

TRABAJO FIN DE GRADO

Problemas de localización competitiva. El modelo de Huff.

Presentado por:
María Cristina Molero del Río

Supervisado por:
DR. RAFAEL BLANQUERO BRAVO
DR. EMILIO CARRIZOSA PRIEGO



FACULTAD DE MATEMÁTICAS
Departamento de Estadística e Investigación Operativa
Sevilla, Junio 2016

Índice general

Abstract	5
Introducción	7
1. Modelos de localización competitiva	9
1.1. Reglas de competencia	9
1.2. Patrones de elección	10
1.3. Función de atracción	11
1.4. Características del espacio de decisión	11
1.5. Criterio de decisión	12
2. Modelo continuo	13
2.1. Introducción al problema	13
2.2. Procedimientos de solución	16
2.2.1. Algoritmo de Weiszfeld generalizado	16
2.2.1.1. Explicación del algoritmo y convergencia	16
2.2.1.2. Experiencia computacional	22
2.2.2. Procedimiento de ramificación y acotación	24
2.2.2.1. Algoritmo propuesto	24
2.2.2.2. Cotas superiores	26
3. Modelo discreto	31
3.1. Introducción al problema	31
3.2. Procedimientos de solución	33
3.2.1. Concavidad de la función objetivo	33
3.2.2. Reformulación del problema	35
3.3. Experiencia computacional	37
Anexo I. Modelo continuo	41
Anexo II. Modelo discreto	45
Bibliografía	47

Abstract

The ability of a firm to produce goods and/or services and market them effectively depends largely on the location of its facilities.

Location theory deals with the modelling, formulation and solution of mathematical problems related to sitting facilities in a given space.

In this text, we focus on competitive location models. The root of competitive location models is Hotelling's seminal article 'Stability in Competition' in 1929. Frank Plastria defines a location model as competitive "when it explicitly incorporates the fact that other facilities are already - or will be - present in the market and that the new facility or facilities will have to compete with them for its or their market share".

This work starts with a short introduction to competitive location models, including a brief description of the elements of these models. Next, we address Huff's model in the continuous space, exposing two different techniques to solve this problem, the generalized Weiszfeld algorithm and a standard branch and bound algorithm. Then, we develop a discrete competitive location model. Resolution methods are described. Computational experiments are reported both for the continuous and the discrete model. The remainder of the text contains the codes of R and AMPL used for solving the posed problems.

Introducción

La mayoría de las empresas, tanto del sector privado como del sector público, se ha enfrentado alguna vez al problema de la localización de instalaciones. Las industrias deben determinar la ubicación de las plantas de fabricación y montaje, así como de los almacenes. Los comercios deben colocar sus tiendas. El Gobierno debe localizar oficinas y otros servicios públicos como escuelas, hospitales, estaciones de bomberos, bases de ambulancias, estaciones de inspección de vehículos, vertederos, etc.

En cualquier caso, la capacidad de una empresa para producir y comercializar sus productos de manera efectiva o de una agencia para ofrecer servicios de alta calidad depende, en parte, de la ubicación de sus instalaciones y de la relación con otras instalaciones y con sus clientes. Es aquí cuando surge la teoría de localización.

La teoría de localización se dedica a modelar, formular y resolver problemas matemáticos relacionados con la localización de servicios en un espacio dado.

En este trabajo nos centramos en los modelos de localización competitiva. Éstos comenzaron a desarrollarse en 1929 cuando Hotelling publicó su artículo ‘Stability in Competition’. Parafraseando a Frank Plastria, “un modelo de localización competitiva es aquél que incorpora el hecho de que otras instalaciones ya están o estarán presentes en el mercado y las nuevas tendrán que competir con éstas por la cuota de mercado”.

La estructura de este trabajo se organiza de la siguiente forma. En el *Capítulo 1* introducimos al lector los modelos de localización competitiva, incluyendo una breve descripción de los distintos elementos que componen estos modelos. En el *Capítulo 2*, comenzamos planteando el modelo de Huff en un espacio de localización continuo. Seguidamente, exponemos dos técnicas de resolución a estos problemas entre las que destaca el algoritmo de Weiszfeld generalizado, el cual usaremos para dar solución a un problema real. En el último capítulo, el *Capítulo 3*, desarrollamos un modelo discreto y, de igual forma al capítulo anterior, se presentan dos métodos para la resolución del mismo y se aplica el modelo a unos datos determinados. Por último, en los *Anexos I y II*, se encuentra la implementación de los programas correspondientes a los algoritmos descritos en los *Capítulos 2 y 3*, respectivamente.

Capítulo 1

Modelos de localización competitiva

Un modelo de localización es un modelo que pretende dar solución al problema de localizar nuevas instalaciones o plantas en una determinada área de mercado.

El problema básico que propone un modelo de localización competitiva es la localización óptima de nuevas plantas en un mercado donde existe la competencia. Un nuevo competidor entrará en el mercado y deberá decidir dónde ubicarse estratégicamente para conseguir su máximo beneficio.

En la literatura existe una gran variedad de modelos de localización competitiva. Por ello, cabe estudiar los distintos elementos que pueden componer un modelo.

En este capítulo analizamos brevemente los elementos fundamentales de todo problema de localización competitiva. Ver [9].

1.1. Reglas de competencia

Las plantas a ubicar competirán con otras para capturar el mercado. La manera en la que se producirá la competencia y las posibles formas de reaccionar ésta tras la localización de las nuevas plantas, pueden darse de distintas formas, que revisamos a continuación.

Competencia estática

Esta forma de competencia es la más simple y constituye la base sobre la cual se construyen modelos más complejos.

En ella se presupone que la competencia ya está presente en el mercado, por lo que la localización y las características de la misma se conocen de antemano y están fijadas.

Competencia con previsión

En este caso, la empresa que entra en el mercado conoce la futura apertura de otras plantas competidoras, además de las ya existentes, así que deberá tomar decisiones en vistas a la situación futura que se presentará.

Modelos dinámicos y equilibrio competitivo

Cuando estamos en una dinámica de mercado, lo más probable es que la competencia existente varíe sus estrategias al haberse producido la apertura de nuevas plantas. Esto implicará un cambio en el ambiente competitivo. Esta serie de cambios lleva a modelos dinámicos cuyo objetivo es describir los ciclos de acción/reacción de los distintos competidores, en los que resulta de gran interés encontrar situaciones de equilibrio.

1.2. Patrones de elección

Este elemento tiene en cuenta el comportamiento usual del cliente en el mercado. Distinguimos entre dos alternativas: el patrón determinista y el patrón probabilístico. Las describimos a continuación.

Determinista

Este patrón establece que la demanda completa de cada cliente es servida por la planta que más le atrae.

Una elección determinista no especifica qué ocurre cuando un cliente se siente igual de atraído por varias plantas, incluyendo una nueva, siendo posible considerar diferentes alternativas:

- Dividir la demanda entre las plantas que le atraen.
- Destinar todo a las nuevas plantas.
- Destinar todo a las plantas de la competencia.

Probabilístico

Este comportamiento tiene lugar cuando el cliente reparte su demanda entre las diferentes plantas de acuerdo a unas probabilidades que vienen determinadas, de alguna manera, por la atracción que experimenta hacia ellas.

1.3. Función de atracción

La función de atracción nos da información sobre cómo se ve atraído un cliente por una planta.

Parece razonable asumir que la distancia entre la planta y el cliente influye en gran medida en dicha atracción; de hecho, deben ser inversamente proporcionales. Pero, en muchos casos, no solamente influye esta distancia sino que puede ocurrir que las plantas tengan distintas características y difieran en otros aspectos tales como la superficie de la planta, el precio de los productos o la posibilidad de devolución, entre otras. Estas diferencias serán tenidas en cuenta por los clientes y deberán ser incluidas en esta función de atracción junto con la distancia. Por tanto, la función de atracción tomará la forma $\alpha = \alpha(d_{ij})$ con α decreciente.

En el siguiente capítulo usaremos la formulación de la función de atracción dada por Huff, $\alpha(d_{ij}) = \frac{A_j}{d_{ij}^\lambda}$, siendo A_j la medida de atracción de la planta j , d_{ij} la distancia entre el cliente i y la planta j y λ un determinado escalar positivo. En el último capítulo consideraremos un modelo de Huff “aditivo”, cuya función de atracción admite la forma aditiva $\alpha(d_{ij}) = A_j - \lambda d_{ij}$.

1.4. Características del espacio de decisión

Espacio de localización

Determina la forma en la que se distribuye el área de mercado.

Espacio continuo

Está determinado por un sistema de coordenadas en el que cualquier valor real es admisible.

Podemos considerar desde un problema en el espacio unidimensional —en una recta— hasta un problema en la esfera, pasando por un problema en el plano. Este último es el que más nos interesa.

Será necesario especificar el tipo de distancia que se utilizará. Aunque usualmente se considera la distancia euclídea, también puede considerarse cualquier norma p o norma poliédrica.

Espacio discreto

Hay una lista finita de lugares candidatos y se supone siempre que el mercado consistirá en un conjunto finito de puntos de demanda.

La dificultad en la práctica reside en la gran cantidad de datos que han de ser recogidos: es necesario disponer de la matriz de distancias entre todos los puntos candidatos y los puntos de demanda.

