	1	53	28	10	68	90	10	83	99	6
2	61	2	18	97	5	37	20	5	53	33	6	68	96	5	84	35	5
2	68	8	19	4	7	37	21	5	53	48	9	68	99	8	84	43	5
2	80	2	19	14	7	37	31	2	53	69	3	69	1	9	84	56	9
2	85	1	19	19	4	37	37	7	53	72	2	69	26	1	84	62	4
2	86	8	19	27	8	37	62	10	53	75	7	69	32	10	84	84	1
2	96	10	19	43	5	37	63	8	53	80	8	69	41	1	84	86	10
3	5	9	19	44	4	37	76	7	53	83	6	69	43	9	84	90	10
3	21	5	19	53	4	37	81	3	53	85	9	69	47	3	85	3	9
3	41	8	19	69	2	37	92	6	53	89	9	69	56	4	85	28	7
3	42	2	19	80	8	37	94	1	53	90	6	69	72	7	85	41	5
3	44	4	20	3	8	37	99	1	54	9	9	69	75	5	85	51	7
3	45	7	20	9	4	38	3	5	54	21	3	69	76	5	85	56	10
3	56	9	20	20	4	38	8	8	54	28	10	69	91	7	85	60	10
3	61	4	20	45	5	38	52	3	54	65	6	70	7	3	85	91	7
3	75	8	20	61	6	38	54	10	54	68	1	70	13	2	85	92	10
3	90	9	20	67	5	38	57	9	54	69	9	70	15	10	86	4	7
3	93	8	20	72	8	38	58	3	54	70	10	70	22	2	86	6	9
3	95	5	21	9	8	38	63	1	54	71	10	70	38	8	86	10	10
4	14	1	21	19	9	38	70	8	54	76	4	70	52	1	86	28	8
4	17	7	21	30	5	38	78	2	54	87	3	70	57	7	86	34	2
4	18	8	21	39	6	38	81	6	54	96	2	70	58	9	86	40	4
4	30	1	21	55	7	38	86	5	54	98	8	70	63	5	86	44	9
4	33	10	21	68	6	39	50	7	54	100	6	70	64	1	86	48	4
4	51	5	21	91	7	39	59	5	55	4	9	70	67	6	86	50	7
4	85	10	21	95	1	39	66	2	55	14	2	70	80	5	86	64	4
4	95	8	22	0	2	39	70	1	55	22	8	70	89	10	86	70	8
4	97	2	22	11	2	39	74	9	55	27	3	71	3	3	86	91	7
5	2	7	22	12	3	39	78	8	55	30	8	71	24	8	86	97	6
5	7	1	22	27	7	39	83	9	55	33	7	71	31	3	87	5	3
5	10	9	22	37	1	40	7	7	55	43	7	71	50	5	87	7	10
5	21	4	22	52	5	40	8	3	55	56	4	71	51	7	87	8	5
5	23	9	22	67	9	40	18	9	55	62	6	71	62	6	87	12	9
5	30	6	23	6	8	40	30	4	55	68	4	71	68	10	87	27	3
5	47	4	23	9	1	40	39	4	55	73	6	71	74	9	87	33	7
5	49	2	23	17	3	40	42	2	55	81	8	71	75	3	87	61	10
5	53	2	23	35	3	40	52	7	55	83	9	71	78	3	87	63	4
5	69	7	23	37	9	40	53	9	55	87	1	71	83	1	88	1	3
5	74	3	23	50	10	40	55	1	55	94	4	71	87	10	88	4	7
5	80	7	23	53	3	40	70	3	56	0	8	72	25	6	88	29	5
6	27	10	23	64	5	40	71	6	56	16	6	72	35	7	88	44	8

A-1000\_2: conjunto de 1000 puntos con pesos no unitarios (continuación)

<i>x</i>	<i>y</i>	<i>w</i>															
6	32	8	23	84	2	40	75	9	56	18	5	72	48	7	88	47	5
6	40	6	24	12	7	40	98	5	56	27	8	72	52	10	88	52	10
6	41	5	24	15	10	41	4	8	56	64	5	72	97	6	88	58	1
6	45	6	24	24	2	41	11	5	56	77	4	73	23	6	88	66	9
6	68	1	24	40	1	41	12	5	57	6	5	73	36	5	88	71	4
6	73	6	24	44	7	41	22	2	57	12	7	73	37	9	88	72	1
6	78	4	24	57	10	41	28	8	57	17	2	73	44	4	88	78	5
6	86	10	24	58	9	41	31	3	57	25	4	73	53	9	88	91	6
7	8	1	24	60	5	41	39	7	57	47	1	73	57	2	88	99	1
7	24	1	24	65	9	41	50	2	57	57	3	73	65	5	89	11	3
7	32	6	24	68	9	41	60	8	57	70	8	73	76	9	89	20	4
7	34	1	24	70	4	41	63	6	57	85	8	73	89	10	89	23	9
7	39	1	24	89	1	41	71	2	57	88	4	73	91	6	89	24	2
7	41	7	24	92	3	41	75	9	57	93	8	74	7	9	89	25	4
7	44	2	25	25	8	41	88	6	57	96	8	74	11	7	89	33	7
7	52	1	25	32	5	41	98	10	58	2	2	74	23	8	89	36	6
7	59	10	25	34	4	42	5	10	58	6	3	74	42	5	89	37	1
7	89	1	25	43	8	42	12	5	58	29	9	74	46	7	89	70	6
8	5	6	25	45	3	42	34	6	58	37	3	74	51	2	89	80	8
8	13	2	25	81	1	42	36	6	58	40	6	74	52	9	89	91	8
8	16	4	26	2	10	42	38	3	58	42	1	74	69	9	90	10	1
8	23	10	26	4	8	42	39	9	58	53	3	75	11	2	90	12	10
8	40	3	26	9	9	42	46	8	58	55	6	75	13	9	90	13	9
8	51	4	26	41	6	42	49	2	58	61	3	75	20	7	90	19	8
8	63	7	26	59	1	42	55	10	58	64	7	75	26	6	90	22	1
8	66	6	26	60	8	42	66	7	58	70	1	75	38	6	90	35	2
8	78	10	26	61	6	42	82	5	58	94	1	75	42	9	90	50	7
8	85	2	26	84	3	42	84	9	58	95	1	75	47	9	90	52	6
8	93	8	26	88	9	42	98	4	59	14	1	75	54	4	90	58	6
8	97	1	26	89	5	43	1	7	59	15	2	75	57	3	90	59	6
9	1	5	26	93	5	43	14	8	59	17	8	75	74	6	90	73	5
9	11	2	27	4	7	43	17	8	59	20	7	75	98	4	90	93	9
9	26	9	27	15	3	43	18	10	59	25	7	76	10	10	90	100	7
9	27	1	27	20	9	43	26	3	59	33	9	76	38	3	91	1	2
9	32	6	27	52	10	43	27	8	59	37	9	76	39	5	91	39	1
9	41	2	27	56	10	43	31	2	59	43	3	76	42	10	91	63	1
9	46	1	27	86	1	43	45	7	59	67	7	76	44	1	91	64	2
9	53	6	28	45	6	43	54	8	59	87	10	76	58	8	91	71	6
9	58	7	28	54	7	43	69	10	60	6	1	76	60	8	91	80	8
9	59	3	28	55	10	43	88	4	60	8	10	76	63	1	91	88	9
10	15	7	28	59	10	43	89	4	60	11	8	76	76	8	91	89	8
10	26	4	28	62	6	44	1	9	60	15	9	76	82	2	92	0	5
10	35	6	28	75	3	44	5	1	60	26	7	76	83	6	92	14	4
10	39	1	28	88	5	44	9	1	60	39	2	76	88	3	92	29	8
10	49	2	28	90	9	44	26	8	60	47	7	76	89	1	92	50	3
10	52	9	29	24	6	44	28	10	60	53	8	76	96	4	92	56	5
11	4	7	29	40	9	44	31	10	60	60	5	76	98	3	92	64	1
11	14	2	29	41	5	44	53	5	60	79	4	76	100	2	92	71	7
11	31	3	29	43	1	44	83	4	60	86	10	77	17	9	92	77	1
11	46	5	29	50	8	44	90	1	60	88	10	77	19	3	92	79	6

A-1000\_2: conjunto de 1000 puntos con pesos no unitarios (continuación)

<i>x</i>	<i>y</i>	<i>w</i>															
11	52	2	29	62	3	44	94	7	60	96	5	77	35	7	92	80	3
11	59	3	29	75	6	45	21	9	61	1	6	77	40	9	93	5	6
11	63	2	29	80	9	45	24	8	61	19	8	77	48	7	93	7	3
11	65	5	29	83	4	45	25	8	61	25	2	77	58	10	93	11	2
11	78	4	30	1	6	45	33	1	61	30	6	77	80	2	93	18	6
11	82	3	30	5	6	45	38	8	61	62	9	77	82	9	93	19	3
11	91	9	30	9	9	45	43	9	61	74	2	77	93	10	93	30	4
11	96	1	30	15	9	45	46	7	61	76	9	78	7	3	93	39	3
11	98	9	30	30	4	45	47	3	61	82	9	78	20	10	93	40	1
12	2	3	30	40	5	45	67	2	61	86	7	78	27	3	93	41	1
12	6	10	30	45	9	45	84	6	61	91	10	78	37	8	93	48	3
12	18	3	30	48	4	46	9	6	61	99	3	78	42	1	93	73	8
12	21	2	30	68	6	46	10	10	62	7	1	78	50	5	93	74	6
12	27	3	30	94	10	46	21	1	62	15	4	78	61	10	93	81	5
12	47	7	31	7	2	46	64	10	62	35	6	78	71	2	94	1	7
12	49	9	31	17	7	46	72	4	62	47	4	78	72	10	94	22	5
12	50	10	31	51	6	46	90	8	62	57	1	78	91	8	94	26	6
12	59	3	31	53	2	46	96	6	62	68	4	78	93	2	94	39	2
12	64	4	31	58	9	46	98	2	62	93	9	79	20	7	94	46	3
12	81	4	31	69	1	47	14	1	62	94	4	79	24	5	94	49	5
12	88	1	31	73	8	47	15	4	63	14	3	79	32	6	94	82	8
13	1	8	31	98	6	47	30	4	63	23	6	79	33	3	95	3	4
13	2	6	32	12	10	47	50	5	63	54	5	79	72	3	95	7	3
13	3	10	32	13	7	47	54	1	63	80	7	79	78	7	95	9	10
13	10	2	32	22	3	47	65	1	63	91	7	79	89	5	95	21	8
13	12	2	32	42	6	47	84	5	64	18	1	79	95	4	95	27	3
13	17	1	32	55	10	47	91	2	64	22	9	79	96	7	95	44	1
13	21	9	32	65	10	47	95	1	64	34	3	80	5	3	95	54	6
13	44	1	32	77	3	48	26	4	64	90	5	80	11	9	95	72	5
13	55	5	32	78	5	48	40	6	64	96	3	80	23	10	95	98	6
13	72	2	32	81	8	48	59	2	65	7	3	80	31	7	95	99	1
13	87	7	32	83	9	48	62	3	65	12	6	80	40	2	96	9	6
13	94	10	32	84	4	48	91	6	65	23	5	80	42	8	96	10	1
13	100	2	33	10	8	49	17	10	65	38	5	80	43	10	96	12	5
14	2	10	33	11	7	49	22	3	65	41	9	80	50	7	96	16	10
14	26	1	33	15	4	49	44	5	65	48	7	80	92	2	96	27	10
14	27	2	33	23	10	49	45	7	65	49	10	80	94	7	96	49	9
14	39	10	33	30	1	49	58	3	65	52	4	80	99	1	96	54	1
14	42	10	33	43	5	49	65	9	65	53	8	81	7	10	96	57	9
14	44	4	33	46	8	49	69	10	65	68	7	81	14	9	96	62	7
14	48	4	33	60	10	49	77	4	65	73	1	81	24	10	96	68	5
14	60	10	33	80	10	49	86	3	65	75	6	81	38	7	96	76	1
14	79	1	33	84	8	50	15	1	65	92	6	81	40	10	97	8	6
14	84	2	33	98	6	50	23	5	65	93	2	81	49	9	97	20	2
14	87	6	34	16	8	50	40	2	66	5	4	81	56	3	97	22	8
15	2	5	34	19	4	50	43	10	66	11	10	81	58	2	97	62	7
15	14	7	34	29	8	50	48	10	66	12	2	81	60	8	97	65	9
15	40	4	34	30	5	50	49	9	66	17	3	81	75	2	97	82	10
15	59	1	34	42	2	50	54	5	66	28	3	81	90	5	97	85	6
15	62	3	34	58	2	50	65	4	66	52	3	81	91	2	97	87	1

A-1000\_2: conjunto de 1000 puntos con pesos no unitarios (continuación)

<i>x</i>	<i>y</i>	<i>w</i>															
15	66	6	34	59	3	50	75	8	66	72	6	82	0	1	97	97	7
15	78	2	34	66	3	50	76	8	66	89	2	82	10	9	98	13	3
15	83	6	34	72	1	50	81	4	67	2	2	82	18	2	98	17	2
16	0	3	34	74	8	50	100	4	67	19	4	82	19	2	98	28	2
16	1	8	34	78	7	51	8	8	67	21	7	82	62	8	98	40	5
16	33	4	34	85	3	51	18	4	67	34	5	82	64	1	98	44	2
16	72	6	34	90	4	51	27	2	67	44	9	82	72	10	99	7	10
16	76	9	34	94	2	51	55	3	67	50	3	82	79	7	99	17	3
16	84	2	34	95	10	51	59	8	67	59	7	82	82	8	99	46	5
17	9	3	35	9	2	51	61	4	67	63	5	82	98	8	99	51	9
17	19	1	35	15	6	51	86	4	67	66	2	83	13	1	99	56	10
17	20	4	35	20	3	51	91	7	67	72	7	83	16	5	99	86	4
17	36	2	35	42	2	52	9	10	67	74	6	83	18	2	100	8	5
17	94	6	35	45	1	52	12	2	67	84	10	83	20	6	100	22	7
18	4	2	35	62	8	52	23	5	67	89	6	83	29	4	100	55	6
18	19	7	35	69	3	52	30	8	68	4	9	83	51	4	100	86	1
18	21	1	35	70	10	52	33	9	68	14	8						

## Anexo II: Programas

En el cuadro siguiente figuran los programas que se han utilizado en el presente trabajo. A continuación se exponen los programas una única vez aunque figuren como subprogramas varias veces. Se han ordenado según figura en la cabecera del cuadro; en primer lugar los programas principales y a continuación los módulos y las rutinas.

