Universidad Miguel Hernández de Elche

Facultad de Ciencias Sociales y Jurídicas

Curso 2018 - 2019

Trabajo de Fin de Grado

Problemas de Localización Discreta

Alumno: Adrián Oltra Sánchez

Tutor: Mercedes Landete Ruiz

Abstract

El descubrimiento problemas de localización discreta fue un gran avance en la sociedad moderna. Nos permitió resolver dudas sobre si sería factible o no establecer una planta en un lugar determinado, a poder elegir el lugar correcto frente a una multitud de opciones. A pesar de que se ha avanzado mucho en la materia, sus investigaciones hoy en día aún continuan. En este trabajo continuaremos con ese progreso. Veremos un modo diferente, de la que estamos acostumbrados, de aplicar esos problemas en nuestro ordenador. Veremos que su aplicación puede ser tan útil como en los softwares más exclusivos.

Índice general

Capítulo 1

Introducción:

Durante este trabajo final de grado (TFG) vamos a estudiar una serie de problemas denominados de localización discreta. Los problemas de localización discreta tratan sobre la elección de una serie de ubicaciones donde estableceremos lo que se denomina como *planta*. A su vez vamos a disponer de una serie de clientes que querrán satisfacer su demanda. Por lo tanto hemos de resolver dos cuestiones, qué plantas se abrirán y su asociación con los clientes que así tendrán su demanda satisfecha.

En la literatura existen una gran variedad de problemas de localización discreta. Estos problemas se pueden devidir en dos tipos en función de su finalidad: los que tienen como objetivo tipo mediana y los que cuyo objetivo es p-centro. Los problemas que tienen como objetivo tipo mediana tratan de reducir los costes derivados de transporte a los clientes y los problemas de objetivo p-centro tratan de minimizar la máxima distancia al cliente más lejano.

Dentro de los problemas cuyo obtivo es tipo mediana nosotros vamos a tratar los más clásicos, problema de localización de plantas simples (SPLP) y problema de localización de plantas con p-mediana (p-mediana). La diferencia entre uno y otro radica en que en el problema de localización de plantas con p-mediana nosotros podremos escoger el número de plantas que se abrirán mientras que el problema de localización de plantas simples no. Adicional a lo anterior podemos encontrar derivaciones de estos problemas: problema de localización de plantas con capacidades, problema de localización de plantas con multietapa y problema de localización de concentradores. Este tipo de problemas tiene en común que su función principal es la de minimizar la distancia entre las posibles plantas y todos los nodos de demanda que son asignados. Un ejemplo sería una oficina de correos, la cual interesa más que esté cerca del mayor cúmulo de población posible para un mejor reparto del correo.

En los problemas de objetivo p-centro trataremos el problema de localización de p-centro. En este a diferencia de los anteriores lo que se buscará es minimizar la distancia entre las plantas y el nodo de demanda de mayor distancia. Un ejemplo para este tipo sería un hospital. Si se usaran problemas de objetivo tipo mediana los hospitales siempre estarían situados enmedio de grandes poblaciones incrementando la distancia que habría frente a poblaciones más alejadas por lo tanto no podrian ser atentidos correctamente. Al aplicar el problema de p-centro nos centramos en poder dar servicio a todos los clientes.

Este trabajo veremos que se divide en dos partes: una tendrá un marco teórico y la otra práctico. En la parte teórica serán explicados todos y cada uno de los problemas mencionados anteriormente. Se explicará en qué consiste cada uno, su nomenclatura así como las variables que lo componen y la formulación con sus respectivas restricciones.

En la parte práctica los problemas de localización de plantas y problemas de localización de plantas con tendrá dos ejemplos. En uno usaremos una base de datos real y en el otro uno ficticio. En el caso de los problemas de localización de plantas con capacidades y problema de localización del p-centro solo tendrán un ejemplo real. Los problemas serán escritos en lenguaje R y el código será explicado para su mayor entendimiento. Al finalizar los resultados serán comentados. El problema de localización de plantas con multietapa al igual que el problema de localización de concentradores no tendrán una parte práctica debido a su complejidad. En el caso del problema de localización de plantas con multietapa sería muy complejo ya que se trata de una conglomeración de distintos problemas de objetivo tipo mediana. Es decir, al tratarse de varias etapas en cada una se puede usar un tipo de problema distinto.