Red

Una red $G = (V, E)$ representa el mercado. V es el conjunto de nodos y E es el conjunto de arcos que conectan cada par de nodos y representan la conexión geográfica entre ellos. Para cada par de nodos (i, j) se representa por d_{ij} la distancia entre ellos. La situación de la demanda y la localización de las plantas, tanto de las existentes como de las nuevas, no sólo podrá encontrarse en los nodos del grafo sino también en sus arcos.

En el caso en el que el conjunto de puntos de demanda y el de plantas sean conjuntos finitos de puntos del grafo, estos problemas pueden considerarse problemas discretos.

Plantas

Es importante conocer el número y la ubicación de las plantas o instalaciones que cada uno de los competidores posee en el mercado, así como el número de nuevos servicios a localizar.

1.5. Criterio de decisión

El objetivo es maximizar la cuota total de mercado capturado por el nuevo competidor; esto es, la suma del volumen de la demanda de todos los clientes que son servidos por las nuevas plantas.

Capítulo 2

Modelo continuo

En este capítulo desarrollaremos un modelo de localización competitiva para el caso continuo basándonos en lo propuesto por Tammy y Zvi Drezner en [5].

2.1. Introducción al problema

Nuestro objetivo es encontrar la mejor ubicación de nuevas plantas que maximicen el beneficio de las mismas en un ambiente donde la competencia ya está presente.

Encuadraremos nuestro problema en un marco de competencia estática y un patrón de elección probabilístico. Concretamente, usaremos el modelo de Huff.

Por simplificar, el modelo se planteará para el caso en el que se va a localizar una sola planta.

Modelo de Huff

El modelo de Huff ([7],[8]) es un modelo probabilístico que plantea que la forma en la que un cliente se ve atraído por una planta determinada es directamente proporcional a una cierta medida de atracción de la planta e inversamente proporcional a una potencia de la distancia cliente-planta, $\alpha(d_{ij}) = \frac{A_j}{d_{ij}^\lambda}$.

Formulación matemática del modelo de Huff

Con el fin de obtener una formulación matemática del mismo definimos lo siguiente:

Parámetros

n	número de puntos de demanda
p	número de plantas existentes
(x_i, y_i)	localización del punto de demanda $i = 1, \dots, n$
$b_i > 0$	número de clientes o demanda disponible en el punto de demanda $i, i = 1, \dots, n$
A_j	medida de atracción de la planta existente $j, j = 1, \dots, p$
(x'_j, y'_j)	localización de la planta existente $j, j = 1, \dots, p$
d_{ij}	distancia euclídea entre el punto de demanda i y la planta existente j para $i = 1, \dots, n, j = 1, \dots, p$
A	medida de atracción de la nueva planta a ubicar
$\lambda > 0$	parámetro de decadencia de la distancia ($\lambda = 2$, modelo gravitacional)

Variables de decisión

$X = (x, y)$	localización de la nueva planta
$d_i(X) = d_i(x, y)$	distancia euclídea entre el punto de demanda i y la nueva planta para $i = 1, \dots, n$

Función objetivo

$M(X)$ cuota de mercado capturado por la nueva planta localizada en X

Por el modelo de Huff, la cuota de mercado, $M(X)$, se plantea como el valor esperado del volumen de demanda servida por esta nueva planta:

$$M(X) = \sum_{i=1}^n b_i \frac{\frac{A}{d_i^\lambda(X)}}{\frac{A}{d_i^\lambda(X)} + \sum_{j=1}^p \frac{A_j}{d_{ij}^\lambda}}$$

Se tiene que $M(X) = \sum_{i=1}^n b_i \frac{\frac{A}{d_i^\lambda(X)}}{\frac{A}{d_i^\lambda(X)} + \sum_{j=1}^p \frac{A_j}{d_{ij}^\lambda}} = \sum_{i=1}^n b_i \frac{1}{1 + \sum_{j=1}^p \frac{A_j}{A} \frac{d_i^\lambda(X)}{d_{ij}^\lambda}}$

Luego, podemos reescribir $M(X)$ como:

$$M(X) = \sum_{i=1}^n b_i \frac{1}{1 + h_i d_i^\lambda(X)} \text{ siendo } h_i = \sum_{j=1}^p \frac{A_j}{A d_{ij}^\lambda}.$$

Dibujemos la gráfica de M .

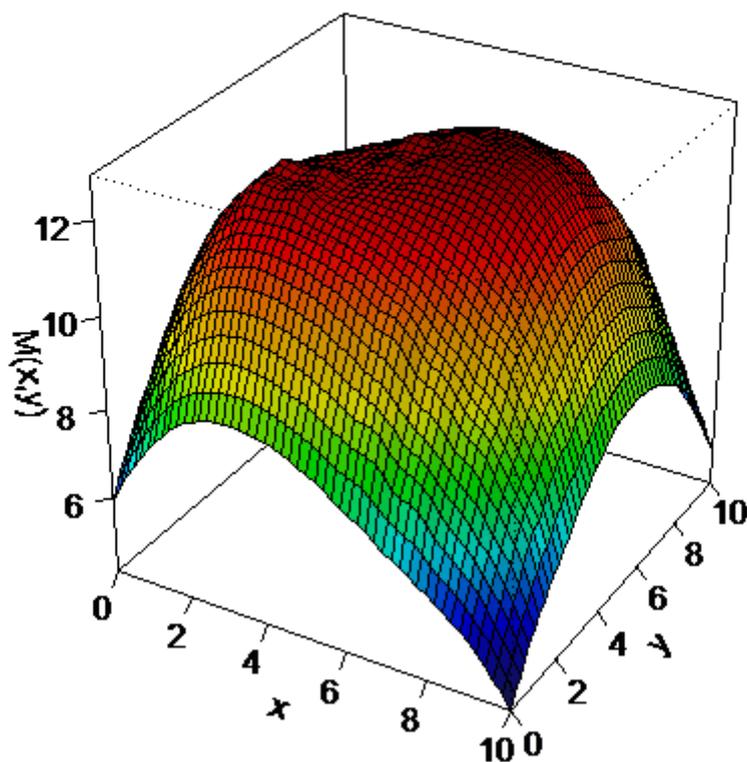


Figura 2.1: Gráfica de M tomando $\lambda = 1 = b_i = h_i \forall i$

El objetivo es localizar en el plano la nueva planta de tal forma que se maximice la cuota de mercado total capturado por dicha planta, M , usando la expresión anterior.

En conclusión, hemos de resolver el siguiente problema de optimización:

$$\max_{X \in \mathbb{R}^2} M(X) \quad (2.1)$$

Para ello, se abordan distintos métodos de solución que son estudiados en la siguiente sección.

2.2. Procedimientos de solución

Para la resolución del problema (2.1) trataremos una generalización del algoritmo de Weiszfeld y un método de ramificación y acotación.

2.2.1. Algoritmo de Weiszfeld generalizado

2.2.1.1. Explicación del algoritmo y convergencia

Analizaremos el algoritmo de Weiszfeld y lo adaptaremos a nuestro problema, probando que el método converge a un máximo local o a un punto de silla.

Forma general del Algoritmo de Weiszfeld

Supongamos que el problema de optimización puede formularse como maximizar o minimizar una función objetivo la cual es suma de términos que son función de la distancia euclídea entre la planta y el punto de demanda:

$$f(X) = f(x, y) = \sum_{i=1}^n f_i(d_i(x, y))$$

Imponiendo las condiciones necesarias de optimalidad en los puntos de diferenciabilidad, $\frac{\partial f}{\partial x} = 0$ y $\frac{\partial f}{\partial y} = 0$, obtenemos:

$$\frac{\partial f}{\partial x} = \sum_{i=1}^n \frac{\partial f_i}{\partial d_i} \frac{\partial d_i}{\partial x} = \sum_{i=1}^n \frac{\partial f_i}{\partial d_i} \frac{2(x - x_i)}{2\sqrt{(x - x_i)^2 + (y - y_i)^2}} = \sum_{i=1}^n \frac{\partial f_i}{\partial d_i} \frac{x - x_i}{d_i(X)} = 0$$

$$\frac{\partial f}{\partial y} = \sum_{i=1}^n \frac{\partial f_i}{\partial d_i} \frac{\partial d_i}{\partial y} = \sum_{i=1}^n \frac{\partial f_i}{\partial d_i} \frac{2(y - y_i)}{2\sqrt{(x - x_i)^2 + (y - y_i)^2}} = \sum_{i=1}^n \frac{\partial f_i}{\partial d_i} \frac{y - y_i}{d_i(X)} = 0$$

Luego:

$$\sum_{i=1}^n \frac{\partial f_i}{\partial d_i} \frac{x - x_i}{d_i(X)} = x \sum_{i=1}^n \frac{\partial f_i}{\partial d_i} \frac{1}{d_i(X)} - \sum_{i=1}^n \frac{\partial f_i}{\partial d_i} \frac{x_i}{d_i(X)} = 0$$

$$\sum_{i=1}^n \frac{\partial f_i}{\partial d_i} \frac{y - y_i}{d_i(X)} = y \sum_{i=1}^n \frac{\partial f_i}{\partial d_i} \frac{1}{d_i(X)} - \sum_{i=1}^n \frac{\partial f_i}{\partial d_i} \frac{y_i}{d_i(X)} = 0$$

Despejando x e y :

$$x = \frac{\sum_{i=1}^n \frac{\partial f_i}{\partial d_i} \frac{x_i}{d_i(X)}}{\sum_{i=1}^n \frac{\partial f_i}{\partial d_i} \frac{1}{d_i(X)}} \quad (2.2)$$

$$y = \frac{\sum_{i=1}^n \frac{\partial f_i}{\partial d_i} \frac{y_i}{d_i(X)}}{\sum_{i=1}^n \frac{\partial f_i}{\partial d_i} \frac{1}{d_i(X)}} \quad (2.3)$$

Estas ecuaciones de punto fijo son las que usaremos de forma recursiva para obtener la solución.