**Cuadro de Programas**

<b>Programa principal</b>	<b>Módulos</b>	<b>Rutinas</b>	<b>Acción</b>
MFP		movimientos	Coloca $lq / lp$ y $m \leq 1$
	Mfa_Lq		Xq y dq
$lq / lp$	Mfa_Lp		Xp y dp
GMFP		movimientos	Coloca $lq / l1$ y $m \leq 1$
	MGP1q_q		Xq y dq
$lq / l1$	MGP1q_1		X1 y d1
MWP		movimientos	Coloca $lq / l1$ y $m \leq 1$
$l2 / l1$			Xs y ds
Gate(1,q)		movimientos	Coloca $lq / l1$ y $m \leq 1$
		rectars	xr1
		rectars1q	xr y dr
	f_lq	weiszfeldq	Halla $WPq(x,y) \rightarrow Xq$ y dq
	f_l1q	medianas	Halla medX y MedY
		malla	
		derqx	Derivadas laterales en X
		derqy	Derivadas laterales en Y
$lq / l1$		dist_seg_hq	X1 y d1
		dist_seg_vq	X1 y d1
		distancia1q	X1 y d1
Gdaq		movimientos	Coloca $lq / l1$ y $m \leq 1$
		rectars	xr1
		rectars1q	xr y dr
	f_lqars	arsweis	Xq y dq
	f_l1q	medianas	Halla medX y MedY
		malla	
		derqx	Derivadas laterales en X
		derqy	Derivadas laterales en Y
$lq / l1$		dist_seg_hq	X1 y d1
		dist_seg_vq	X1 y d1
		distancia1q	X1 y d1

### Cuadro de Programas (Continuación)

Programa principal	Módulos	Rutinas	Acción
Garsq	lq / l1	movimientos	Coloca lq / l1 y m<=1
		rectars1q	xr1
		actugatesq	Xq y dq
		arsdist1q	X1 y d1
Gate(p,inf)	linf / lp	movimientos	Coloca linf / l1 y m<=1
		rectarspinf	xr y dr
		weiszfeldq	WPinf(x,y) -----> Xinf y dinf
		weiszfeldq	Halla Wpp(x,y) --> Xp y dp
Gate(1,inf)	linf / l1	movimientos	Coloca linf / l1 y m<=1
		Punto_rectainf	xr1
		rectarspinf	xr y dr
		weiszfeldq	WPinf(x,y) -----> Xinf y dinf
		medianas	Halla medX y MedY
		malla	
		der1infx	Derivadas laterales en X
		der1infy	Derivadas laterales en Y
		dist_seg_hinf	X1 y d1
		dist_seg_vinf	X1 y d1
		distancia1inf	X1 y d1
Gate(p,q)	lq / lp	movimientos	Coloca lq / lp y m<=1
		rectarspq	xr y dr
		gates_pq	Obtiene los gates
		arsweis	Xq y dq
		gates_pq	Obtiene los gates
		arsweis	Xp y dp

Las entradas habituales de los programas son:

- m: pendiente de la recta.
- u: matriz con coordenadas y ponderaciones de los puntos.
- r: valor de  $p$  en  $l_p$ .
- s: valor de  $q$  en  $l_q$ .
- pos: posición relativa de  $l_p$  y  $l_q$  (1 si  $l_q / l_p$ , 0 en caso contrario).
- p: puntos de u en  $l_p$ .
- q: puntos de u en  $l_q$ .

## PROGRAMAS

### Programa MFP

```
function [sal]=MFP(m,u,r,s,pos)
N=length(u(1,:));mseg=m;useg=u;
[v,A,md,m]=movimientos(m,u,pos);u=v;
n1=0;p=[]; q=[];
for j=1:N % Ubicando y ordenando en lq y lp
    if u(2,j)>=m*u(1,j)
        u(4,j)=s;
        q=[q,u(1:4, j)];
    else
        u(4,j)=r;n1=n1+1;
        p=[p,u(1:4, j)];
    end
end
n2=N-n1;u=[p,q];t00=cputime;vuel=10; % Número de vueltas
for vueltas=1:vuel % Buscando el punto de lq
    [resq,zq,dq,itq]=Mfa_Lq(m,u,n1,n2,r,s);
end
tq=(cputime-t00)/vuel;t0=cputime;
for vueltas=1:vuel % Buscando el punto de lp
    [resp,zp,dp,itp]=Mfa_Lp(m,u,n1,n2,r,s);
end
tp=(cputime-t0)/vuel;tt=(cputime-t00)/vuel;
v=[resq(1) resp(1);resq(2) resp(2)]; % Deshacemos movimientos
sol=inv(A)*(v-md*ones(1,2));
resq=sol(:,1);resp=sol(:,2);
u=useg;m=mseg;
sal=[m s N resq(1) resq(2) dq itq tq resp(1) resp(2) dp itp tp tt];
```

### Programa GMFP

```
function [sal]=GMFP(m,u,s,pos)
N=length(u(1,:));mseg=m;useg=u;
[v,A,md,m]=movimientos(m,u,pos);u=v;
g=fsolve(@(g) 1-m*g^(s-1)+(m-1)*((1+g^s)^((s-1)/s)),0.1,++
optimset('Display','off','TolFun',1e-20));
g=-g;n1=0;p=[]; q=[];
for j=1:N % Ubicando y ordenando en l1 y lq
    if u(2,j)>=m*u(1,j)
        u(4,j)=2;
        q=[q,u(1:4, j)];
    else
        u(4,j)=1;n1=n1+1;
        p=[p,u(1:4, j)];
    end
end
n2=N-n1;u=[p,q];t00=cputime;vuel=10; % Número de vueltas
for vueltas=1:vuel % Buscando el punto de lq
    [res2,z2,d2,it2]=MGP1q_q(m,u,g,s,n1,n2);
end
t2=(cputime-t00)/vuel;t0=cputime;
for vueltas=1:vuel % Buscando el punto de l1
    [res1,z0,d1,it1]=MGP1q_l(m,u,g,s,n1,n2);
```

```

end
t1=(cputime-t0)/vuel;tt=(cputime-t00)/vuel;
v=[res2(1) res1(1);res2(2) res1(2)]; % Deshacemos movimientos
sol=inv(A)*(v-md*ones(1,2));
res2=sol(:,1);res1=sol(:,2);
u=useg;m=mseg;
sal=[m s N res2(1) res2(2) d2 it2 t2 res1(1) res1(2) d1 it1 t1 tt];

```

## Programa MWP

```

function [sal]=MWP(m,u,pos)
N=length(u(1,:));useg=u;mseg=m;
[v,A,md,m]=movimientos(m,u,pos);u=v;
g=-(sqrt(2*m)+m)/(1+sqrt(2*m));
n1=0;p=[]; q=[];
for j=1:N % Ubicando y ordenando en l1 y l2
    if u(2,j)>=m*u(1,j)
        u(4,j)=2;
        q=[q,u(1:4,j)];
    else
        u(4,j)=1;n1=n1+1;
        p=[p,u(1:4,j)];
    end
end
u=[p,q];n2=N-n1; t0=cputime;vuel=10; % Número de vueltas
uu=u;
for vueltas=1:vuel
    for j=n1+1:n2
        uu(1,j)=(u(2,j)-g*u(1,j))/(m-g);
        uu(2,j)=m*uu(1,j);
    end
    x0=u(1,:)*u(3,:)/sum(u(3,:));y0=u(2,:)*u(3,:)/sum(u(3,:));
    TOL=0.000001;T=1;D0=0.1;it=0;
    while T>=TOL
        if y0<m*x0
            r=1;
        else
            r=2;
            g1=(y0-g*x0)/(m-g);
            g2=m*g1;
        end
        numx=0;numy=0;denom=0;D=0;
        for h=1:n1
            if r==1
                d1=1/(abs(x0-u(1,h))+abs(y0-u(2,h)));
                numx=numx+u(1,h)*u(3,h)*d1;
                numy=numy+u(2,h)*u(3,h)*d1;
                denom=denom+u(3,h)*d1;
                D=D+u(3,h)/d1;
            else
                if u(2,h)>=g2 % está en S1h
                    d12=1/(sqrt((x0-u(2,h)/m)^2+(y0-u(2,h))^2)+
                        (u(1,h)-u(2,h)/m));
                elseif u(1,h)<=g1 % está en S1v
                    d12=1/(sqrt((x0-u(1,h))^2+(y0-m*u(1,h))^2)+
                        (m*u(1,h)-u(2,h)));
                else % está en SR

```

```

        d12=1/(sqrt((x0-g1)^2+(y0-g2)^2)+(u(1,h)-g1)+
            (g2-u(2,h)));
    end
    numx=numx+u(1,h)*u(3,h)*d12;
    numy=numy+u(2,h)*u(3,h)*d12;
    denom=denom+u(3,h)*d12;
    D=D+u(3,h)/d12;
end
end
for h=n1+1:N
    if r==1
        if uu(2,h)<=y0 % está en S2h
            d21=1/(sqrt((u(1,h)-y0/m)^2+(u(2,h)-y0)^2)+(x0-y0/m));
        elseif uu(1,h)>=x0 % está en S2v
            d21=1/(sqrt((u(1,h)-x0)^2+(u(2,h)-m*x0)^2)+(m*x0-y0));
        else % está en Sp
            d21=1/(sqrt((u(1,h)-uu(1,h))^2+(u(2,h)-uu(2,h))^2)+
                abs(x0-uu(1,h))+abs(y0-uu(2,h)));
        end
        numx=numx+u(1,h)*u(3,h)*d21;
        numy=numy+u(2,h)*u(3,h)*d21;
        denom=denom+u(3,h)*d21;
        D=D+u(3,h)/d21;
    else
        d2=1/sqrt((x0-u(1,h))^2+(y0-u(2,h))^2);
        numx=numx+u(1,h)*u(3,h)*d2;
        numy=numy+u(2,h)*u(3,h)*d2;
        denom=denom+u(3,h)*d2;
        D=D+u(3,h)/d2;
    end
end
x1=numx/denom;y1=numy/denom;
T=abs(D-D0);
D0=D;x0=x1;y0=y1;it=it+1;
end
end
tt=(cputime-t0)/vuel;
v=[x0;y0]; % Deshacemos movimientos
sol=inv(A)*(v-md*ones(1,1));
x0=sol(1);y0=sol(2);u=useg;m=mseg;
sall=-99*ones(1,8);
if r==2
    sal0=[m 2 x0 y0 D it];
    sal=[sal0,sall,-99,tt];
else
    sal0=[m 2];sal2=[x0 y0 D it -99 tt];
    sal=[sal0,sall,sal2] ;
end
end

```

### Programa Gate(1,q)

```

function [sal]=ffinalq(m,u,s,pos)
N=length(u(1,:));useg=u;mseg=m;
[v,A,md,m]=movimientos(m,u,pos);u=v;
g=fsolve(@(g) 1-m*g^(s-1)+(m-1)*((1+g^s)^((s-1)/s)),0.1,+,
    optimset('Display','off','TolFun',1e-20));
g=-g;n1=0;p=[]; q=[];

```

```

for j=1:N
    if u(2,j)>=m*u(1,j)
        u(4,j)=2;
        q=[q,u(1:4, j)];
    else
        u(4,j)=1;
        p=[p,u(1:4, j)];
    end
end
t00=cputime;vuel=10;
for vueltas=1:vuel
    [xr1,dr1]=rectars(m,p,q,1,s,10^(-6));
    [xr,dr]=rectars1q(m,p,q,g,s,10^(-6));
end
tr=(cputime-t00)/vuel;t0=cputime;
for vueltas=1:vuel
    [res2,d2,it2]=f_lq(m,g,p,q,xr,s);
end
t2=(cputime-t0)/vuel;t0=cputime;
for vueltas=1:vuel
    [x0,y0,d1,mensa]=f_l1q(m,g,p,q,xr1,s);
end
t1=(cputime-t0)/vuel; tt=(cputime-t00)/vuel;
v=[res2(1) xr x0;res2(2) m*xr y0];
sol=inv(A)*(v-md*ones(1,3));
res2=sol(:,1);xr=sol(1,2);x0=sol(1,3);y0=sol(2,3);
u=useg;m=mseg;
sal=[m s N res2(1) res2(2) d2 it2 t2 xr dr tr x0 y0 d1 mensa t1 tt];