Este trabajo final de grado no se habría podido realizar sino es gracias a la aplicación del conocimento adquerido durante el grado de Estadística Empresarial en la Universidad Miguel Hernández. Todas las asginaturas que componen el grado han ayudado en mayor o menor grado pero debo resaltar unas cuantas. Optimización: optimización de recursos, modelos de optimización y gestión y planificación de la producción componen el grupo de asignaturas sobre la optimización. En técnicas estadísticas serían estadística, minería de datos y técnicas estadísticas en análisis de mercados que contribuyeron en gran medida en el aprendizaje del software R junto a gestión de cartera e inversiones.

En la actualidad existen diversos software, como LINGO o IBM ILOG CPLEX Optimization Studio, que son más comunes a la hora de solucionar estos problemas de localización discreta. En internet se puede observar muy poco sobre la aplicación del software R haciendo que sea esto último lo más destacable del

TFG. Se decidió el uso de este software porque es el más usado durante el grado, muy completo en materia estadística y gratuito. Al ser accesible a todo el mundo se demuestra que también podemos solucionar este tipo de problemas sin necesidad de pagar un software exclusivo. También cabe destacar el uso de LaTeXpara la elaboración de este trabajo final de grado al tratarse de uno de los mejores programas de edición de textos científicos.

Para el desarrollo de este trabajo se he precisado de búsqueda de información para completarlo. En la **bibliografía** destacamos el uso de [9] para el entendimiento de los problemas de localización discreta y elaboración de la introducción. Las bibliografías [2], [4], [7] y [9] se utilizarón para la elaboración del marco teórico y del modelo de problema de localización de plantas simples. Para el modelo del problema de localización de plantas de la p-mediana se usó el [3] y [8]. Las bibliografías [5] y [11] fueron utilizadas en el problema de localización de plantas con capacidades. Para el modelo del problema de localización de plantas con multietapa se usó el [2]. En el problema de localización de concentradores fueron usadas las bilibografías [6] y [10]. Por último la bibliografía [1] fue utilizada en el problema de localización del p-centro.

Capítulo 2

Objetivo tipo mediana

Dentro de los problemas cuyo objetivo es tipo mediana hemos podido encontrar que cada uno usa una nomenclatura diferente dependiendo también del autor. En este caso se ha procedido a recopilar todas aquellas variables que sean comunes en todos y cada uno de los problemas para mejorar su entendimiento. A continuación las veremos.

I=$\{1, ..., m\}$: Conjunto de posibles ubicaciones de las plantas.

J=$\{1, ..., n\}$: Conjunto de posibles clientes.

m: Número de posibles plantas.

n: Número de posibles clientes.

2.1. Problema de localización de plantas simples

Imaginemos que estamos en una población y tenemos que abrir una planta. No sabemos donde hacerlo pero se nos ha dado una lista de posibles sitios donde hacerlo. En este supuesto vamos a tener capacidad ilimitada, es decir, no tenemos límite en la cantidad de producto que podamos almacenar. Lo único que buscamos es satisfacer la demanda de todos los clientes que tengamos. También tenemos costes derivados de la producción, mantenimiento de la planta, administrativo y de transporte al cliente. En la aplicación todos estos costes irán incluidos dentro de los costes de transporte al cliente. La apertura de cualquier planta tendrá asociada un coste fijo. El objetivo principal de este problema es la reducción de costes, para ello deberemos elegir la mejor ubicación donde poder situar nuestra planta y así poder satisfacer la demanda de nuestros clientes.

2.1.1. Variables y formulación

b_j: Número de unidades de producto solicitadas del cliente j. $\forall$ j ϵ J

t_{ij}: Coste de transporte de la planta i al cliente j. $\forall$ i ϵ I, j ϵ J

f_i: Coste fijo de establecer la planta i. $\forall$ i ϵ I

Las variables de decisión para el problema de la SPLP son las siguientes.

$$y_i = \begin{cases} 1 & \text{si se ubica la planta en la localización i } \epsilon \text{ I} \\ 0 & \text{en caso contrario} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{si al cliente j } \epsilon \text{ J se le asigna la planta i } \epsilon \text{ I} \\ 0 & \text{en caso contrario} \end{cases}$$

Una vez conocida todas las variables que vamos a utilizar, procederemos a ver la formulación del problema de **SPLP**.