Algoritmo de Weiszfeld generalizado para el problema de Huff

Particularicemos las ecuaciones de punto fijo anteriores, (2.2) y (2.3), para nuestro problema en el que

$$f_i(d_i) = \frac{b_i}{1 + h_i d_i^\lambda}.$$

Como

$$f'_i(d_i) = -\frac{\lambda b_i h_i d_i^{\lambda-1}}{(1 + h_i d_i^\lambda)^2}$$

obtenemos:

$$x = \frac{\sum_{i=1}^n w_i(X) x_i}{\sum_{i=1}^n w_i(X)} \quad (2.4)$$

$$y = \frac{\sum_{i=1}^n w_i(X) y_i}{\sum_{i=1}^n w_i(X)} \quad (2.5)$$

donde

$$w_i(X) = \frac{b_i h_i \lambda d_i^{\lambda-2}(X)}{(1 + h_i d_i^\lambda(X))^2}$$

Convergencia del procedimiento de Weiszfeld generalizado

A continuación, se probará que las iteraciones del algoritmo de Weiszfeld para nuestro modelo convergen a un máximo local o a un punto de silla de la cuota de mercado capturado.

En primer lugar, probamos que el valor de la función objetivo $M(x)$ aumenta en cada iteración del algoritmo de Weiszfeld.

Lema 2.2.1. Dado $X \in \mathbb{R}^2$, definimos la función $F(Z)$ para cualquier $Z \in \mathbb{R}^2$ como:

$$F(Z) = \sum_{i=1}^n \frac{b_i}{1 + h_i d_i^\lambda(X)} \left(2 - \frac{1 + h_i d_i^{\lambda-2}(X) d_i^2(Z)}{1 + h_i d_i^\lambda(X)} \right).$$

Entonces, $F(X) = M(X)$.

Demostración

$$\begin{aligned} F(X) &= \sum_{i=1}^n \frac{b_i}{1 + h_i d_i^\lambda(X)} \left(2 - \frac{1 + h_i d_i^{\lambda-2}(X) d_i^2(X)}{1 + h_i d_i^\lambda(X)} \right) = \\ &= \sum_{i=1}^n \frac{b_i}{1 + h_i d_i^\lambda(X)} \left(2 - \frac{1 + h_i d_i^\lambda(X)}{1 + h_i d_i^\lambda(X)} \right) = \\ &= \sum_{i=1}^n \frac{b_i}{1 + h_i d_i^\lambda(X)} (2 - 1) = \sum_{i=1}^n \frac{b_i}{1 + h_i d_i^\lambda(X)} = M(X) \end{aligned}$$

□

Lema 2.2.2. Sean X y $T(X)$ los puntos obtenidos en las iteraciones del algoritmo de Weiszfeld actual y siguiente, respectivamente. Entonces, el máximo de $F(Z)$ se alcanza para $Z = T(X)$.

Demostración

En primer lugar, veremos que $\max_Z F(Z)$ es equivalente a $\min_Z \sum_{i=1}^n w_i(X) d_i^2(Z)$ para un X dado.

$$\begin{aligned} \max_Z F(Z) &= \max_Z \sum_{i=1}^n \frac{b_i}{1 + h_i d_i^\lambda(X)} \left(2 - \frac{1 + h_i d_i^{\lambda-2}(X) d_i^2(Z)}{1 + h_i d_i^\lambda(X)} \right) \\ &= \max_Z \left\{ \sum_{i=1}^n \frac{2b_i}{1 + h_i d_i^\lambda(X)} - \sum_{i=1}^n \frac{b_i}{(1 + h_i d_i^\lambda(X))^2} - \sum_{i=1}^n \frac{b_i h_i d_i^{\lambda-2}(X)}{(1 + h_i d_i^\lambda(X))^2} d_i^2(Z) \right\} \\ &= \max_Z \left\{ \sum_{i=1}^n \frac{2b_i + 2b_i h_i d_i^\lambda(X) - b_i}{(1 + h_i d_i^\lambda(X))^2} - \sum_{i=1}^n \frac{b_i h_i d_i^{\lambda-2}(X)}{(1 + h_i d_i^\lambda(X))^2} d_i^2(Z) \right\} \\ &= \max_Z \left\{ \sum_{i=1}^n \frac{b_i (1 + 2h_i d_i^\lambda(X))}{(1 + h_i d_i^\lambda(X))^2} - \sum_{i=1}^n \frac{b_i h_i d_i^{\lambda-2}(X)}{(1 + h_i d_i^\lambda(X))^2} d_i^2(Z) \right\} \\ &= \min_Z \sum_{i=1}^n \frac{b_i h_i d_i^{\lambda-2}(X)}{(1 + h_i d_i^\lambda(X))^2} d_i^2(Z) = \min_Z \sum_{i=1}^n w_i(X) d_i^2(Z) \end{aligned}$$

Hemos convertido nuestro problema en uno bien conocido, como es el problema de Weber con distancias al cuadrado, cuya única solución óptima es el centro de gravedad que viene dado justamente por las ecuaciones (2.4) y (2.5).

Por tanto, el máximo de $F(Z)$ para un X dado está determinado por las ecuaciones (2.4) y (2.5), que se corresponden con la siguiente iteración del algoritmo de Weiszfeld. Así, se tiene que el máximo de $F(Z)$ se alcanza para $Z = T(X)$.

□

Teorema 2.2.1. *Sea X el punto obtenido en la iteración actual del algoritmo de Weiszfeld. Entonces $M(T(X)) \geq M(X)$ y la desigualdad estricta se tiene cuando $T(X) \neq X$.*

Demostración

Según los lemas 2.2.1 y 2.2.2:

$$M(T(X)) = \max_Z \{F(Z)\} \geq F(X) = M(X)$$

dándose la igualdad sólo si $T(X) = X$, ya que el óptimo del problema de Weber con las distancias al cuadrado es único.

□

Corolario 2.2.1. *Si $M(T(X)) = M(X)$, entonces $T(X) = X$.*

Demostración

Probar el enunciado sería equivalente a estudiar qué ocurre cuando $T(X) \neq X$. Por el Teorema 2.2.1 se tendría que $M(T(X)) > M(X)$ y, con ello, que $M(T(X)) \neq M(X)$, como queríamos probar.

□

Lema 2.2.3. *Sea $\{X_k\}_{k=1}^{\infty}$ la sucesión de puntos proporcionada por el algoritmo de Weiszfeld. Entonces el límite $\lim_{k \rightarrow \infty} \{M(X_k)\}$ existe.*

Demostración

La existencia surge directamente del hecho de ser $M(X_k)$ una función monótona creciente acotada superiormente por el óptimo global.

□

Corolario 2.2.2.

$$\lim_{k \rightarrow \infty} \{M(X_{k+1}) - M(X_k)\} = 0$$

Lema 2.2.4. *La sucesión $\{X_k\}_{k=1}^{\infty}$ está en el cierre convexo de los puntos de demanda.*

Demostración

Cada término de la sucesión es una combinación convexa de los puntos de demanda.

□

Lema 2.2.5. *Si $T(X) = X$, entonces X es un punto estacionario de la función M .*

Demostración

Un punto estacionario de una función es aquél que anula su gradiente. Por tanto, si $T(X) = X$, por definición de T , las derivadas parciales primeras se anulan en X y ya se tiene.

□

Lema 2.2.6. *La función $T(X)$ es continua.*

Demostración

Trivial por cómo están definidas x e y en (2.4) y (2.5), respectivamente.

□

Teorema 2.2.2. *La sucesión $\{X_k\}_{k=1}^{\infty}$ converge a un máximo local de $M(X)$ o a un punto de silla de $M(X)$.*

Demostración

Por el Lema 2.2.4, la sucesión $\{X_k\}_{k=1}^{\infty}$ está acotada. Por tanto, usando el teorema de Bolzano-Weierstrass, existe una subsucesión $\{X_{k_i}\}_{i=1}^{\infty}$ convergente.

Sea \bar{X} dicho punto límite. Por el Corolario 2.2.2, $M(T(\bar{X})) = M(\bar{X})$; entonces, según el Corolario 2.2.1, $T(\bar{X}) = \bar{X}$.

Como dijimos en el Lema 2.2.6, T es continua; por tanto, conforme X_k se aproxime a \bar{X} , X_{k+1} lo hará hacia $T(\bar{X}) = \bar{X}$.

Obtenemos así la convergencia de la sucesión X_k a \bar{X} , que sabemos que es un punto estacionario por el Lema 2.2.5.

Sin embargo, la sucesión $M(X_k)$ es monótona creciente por el Teorema 2.2.1; así que \bar{X} ha de ser un máximo local o un punto de silla.

□

Pseudocódigo

Hemos probado que, partiendo de un punto inicial, en cada iteración, el algoritmo de Weiszfeld se va acercando a un máximo local o a un punto de silla. Si nuestra función fuera cóncava, ya tendríamos la solución óptima. Sin embargo, en general, no lo es y puede presentar más de un máximo local. Por tanto, acercarse a uno de ellos no nos garantiza la obtención de la solución que buscamos.