```

## Programa Gdaq

```

function [sal]=gdaq(m,u,s,pos)
% ffinalq con arsweis en lugar de Weiszfeld
N=length(u(1,:));useg=u;mseg=m;
[v,A,md,m]=movimientos(m,u,pos);u=v;
g=fsolve(@(g) 1-m*g^(s-1)+(m-1)*((1+g^s)^((s-1)/s)),0.1,++
    optimset('Display','off','TolFun',1e-20));
g=-g;n1=0;p=[]; q=[];
for j=1:N
    if u(2,j)>=m*u(1,j)
        u(4,j)=2;
        q=[q,u(1:4, j)];
    else
        u(4,j)=1;
        p=[p,u(1:4, j)];
    end
end
ww1=sum(p(3,:));ww2=sum(q(3,:));
t00=cputime;vuel=10;
for vueltas=1:vuel
    [xr1,dr1]=rectars(m,p,q,1,s,10^(-6));
    [xr,dr]=rectars1q(m,p,q,g,s,10^(-6));
end
tr=(cputime-t00)/vuel;t0=cputime;
for vueltas=1:vuel
    [res2,d2,it2]=f_lqars(m,g,p,q,xr,s);
end

```

```

t2=(cputime-t0)/vuel;t0=cputime;
for vueltas=1:vuel % Buscando el punto de l1
    [x0,y0,d1,mensa]=f_llq(m,g,p,q,xr1,s);
end
t1=(cputime-t0)/vuel;tt=(cputime-t00)/vuel;
v=[res2(1) xr x0;res2(2) m*xr y0]; % Deshacemos movimientos
sol=inv(A)*(v-md*ones(1,3));
res2=sol(:,1);xr=sol(1,2);x0=sol(1,3);y0=sol(2,3);
u=useg;m=mseg;
sal=[m s N res2(1) res2(2) d2 it2 t2 xr dr tr x0 y0 d1 mensa t1 tt];

```

## Programa Garsq

```

function [sal]=garsq(m,u,s,pos)
N=length(u(1,:));mseg=m;useg=u;
[v,A,md,m]=movimientos(m,u,pos);u=v;
g=fsolve(@(g) 1-m*g^(s-1)+(m-1)*((1+g^s)^((s-1)/s)),0.1,++
    optimset('Display','off','TolFun',1e-20));
g=-g;n1=0;p=[];q=[];
for j=1:N % Ubicando y ordenando en l1 y lq
    if u(2,j)>=m*u(1,j)
        u(4,j)=2;
        q=[q,u(1:4, j)];
    else
        u(4,j)=1;n1=n1+1;
        p=[p,u(1:4, j)];
    end
end
u=[p,q];n2=N-n1;t00=cputime;vuel=10; % Número de vueltas
for vueltas=1:vuel % Buscando el punto de la
recta
    [xr,dr]=rectarslq(m,p,q,g,s,10^(-6));
end
tr=(cputime-t00)/vuel;t0=cputime;
it22=0;d22=10^9;d=[-1 -1;-1 1;1 1;1 -1]; % Buscando el punto de lq
for vueltas=1:vuel
    for i=1:5 %5 vueltas para minimizar d22
        res2(1)=q(1,:)*q(3,:)/sum(q(3,:));
        res2(2)=q(2,:)*q(3,:)/sum(q(3,:));
        w1=sum(p(3,:));w2=sum(q(3,:));w12=w1+w2;
        x0=(w2*res2(1)+w1*xr)/w12;y0=(w2*res2(2)+w1*m*xr)/w12;
        d0=actugatesq(p,q,m,g,s,x0,y0); % Evaluamos la distancia
        TOL=0.000001;T=1;it2=0;e=1;
        while T>=TOL
            v=rand(2,1);flag=0;
            ux=fix(10*v(1))/10*e;uy=fix(10*v(2))/10*e;
            ux=min(abs(ux),(y0/m-x0)/3);uy=min(abs(uy),(y0-m*x0)/3);
            for j=1:4
                xj=x0+d(j,1)*ux;yj=y0+d(j,2)*uy;
                dj=actugatesq(p,q,m,g,s,xj,yj); % Evaluamos la distancia
                if dj<d0
                    x0=xj;y0=yj;T=abs(dj-d0);d0=dj;
                    flag=1;
                    break
                end
            end
            if flag==0;e=e/2;end
        end
    end
end

```

```

        it2=it2+1;it22=it22+1;
        if d0<d22
            d22=d0;x22=x0;y22=y0;
        end
        if it2>=200;break;end
    end
end
end
t2=(cputime-t0)/vuel;t0=cputime;
for vueltas=1:vuel % Buscando el punto de l1
    d11=10^9;it11=0;d=[-1 1;-1 -1;1 1;1 -1];
    qq=q;
    for j=1:n2
        qq(1,j)=(q(2,j)-g*q(1,j))/(m-g);
        qq(2,j)=m*qq(1,j);
    end
    uu=[p(1:3,:),qq(1:3,:)];M=medianas(uu);
    mxp=min(M(1,:));myp=max(M(2,:)); % Encontrada la mediana
    if myp>=m*mxp
        x11=xr;y11=m*xr-10^(-6);
        d11=arsdistlq(p,q,m,s,qq,x11,y11);
        break
    end
    for i=1:5 % 5 vueltas para minimizar d11
        x0=(3*mxp+myp/m)/4;y0=(3*myp+m*mxp)/4;
        d1=arsdistlq(p,q,m,s,qq,x0,y0);d0=d1; % Evaluamos la distancia
        TOL=0.000001;T=1;it1=0;e=1;
        while T>=TOL
            v=rand(2,1); flag=0;
            ux=fix(10*v(1))/10*e; uy=fix(10*v(2))/10*e;
            ux=min(min(abs(ux),(x0-y0/m)/3),mxp-x0);
            uy=min(min(abs(uy),(m*x0-y0)/3),y0-myp);
            for j=1:4
                xj=x0+d(j,1)*ux;yj=y0+d(j,2)*uy;
                dj=arsdistlq(p,q,m,s,qq,xj,yj); % Evaluamos la distancia
                if dj<d0
                    x0=xj;y0=yj;T=abs(dj-d0);d0=dj;
                    flag=1;
                    break
                end
            end
            end
            if flag==0;e=e/2;end
            it1=it1+1;it11=it11+1;
            if d0<d11
                d11=d0;x11=x0;y11=y0;
            end
            if it1>=200;break;end
        end
    end
end
end
t1=(cputime-t0)/vuel;tt=(cputime-t00)/vuel;
v=[x22 xr x11;y22 m*xr y11]; % Deshacemos movimientos
sol=inv(A)*(v-md*ones(1,3));
u=useg;m=mseg;
x22=sol(1,1);y22=sol(2,1);xr=sol(1,2);x11=sol(1,3);y11=sol(2,3);
sal=[m s N x22 y22 d22 it22 t2 xr dr tr x11 y11 d11 it11 t1 tt];

```

## Programa Gate( $1, \infty$ )

```

function [sal]=pinfl(m,u,pos)
% Busca la solución para l-inf /l1, con 0<m<1
N=length(u(1,:));Useg=u;Mseg=m;
[v,A,md,m]=movimientos(m,u,pos);u=v;
n1=0;p=[];q=[];g=-1;
for j=1:N % Ubicando y ordenando puntos
    if u(2,j)>=m*u(1,j)
        u(4,j)=2;
        q=[q,u(1:4,j)];
    else
        u(4,j)=1;
        p=[p,u(1:4,j)];n1=n1+1;
    end
end
u=[p,q];useg=u;mseg=m;r=1;
% Hallando el punto de la recta
a=min(u(1,:));b=max(u(1,:));options=optimset('TolX',1e-06);
[xr1 dr1]=fminbnd(@(x) Punto_rectainf(x,m,u,N,n1),a,b,options);
[xr dr]=rectarspinf(m,p,q,r,10^(-6));
u(2,1:n1)=m*u(1,1:n1);
B=[1 -1;1 1];u(1:2,:)=B*u(1:2,:); % Transformando a l1
[res2,itw]=weiszfeldq(u,1,10^(-6)); % Algoritmo de Weiszfeld
res2=inv(B)*res2'; % Transformación inversa a l-
inf
u=useg;
g1=(res2(2)+res2(1))/(m+1);g2=m*g1;
for j=1:n1 % Actualizamos gates
    if u(2,j)>=g2
        u(1,j)=u(2,j)/m;
    elseif u(1,j)>=g1
        u(1,j)=g1;
    end
    u(2,j)=m*u(1,j);
end
C=u(1:2,:)-res2*ones(1,N); % Evaluamos la distancia
d2=max(abs(C))*u(3,:)'+(abs(p(1,1:n1)-u(1,1:n1))+
    abs(p(2,1:n1)-u(2,1:n1)))*u(3,1:n1)';
% Buscando ls solución en l-inf con ARS
dini=d2;d22=d2;res22=res2;resini=res2;
for i=1:5
    res0=resini;d0=dini;
    TOL=10^(-6);T=1;it2=0;d=[1 -1 1 -1;-1 -1 1 1];e=5;
    while T>=TOL
        v=rand(2,1);ux=e*v(1);uy=e*v(2);aviso=0;
        for k=1:4
            res2(1)=res0(1)+ux*d(1,k);res2(2)=res0(2)+uy*d(2,k);
            g1=(res2(2)+res2(1))/(m+1);g2=m*g1;
            u=useg;
            for j=1:n1
                if u(2,j)>=g2
                    u(1,j)=u(2,j)/m;
                elseif u(1,j)>=g1
                    u(1,j)=g1;
                end
                u(2,j)=m*u(1,j);
            end
        end
    end
end

```

```

C=u(1:2,:)-res2*ones(1,N); % Evaluamos la distancia
d2=max(abs(C))*u(3,:)'+((abs(p(1,1:n1)-u(1,1:n1)).^r+
abs(p(2,1:n1)-u(2,1:n1)).^r).^^(1/r))*u(3,1:n1)';
if d2<d0
res0=res2;aviso=1;
T=abs(d2-d0);d0=d2;
break
end
end
if aviso==0;e=e/1.2;end
it2=it2+1;
if it2>=200;break;end
end
if d0<d22;d22=d0;res22=res0;end
end
res2=res22;d2=d22;
n2=N-n1;qq=q; % Buscando la solución en l1
for j=1:n2
qq(1,j)=(q(2,j)-g*q(1,j))/(m-g);
qq(2,j)=m*qq(1,j);
end
uu=[p(1:3,:),qq(1:3,:)];
M=medianas(uu);
mxp=min(M(1,:));myp=max(M(2,:)); % Encontrada la mediana
if myp<m*mxp % Obteniendo la malla
[xval yval]=malla(n1,n2,m,p,qq,mxp,myp);
[x0,cx,ux]=der1linfx(n1,n2,m,p,q,qq,xval); % Calculando la derivadas
[y0,cy,uy]=der1linfy(n1,n2,m,p,q,qq,yval);
mensa=cx+2*cy;
switch mensa
case 0
d1=(abs(x0-p(1,:))+abs(y0-p(2,:)))*p(3,:);
for j=1:n2
if qq(2,j)<=y0 % Está en Sh
d1=d1+(x0-q(1,j))*q(3,j);
elseif qq(1,j)>=x0 % Está en Sv
d1=d1+(max(abs(q(1,j)-x0),abs(q(2,j)-m*x0))+
m*x0-y0)*q(3,j);
else
d1=d1+(max(abs(q(1,j)-qq(1,j)),abs(q(2,j)-qq(2,j))))+
abs(x0-qq(1,j))+abs(y0-qq(2,j))*q(3,j);
end
end
case 1
Dl=0;wh=0;wp=0;wv=0;Spp=[];Sv=[];
for j=1:n2
if qq(2,j)<=y0 % Está en Sh
Dl=Dl+max(abs(q(1,j)-y0/m),abs(q(2,j)-y0))*q(3,j);
wh=wh+q(3,j);
elseif qq(1,j)>=x0 % Está en Sv
Sv=[Sv,q(:,j)];wv=wv+q(3,j);
else
Dl=Dl+max(abs(q(1,j)-qq(1,j)),abs(q(2,j)-qq(2,j)))*q(3,j);
wp=wp+1;Spp=[Spp,qq(:,j)];
end
end
end
v=[p,Spp];inf=xval(ux);sup=xval(ux+1)
options=optimset('TolFun',1e-6);
[x1,d1]=fminbnd(@(x) dist_seg_hinf(x,m,v,n1,wp,Sv,wv,y0,wh),++