Formulación:

$$min \sum_{i \epsilon I} \sum_{j \epsilon J} (b_j t_{ij}) x_{ij} + \sum_{i \epsilon I} f_i y_i \quad (1.1)$$

Restricciones:

$$\sum_{i \epsilon I} x_{ij} = 1, \forall \text{ j} \, \epsilon \text{ J} \quad (1.2)$$

$$x_{ij} \leq y_i, \forall \text{ i} \, \epsilon \text{ I}, \forall \text{ j} \, \epsilon \text{ J} \quad (1.3)$$

$$y_i \in \{0{,}1\}, x_{ij} \in \{0{,}1\}, \forall \text{ i} \, \epsilon \text{ I}, \forall \text{ j} \, \epsilon \text{ J} \quad (1.4)$$

En la función objetivo (1.1) minimizamos los costes de envío de productos a los clientes que sean asigandos a una planta y el coste de apertura de las mismas. La restricción (1.2) obliga a que a todos los clientes son asignados a una planta, es decir que ningún cliente se queda sin recibir servicio. Para ello utilizamos una restricción de igualdad en la que nuestra variable $\mathbf{x_{ij}}$ debe ser 1 obligatoriamente. En la restricción (1.3) establecemos una relación de desigualdad entre dos de nuestras variables $\mathbf{x_{ij}}$ y $\mathbf{y_i}$. Se pretende que a ninguno de nuestros clientes se le pueda asignar una planta que no esté operativa. Es obligatorio que una planta esté abierta para que esta pueda ser fijada para cualquier cliente. Para ello como se ha dicho previamente se utiliza una relación de desigualdad en la que nuestra variable $\mathbf{x_{ij}}$ nunca puede ser menor o igual que la variable que determina si las plantas están abiertas o no $\mathbf{y_i}$.
Por último la restricción (1.4) especifica que las variables de decisión son binarias.

2.2. Problema de localización de plantas con p-mediana

Este problema es parecido al SPLP. La diferencia con el problema anterior es que podremos decidir de antemano el número de plantas que se abrirán. Entonces los clientes serán asociados a cualquiera de las plantas que resulten operativas. Tendremos costes asociados al transporte de los productos a los clientes. Al igual que en el problema de la SPLP vamos a tener capacidad ilimitada, esto significa que podemos asociar a los clientes a su planta mas cercana. Al nosotros poder decidir el número de plantas que se abrirán no tendremos costes asociados a la apertura de las plantas ya que serán los mismo para todos, por lo que no los tendremos en cuenta. El objetivo principal de este problema es determinar que planta serán abiertas y a qué clientes abastecerán mientras se reducen costes.

2.2.1. Variables y formulación

b_j: Número de unidades de producto solicitadas del cliente j. $\forall$ j ϵ J

t_{ij}: Coste de transporte de la planta i al cliente j. $\forall$ i ϵ I, j ϵ J

Las variables de decisión para el problema de la p - mediana son las siguientes.

$$y_i = \begin{cases} 1 & \text{si se ubica la planta en la localización i } \epsilon \text{ I} \\ 0 & \text{en caso contrario} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{si al cliente j } \epsilon \text{ J se le asigna la planta i } \epsilon \text{ I} \\ 0 & \text{en caso contrario} \end{cases}$$

Definida toda la nomenclatura y las variables, el problema de la p - mediana puede formularse de la siguiente manera.

Formulación:

$$min \sum_{i \epsilon I} \sum_{j \epsilon J} (b_j t_{ij}) x_{ij} \quad (2.1)$$

Restricciones:

$$\sum_{i \epsilon I} x_{ij} = 1, \forall \text{ j } \epsilon \text{ J} \quad (2.2)$$

$$\sum_{i \epsilon I} y_i = p \quad (2.3)$$

$$x_{ij} \leq y_i, \forall \text{ i } \epsilon \text{ I}, \forall \text{ j } \epsilon \text{ J} \quad (2.4)$$

$$y_i \in \{0,1\}, x_{ij} \in \{0,1\}, \forall \text{ i } \epsilon \text{ I}, \forall \text{ j } \epsilon \text{ J} \quad (2.5)$$

En la función objetivo (2.1) minimizamos los costes de envío de producto a los clientes. La restricción (2.2) nos asegura que cada cliente es asignado a una planta, es idéntica a la restricción(1.2) del problema anterior. La restricción (2.3) nos garantiza que exactamente habrán *p* plantas, que es la cantidad de agrupaciones de clientes que tendremos asociado a las plantas. Para ello igualamos nuestra variable $\mathbf{y_i}$ a la constante *p*. La restricción (2.4) asegura que los clientes sean asignados a una planta que esté operativa. De nuevo aplicamos lo visto en la restricción (1.3). Por último las restricciones (2.5) especifican que las variables de decisión son binarias.