Una posible solución a este problema consiste en aplicar la técnica de multiarranque, es decir, repetir la búsqueda de Weiszfeld partiendo de diversos puntos iniciales que convergerán a diferentes máximos locales y, de entre ellos, quedarnos con aquél que haga máxima la función objetivo. Es así como hemos programado el algoritmo de Weiszfeld generalizado, tal y como se describe a continuación.

Fase 1: INICIALIZACIÓN

- Fijar la precisión requerida $\varepsilon > 0$.
- Seleccionar aleatoriamente r puntos pertenecientes a la envolvente convexa de los puntos de demanda.

Fase 2: ALGORITMO WEISZFELD

Para cada p_j , con $j = 1, \dots, r$, repetir:

1. $k := 0$

$$2. x_k := p_j$$

$$3. x_{k+1} := \left(\frac{\sum_{i=1}^n w_i(x_k)x_i}{\sum_{i=1}^n w_i(x_k)}, \frac{\sum_{i=1}^n w_i(x_k)y_i}{\sum_{i=1}^n w_i(x_k)} \right)$$

$$4. \text{ Si } |M(x_{k+1}) - M(x_k)| \leq \varepsilon$$

entonces STOP y almacenar $x_j := x_{k+1}$ y $M_j := M(x_{k+1})$;

en otro caso,

hacer $k := k + 1$ y volver a 3.

Fase 3: SOLUCIÓN

$$z_{opt} := \max_j \{M_j\}$$

$$x_{opt} := \{x_j : M_j = z_{opt}\}$$

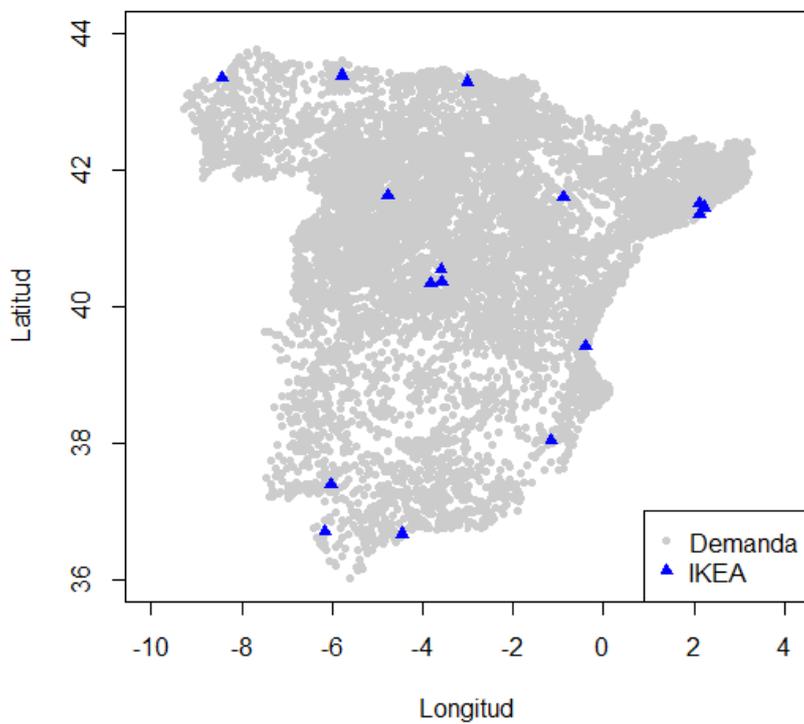
2.2.1.2. Experiencia computacional

En esta apartado presentaremos un problema con unos datos determinados y lo resolveremos con el algoritmo propuesto. Para ello, programaremos el algoritmo de Weiszfeld generalizado en el software R. El código del mismo se encuentra en el *Anexo I*.

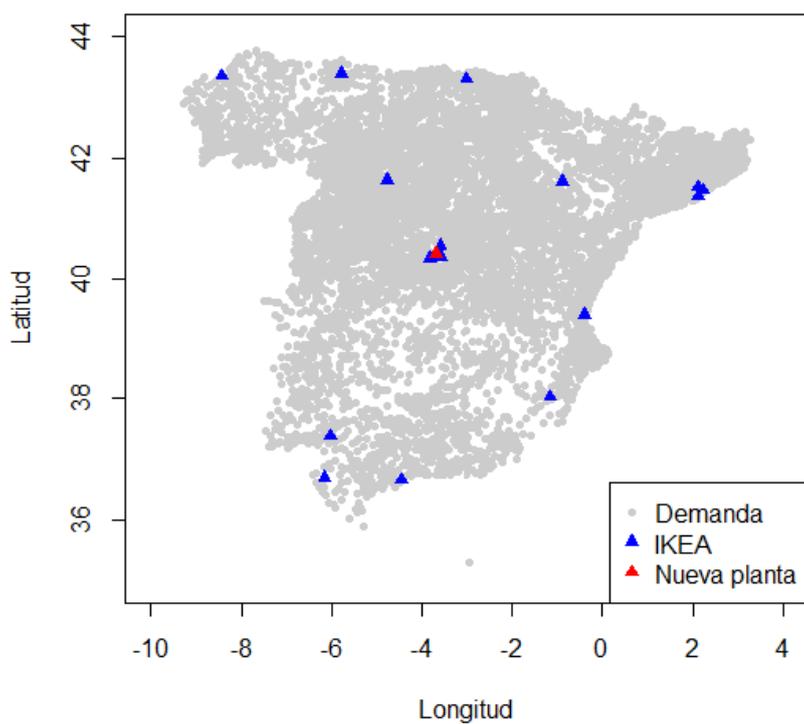
Planteamiento del problema

Supongamos que, en España, una nueva empresa quiere entrar en el mercado mobiliario. Para ello, tendrá que competir con la cadena de muebles IKEA. Usando *Google maps* hemos geolocalizado los dieciséis IKEA's en España ($p = 16$). Además, hemos usado una base de datos de *Business Intelligence* para obtener las coordenadas geográficas de todos los municipios de España ($n = 7955$) junto con el número de habitantes de cada municipio ($b_i, i = 1, \dots, n$). Teniendo en cuenta que IKEA lleva tiempo en el mercado, vamos a considerar que cada una de sus plantas tiene atracción la unidad ($A_j = 1, j = 1, \dots, p$) y que la nueva planta competidora tiene justo la mitad ($A = 0,5$). Por último, tomaremos $\lambda = 2$, es decir, plantearemos el modelo de Huff gravitacional.

El estado actual del mercado es el siguiente:



Solución del problema



Weiszfeld	
Localización	$(-3.700739, 40.415987)$
Cuota de mercado	4182210

Tabla 2.2: Solución óptima del problema planteado.

Notemos que hemos tomado $\varepsilon = 0,0001$ y $r = 50$ para el algoritmo de Weiszfeld generalizado.

2.2.2. Procedimiento de ramificación y acotación

La segunda técnica que se presenta para resolver el problema de Huff es un algoritmo de ramificación y acotación. Consiste en la triangulación del espacio de decisión y el consiguiente establecimiento de cotas superiores e inferiores para la función objetivo en cada triángulo.

Para conseguir una triangulación del espacio de decisión tomamos como referencia los puntos de demanda y obtenemos su diagrama de Voronoi. El grafo dual del mismo constituirá la triangulación, llamada triangulación de Delauney. En ella, los vértices de los triángulos son los puntos de demanda.

Una cota inferior para el valor máximo posible de la función objetivo en un triángulo es el valor de la función objetivo en cualquier punto del triángulo. Se propone usar un punto conocido del triángulo, como es su centro de gravedad.

Al estar maximizando, el estudio de las cotas superiores de la función objetivo en cada triángulo requiere más detalle que el de las cotas inferiores. Es por ello que nos detendremos en ellas más tarde.

2.2.2.1. Algoritmo propuesto

A continuación, se muestra una breve descripción del algoritmo de ramificación y acotación. Éste ha sido una adaptación del que aparece propuesto para un problema de minimización en [4].

Fase 1: INICIALIZACIÓN

- Triangular el espacio de decisión, τ .
- Establecer una precisión $\varepsilon > 0$.
- Inicializar la lista de "triángulos restantes", $\Lambda = \emptyset$.
- Inicializar $LB = -\infty$.
- Para cada $t \in \tau$ hacer:
 - (a) Evaluar la función objetivo en su centro de gravedad. Si este valor es mayor que LB , entonces actualizar LB y almacenar en x_{LB} dicho punto.
 - (b) Calcular una cota superior del triángulo t , $UB(t)$.
 - (c) Si $UB(t) \geq (1 + \varepsilon)LB$, insertar t en Λ .

Fase 2: PROCEDIMIENTO DE RAMIFICACIÓN Y ACOTACIÓN

Repetir hasta que se satisfaga la condición de parada.