```

```

                                inf,sup,options);
    d1=d1+D1;x0=x1;
case 2
D1=0;wh=0;wp=0;wv=0;Spp=[];Sh=[];
for j=1:n2
    if qq(1,j)>=x0                                % Está en Sv
        D1=D1+max(abs(q(1,j)-x0),abs(q(2,j)-m*x0))*q(3,j);
        wv=wv+q(3,j);
    elseif qq(2,j)<=y0                            % Está en Sh
        Sh=[Sh,q(:,j)];wh=wh+q(3,j);
    else
        D1=D1+max(abs(q(1,j)-qq(1,j)),abs(q(2,j)-q(2,j)))*q(3,j);
        wp=wp+1;Spp=[Spp,qq(:,j)];
    end
end
v=[p,Spp];inf=yval(uy);sup=yval(uy+1);
options=optimset('TolFun',1e-6);
[y1,d1]=fminbnd(@(y) dist_seg_vinf(y,m,v,n1,wp,Sh,wh,x0,wv),++,
                inf,sup,options);
    d1=d1+D1;y0=y1;
otherwise
D1=0;wh=0;wp=0;wv=0;Spp=[];Sh=[];Sv=[];
x0=xval(ux);y0=yval(uy);
for j=1:n2
    if qq(1,j)>=x0                                % Está en Sv
        Sv=[Sv,q(:,j)];wv=wv+q(3,j);
    elseif qq(2,j)<=y0                            % Está en Sh
        Sh=[Sh,q(:,j)];wh=wh+q(3,j);
    else
        D1=D1+max(abs(q(1,j)-qq(1,j)),abs(q(2,j)-q(2,j)))*q(3,j);
        wp=wp+1;Spp=[Spp,qq(:,j)];
    end
end
v=[p,Spp];options=optimset('TolX',1e-06);
[X,d1]=fminsearch(@(a) distancia1inf(a,m,v,n1,wp,Sh,wh,Sv,wv),++,
                  [x0,y0],options);
    d1=d1+D1;x0=X(1);y0=X(2);
end
else
    x0=mxp;y0=mysp;
end
if y0>=m*x0
    mensa=99;x0=xr1;y0=m*x0-10^(-4);
    d1=(abs(x0-p(1,:))+abs(y0-p(2,:)))*p(3,:);
    for j=1:n2
        if qq(2,j)<=y0                            % Está en Sh
            d1=d1+(x0-q(1,j))*q(3,j);
        elseif qq(1,j)>=x0                        % Está en Sv
            d1=d1+(max(abs(q(1,j)-x0),abs(q(2,j)-m*x0))+m*x0-y0)*q(3,j);
        else
            d1=d1+(max(abs(q(1,j)-qq(1,j)),abs(q(2,j)-qq(2,j)))+
                abs(x0-qq(1,j))+abs(y0-qq(2,j)))*q(3,j);
        end
    end
end
v=[res2(1) xr x0;res2(2) m*xr y0];                % Deshacemos movimientos
sol=inv(A)*(v-md*ones(1,3));
res2=sol(:,1);xr=sol(1,2);x0=sol(1,3);y0=sol(2,3);
u=Useg;m=Mseg;

```

```
sal=[m s N res2(1) res2(2) d2 it2 t2 xr dr tr x0 y0 d1 mensa t1 tt];
```

### Programa Gate ( $p, \infty$ )

```
function [sal]=pinfq(m,u,r,pos)
N=length(u(1,:));Useg=u;Mseg=m; % Hay dos conjuntos de
seguridad
[v,A,md,m]=movimientos(m,u,pos);u=v;
if r==1
    g=10^9;
elseif m==1
    g=1;
else
    g=fsolve(@(g) 1+m*g^(r-1)-((1+g^r)^(r-1)/r),m*(1-m^r)^(-1/r)+0.1,++
        optimset('Display','off','TolFun',1e-20));
end
n1=0;p=[]; q=[]; % Ubicando y ordenando
for j=1:N
    if u(2,j)>=m*u(1,j)
        u(4,j)=2;
        q=[q,u(1:4,j)];
    else
        u(4,j)=1;n1=n1+1;
        p=[p,u(1:4,j)];
    end
end
u=[p,q];useg=u;mseg=m;
t00=cputime;vuel=1; % Buscando el punto de la
recta
a=min(u(1,:));b=max(u(1,:));options=optimset('TolX',1e-06);
[xr dr]=rectarspinf(m,p,q,r,10^(-6));
tr=(cputime-t00)/vuel;t0=cputime; % Buscando el punto de l-inf
w=u(3,:);u(2,1:n1)=m*u(1,1:n1);
B=[1 -1;1 1];u(1:2,:)=B*u(1:2,:); % Transformando a l1
[res2,itw]=weiszfeldq(u,1,10^(-6)); % Algoritmo de Weiszfeld
res2=inv(B)*res2';m=mseg;u=useg; % Transformación inversa a l-
inf
g1=(res2(2)+res2(1))/(m+1);g2=m*g1; % Actualizamos los Gates
if m<1
    h1=(res2(2)-res2(1))/(m-1); h2=m*h1;
else
    h1=-10^9;h2=h1;
end
for j=1:n1 % Actualizamos los Gates
    if u(2,j)>=g2
        u(1,j)=u(2,j)/m;
    elseif u(1,j)>=g1
        u(1,j)=g1;
    elseif u(1,j)>=h1
        % u(1,j) se queda como está
    elseif u(2,j)<=g*(u(1,j)-h1)+h2
        u(1,j)=h1;
    else
        u(1,j)=(u(2,j)-g*u(1,j))/(m-g);
    end
    u(2,j)=m*u(1,j);
end
```

```

C=u(1:2,:)-res2*ones(1,N); % Evaluamos la distancia
d2=max(abs(C))*u(3,:)+((abs(p(1,1:n1)-u(1,1:n1)).^r+
abs(p(2,1:n1)-u(2,1:n1)).^r).^(1/r))*u(3,1:n1)';
% Buscando el punto de l-inf con ARS
dini=d2;d22=d2;res22=res2;resini=res2;it22=0;
for i=1:5
res0=resini;d0=dini;
TOL=0.000001;T=1;it2=0;d=[1 -1 1 -1;-1 -1 1 1];e=500;
while T>=TOL
v=rand(2,1);ux=e*v(1);uy=e*v(2);aviso=0;
for k=1:4
res2(1)=res0(1)+ux*d(1,k);res2(2)=res0(2)+uy*d(2,k);
g1=(res2(2)+res2(1))/(m+1);g2=m*g1;
if m<1
h1=(res2(2)-res2(1))/(m-1); h2=m*h1;
else
h1=-10^9;h2=h1;
end
u=useg;
for j=1:n1 % Actualizamos los Gates
if u(2,j)>=g2
u(1,j)=u(2,j)/m;
elseif u(1,j)>=g1
u(1,j)=g1;
elseif u(1,j)>=h1
% u(1,j) se queda como está
elseif u(2,j)<=g*(u(1,j)-h1)+h2
u(1,j)=h1;
else
u(1,j)=(u(2,j)-g*u(1,j))/(m-g);
end
u(2,j)=m*u(1,j);
end
C=u(1:2,:)-res2*ones(1,N);
d2=max(abs(C))*u(3,:)+((abs(p(1,1:n1)-u(1,1:n1)).^r+
abs(p(2,1:n1)-u(2,1:n1)).^r).^(1/r))*u(3,1:n1)';
if d2<d0
res0=res2;aviso=1;
T=abs(d2-d0);d0=d2;
break
end
end
if aviso==0;e=e/1.2;end
it2=it2+1;it22=it22+1;
if it2>=200;break;end
end
if d0<d22;d22=d0;res22=res0;end
end
res2=res22;d2=d22; % Encontrado el punto de l-inf
t2=(cputime-t0)/vuel;t0=cputime;
u=[p,q]; n2=N-n1;
for vueltas=1:vuel % Buscamos el punto de lp
x0=p(1,:)*p(3,:)/sum(p(3,:));y0=p(2,:)*p(3,:)/sum(p(3,:));
x0=(x0+xr)/2;y0=(y0+m*xr)/2; qq=q;
for j=1:n2
qq(1,j)=(q(2,j)+q(1,j))/(m+1);
qq(2,j)=m*qq(1,j);
if qq(1,j)<=y0/m % Está en SH
u(1,n1+j)=y0/m;

```

```

elseif qq(2,j)<=m*x0 % Está en SC
    u(1,n1+j)=qq(1,j);
elseif q(1,j)<=q(2,j)-(m-1)*x0 % Está en SV (Seguro si m=1)
    u(1,n1+j)=x0;
elseif q(2,j)>=q(1,j)+(m-1)*(y0-g*x0)/(m-g) % Está en SB
    u(1,n1+j)=(q(2,j)-q(1,j))/(m-1);
else
    u(1,n1+j)=(y0-g*x0)/(m-g); % Está en SV2
end
u(2,n1+j)=m*u(1,n1+j);
end
TOL=10^(-6);T=1;precis=10^(-6);it1=0;d0=0.001;
while T>=TOL
    [res1,itw]=weiszfeldq(u,r,precis); % Algoritmo de Weiszfeld
    x0=res1(1);y0=res1(2); % Actualizamos los Gates
    for j=1:n2
        qq(1,j)=(q(2,j)+q(1,j))/(m+1);
        qq(2,j)=m*qq(1,j);
        if qq(1,j)<=y0/m
            u(1,n1+j)=y0/m;
        elseif qq(2,j)<=m*x0
            u(1,n1+j)=qq(1,j);
        elseif q(1,j)<=q(2,j)-(m-1)*x0
            u(1,n1+j)=x0;
        elseif q(2,j)>=q(1,j)+(m-1)*(y0-g*x0)/(m-g)
            u(1,n1+j)=(q(2,j)-q(1,j))/(m-1);
        else
            u(1,n1+j)=(y0-g*x0)/(m-g);
        end
        u(2,n1+j)=m*u(1,n1+j);
    end
    d1=(abs(u(1,:)-x0).^r+abs(u(2,:)-y0).^r)^(1/r)*w'+
        max(abs(q(1,:)-u(1,n1+1:n1+n2)),abs(q(2,:)-
            u(2,n1+1:n1+n2)))*w(1,n1+1:n1+n2)';
    T=abs(d1-d0);d0=d1;
    it1=it1+itw;
    if it1>=10000;break;end
end
u=[p,q];
end
if y0>=m*x0
    mensa='ERR';d1=-1;
else
    mensa='OK';
end
t1=(cputime-t0)/vuel;tt=(cputime-t00)/vuel;
v=[res2(1) xr x0;res2(2) m*xr y0]; % Deshacemos movimientos
sol=inv(A)*(v-md*ones(1,3));
res2=sol(:,1);x0=sol(1,3);y0=sol(2,3);xr=sol(1,2);
u=Useg;m=Mseg;
sal=[m r N res2(1) res2(2) d2 it22 t2 xr dr tr x0 y0 d1 it1 t1 tt];

```

### Programa Gate(p,q)

```

function [sal]=Gates(m,u,r,s,pos)
N=length(u(1,:));mseg=m;useg=u;
[v,A,md,m]=movimientos(m,u,pos);u=v;

```

```

n1=0;p=[]; q=[];
for j=1:N % Ubicando y ordenando en lq(s) y lp(r)
    if u(2,j)>=m*u(1,j)
        u(4,j)=s;
        q=[q,u(1:4, j)];
    else
        u(4,j)=r;n1=n1+1;
        p=[p,u(1:4, j)];
    end
end
u=[p,q];n2=N-n1;t00=cputime;vuel=10;
for vueltas=1:vuel
    if r==1
        g=-m^(-1/(s-1));
        [xr dr]=rectarslq(m,p,q,g,s,10^(-6));
    else
        [xr dr]=rectarspq(m,p,q,r,s,10^(-6));
    end
end
tr=(cputime-t00)/vuel;t0=cputime;
for vueltas=1:vuel % Buscamos el punto en lq
    z=p;pp=p;
    if r==1
        g=10^99;
    else
        g=-m^(-1/(r-1));
    end
    pp(1,:)=(p(2,:)/g-p(1,:))/(m/g-1);pp(2,:)=m*pp(1,:);
    xq0=(q(1,:)*q(3,:)'+pp(1,:)*p(3,:))/sum(u(3,:));
    yq0=(q(2,:)*q(3,:)'+pp(2,:)*p(3,:))/sum(u(3,:));
    xq=xq0;yq=(2*yq0+m*xq0)/3;
    for j=1:n1 % Gates iniciales
        v=[p(1,j) xq;p(2,j) yq];
        if p(1,j)==xq
            z(1,j)=p(1,j);z(2,j)=m*z(1,j);
        elseif p(2,j)==yq
            z(1,j)=p(2,j)/m;z(2,j)=p(2,j);
        else
            [z0]=gates_pq(m,v,r,s);
            z(1,j)=z0;z(2,j)=m*z(1,j);
        end
    end
    res2(1)=xq;res2(2)=yq;
    TOL=10^(-6);T=1;itq=0;D0=0.01;
    while T>=TOL
        u=[z,q];precis=0.001/(itq+1);
        [res2,itw]=arsweis(res2,u,s,precis); % Localización (gates/ptos lq)
        xq=res2(1);yq=res2(2); % Actualizamos punto inicial
        Dq=((abs(q(1,:)-xq).^s+abs(q(2,:)-yq).^s).^(1/s))*u(3,n1+1:N)';
        for j=1:n1 % Actualizamos los gates
            v=[p(1,j) xq;p(2,j) yq];
            if p(1,j)==xq
                z(1,j)=p(1,j);z(2,j)=m*z(1,j);
                d0=abs(yq-p(2,j));
            elseif p(2,j)==yq
                z(1,j)=p(2,j)/m;z(2,j)=p(2,j);
                d0=abs(xq-p(1,j));
            else
                [z0 d0]=gates_pq(m,v,r,s);
            end
        end
    end
end

```

```

                z(1,j)=z0;z(2,j)=m*z(1,j);
            end
            Dq=Dq+d0*z(3,j);
        end
        T=abs(Dq-D0);D0=Dq;res2(1)=xq;res2(2)=yq;
        itq=itq+itw;
        if itq>=10000;break;end
    end
end
tq=(cputime-t0)/vuel;t0=cputime;
for vueltas=1:vuel                                % Buscando el punto en lp
    z=q; g=-m^(-1/(s-1));
    qq=q;qq(1,:)=(q(2,:)/g-q(1,:))/(m/g-1);qq(2,:)=m*qq(1,:);
    xp0=(p(1,:)*p(3,:)'+qq(1,:)*q(3,:))/sum(u(3,:));
    yp0=(p(2,:)*p(3,:)'+qq(2,:)*q(3,:))/sum(u(3,:));
    xp=(2*xp0+yp0/m)/3;yp=(2*yp0+m*xp0)/3;
    for j=1:n2                                      % Gates iniciales
        v=[xp q(1,j);yp q(2,j)];
        if q(1,j)==xp
            z(1,j)=q(1,j);z(2,j)=m*z(1,j);
        elseif q(2,j)==yp
            z(1,j)=q(2,j)/m;z(2,j)=q(2,j);
        end
        [z0]=gates_pq(m,v,r,s);
        z(1,j)=z0;z(2,j)=m*z(1,j);
    end
    res2(1)=xp;res2(2)=yp;
    TOL=10^(-6);T=1;itp=0;D0=0.01;
    while T>=TOL
        u=[p,z];precis=0.001/(itp+1);
        [res2,itw]=arsweis(res2,u,r,precis);        % Localización (gates/ptos lp)
        xp=res2(1);yp=res2(2);                    % Actualizamos punto inicial
        Dp=((abs(p(1,:)-xp).^r+abs(p(2,:)-yp).^r).^(1/r))*u(3,1:n1)';
        for j=1:n2                                  % Actualizamos los gates
            v=[xp q(1,j);yp q(2,j)];
            if q(1,j)==xp
                z(1,j)=q(1,j);z(2,j)=m*z(1,j);
                d0=abs(yp-q(2,j));
            elseif q(2,j)==yp
                z(1,j)=q(2,j)/m;z(2,j)=q(2,j);
                d0=abs(xp-q(1,j));
            else
                [z0 d0]=gates_pq(m,v,r,s);
                z(1,j)=z0;
                z(2,j)=m*z(1,j);
            end
            Dp=Dp+d0*z(3,j);
        end
        T=abs(Dp-D0)/D0;D0=Dp;res2(1)=xp;res2(2)=yp;
        itp=itp+itw;
        if itp>=10000;break;end
    end
end
tp=(cputime-t0)/vuel;tt=(cputime-t00)/vuel;
v=[xq xr xp;yq m*xr yp];                          % Deshacemos movimientos
sol=inv(A)*(v-md*ones(1,3));
xq=sol(1,1);yq=sol(2,1);xr=sol(1,2);xp=sol(1,3);yp=sol(2,3);
u=useg;m=mseg;
sal=[r s m N xq yq Dq itq tq xr dr tr xp yp Dp itp tp tt];

```

## MÓDULOS

### Módulo Mfa\_Lq

```
function [resq,zq,dq,itq]=Mfa_Lq(m,u,n1,n2,r,s)
N=n1+n2;p=u(:,1:n1);q=u(:,n1+1:N);wp=p(3,:);wq=q(3,:);
x0=u(1,:)*u(3,:)/sum(u(3,:));y0=(m*p(1,:)*wp'+q(2,:)*wq')/sum(u(3,:));
z0=p(1,:);
T=1;TOL=10^(-6);e=0.0001;d0=0.0001;itq=0;h=min(1/(s-1),1);
while T>TOL
    itq=itq+1;
    % distancia en lp de Pi a gates
    zp=(z0-p(1,:)).^2+e;
    mzp=(m*z0-p(2,:)).^2+e;
    A=zp.^(r/2)+mzp.^(r/2);
    % distancia en lq de gates a solución
    xz=(z0-x0).^2+e;
    yz=(m*z0-y0).^2+e;
    B=xz.^(s/2)+yz.^(s/2);
    % distancia en lq de solución a Qj
    xq=(q(1,:)-x0).^2+e;
    yq=(q(2,:)-y0).^2+e;
    C=xq.^(s/2)+yq.^(s/2);
    % Nuevos gates
    numz=(x0*xz.^(s/2-1)+m*y0*yz.^(s/2-1))./B.^(1-1/s)+(p(1,:).*zp.^(r/2-1)+
        m*p(2,:).*mzp.^(r/2-1))./A.^(1-1/r);
    denomz=(xz.^(s/2-1)+m^2*yz.^(s/2-1))./B.^(1-1/s)+(zp.^(r/2-1)+
        m^2.*mzp.^(r/2-1))./A.^(1-1/r);
    z1=h*numz./denomz+(1-h)*z0;
    % Nueva solución
    numx=((q(1,:).*xq.^(s/2-1))./C.^(1-1/s))*wq'+
        ((z0.*xz.^(s/2-1))./B.^(1-1/s))*wp';
    denomx=(xq.^(s/2-1))./C.^(1-1/s))*wq'+
        ((xz.^(s/2-1))./B.^(1-1/s))*wp';
    numy=((q(2,:).*yq.^(s/2-1))./C.^(1-1/s))*wq'+
        ((m*z0.*yz.^(s/2-1))./B.^(1-1/s))*wp';
    deny=(yq.^(s/2-1))./C.^(1-1/s))*wq'+((yz.^(s/2-1))./B.^(1-1/s))*wp';
    x1=h*numx./denomx+(1-h)*x0;y1=h*numy./deny+(1-h)*y0;
    x0=x1;y0=y1;z0=z1;
    dq=((abs(q(1,:)-x1).^s+abs(q(2,:)-y1).^s).^(1/s))*wq'+
        ((abs(z0-x1).^s+abs(m*z0-y1).^s).^(1/s))*wp'+
        ((abs(p(1,:)-z0).^r+abs(p(2,:)-m*z0).^r).^(1/r))*wp';
    T=abs(dq-d0);zq=z0;d0=dq;
    if itq==10000;break;end
end
resq=[x0 y0];
```

### Módulo Mfa\_Lp

```
function [resp,zp,dp,itp]=Mfa_Lp(m,u,n1,n2,r,s)
p=u(:,1:n1);q=u(:,n1+1:n1+n2);wp=p(3,:);wq=q(3,:);
x0=u(1,:)*u(3,:)/sum(u(3,:));y0=(m*q(1,:)*wq'+p(2,:)*wp')/sum(u(3,:));
z0=q(1,:);
T=1;TOL=10^(-6);e=0.0001;d0=0.0001;itp=0;h=min(1/(r-1),1);
while T>TOL
    itp=itp+1;
```

```

% distancia en lq de Qj a gates
zq=(z0-q(1,:)).^2+e;
mq=(m*z0-q(2,:)).^2+e;
A=zq.^(s/2)+mq.^(s/2);
% distancia en lp de gates a solución
xz=(z0-x0).^2+e;
yz=(m*z0-y0).^2+e;
B=xz.^(r/2)+yz.^(r/2);
% distancia en lp de solución a Pi
xp=(p(1,)-x0).^2+e;
yp=(p(2,)-y0).^2+e;
C=xp.^(r/2)+yp.^(r/2);
% Nuevos gates
numz=(x0*xz.^(r/2-1)+m*y0*yz.^(r/2-1))./B.^(1-1/r)+
(q(1,).*zq.^(s/2-1)+m*q(2,).*mq.^(s/2-1))./A.^(1-1/s);
denomz=(xz.^(r/2-1)+m^2*yz.^(r/2-1))./B.^(1-1/r)+
(zq.^(s/2-1)+m^2.*mq.^(s/2-1))./A.^(1-1/s);
z1=h*numz./denomz+(1-h)*z0;
% Nueva solución
numx=((p(1,).*xp.^(r/2-1))./C.^(1-1/r))*wp'+
((z0.*xz.^(r/2-1))./B.^(1-1/r))*wq';
denomx=(xp.^(r/2-1))./C.^(1-1/r))*wp'+((xz.^(r/2-1))./B.^(1-1/r))*wq';
numy=((p(2,).*yp.^(r/2-1))./C.^(1-1/r))*wp'+
((m*z0.*yz.^(r/2-1))./B.^(1-1/r))*wq';
denomy=(yp.^(r/2-1))./C.^(1-1/r))*wp'+((yz.^(r/2-1))./B.^(1-1/r))*wq';
x1=h*numx./denomx+(1-h)*x0;y1=h*numy./denomy+(1-h)*y0;
x0=x1;y0=y1;z0=z1;
dp=((abs(p(1,)-x1).^r+abs(p(2,)-y1).^r).^(1/r))*wp'+
((abs(z0-x1).^r+abs(m*z0-y1).^r).^(1/r))*wq'+
((abs(q(1,)-z0).^s+abs(q(2,)-m*z0).^s).^(1/s))*wq';
T=abs(dp-d0);d0=dp;zp=z0;
if itp==10000;break;end
end
resp=[x0 y0];

```

## Módulo MGP1q\_q

```

function [res,z2,d2,it2]=MGP1q_q(m,u,g,s,n1,n2)
p=u(:,1:n1);q=u(:,n1+1:n1+n2);w1=p(3,:);w2=q(3,:);
x0=u(1,).*u(3,)./sum(u(3,:));y0=(m*p(1,).*w1'+q(2,).*w2')/sum(u(3,:));
z0=p(1,);
g1=(y0-g*x0)/(m-g); g2=m*g1;
for j=1:n1 % Obtenemos los gates
    if u(1,j)>=g1
        if u(2,j)<=g2
            z0(j)=g1;
        else
            z0(j)=u(2,j)/m;
        end
    else
        z0(j)=u(1,j);
    end
end
d0=(abs(q(1,)-x0).^s+abs(q(2,)-y0).^s).^(1/s)*w2'+(abs(z0-x0).^s+
abs(m*z0-y0).^s).^(1/s)*w1'+(abs(p(1,)-z0)+abs(p(2,)-m*z0))*w1';
T=1;TOL=10^(-6);e=0.0001;it2=0;h=min(1,1/(s-1));
while T>TOL

```

```

it2=it2+1;
% distancia en lq de gates a solución
xz=(z0-x0).^2+e;
yz=(m*z0-y0).^2+e;
B=xz.^(s/2)+yz.^(s/2);
% distancia en lq de solución a Qj
xq=(q(1,:)-x0).^2+e;
yq=(q(2,:)-y0).^2+e;
C=xq.^(s/2)+yq.^(s/2);
% Nueva solución
numx=((q(1,:).*xq.^(s/2-1))./C.^(1-1/s))*w2'+
      ((z0.*xz.^(s/2-1))./B.^(1-1/s))*w1';
denomx=(xq.^(s/2-1)./C.^(1-1/s))*w2'+((xz.^(s/2-1))./B.^(1-1/s))*w1';
numy=((q(2,:).*yq.^(s/2-1))./C.^(1-1/s))*w2'+
      ((m*z0.*yz.^(s/2-1))./B.^(1-1/s))*w1';
denomy=(yq.^(s/2-1)./C.^(1-1/s))*w2'+((yz.^(s/2-1))./B.^(1-1/s))*w1';
x1=h*numx/denomx+(1-h)*x0;y1=h*numy/denomy+(1-h)*y0;
x0=x1;y0=y1;
g1=(y0-g*x0)/(m-g); g2=m*g1;
for j=1:n1 % Obtenemos los nuevos gates
    if u(1,j)>=g1
        if u(2,j)<=g2
            z0(j)=g1;
        else
            z0(j)=u(2,j)/m;
        end
    else
        z0(j)=u(1,j);
    end
end
d2=((abs(q(1,:)-x0).^s+abs(q(2,:)-y0).^s).^(1/s))*w2'+
    ((abs(z0-x0).^s+abs(m*z0-y0).^s).^(1/s))*w1'+
    (abs(p(1,:)-z0)+abs(p(2,:)-m*z0))*w1';
T=abs(d2-d0);d0=d2;
if it2>=10000;break;end
end
res=[x0 y0];z2=z0;

```

## Módulo MGPIq\_1

```

function [res1,Z0,d1,it1]=MGPIq_1(m,u,g,s,n1,n2)
p=u(:,1:n1);q=u(:,n1+1:n1+n2);qq=q/w1=p(3,:);w2=q(3,:);
for j=1:n2
    qq(1,j)=(q(2,j)-g*q(1,j))/(m-g);qq(2,j)=m*qq(1,j);
end
x0=u(1,:)*u(3,:)/sum(u(3,:));
y0=(m*q(1,:)*u(3,n1+1:n1+n2)+p(2,:)*u(3,1:n1))/sum(u(3,:));
Z0=qq(1,:);
for j=1:n2 % Inicializamos los gates
    if qq(1,j)>=x0 % está en Sv
        Z0(j)=x0;
    elseif qq(2,j)<=y0 % está en Sh
        Z0(j)=y0/m;
    else
        Z0(j)=qq(1,j); %está en Sp
    end
end
end

```

```

d0=(abs(p(1,:)-x0)+abs(p(2,:)-y0))*p(3,:)'+(abs(Z0-x0)+
abs(m*Z0-y0))*q(3,:)'+((abs(Z0-q(1,:)).^s+
abs(m*Z0-q(2,:)).^s).^1/s))*q(3,:)';
T=1;TOL=10^(-6);it1=0;e=0.0001;
while T>TOL
    it1=it1+1;
    % distancia en lp de gates a solución
    xz=(Z0-x0).^2+e;
    yz=(m*Z0-y0).^2+e;
    % distancia en lp de solución a Pi
    xp=(p(1,:)-x0).^2+e;
    yp=(p(2,:)-y0).^2+e;
    % Nueva solución
    numx=((p(1,:).*xp.^(1/2-1))*w1'+((Z0.*xz.^(1/2-1))*w2');
    denomx=(xp.^(1/2-1))*w1'+((xz.^(1/2-1))*w2');
    numy=((p(2,:).*yp.^(1/2-1))*w1'+((m*Z0.*yz.^(1/2-1))*w2');
    deny=(yp.^(1/2-1))*w1'+((yz.^(1/2-1))*w2');
    x1=numx/denomx;y1=numy/denomy;
    x0=x1;y0=y1;
    for j=1:n2 % Actualizamos los gates
        if qq(1,j)>=x0 % está en Sv
            Z0(j)=x0;
        elseif qq(2,j)<=y0 % está en Sh
            Z0(j)=y0/m;
        else
            Z0(j)=qq(1,j); %está en Sp
        end
    end
    d1=(abs(p(1,:)-x0)+abs(p(2,:)-y0))*w1'+(abs(Z0-x0)+
abs(m*Z0-y0))*w2'+((abs(Z0-q(1,:)).^s+
abs(m*Z0-q(2,:)).^s).^1/s))*w2';
    T=abs(d1-d0);d0=d1;
    if it1>=10000;break;end
end
res1=[x0 y0];