1. Seleccionar de Λ el triángulo t_{max} cuya cota superior sea la máxima y eliminarlo de Λ .
2. Si $UB(t_{max}) \leq (1 + \varepsilon)LB$, STOP.
3. Dividir t_{max} por su centro de gravedad en tres subtriángulos t_{max}^1 , t_{max}^2 y t_{max}^3 .
4. Calcular una cota inferior para cada uno de los tres subtriángulos y si alguno de estos valores es mayor que LB , actualizar LB .
5. Calcular una cota superior de la función objetivo para cada subtriángulo t_{max}^i $i = 1, 2, 3$ y si $UB(t_{max}^i) \geq (1 + \varepsilon)LB$, insertar t_{max}^i en Λ .
6. Si LB ha sido actualizado en esta iteración, descartar los triángulos de Λ cuyo UB sea menor que LB .

Fase 3: SOLUCIÓN

$$z_{opt} := UB$$

$$x_{opt} := x_{UB}$$

2.2.2.2. Cotas superiores

En este apartado, recopilamos la construcción de tres cotas superiores distintas para nuestra función objetivo. La primera de ellas la propone Drezner en [5]. Por otro lado, podemos usar la teoría que se conoce sobre Optimización *DC*, como plantean Drezner en [6] y Blanquero y Carrizosa en [3]. Es así como construimos las dos últimas cotas.

UB_1

La base de esta cota consiste en encontrar la mínima distancia entre un punto exterior a un triángulo y todos los puntos del mismo. Parece razonable admitir que esta mínima distancia será hacia un punto del borde del triángulo. Busquémola.

Fijemos, en primer lugar, uno de los tres lados del triángulo.

Sean (x_1, y_1) y (x_2, y_2) los dos vértices de dicho lado y sea (a, b) un punto exterior al triángulo.

Cualquier punto que pase por la recta que une (x_1, y_1) con (x_2, y_2) puede expresarse de la forma $(\alpha x_1 + (1 - \alpha)x_2, \alpha y_1 + (1 - \alpha)y_2)$.

Si $\alpha \in [0, 1]$, el punto pertenece al lado del triángulo; en otro caso, quedaría fuera del triángulo.

Se tiene que la distancia al cuadrado entre el punto (a, b) y un punto cualquiera de la recta es:

$$\begin{aligned} & [\alpha x_1 + (1 - \alpha)x_2 - a]^2 + [\alpha y_1 + (1 - \alpha)y_2 - b]^2 = \\ & = [\alpha(x_1 - x_2) + (x_2 - a)]^2 + [\alpha(y_1 - y_2) + (y_2 - b)]^2 = g(\alpha) \end{aligned}$$

Derivando g con respecto a α , llegamos a que:

$$\alpha^* = \frac{(a - x_2)(x_1 - x_2) + (b - y_2)(y_1 - y_2)}{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Si $\alpha^* \in [0, 1]$, la mínima distancia es hacia el punto dado por ese α^* ; en caso contrario, la distancia mínima se obtiene en uno de los dos vértices del lado que habíamos fijado.

Se obtiene así un cómodo procedimiento para encontrar la mínima distancia, δ_i , al triángulo desde un punto exterior a él:

1. Calcular la distancia hacia todos los vértices del triángulo y llamar δ_i a la más pequeña.
2. Para cada tres parejas de vértices:

Calcular α^* . Si $\alpha^* \in [0, 1]$, calcular la distancia hacia el punto definido por α^* ; y, si es menor que δ_i , actualizar δ_i con el valor obtenido.

Con esto, una cota superior para todos los puntos en un triángulo se calcularía de la forma:

$$UB_1 = \sum_{i=1}^n b_i \frac{1}{1 + h_i \delta_i^\lambda}$$

Para obtener cotas superiores alternativas a UB_1 , escribiremos la función objetivo como

$$M(d) = \sum_{i=1}^n \varphi_i(d) \quad \text{siendo} \quad \varphi_i(d) = \frac{b_i}{1 + h_i d^\lambda}, \quad \text{con} \quad b_i, h_i \geq 0 \text{ y } \lambda \geq 1$$

En primer lugar, se busca una descomposición *dcm* de cada $\varphi_i(d)$.

Definición 2.2.1. *Dado un intervalo no degenerado $K \in \mathbb{R}$, una función $\varphi : K \rightarrow \mathbb{R}$ se denomina diferencia de convexas monótonas (dcm) en K si existen dos funciones $\varphi_i^1, \varphi_i^2 : K \rightarrow \mathbb{R}$ convexas y monótonas en K tales que $\varphi_i(d) = \varphi_i^1(d) - \varphi_i^2(d)$.*

Haciendo uso de las propiedades de la clase de funciones definidas anteriormente, el siguiente paso sería encontrar una mayorante convexa de φ_i^1 y una minorante cóncava de φ_i^2 , que denotamos u_i y l_i , respectivamente. La suma de funciones convexas es convexa, por lo que tendríamos que $u_i - l_i$ es una mayorante convexa de φ_i y, a su vez, $\sum_i (u_i - l_i)$ sería una mayorante convexa de M .

Por tanto:

$$\max_{X \in t} M(X) \leq \max_{X \in t} \sum_i (u_i - l_i) = \max_{v_1, v_2, v_3 \in t} \sum_i (u_i - l_i), \text{ siendo } t \text{ un triángulo cualquiera de la partición } \tau \text{ y } v_1, v_2 \text{ y } v_3 \text{ los vértices de } t.$$

La última igualdad se tiene ya que, por un resultado básico de Programación Matemática, una función convexa en un conjunto convexo que, en nuestro caso, es un triángulo, alcanza sus máximos relativos en los extremos de dicho conjunto.

Concluimos que este procedimiento es bastante útil ya que para obtener una cota superior de M dentro de un triángulo t cualquiera bastaría evaluar la función mayorante convexa obtenida en cada uno de los vértices del triángulo.

Daremos dos descomposiciones *dcm* diferentes para φ_i , que proporcionarán distintas cotas, UB_2 y UB_3 .

UB_2

Derivando dos veces φ_i obtenemos que $\frac{\partial^2 \varphi_i}{\partial^2 d} \geq -b_i \lambda (\lambda - 1) h_i d^{\lambda-2}$ por lo que, integrando dos veces, se obtiene que $\varphi_i(d) + b_i h_i d^\lambda$ es una función convexa y, además, creciente.

Por tanto, podríamos dar la siguiente descomposición *dcm* de $\varphi_i(d)$ en \mathbb{R}_+ :

$$\begin{aligned}\varphi_i^1(d) &= \varphi_i(d) + b_i h_i d^\lambda \\ \varphi_i^2(d) &= b_i h_i d^\lambda\end{aligned}$$

UB_3

Una descomposición *dcm* alternativa de φ_i es la siguiente.

Por el estudio de nuestra función $\varphi_i(d)$, sabemos que posee un único punto de silla, $\bar{d}_i = +\sqrt[\lambda]{\frac{\lambda-1}{\lambda+1} \frac{1}{h_i}}$. A la izquierda de \bar{d}_i , la función es cóncava y, a la derecha, convexa. Considerando la recta tangente a $\varphi_i(d)$ que pase por \bar{d}_i , obtendremos una descomposición *dcm* de φ_i definiendo a trozos como sigue:

$$\begin{aligned}\varphi_i^1 &= \begin{cases} -\varphi_i(\bar{d}) - \varphi'(\bar{d})(d - \bar{d}) & d < \bar{d} \\ 0 & d \geq \bar{d} \end{cases} \\ \varphi_i^2 &= \begin{cases} -\varphi_i(\bar{d}) - \varphi'(\bar{d})(d - \bar{d}) - \varphi(d) & d < \bar{d} \\ -\varphi_i & d \geq \bar{d} \end{cases}\end{aligned}$$

A continuación, representamos φ_i junto a su recta tangente en el punto de inflexión para comprender mejor la construcción de esta última descomposición *dcm*.

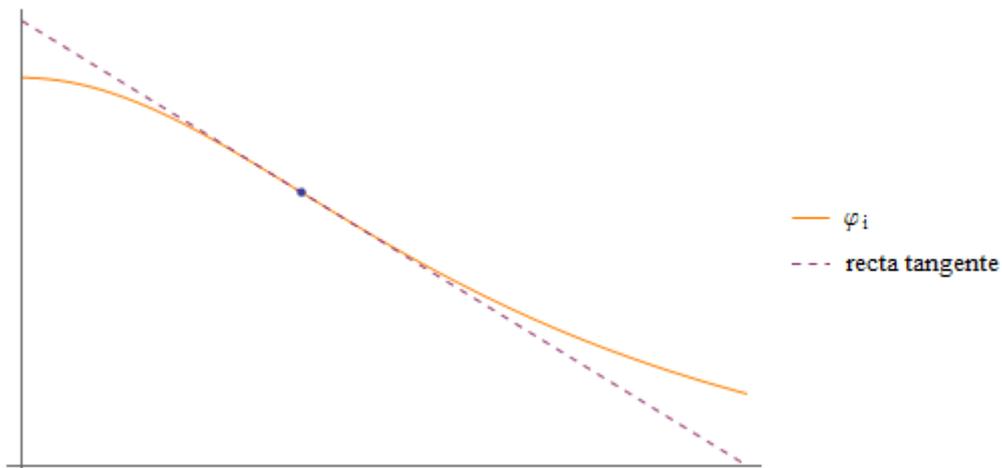


Tabla 2.3: Gráfica de φ_i tomando $\lambda = 1 = b_i = h_i = 1 \forall i$

Para ambas descomposiciones *dcm* se obtiene de forma similar una mayorante convexa y una minorante cóncava de φ_i^1 y φ_i^2 , respectivamente.