```

## Módulo f\_lq

```

function [res2,d2,it2]=f_lq(m,g,p,q,xr,s)
% Busca la solución en lq con Weiszfeld
u=[p,q];w=u(3,:);n1=length(p(1,:));w1=sum(p(3,:));w2=sum(q(3,:));w12=w1+w2;
% Baricentro de los puntos de lq y promedio con Xr
res2(1)=q(1,:)*q(3,:)/sum(q(3,:));res2(2)=q(2,:)*q(3,:)/sum(q(3,:));
res2(1)=(w2*res2(1)+w1*xr)/w12;res2(2)=(w2*res2(2)+w1*m*xr)/w12;
g1=(res2(2)-g*res2(1))/(m-g); g2=m*g1;
for j=1:n1
    if u(1,j)>=g1
        if u(2,j)<=g2
            u(1,j)=g1;u(2,j)=g2;
        else
            u(1,j)=u(2,j)/m;
        end
    else
        u(2,j)=m*u(1,j);
    end
end
end
TOL=0.000001;T=1;precis=0.000001;it2=0;d0=0.001;

```

```

while T>=TOL
    [res2,itw]=weiszfeldq(u,s,precis); % Algoritmo de localización simple
    g1=(res2(2)-g*res2(1))/(m-g); g2=m*g1; % Actualizamos los Gates
    u=[p,q];
    for j=1:n1
        if u(1,j)>=g1
            if u(2,j)<=g2
                u(1,j)=g1;u(2,j)=g2;
            else
                u(1,j)=u(2,j)/m;
            end
        else
            u(2,j)=m*u(1,j);
        end
    end
    x=u(1,:); y=u(2,:); % Evaluamos la distancia
    d2=(abs(x-res2(1)).^s+abs(y-res2(2)).^s).^(1/s)*w'+
        sum((abs(p(1,1:n1)-u(1,1:n1))+abs(p(2,1:n1)-u(2,1:n1))).*w(1,1:n1));
    T=abs(d2-d0);
    it2=it2+itw;d0=d2;
    if it2>=10000;break;end
end

```

## Módulo f\_llq

```

function [x0,y0,d1,mensa]=f_llq(m,g,p,q,xr,s)
% Busca la solución en ll
n1=length(p(1,:));n2=length(q(1,:));qq=q;
for j=1:n2
    qq(1,j)=(q(2,j)-g*q(1,j))/(m-g);
    qq(2,j)=m*qq(1,j);
end
uu=[p(1:3,:),qq(1:3,)];M=medianas(uu);
mxp=min(M(1,:));myp=max(M(2,:)); % Encontrada la mediana
[xval yval]=malla(n1,n2,m,p,qq,mxp,myp); % Obteniendo la malla
[x0,cx,ux]=derqx(n1,n2,m,p,q,qq,xval,s); % Calculando la derivadas
[y0,cy,uy]=derqy(n1,n2,m,p,q,qq,yval,s);
switch cx+2*cy
    case 0
        mensa=0; % Está en un punto
        d1=(abs(x0-p(1,:))+abs(y0-p(2,:)))*p(3,:);
        for j=1:n2
            if qq(2,j)<=y0 % Está en Sh
                d1=d1+((abs(q(1,j)-y0/m)^s+abs(q(2,j)-y0)^s)^(1/s)+
                    (x0-y0/m))*q(3,j);
            elseif qq(1,j)>=x0 % Está en Sv
                d1=d1+((abs(q(1,j)-x0)^s+abs(q(2,j)-m*x0)^s)^(1/s)+
                    (m*x0-y0))*q(3,j);
            else
                d1=d1+((abs(q(1,j)-qq(1,j))^s+abs(q(2,j)-qq(2,j))^s)^(1/s)+
                    (abs(x0-qq(1,j))+abs(y0-qq(2,j))))*q(3,j);
            end
        end
    case 1
        mensa=1; % Está en un segmento horizontal
        D1=0;wh=0;wp=0;wv=0;Spp=[];Sv=[];
        for j=1:n2

```

```

        if qq(2,j)<=y0                % Está en Sh
            D1=D1+(abs(q(1,j)-y0/m)^s+abs(q(2,j)-y0)^s)^(1/s)*q(3,j);
            wh=wh+q(3,j);
        elseif qq(1,j)>=x0            % Está en Sv
            Sv=[Sv,q(:,j)];wv=wv+q(3,j);
        else
            D1=D1+(abs(q(1,j)-qq(1,j))^s+
                abs(q(2,j)-qq(2,j))^s)^(1/s)*q(3,j);
            wp=wp+1;Spp=[Spp,qq(:,j)];
        end
    end
    v=[p,Spp];inf=xval(ux);sup=xval(ux+1);
    options=optimset('TolFun',1e-6);
    [x1,d1]=fminbnd(@(x) dist_seg_hq(x,m,v,s,nl,wp,Sv,wv,y0,wh),++
        inf,sup,options);
    d1=d1+D1;x0=x1;
case 2
    mensa=2;                        % Está en un segmento vertical
    D1=0;wh=0;wp=0;wv=0;Spp=[];Sh=[];
    for j=1:n2
        if qq(1,j)>=x0                % Está en Sv
            D1=D1+(abs(q(1,j)-x0)^s+abs(q(2,j)-m*x0)^s)^(1/s)*q(3,j);
            wv=wv+q(3,j);
        elseif qq(2,j)<=y0            % Está en Sh
            Sh=[Sh,q(:,j)];wh=wh+q(3,j);
        else
            D1=D1+(abs(q(1,j)-qq(1,j))^s+
                abs(q(2,j)-qq(2,j))^s)^(1/s)*q(3,j);
            wp=wp+1;Spp=[Spp,qq(:,j)];
        end
    end
    v=[p,Spp];inf=yval(uy);sup=yval(uy+1);
    options=optimset('TolFun',1e-6);
    [y1,d1]=fminbnd(@(y) dist_seg_vq(y,m,v,s,nl,wp,Sh,wh,x0,wv),++
        inf,sup,options);
    d1=d1+D1;y0=y1;
otherwise
    mensa=12;                        % Está en un rectángulo
    D1=0;wh=0;wp=0;wv=0;Spp=[];Sh=[];Sv=[];
    x0=xval(ux);y0=yval(uy);
    for j=1:n2
        if qq(1,j)>=x0                % Está en Sv
            Sv=[Sv,q(:,j)];wv=wv+q(3,j);
        elseif qq(2,j)<=y0            % Está en Sh
            Sh=[Sh,q(:,j)];wh=wh+q(3,j);
        else
            D1=D1+(abs(q(1,j)-qq(1,j))^s+
                abs(q(2,j)-qq(2,j))^s)^(1/s)*q(3,j);
            wp=wp+1;Spp=[Spp,qq(:,j)];
        end
    end
    v=[p,Spp];options=optimset('TolFun',1e-06);
    [X,d1]=fminsearch(@(a) distancialq(a,m,v,s,nl,wp,Sh,wh,Sv,wv),++
        [x0,y0],options);
    d1=d1+D1; x0=X(1);y0=X(2);
end
if y0>=m*x0
    mensa=99;x0=xr;y0=m*x0-10^(-4);
    d1=(abs(x0-p(1,:))+abs(y0-p(2,:)))*p(3,:);

```

```

for j=1:n2
    if qq(2,j)<=y0 % Está en Sh
        d1=d1+((abs(q(1,j)-y0/m)^s+abs(q(2,j)-y0)^s)^(1/s)+
            (x0-y0/m))*q(3,j);
    elseif qq(1,j)>=x0 % Está en Sv
        d1=d1+((abs(q(1,j)-x0)^s+abs(q(2,j)-m*x0)^s)^(1/s)+
            (m*x0-y0))*q(3,j);
    else
        d1=d1+((abs(q(1,j)-qq(1,j))^s+abs(q(2,j)-qq(2,j))^s)^(1/s)+
            (abs(x0-qq(1,j))+abs(y0-qq(2,j))))*q(3,j);
    end
end
end
end

```

## Módulo f\_lqars

```

function [res2,d2,it2]=f_lqars(m,g,p,q,xr,s)
% Busca la solución en lq con ARS
u=[p,q];w=u(3,:);n1=length(p(1,:));w1=sum(p(3,:));w2=sum(q(3,:));w12=w1+w2;
% Baricentro de los puntos de lq y promedio con Xr
res2(1)=q(1,:)*q(3,:)/sum(q(3,:));res2(2)=q(2,:)*q(3,:)/sum(q(3,:));
res2(1)=(w2*res2(1)+w1*xr)/w12;res2(2)=(w2*res2(2)+w1*m*xr)/w12;
g1=(res2(2)-g*res2(1))/(m-g); g2=m*g1;
for j=1:n1
    if u(1,j)>=g1
        if u(2,j)<=g2
            u(1,j)=g1;u(2,j)=g2;
        else
            u(1,j)=u(2,j)/m;
        end
    else
        u(2,j)=m*u(1,j);
    end
end
end
TOL=10^(-6);T=1;precis=10^(-6);it2=0;d0=0.001;
while T>=TOL
    [res2,itw]=arsweis(res2,u,s,precis); % Localización simple con ARS
    g1=(res2(2)-g*res2(1))/(m-g); g2=m*g1; % Actualizamos los Gates
    u=[p,q];
    for j=1:n1
        if u(1,j)>=g1
            if u(2,j)<=g2
                u(1,j)=g1;u(2,j)=g2;
            else
                u(1,j)=u(2,j)/m;
            end
        else
            u(2,j)=m*u(1,j);
        end
    end
end
x=u(1,:); y=u(2,:); % Evaluamos la distancia
d2=(abs(x-res2(1)).^s+abs(y-res2(2)).^s)^(1/s)*w'+
    sum((abs(p(1,1:n1)-u(1,1:n1))+abs(p(2,1:n1)-u(2,1:n1))).*w(1,1:n1));
T=min(abs(d2-d0),abs(res2(2)-m*res2(1))/sqrt(1+m^2));
it2=it2+itw;d0=d2;
if it2>=10000;break;end
end
end

```

## RUTINAS

### Rutina movimientos

```
function [v,A,md,m]=movimientos(m,u,pos)
% Programa que transforma los datos de manera que,  $0 < m \leq 1$ ,  $lq / l1$ 
% pos=1 si  $lq/l1$ , pos=0 en caso contrario
v=u;md=[0;0];N=length(u(1,:));
if pos==1
    if m<=1
        A=eye(2);
    else
        A=[0 -1;-1 0]; % Simetría axial(Y) y giro(+90°)
        m=1/m;
        md(1)=max(u(2,:));md(2)=m*md(1);
    end
elseif m<=1
    A=[-1 0;0 -1]; % Simetría respecto del origen
    md(1)=max(u(1,:));md(2)=m*md(1);
else
    A=[0 1;1 0]; % Simetría respecto de la bisectriz
    m=1/m;
end
v(1:2,:)=A*u(1:2,:)+md*ones(1,N); % Trasladamos al primer cuadrante
```

### Rutina rectars

```
function [xr dr]=rectars(m,p,q,r,s,precis)
% Obtiene el punto de la recta cuya distancia-p a p y distancia-q a q es
mínima
u=[p,q];xr=u(1,:)*u(3,:)/sum(u(3,:));
dr=(abs(xr-p(1,:)).^r+abs(m*xr-p(2,:)).^r).^(1/r)*p(3,:)+
    (abs(xr-q(1,:)).^s+abs(m*xr-q(2,:)).^s).^(1/s)*q(3,:);
T=1;it2=0;e=1;d=[-1 1];
while T>=precis
    v=rand(2,1);
    ux=fix(10*v(1))/10*e;
    flag=0;
    for j=1:2
        xj=xr+d(j)*ux;
        dj=(abs(xj-p(1,:)).^r+abs(m*xj-p(2,:)).^r).^(1/r)*p(3,:)+
            (abs(xj-q(1,:)).^s+abs(m*xj-q(2,:)).^s).^(1/s)*q(3,:);
        if dj<dr
            xr=xj;T=abs(dj-dr);
            dr=dj;
            flag=1;e=1.1*e;
            break
        end
    end
    if flag==0;e=e/2;end
    it2=it2+1;
    if it2>=200;break;end
end
```

## Rutina rectarslq

```
function [xr dr]=rectarslq(m,p,q,g,s,precis)
% Obtine el punto de la recta cuya distancia a p y q es mínima
qq=q;pp=p;n1=length(p(1,:));n2=length(q(1,:));
for j=1:n2
    qq(1,j)=(q(2,j)-g*q(1,j))/(m-g);
    qq(2,j)=m*qq(1,j);
end
for j=1:n1
    pp(1,j)=p(1,j);
    pp(2,j)=m*pp(1,j);
end
uu=[pp(1:3,:),qq(1:3,:)];
M=medianas(uu);
xr=(min(M(1,:))+max(M(1,:)))/2;
for j=1:n1
    if p(1,j)>=xr
        if p(2,j)<=m*xr
            pp(1,j)=xr;pp(2,j)=m*xr;
        else
            pp(1,j)=p(2,j)/m;pp(2,j)=p(2,j);
        end
    else
        pp(2,j)=m*p(1,j);pp(1,j)=p(1,j);
    end
end
dr=((abs(xr-pp(1,:)).^s+abs(m*xr-pp(2,:)).^s).^s+abs(pp(1,:)-p(1,:))+
    abs(pp(2,:)-p(2,:)))*p(3,:)+(abs(xr-q(1,:)).^s+
    abs(m*xr-q(2,:)).^s).^s*p(3,:);
T=1;it2=0;e=1;d=[-1 1];
while T>=precis
    v=rand(2,1);
    ux=fix(10*v(1))/10*e;
    flag=0;
    for j=1:2
        xj=xr+d(j)*ux;
        for h=1:n1
            if p(1,h)>=xj
                if p(2,h)<=m*xj
                    pp(1,h)=xj;pp(2,h)=m*xj;
                else
                    pp(1,h)=p(2,h)/m;pp(2,h)=p(2,h);
                end
            else
                pp(2,h)=m*p(1,h);pp(1,h)=p(1,h);
            end
        end
        dj=((abs(xj-pp(1,:)).^s+abs(m*xj-pp(2,:)).^s).^s+
            abs(pp(1,:)-p(1,:))+abs(pp(2,:)-p(2,:)))*p(3,:)+
            (abs(xj-q(1,:)).^s+abs(m*xj-q(2,:)).^s).^s*p(3,:);
        if dj<dr
            xr=xj;T=abs(dj-dr);
            dr=dj;
            flag=1;e=1.1*e;
            break
        end
    end
end
```

```

        if flag==0;e=e/2;end
        it2=it2+1;
        if it2>=200;break;end
end

```

### Rutina weiszfeldq

```

function [res2,itw]=weiszfeldq(u,s,precis)
% Algoritmo de localización de Weiszfeld con suavizado hiperbólico e=0.0001
% Para s=1 halla la mediana directamente
if s==1
    uu=u(1:3,:);
    M=medianas(uu);
    res2=[max(M(1,:)),min(M(2,:))];
    itw=1;
    return
end
h=min(2/(s-1),1); % Factor multiplicador
x=u(1,:);y=u(2,:);w=u(3,:);e=0.0001;
x0=sum(x.*w)/sum(w);y0=sum(y.*w)/sum(w); % Centro de gravedad
itw=0;t=5;d0=(abs(x-x0).^s+abs(y-y0).^s).^(1/s)*w';
while t>precis
    raiz=((x-x0).^2+e).^(s/2)+((y-y0).^2+e).^(s/2)).^(1-1/s);
    xx=((x-x0).^2+e).^(1-s/2);
    yy=((y-y0).^2+e).^(1-s/2);
    x1=sum(w.*x./(xx.*raiz))/sum(w./(xx.*raiz));
    y1=sum(w.*y./(yy.*raiz))/sum(w./(yy.*raiz));
    itw=itw+1;
    x2=(1-h)*x0+h*x1;y2=(1-h)*y0+h*y1;
    x1=x2;y1=y2;
    d1=(abs(x-x1).^s+abs(y-y1).^s).^(1/s)*w';
    t=abs(d1-d0);
    x0=x1;y0=y1;d0=d1;res2=[x0 y0];
    if itw>=10000;break;end
end

```

### Rutina medianas

```

function M=medianas(u)
% u es una matriz con coordenadas y ponderación en cada columna
% u=[x1 x2 x3...;y1 y2 y3...;.....;z1 z2 z3...;w1 w2 w3....]
fil=length(u(:,1));col=length(u(1,:));M=zeros(fil-1,2);
N=sum(u(fil,:));
for i=1:fil-1
    v=[u(i,:);u(fil,:)]';
    v=sortrows(v,1);w=v(:,2)';
    W=zeros(1,col);
    x0=-10;
    for j=1:col
        W(j)=sum(w(1:j));
        if W(j)<N/2
            continue
        elseif W(j)==N/2
            x0=v(j,1);
            continue
        end
    end
end

```

```

        else
            x1=v(j,1);
            break
        end
    end
    if x0== -10
        x0=x1;
    end
    M(i,1)=x0;M(i,2)=x1;
end

```

### Rutina malla

```

function [xval yval]=malla(n1,n2,m,p,qq,mxp,myp)
xval=[mxp];yval=[myp];
for j=1:n1
    if p(1,j)>=myp/m
        if p(1,j)<=mxp
            xval=[p(1,j),xval];
        end
    end
    if p(2,j)<=m*mxp
        if p(2,j)>=myp
            yval=[yval,p(2,j)];
        end
    end
end
for j=1:n2
    if qq(1,j)>=myp/m
        if qq(1,j)<=mxp
            xval=[qq(1,j),xval];
        end
    end
    if qq(2,j)<=m*mxp
        if qq(2,j)>=myp
            yval=[yval,qq(2,j)];
        end
    end
end
xval=[myp/m,xval,mxp];yval=[myp,yval,m*mxp];
xval=sort(xval);yval=sort(yval); % Eliminando los duplicados
xfval=(xval(1));yfval=(yval(1));
for i=2:length(xval)
    if abs(xval(i)-xval(i-1))>0.001
        xfval=[xfval,xval(i)];
    end
end
for i=2:length(yval)
    if abs(yval(i)-yval(i-1))>0.001
        yfval=[yfval,yval(i)];
    end
end
xval=xfval; yval=yfval;

```

## Rutina derqx

```
function [x0,cx,ux]=derqx(n1,n2,m,p,q,qq,xval,s)
% Calcula derivadas laterales en X
cx=-5;ux=0;
for i=1:length(xval)
    a=0;c=0;
    x0=xval(i);dx=sign(x0-p(1,:))*p(3,1:n1)'+sum(q(3,1:n2));
    for j=1:n1
        if p(1,j)==x0
            a=a+p(3,j);
        end
    end
    for j=1:n2
        if qq(1,j)>x0 % Está en Sv*
            dx=dx+q(3,j)*(m-1+(sign(x0-q(1,j))*abs(x0-q(1,j))^(s-1)-
                m*abs(m*x0-q(2,j))^(s-1))/(abs(x0-q(1,j))^s+
                abs(m*x0-q(2,j))^s)^(1-1/s));
        end
        if abs(q(1,j)-x0)<0.001 && abs(q(2,j)-m*x0)<0.001
            c=c+q(3,j);
        end
    end
    dix=dx-a+c*(m-1-(1+m^s)^(1/s));ddx=dx+a;
    if dix*ddx<=0 % Solución EX en X
        cx=0;
        break
    elseif ddx<0
        continue
    else
        if i==1
            cx=0; % Solución EX en X
        else
            cx=1;ux=i-1; % Solución SH
        end
        break
    end
end
end
if cx==5
    cx=0; % Solución EX en X
end
end
```

## Rutina derqy

```
function [y0,cy,uy]=derqy(n1,n2,m,p,q,qq,yval,s)
% Calcula derivadas laterales en Y
cy=-5;uy=0;
for i=length(yval):-1:1
    b=0;d=0;
    y0=yval(i);dy=sign(y0-p(2,:))*p(3,1:n1)'+sum(q(3,1:n2));
    for j=1:n1
        if p(2,j)==y0
            b=b+p(3,j);
        end
    end
    for j=1:n2
        if qq(2,j)<y0 % Está en Sh*
```

```

                dy=dy+q(3,j)*(1-1/m+(1/m*abs(y0/m-q(1,j))^(s-1)+
                    sign(y0-q(2,j))*abs(y0-q(2,j))^(s-1))/(abs(y0/m-q(1,j))^s+
                    abs(y0-q(2,j))^s)^(1-1/s));
            end
            if abs(q(1,j)-y0/m)<0.001 && abs(q(2,j)-y0)<0.001
                d=d+q(3,j);
            end
        end
        diy=dy-b;ddy=dy+b+d*(1-1/m+(1/m^s+1)^(1/s));
        if diy*ddy<=0
            cy=0; % Solución EX en Y
            break
        elseif ddy>0
            continue
        else
            if i==length(yval)
                cy=0; % Solución EX en Y
            else
                cy=1;uy=i; % Solución SV
            end
            break
        end
    end
end
if cy== -5cy=0;end % Solución EX en Y

```

#### Rutina dist\_seg\_hq

```

function d1=dist_seg_hq(x,m,v,s,n1,wp,Sv,wv,y0,wh)
xi=v(1,:)-x;yi=v(2,:)-y0;w=v(3,1:n1+wp);
if isempty(Sv)
    d11=0;
else
    sx=Sv(1,:)-x;sy=Sv(2,:)-m*x;
    d11=sum((abs(sx).^s+abs(sy).^s)^(1/s).*Sv(3,:));
end
d1=wh*(x-y0/m)+(abs(xi)+abs(yi))*w'+wv*(m*x-y0)+d11;

```

#### Rutina dist\_seg\_vq

```

function d1=dist_seg_vq(y,m,v,s,n1,wp,Sh,wh,x0,wv)
xi=v(1,:)-x0;yi=v(2,:)-y;w=v(3,1:n1+wp);
if isempty(Sh)
    d11=0;
else
    sx=Sh(1,:)-y/m;sy=Sh(2,:)-y;
    d11=sum((abs(sx).^s+abs(sy).^s)^(1/s).*Sh(3,:));
end
d1=(abs(xi)+abs(yi))*w'+wv*(m*x0-y)+wh*(x0-y/m)+d11;

```

#### Rutina distancia1q

```

function d1=distancia1q(a,m,v,s,n1,wp,Sh,wh,Sv,wv)
xi=v(1,:)-a(1);yi=v(2,:)-a(2);w=v(3,1:n1+wp);
if isempty(Sv)

```

```

        dlw=0;
    else
        svx=Sv(1,:)-a(1);svy=Sv(2,:)-m*a(1);
        dlw=sum((abs(svx).^s+abs(svy).^s).^(1/s).*Sv(3,:));
    end
    if isempty(Sh)
        dlh=0;
    else
        shx=Sh(1,:)-a(2)/m;shy=Sh(2,:)-a(2);
        dlh=sum((abs(shx).^s+abs(shy).^s).^(1/s).*Sh(3,:));
    end
    dl=wh*abs(a(1)-a(2)/m)+(abs(xi)+abs(yi))*w'+wv*abs(m*a(1)-a(2))+dlw+dlh;

```

### Rutina arsweis

```

function [res2,itw]=arsweis(res2,u,s,precis)
% ARS aplicado a Weber en un semiplano
x=u(1,:);y=u(2,:);itw=0;d22=10^9;d=[-1 -1;1 -1;1 1;-1
1];x22=res2(1);y22=res2(2);
for i=1:5
    x0=res2(1);y0=res2(2);
    d0=(abs(x-x0).^s+abs(y-y0).^s).^(1/s)*u(3,:)';T=1;it2=0;e=2;
    while T>=precis
        v=rand(2,1);ux=e*v(1);uy=e*v(2);aviso=0;
        for j=1:4
            xj=x0+d(j,1)*ux;yj=y0+d(j,2)*uy;
            dj=(abs(x-xj).^s+abs(y-yj).^s).^(1/s)*u(3,:)';
            if dj<d0
                x0=xj;y0=yj;aviso=1;
                T=abs(dj-d0);d0=dj;
            end
        end
        if aviso==0;e=e/2;end
        it2=it2+1;itw=itw+1;
        if d0<d22;d22=d0;x22=x0;y22=y0;end
        if it2>=200;break;end
    end
end
res2=[x22 y22];

```

### Rutina actugatesq

```

function d0=actugatesq(p,q,m,g,s,x0,y0)
g1=(y0-g*x0)/(m-g); g2=m*g1; % Actualizamos el Gate y los puntos de cruce
u=[p,q];n1=length(p(1,:));
for j=1:n1
    if u(1,j)>=g1
        if u(2,j)<=g2
            u(1,j)=g1;u(2,j)=g2;
        else
            u(1,j)=u(2,j)/m;
        end
    else
        u(2,j)=m*u(1,j);
    end
end

```

```

end
x=u(1,:); y=u(2,:); d0=((abs(x-x0).^s+abs(y-y0).^s).^(1/s))*u(3,:)'+
    (abs(p(1,1:n1)-u(1,1:n1))+abs(p(2,1:n1)-u(2,1:n1)))*u(3,1:n1)';

```

### Rutina arsdist1q

```

function d1=arsdist1q(p,q,m,s,qq,x0,y0)
n2=length(q(1,:));
d1=(abs(x0-p(1,:))+abs(y0-p(2,:)))*p(3,:);
for j=1:n2
    if qq(2,j)<=y0-0.001 % está en Sh
        d1=d1+((abs(q(1,j)-y0/m)^s+abs(q(2,j)-y0)^s).^(1/s)+(x0-y0/m))*q(3,j);
    elseif qq(1,j)>=x0+0.001 % está en Sv
        d1=d1+((abs(q(1,j)-x0)^s+abs(q(2,j)-m*x0)^s).^(1/s)+(m*x0-y0))*q(3,j);
    else
        d1=d1+((abs(q(1,j)-qq(1,j))^s+abs(q(2,j)-qq(2,j))^s).^(1/s)+
            (abs(x0-qq(1,j))+abs(y0-qq(2,j))))*q(3,j);
    end
end
end