Por un lado, como $\varphi_i^1(d)$ es no decreciente, φ_i^1 es la composición de una función no decreciente y una función convexa, como es la distancia; luego, φ_i^1 es también una función convexa. Por tanto, la propia φ_i^1 se toma como una mayorante convexa de ella misma, es decir, $u_i(d) = \varphi_i^1(d)$.

Por otro lado, $\varphi_i^2(d)$ es también una función no decreciente por lo que, a su vez, φ_i^2 es convexa por ser composición de una función no decreciente y una convexa. Sea $x_0 \in t$, para cualquier p_i subgradiente de φ_i^2 en $d_0 = d_i(x_0) \in \mathbb{R}_+$ se tiene que $\varphi_i^2(d) \geq \varphi_i^2(d_0) + p_i(d - d_0)$. Esta función escrita en la parte derecha de la desigualdad constituiría una minorante cóncava de φ_i^2 , es decir, $l_i(d) = \varphi_i^2(d_0) + p_i(d - d_0)$.

Capítulo 3

Modelo discreto

Dedicaremos este capítulo al estudio de un modelo de localización competitiva discreto. Tomaremos como base lo descrito por Benati y Hansen en [2].

3.1. Introducción al problema

En este capítulo volveremos a desarrollar un problema de localización competitiva estática con un patrón de elección probabilístico. La diferencia con el capítulo anterior reside en que, ahora, el espacio de localización es discreto.

Sea $G = (V, E)$ un grafo completo no dirigido que representa el área de mercado. V , el conjunto de nodos, son los únicos puntos del grafo donde pueden localizarse la demanda y las plantas, tanto existentes como nuevas. Las p plantas existentes se localizan en el conjunto $V_p \subset V$ y cada una tiene una medida de atracción asociada A_j . Para cada punto de demanda $i \in V$ también se define una cantidad b_i que indica el número de clientes o la demanda disponible en i . E , el conjunto de aristas, representa las conexiones geográficas entre los distintos nodos sobre las cuales se define una distancia. Además, se definirá el escalar positivo λ que medirá el grado de participación de la distancia.

Supongamos que un nuevo competidor Y quiere entrar en el mercado y establecer r nuevas plantas. El objetivo será maximizar la cuota de mercado capturada por el nuevo competidor Y , es decir, maximizar los beneficios de las nuevas plantas, escogiendo un conjunto $Y_r \subset V$ de r nodos donde se situarán las r plantas.

Como ya adelantamos en el capítulo introductorio, la función de atracción para el modelo de Huff aditivo es $\alpha(d_{ij}) = A_j - \lambda d_{ij}$.

En [2] se obtiene que, entonces, la probabilidad de que un cliente i acuda a una planta j es $p_{ij} = \frac{\exp(A_j - \lambda d_{ij})}{\sum_{k \in V'} \exp(A_k - \lambda d_{ik})}$ siendo $V' \subset V$ el conjunto de plantas localizadas.

Por tanto, dado un conjunto Y_r , la cuota de mercado capturado por Y_r , $M(Y_r)$, se plantea como el valor esperado del volumen de demanda servida por las nuevas plantas, es decir:

$$M(Y_r) = \sum_i b_i \frac{\sum_{j \in Y_r} \exp(A_j - \lambda d_{ij})}{\sum_{j \in Y_r} \exp(A_j - \lambda d_{ij}) + \sum_{j \in V_p} \exp(A_j - \lambda d_{ij})}$$

Podemos reescribir $M(Y_r)$ como:

$$M(Y_r) = \sum_i b_i \frac{\sum_{j \in Y_r} a_{ij}}{\sum_{j \in Y_r} a_{ij} + h_i},$$

siendo $a_{ij} = \exp(A_j - \lambda d_{ij})$ y $h_i = \sum_{j \in V_p} \exp(A_j - \lambda d_{ij})$.

El objetivo es localizar en el grafo G las r nuevas plantas de tal forma que se maximice el beneficio de las mismas.

Así pues, definiendo las variables de decisión:

$$x_j = \begin{cases} 1 & \text{si abrimos una nueva planta en el nodo } j \in V \\ 0 & \text{c.c} \end{cases}$$

nos planteamos resolver el siguiente problema de programación fraccional lineal entera:

$$(P) \quad \begin{aligned} & \text{máx} \sum_{i \in V} b_i \frac{\sum_{j \in V} a_{ij} x_j}{\sum_{j \in V} a_{ij} x_j + h_i} \\ & \text{s.a} \sum_j x_j = r \\ & x_j \in \{0, 1\} \forall j \in V \end{aligned}$$

En la siguiente sección, veremos que este problema puede resolverse mediante diferentes métodos, aprovechando alguna propiedad de la función objetivo.

3.2. Procedimientos de solución

(P) es un problema NP-hard como puede verse en [1].

En esta sección discutiremos dos formas de resolver (P). La primera de ellas será plantear un algoritmo de ramificación y acotación en el que destaca la concavidad de la relajación continua de la función objetivo de (P); la segunda, reformular nuestro problema como un problema de programación lineal entera mixto.

3.2.1. Concavidad de la función objetivo

Teorema 3.2.1. *La relajación continua de la función objetivo de (P) es cóncava.*

Demostración

Sea M_r la relajación continua de la función objetivo M , es decir, admitimos valores continuos para las variables:

$$M_r : \mathbb{R}_+^n \rightarrow \mathbb{R}$$

$$(x_1, \dots, x_n) \mapsto \sum_{i \in V} b_i \frac{\sum_{j \in V} a_{ij} x_j}{\sum_{j \in V} a_{ij} x_j + h_i},$$

siendo n en número de nodos de V .

Veamos que M_r es cóncava.

Fijado $h_i > 0$, definimos la función

$$\psi_i : \mathbb{R}_+ \rightarrow \mathbb{R}$$

$$t \mapsto \frac{t}{t + h_i}$$

la cual es cóncava pues $\psi_i \in C^\infty(\mathbb{R}_+)$ y su segunda derivada es $\psi_i'' = -\frac{2h_i}{(t + h_i)^3} < 0$.

Además, sabemos que la composición de una función cóncava con una función lineal es cóncava siempre y cuando pueda darse tal composición. En nuestro caso, la

función

$$\begin{aligned} \gamma_i: \mathbb{R}_+^n &\rightarrow \mathbb{R} \\ (x_1, \dots, x_n) &\mapsto \frac{\sum_{j \in V} a_{ij} x_j}{\sum_{j \in V} a_{ij} x_j + h_i} = (\psi_i \circ g)(x_1, \dots, x_n) \end{aligned}$$

es cóncava siendo $g(x_1, \dots, x_n) = \sum_{j \in V} a_{ij} x_j$ lineal.

Por último, vemos que M_r no es más que una combinación lineal con coeficientes positivos de funciones cóncavas:

$$\begin{aligned} M_r: \mathbb{R}_+^n &\rightarrow \mathbb{R} \\ (x_1, \dots, x_n) &\mapsto \sum_{i \in V} b_i \frac{\sum_{j \in V} a_{ij} x_j}{\sum_{j \in V} a_{ij} x_j + h_i} = \sum_{i \in V} b_i \gamma_i(x_1, \dots, x_n) \text{ con } b_i > 0 \end{aligned}$$

Deducimos finalmente que M_r es cóncava en \mathbb{R}_+ .

□

Notemos que, siendo M_r continua y la relajación continua del dominio un conjunto convexo, el valor óptimo del problema relajado podrá ser calculado mediante métodos de gradiente, y constituirá una cota superior del valor óptimo del problema entero.

Este resultado podrá ser explotado en un algoritmo de ramificación y acotación como el que se describe a continuación.

Algoritmo de ramificación y acotación

Este algoritmo de ramificación y acotación consiste en una sucesión de problemas relajados a los que se les va añadiendo nuevas restricciones sobre las variables de decisión de tal forma que en algún paso se consigan soluciones enteras. Estas nuevas restricciones mantendrán la convexidad del conjunto factible de soluciones y, por tanto, los problemas que surgen podrán resolverse fácilmente.

Fase 1: INICIALIZACIÓN

- Resolver (P_r) , la relajación continua de (P) .
 - a) Si la solución óptima cumple las restricciones de integridad, STOP.

- b) En otro caso, inicializar la cota inferior para el valor objetivo óptimo de (P) , $z_I = -\infty$.

Fase 2: RAMIFICACIÓN

1. Seleccionar un subproblema no sondeado arbitrariamente, inicialmente (P_r) .
2. Elegir x_k una componente cualquiera no entera de la solución óptima del subproblema seleccionado.
3. Ramificar el subproblema actual en dos nuevos subproblemas, obtenidos al añadir las restricciones $x_k = 0$ y $x_k = 1$.

Fase 3: ACOTACIÓN

- Para cada nuevo subproblema, determinar una cota superior z_S para el valor objetivo óptimo de (P) .

Fase 4: SONDEO

- Analizar los subproblemas que puedan proporcionar la solución óptima de (P) y marcar como terminales aquéllos que:
 - a) Son infactibles.
 - b) $z_S \leq z_I$.
 - c) z_S se alcanza en un punto factible para (P) y $z_S > z_I$. En este último caso, hacer $z_I = z_S$ y almacenar la solución óptima.

Fase 5: CONVERGENCIA Y SOLUCIÓN

- Si todos los subproblemas son terminales, STOP. La solución óptima es el último punto obtenido en c) en la *Fase 4*, con $z_{opt} = z_I$.

3.2.2. Reformulación del problema

El propósito de esta subsección es reformular nuestro problema de programación fraccional lineal entera en un problema programación lineal entera mixto, para su consiguiente implementación en un optimizador de números enteros.

El primer paso es eliminar la parte fraccionaria de la función objetivo. Para ello, introducimos las variables y_i , definidas como:

$$y_i = \frac{\sum_{j \in V} a_{ij} x_j}{\sum_{j \in V} a_{ij} x_j + h_i} \in [0, 1]$$

Por ser el denominador una cantidad positiva, podemos escribir:

$$\left(\sum_{j \in V} a_{ij} x_j + h_i \right) y_i = \sum_{j \in V} a_{ij} x_j$$

y, por tanto,

$$\sum_{j \in V} a_{ij} y_i x_j + h_i y_i = \sum_{j \in V} a_{ij} x_j.$$

Introduciendo a continuación las variables $w_{ij} = y_i x_j$ y teniendo en cuenta que $h_i > 0 \forall i$, podemos despejar y_i de tal forma que nos quede:

$$y_i = \frac{1}{h_i} \left(\sum_{j \in V} a_{ij} x_j - \sum_{j \in V} a_{ij} w_{ij} \right)$$

Para conseguir un problema lineal, no podemos añadir como restricción $w_{ij} = y_i x_j$, así que la escribiremos de manera equivalente con el siguiente conjunto de restricciones:

$$\begin{cases} w_{ij} \leq x_j \\ w_{ij} \geq y_i - (1 - x_j) \\ w_{ij} \leq y_i \end{cases}$$

Por tanto, resolver (P) es equivalente a resolver:

$$\begin{aligned}
 (P_1) \quad & \text{máx} \quad \sum_{j \in V} \left(\sum_{i \in V} b_i \frac{a_{ij}}{h_i} \right) x_j - \sum_{i \in V} \sum_{j \in V} b_i \frac{a_{ij}}{h_i} w_{ij} \\
 & \text{s.a} \quad \sum_{j \in V} x_j = r \\
 & w_{ij} \leq x_j \quad \forall i, j \in V \\
 & w_{ij} \geq \sum_k \frac{a_{ik}}{h_i} x_k - \sum_k \frac{a_{ik}}{h_i} w_{ik} - (1 - x_j) \quad \forall i, j \in V \\
 & w_{ij} \leq \sum_k \frac{a_{ik}}{h_i} x_k - \sum_k \frac{a_{ik}}{h_i} w_{ik} \quad \forall i, j \in V \\
 & w_{ij} \in \mathbb{R}^+ \quad \forall i, j \in V \\
 & x_j \in \{0, 1\} \quad \forall j \in V
 \end{aligned}$$

Por tanto, ya tenemos la transformación de (P) en un problema de programación lineal entera mixto.

3.3. Experiencia computacional

Esta sección está dedicada a la presentación de un problema de localización discreto y a su consiguiente resolución.

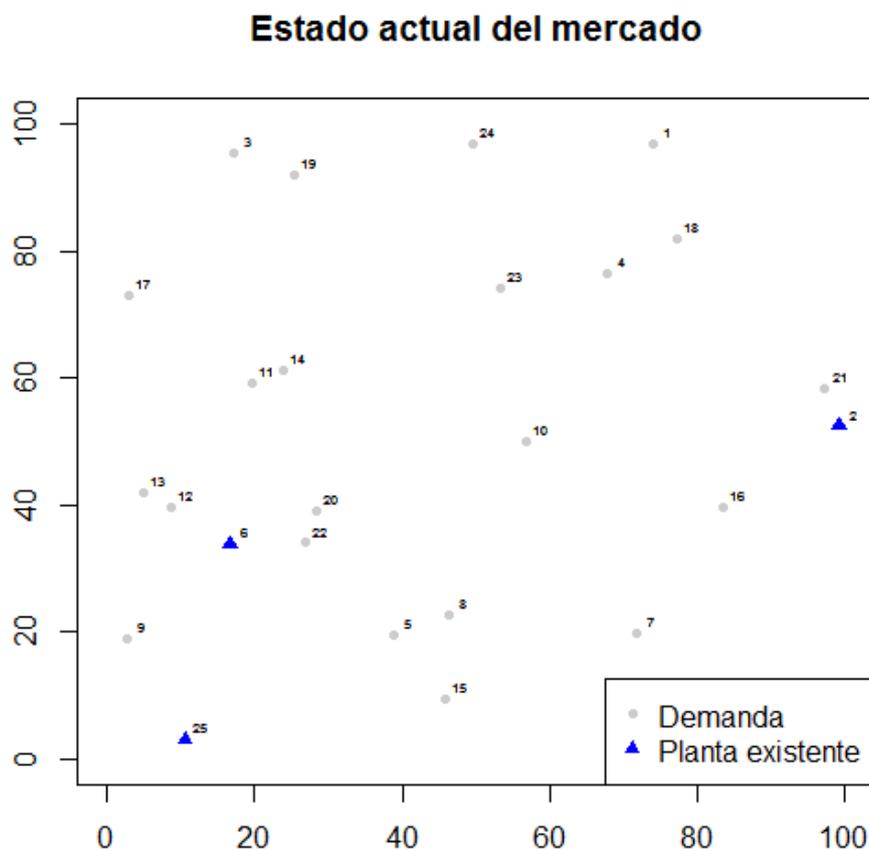
Para ello, usaremos la herramienta NEOS Server: un servicio online que contiene distintos optimizadores. Como hemos demostrado anteriormente, (P) y (P_1) son equivalentes, por lo que podemos elegir entre implementar el código AMPL de (P) , o bien, el de (P_1) y resolver, posteriormente, con un optimizador global *Baron* o con un optimizador lineal *Gurobi*, respectivamente. En nuestro ejemplo, hemos resuelto (P) con *Baron*. En cualquier caso, véase *Anexo II* para visualizar el código AMPL de ambos problemas.

Planteamiento del problema

El experimento es el siguiente. Construimos un grafo completo formado por 25 nodos ($n = 25$), seleccionados aleatoriamente de un cuadrado de lado 100 unidades. Cada nodo llevará asociado un poder de compra $b_i = 1 \forall i, j = 1, \dots, n$. A continuación, se calcula la distancia euclídea entre cada par de nodos $(d_{ij}, i, j = 1, \dots, n)$. Después tomamos, también de forma aleatoria, un subconjunto de nodos del grafo donde colocaremos las 3 plantas existentes ($p = 3$). El nuevo competidor quiere localizar 3 plantas nuevas ($r = 3$). Supondremos que un nodo puede soportar más de una planta

y, además, que la atracción de todas las plantas, tanto nuevas como existentes, será 1. En cuanto a λ , le daremos distintos valores y compararemos resultados.

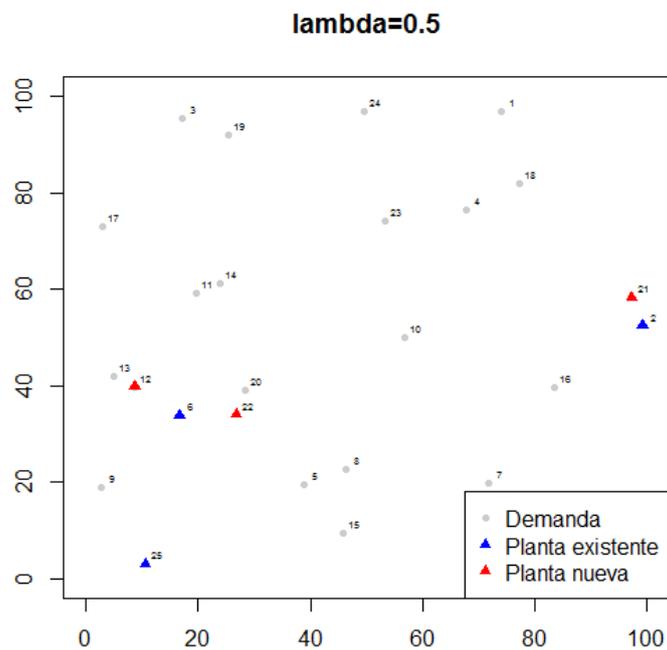
El estado actual del mercado se muestra en el siguiente gráfico.



No mostramos el mercado en forma de grafo para su mejor visualización.

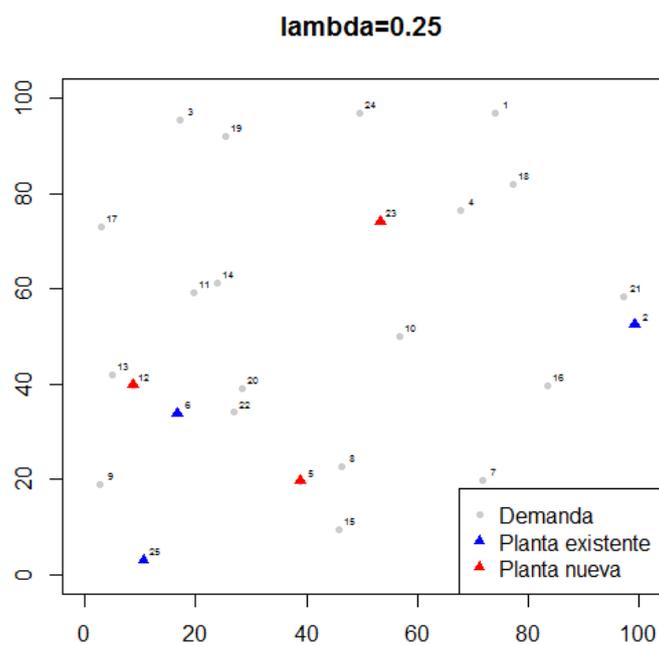
Solución del problema

Se obtiene, en primer lugar, la solución óptima para el valor de $\lambda = 0.5$.