```

### Rutina rectarspinf

```

function [xr dr]=rectarspinf(m,p,q,r,precis)
% Obtine el punto de la recta cuya distancia a p y q (inf) es mínima
k1=3*m/2;mq=-1;g2=0;
if r==1
    g1=10^99;mp=10^99;
else
    g1=fsolve(@(g) 1+m*g^(r-1)-((1+g^r)^(r-1)/r),k1,++
        optimset('Display','off','TolFun',1e-20));
    mp=-m*abs(m)^(-r/(r-1));
end
% g1<g2, es decir, g1 define la zona "horizontal"
qq=q;pp=p;n1=length(p(1,:));n2=length(q(1,:));
for j=1:n2
    qq(1,j)=(q(2,j)-mq*q(1,j))/(m-mq);qq(2,j)=m*qq(1,j);
end
for j=1:n1
    pp(1,j)=(p(2,j)-mp*p(1,j))/(m-mp);pp(2,j)=m*pp(1,j);
end
uu=[pp(1:3,:),qq(1:3,)];M=medianas(uu);xr=(min(M(1,:))+max(M(1,:)))/2;
for j=1:n1
    if p(2,j)>=m*xr+g2*(p(1,j)-xr)
        pp(1,j)=(p(2,j)-g2*p(1,j))/(m-g2);
    elseif p(2,j)<=m*xr+g1*(p(1,j)-xr)
        pp(1,j)=xr;
    else
        pp(1,j)=(p(2,j)-g1*p(1,j))/(m-g1);
    end
    pp(2,j)=m*pp(1,j);
end
end
resj(1)=xr;resj(2)=m*xr;
Cp=pp(1:2,:)-resj'*ones(1,n1);Cq=q(1:2,:)-resj'*ones(1,n2);
dr=max(abs(Cp))*p(3,:)+max(abs(Cq))*q(3,:)+((abs(pp(1,:)-p(1,:)).^r+
    abs(pp(2,:)-p(2,:)).^r).^(1/r))*p(3,:);

```

```

T=1;itg=0;d=[-1 1];e=1;
while T>=precis
    aviso=0;
    for j=1:2
        xj=xr+d(j)*e;
        for h=1:n1
            if p(2,h)>=m*xj+g2*(p(1,h)-xj)
                pp(1,h)=(p(2,h)-g2*p(1,h))/(m-g2);
            elseif p(2,h)<=m*xj+g1*(p(1,h)-xr)
                pp(1,h)=xj;
            else
                pp(1,h)=(p(2,h)-g1*p(1,h))/(m-g1);
            end
            pp(2,h)=m*pp(1,h);
        end
        resj(1)=xj;resj(2)=m*xj;
        Cp=pp(1:2,:) -resj'*ones(1,n1);Cq=q(1:2,:)-resj'*ones(1,n2);
        dj=max(abs(Cp))*p(3,:)+max(abs(Cq))*q(3,:)+
            ((abs(pp(1,:)-p(1,:)).^r+abs(pp(2,:)-p(2,:)).^r).^(1/r))*p(3,:);
        if dj<dr
            xr=xj;T=abs(dj-dr);dr=dj;aviso=1;
            break
        end
    end
    end
    if aviso==0;e=e/2;end
    itg=itg+1;
    if itg>=200;break;end
end

```

### Rutina Punto\_rectainf

```

function dr=Punto_rectainf(x,m,u,N,n1)
xi=x-u(1,:); yi=m*x-u(2,:); w=u(3,:);
dr=(abs(xi(1:n1))+abs(yi(1:n1)))*w(1:n1)'+
    max([abs(xi(n1+1:N));abs(yi(n1+1:N))])*w(n1+1:N)';

```

### Rutina derl infx

```

function [x0,cx,ux]=derl infx(n1,n2,m,p,q,qq,xval)
% Calcula la derivadas laterales en X para p=1 y q=inf
cx=-5;ux=0;
for i=1:length(xval)
    a=0;w0=0;w1=0;w2=0;wd=0;
    x0=xval(i);dx=sign(x0-p(1,:))*p(3,1:n1)';
    for j=1:n1
        if p(1,j)==x0
            a=a+p(3,j);
        end
    end
    end
    for j=1:n2
        if qq(1,j)<x0 % Está en Srh ó Sh ó Sc
            dx=dx+q(3,j);
        end
        if abs(qq(1,j)-x0)<0.0001 && abs(qq(2,j)-m*x0)<0.0001 % Está en r1
            if abs(q(1,j)-x0)<0.0001 % Está en el vértice de r1 y r

```

```

        w0=w0+q(3,j);
    else
        w1=w1+q(3,j);           % Está en r1 pero no está en r
    end
end
if q(1,j)>x0 && q(2,j)<=q(1,j)+(m-1)*x0 % Puede estar en r2 ó V2
    if abs(q(2,j)-q(1,j)-(m-1)*x0)<0.0001
        w2=w2+q(3,j);         % Está en r2
    else
        wd=wd+q(3,j);         % Está en V2
    end
end
end
dix=dx-a+(m-1)*(w0+w2+wd); ddx=dx+a+w0+w1+(m-1)*wd;
if dix*ddx<-0.001           % Solución EX en X
    cx=0;
    break
elseif abs(ddx)<0.001
    ux=i;cx=0;
    break
elseif ddx<0
    continue
else
    if i==1
        cx=0;                 % Solución (posiblemente EX) en X
    else
        cx=1;ux=i-1;         % Solución SH
    end
    break
end
end
if cx== -5
    cx=0;                     % Solución (posiblemente EX) en X
end
end

```

### Rutina derl infy

```

function [y0,cy,uy]=derl infy(n1,n2,m,p,q,qq,yval)
% Calcula derivadas laterales en Y para p=1 y q=inf.
cy=-5;uy=0;
for i=length(yval):-1:1
    b=0;d=0;
    y0=yval(i);dy=sign(y0-p(2,:))*p(3,1:n1)';
    for j=1:n1
        if p(2,j)==y0
            b=b+p(3,j);
        end
    end
    for j=1:n2
        if qq(2,j)>y0           % No está en Sh, ni en rh
            dy=dy-q(3,j);
        end
        if abs(qq(1,j)-y0/m)<0.001 && abs(qq(2,j)-y0)<0.001 % Está en rh
            d=d+q(3,j);
        end
    end
end
diy=dy-b-d;ddy=dy+b;

```

```

% Poner la condiciones para solución en Y
if diy*ddy<-0.001
    cy=0; % Solución EX en Y
    break
elseif abs(diy)<0.001
    cy=0;uy=i;
    break
elseif ddy>0
    continue
else
    if i==length(yval)
        cy=0; % Solución (Posiblemente EX) en Y
    else
        cy=1;uy=i; % Solución SV
    end
    break
end
end
if cy== -5
    cy=0; % Solución (posiblemente EX) en Y
end

```

#### Rutina dist\_seg\_hinf

```

function dl=dist_seg_hinf(x,m,v,n1,wp,Sv,wv,y0,wh)
xi=v(1,:)-x;yi=v(2,:)-y0;w=v(3,1:n1+wp);
if isempty(Sv)
    dl=0;
else
    S=max(abs([Sv(1,:);Sv(2,:)]-[x;m*x]*ones(1,length(Sv(1,:)))));
    dl=S*Sv(3,:);
end
dl=wh*(x-y0/m)+(abs(xi)+abs(yi))*w'+wv*(m*x-y0)+dl;

```

#### Rutina dist\_seg\_vinf

```

function dl=dist_seg_vinf(y,m,v,n1,wp,Sh,wh,x0,wv)
xi=v(1,:)-x0;yi=v(2,:)-y;w=v(3,1:n1+wp);
if isempty(Sh)
    dl=0;
else
    S=max(abs([Sh(1,:);Sh(2,:)]-[y/m;y]*ones(1,length(Sh(1,:)))));
    dl=S*Sh(3,:);
end
dl=(abs(xi)+abs(yi))*w'+wv*(m*x0-y)+wh*(x0-y/m)+dl;

```

#### Rutina distancialinf

```

function dl=distancialinf(a,m,v,n1,wp,Sh,wh,Sv,wv)
xi=v(1,:)-a(1);yi=v(2,:)-a(2);w=v(3,1:n1+wp);
if isempty(Sv)
    dl=0;
else
    S=max(abs([Sv(1,:);Sv(2,:)]-[a(1);m*a(1)]*ones(1,length(Sv(1,:)))));

```

```

        dlv=S*Sv(3,:)' ;
    end
    if isempty(Sh)
        dlh=0;
    else
        S=max(abs([Sh(1,:);Sh(2,:)]-[a(2)/m;a(2)]*ones(1,length(Sh(1,:)))));
        dlh=S*Sh(3,:)' ;
    end
    end
    d1=wh*abs(a(1)-a(2)/m)+(abs(xi)+abs(yi))*w'+wv*abs(m*a(1)-a(2))+dlv+dlh;

```

## Rutina rectarspq

```

function [xr dr]=rectarspq(m,p,q,r,s,precis)
% Obtine el punto de la recta cuya distancia a p y q es mínima
k2=m/2;k1=3*m/2;cota=(1+m^s)^(1/s);
mq=-m*abs(m)^(-s/(s-1));mp=-m*abs(m)^(-r/(r-1));
if r==1
    g1=10^99;g2=0;
else
    g1=fsolve(@(g) 1+m*g^(r-1)-cota*((1+g^r)^((r-1)/r)),k1,++
        optimset('Display','off','TolFun',1e-20));
    g2=fsolve(@(g) 1+m*g^(r-1)-cota*((1+g^r)^((r-1)/r)),k2,++
        optimset('Display','off','TolFun',1e-20));
end
% g1<g2, es decir, g1 define la zona "horizontal"
qq=q;pp=p;n1=length(p(1,:));n2=length(q(1,:));
for j=1:n2
    qq(1,j)=(q(2,j)-mq*q(1,j))/(m-mq);qq(2,j)=m*qq(1,j);
end
for j=1:n1
    pp(1,j)=(p(2,j)-mp*p(1,j))/(m-mp);pp(2,j)=m*pp(1,j);
end
uu=[pp(1:3,:),qq(1:3,:)];M=medianas(uu);xr=(min(M(1,:))+max(M(1,:)))/2;
for j=1:n1
    if p(2,j)>=m*xr+g2*(p(1,j)-xr)
        pp(1,j)=(p(2,j)-g2*p(1,j))/(m-g2);
    elseif p(2,j)<=m*xr+g1*(p(1,j)-xr)
        pp(1,j)=xr;
    else
        pp(1,j)=(p(2,j)-g1*p(1,j))/(m-g1);
    end
    end
    pp(2,j)=m*pp(1,j);
end
dr=((abs(xr-pp(1,:)).^s+abs(m*xr-pp(2,:)).^s).^^(1/s))*p(3,:)' +
    ((abs(pp(1,:)-p(1,:)).^r+abs(pp(2,:)-p(2,:)).^r).^^(1/r))*p(3,:)' +
    ((abs(xr-q(1,:)).^s+abs(m*xr-q(2,:)).^s).^^(1/s))*q(3,:)' ;
T=1;itg=0;d=[-1 1];e=1;
while T>=precis
    aviso=0;
    for j=1:2
        xj=xr+d(j)*e;
        for h=1:n1
            if p(2,h)>=m*xj+g2*(p(1,h)-xj)
                pp(1,h)=(p(2,h)-g2*p(1,h))/(m-g2);
            elseif p(2,h)<=m*xj+g1*(p(1,h)-xr)
                pp(1,h)=xj;
            else
                aviso=1;
            end
        end
    end
    e=e/2;
end

```

```

                pp(1,h)=(p(2,h)-g1*p(1,h))/(m-g1);
            end
            pp(2,h)=m*pp(1,h);
        end
        dj=((abs(xj-pp(1,:)).^s+abs(m*xj-pp(2,:)).^s).^(1/s))*p(3,:)' +
            ((abs(pp(1,:)-p(1,:)).^r+abs(pp(2,:)-p(2,:)).^r).^(1/r))*p(3,:)' +
            ((abs(xj-q(1,:)).^s+abs(m*xj-q(2,:)).^s).^(1/s))*q(3,:);
        if dj<dr
            xr=xj;T=abs(dj-dr);dr=dj;aviso=1;
            break
        end
    end
    if aviso==0;e=e/2;end
    itg=itg+1;
    if itg>=200;break;end
end

```

### Rutina gates\_pq

```

function [z0 d0]=gates_pq(m,v,r,s)
% Obtiene el gate entre P y Q definidos en v
%v=[p(1,j)=xp xq o bien v=[xp xq=q(1,j)
% p(2,j)=yp yq] yp yq=q(2,j)]
xp=v(1,1);yp=v(2,1); % El punto P que no está en la recta
xq=v(1,2);yq=v(2,2);e=abs(xp-xq)/2;
z0=(xp+xq)/2;d=[-1 1];
d0=((abs(z0-xp)^r+abs(m*z0-yp)^r).^(1/r))+((abs(z0-xq)^s+abs(m*z0-yq)^s).^(1/s));
T=1;TOL=10^(-6);itg=0;
while T>TOL
    itg=itg+1;aviso=0;
    for j=1:2
        z1=z0+d(j)*e;
        d1=(abs(z1-xp)^r+abs(m*z1-yp)^r).^(1/r)+
            (abs(z1-xq)^s+abs(m*z1-yq)^s).^(1/s);
        if d1<d0
            T=abs(d1-d0);d0=d1;z0=z1;
            aviso=1;
            break
        end
    end
    if aviso==0;e=e/2;end
    if itg>=50;break;end
end

```