Observamos que las tres plantas nuevas se han situado en las proximidades de las ya existentes con el fin de competir con ellas.

Si λ decrece, los clientes están dispuestos a recorrer mayores distancias. Hagamos, por tanto, el mismo experimento tomando $\lambda = 0.25$ y veamos qué solución óptima se obtiene.



Vemos que una de las plantas se ha mantenido en el mismo lugar que en el caso anterior; sin embargo, las otras dos se han alejado de la competencia buscando hacerse con el mercado en nuevas zonas.

A continuación, mostramos una tabla con la solución óptima obtenida para cada valor de λ .

	$\lambda = 0,5$	$\lambda = 0,25$
Sol. Óptima	$x_i^* = \begin{cases} 1 & \text{si } i = 12, 21, 22. \\ 0 & \text{caso contrario} \end{cases}$	$x_i^* = \begin{cases} 1 & \text{si } i = 5, 12, 23. \\ 0 & \text{caso contrario} \end{cases}$
Cuota de mercado	18,4233	16,5916

Tabla 3.1: Solución óptima obtenida para cada valor de λ .

Anexo I. Modelo continuo

En este apartado, incluimos la implementación de los programas usados en el *Capítulo 2*.

Código R

```
weiszfeldgeneralizado <- function(Xd,b,A_e,F,A_n,lambda,epsilon,r) {

xmin <- floor(min(Xd[,1]))
xmax <- ceiling(max(Xd[,1]))
ymin <- floor(min(Xd[,2]))
ymax <- ceiling(max(Xd[,2]))

startingvalues <- matrix(rep(0,2*r),c(r,2))
startingvalues[,1] <- runif(r,min=xmin,max=xmax)
startingvalues[,2] <- runif(r,min=ymin,max=ymax)

n<- nrow(Xd)
X<-matrix(rep(0,2*r),c(r,2))
market_share <- c()
Dist<-matrix(1:n*r,n,r)
h<-c()

for (j in 1:r) {
print(j)

w <- weiszfeld(startingvalues[j,],Xd,b,A_e,F,A_n,lambda,epsilon)

X[j,]<-w[[1]]
Dist[,j]<-w[[2]]
market_share[j]<-w[[3]]
}

z_opt <- max(market_share)
k <- which.max(market_share)
optimal_solution<- X[k,]
```

```

dopt<- Dist[,k]

plot(c(xmin,xmax),c(ymin,ymax),type = "n",xlab="",ylab="",xlim=
c(xmin,xmax),ylim=c(ymin,ymax))
points(c(Xd[,1]),c(Xd[,2]),col="gray80",pch=20)
points(c(F[,1]),c(F[,2]),col = "blue",pch=17)
points(optimal_solution[1],optimal_solution[2],col="red",pch=17)

L<-list(X,market_share,optimal_solution,z_opt,dopt);
return(L)
}

```

```

weiszfeld <- function(X0,Xd,b,A_e,F,A_n,lambda,epsilon){

n <- nrow(Xd)
k <- 1
X<- matrix(X0,1,2)
h<-c()
w<-c()

f <- nrow(F)
dij<-matrix(0,n,f)
for (i in 1:n) {
for (j in 1:f) {
dij[i,j]<-sqrt((Xd[i,1]-F[j,1])^2+(Xd[i,2]-F[j,2])^2)}
}

repeat{
for (i in 1:n) {h[i] <- sum(A_e*dij[i,]^(-lambda))/A_n
w[i] <- b[i]*h[i]*sqrt((Xd[i,1]-X[k,1])^2
+(Xd[i,2]-X[k,2])^2)^(lambda-2)/(1+h[i]*sqrt((Xd[i,1]-X[k,1])^2
+(Xd[i,2]-X[k,2])^2)^lambda)^2

M <- function(s) {z <-0; for (i in 1:n) {
z<-z+b[i]/(1+h[i]*sqrt((Xd[i,1]-s[1])^2+(Xd[i,2]-s[2])^2)^lambda)}
return(z)};

S <- sum(w)
}

X<-rbind(X,c(sum(w*Xd[,1])/S,sum(w*Xd[,2])/S))

if (abs(M(X[k+1,])-M(X[k,]))<=epsilon) break;

k<-k+1
}

```

```
m_s<-M(X[k+1,])
size <- nrow(X)
Xopt <-X[+size,]
D<-c()
for (i in 1:n) {
D[i] <- sqrt((Xd[i,1]-Xopt[1])^2+(Xd[i,2]-Xopt[2])^2)
}

L<-list(Xopt,D,m_s)
return(L)
}
```


Anexo II. Modelo discreto

En este apartado, incluimos la implementación de los programas usados en el *Capítulo 3*.

Código AMPL

(P)

```
param n; #number of nodes
param r; #number of new facilities
param p; #number of existing facilities
param lambda; #coefficient of spatial friction
set nodes:= 1..n; #nodes=set of vertices
set existing within nodes; #existing=subset of nodes where the
existing facilities are located
param b{nodes}; #b[i]=number of customers located in the node i
param dij{nodes,nodes}; #dij[i,j]=distance between nodes i and j
param Ae{existing}; #Ae[j]=attractiveness of the existing facility
in node j
param An{nodes}; #An[j]=attractiveness of a new facility in node j
param a{i in nodes,j in nodes}:= exp(An[j]-lambda*dij[i,j]);
param h{i in nodes}:=sum{j in existing} exp(Ae[j]-lambda*dij[i,j]);
var x{nodes} binary; #x[j]=1 if a new facility is located in j,
```

```
x[j]=0 otherwise  
  
maximize marketshare: sum{i in nodes} b[i]*(sum{j in nodes}  
a[i,j]*x[j])/(h[i]+sum{j in nodes} a[i,j]*x[j]);  
  
subject to r_1: sum{j in nodes} x[j]=r;
```

(P₁)

```

param n; #number of nodes
param r; #number of new facilities
param p; #number of existing facilities
param lambda; #coefficient of spatial friction
set nodes:= 1..n; #nodes=set of vertices
set existing within nodes; #existing=subset of nodes where the
existing facilities are located
param b{nodes}; #b[i]=number of customers located in the node i
param dij{nodes,nodes}; #dij[i,j]=distance between nodes i and j
param Ae{existing}; #Ae[j]=attractiveness of the existing facility
in node j
param An{nodes}; #An[j]=attractiveness of a new facility in node j
param a{i in nodes,j in nodes}:= exp(An[j]-lambda*dij[i,j]);
param h{i in nodes}:=sum{j in existing} exp(Ae[j]-lambda*dij[i,j]);
var x{nodes} binary; #x[j]=1 if a new facility is located in j,
x[j]=0 otherwise
var w{1..n,1..n} >= 0;
maximize marketshare: sum{j in nodes} (sum{i in nodes} b[i]*a[i,j]/
h[i])*x[j]- sum{i in nodes,j in nodes} b[i]*a[i,j]*w[i,j]/h[i];
subject to r_1: sum{j in nodes} x[j]=r;
subject to r_2{i in nodes,j in nodes}: w[i,j] <= x[j];
subject to r_3{i in nodes,j in nodes}: w[i,j] >= sum{k in nodes}
a[i,k]*x[k]/h[i]-sum{k in nodes} a[i,k]/h[i]*w[i,k] -(1-x[j]);
subject to r_4{i in nodes,j in nodes}: w[i,j] <= sum{k in nodes}
a[i,k]*x[k]/h[i]-sum{k in nodes} a[i,k]/h[i]*w[i,k];

```


Bibliografía

- [1] Stefano Benati. The maximum capture problem with heterogeneous customers. *Computers & operations research*, 26(14):1351–1367, 1999.
- [2] Stefano Benati and Pierre Hansen. The maximum capture problem with random utilities: Problem formulation and algorithms. *European Journal of Operational Research*, 143(3):518–530, 2002.
- [3] Rafael Blanquero and Emilio Carrizosa. Continuous location problems and big triangle small triangle: constructing better bounds. *Journal of Global Optimization*, 45(3):389–402, 2009.
- [4] Rafael Blanquero and Emilio Carrizosa. Solving the median problem with continuous demand on a network. *Computational Optimization and Applications*, 56(3):723–734, 2013.
- [5] Tammy Drezner and Zvi Drezner. Finding the optimal solution to the huff based competitive location model. *Computational Management Science*, 1(2):193–208, 2004.
- [6] Zvi Drezner. A general global optimization approach for solving location problems in the plane. *Journal of Global Optimization*, 37(2):305–319, 2007.
- [7] David L Huff. Defining and estimating a trading area. *The Journal of Marketing*, 28(3):34–38, 1964.
- [8] David L Huff. A programmed solution for approximating an optimum retail location. *Land Economics*, 42(3):293–303, 1966.
- [9] Frank Plastria. Static competitive facility location: an overview of optimisation approaches. *European Journal of Operational Research*, 129(3):461–470, 2001.