2.3. Problema de localizacićon de plantas con capacidades

Al tratarse de una extensión del problema de la **SPLP** el objetivo principal es el mismo, la reducción de costes y poder satisfacer la demanda de los clientes. Para ello vamos a tener costes de apertura de las diferentes plantas así como costes de producción y transporte. La parte novedosa de este apartado es que ya no vamos a tener una capacidad ilimitada en nuestras diversas plantas. Hay que incidir en que cada una tendrá una capacidad propia, no es la misma para todas. Por tanto, aunque nuestro objetivo es reducir costes ya no se asociará siempre cada cliente a su planta más cercana, como ocurria en problemas anteriores, pues hay que tener en cuenta la capacidad de cada planta.

2.3.1. Variables y formulación

f_i: Coste de apertura de la planta i. $\forall$ i ϵ I

v_i: Valor máximo de producción. $\forall$ i ϵ I

b_j: Número de unidades de producto solicitadas del cliente j. $\forall$ j ϵ J

t_{ij}: Coste de transporte de la planta i al cliente j. $\forall$ i ϵ I, j ϵ J

Las variables de decisión para el problema de localización de plantas con **capacidades** son las siguientes.

$$y_i = \begin{cases} 1 & \text{si se ubica la planta en la localización i } \epsilon \text{ I} \\ 0 & \text{en caso contrario} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{si al cliente j } \epsilon \text{ J se le asigna la planta i } \epsilon \text{ I} \\ 0 & \text{en caso contrario} \end{cases}$$

Definida toda la nomenclatura y las variables, el problema de localización con **capacidades** puede formularse de la siguiente manera.

Formulación:

$$min \sum_{i \epsilon I} \sum_{j \epsilon J} (b_j t_{ij}) x_{ij} + \sum_{i \epsilon I} f_i y_i \quad (3.1)$$

Restricciones:

$$\sum_{i \epsilon I} x_{ij} = 1, \forall \text{ j } \epsilon \text{ J} \quad (3.2)$$

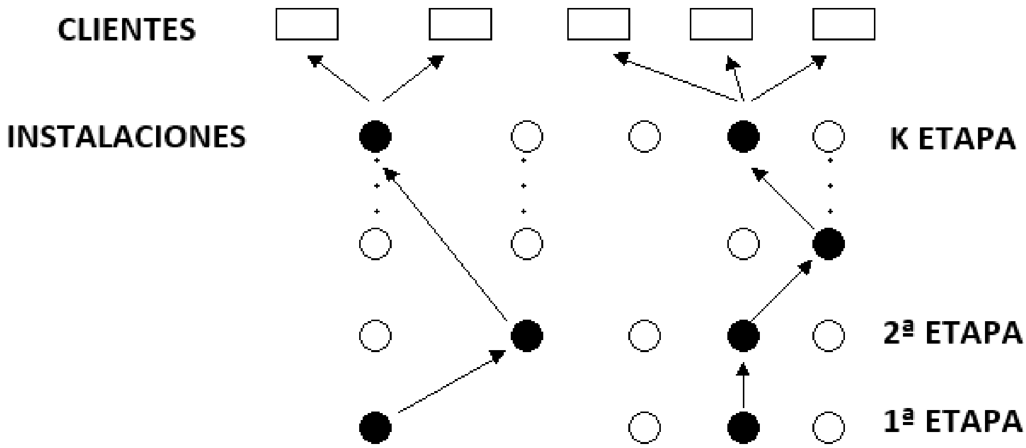
$$\sum_{j \epsilon J} b_j x_{ij} \leq v_i y_i, \forall \text{ i } \epsilon \text{ I} \quad (3.3)$$

$$y_i \in \{0,1\}, x_{ij} \in \{0,1\}, \forall \text{ i } \epsilon \text{ I}, \forall \text{ j } \epsilon \text{ J} \quad (3.4)$$

En la función objetivo (3.1) minimizamos los costes de envío de productos a los clientes así como los costes de apertura de las plantas. La restricción (3.2) nos asegura que a cada cliente se le es asignado una instalación, es idéntica a las anteriores (1.2) y (2.2). La restricción (3.3) consiste en que no podemos enviarle al cliente más productos de los que son producidos en una planta. Por último las restricciones (3.4) definen que las variables de decisión son binarias.

2.4. Problema de localización de plantas con multietapa

En este apartado tendremos de nuevo un conjunto de posibles plantas y clientes. Pero a diferencia que en los casos anteriores cada cliente no será servido por una sola planta sino por una secuencia de ellas. Estas secuencias estarán definidas por la jerarquá de la producción y distribución de las plantas. Hay que hacer meción a que cada planta tiene un coste, además también hay costes derivados del transporte y mantenimiento de los diferentes caminos. El objetivo principal es el de minimizar los costes de las secuencias de las plantas para poder dar servicio a los clientes y poder satisfacer su demanda.



(Imagen obtenida de www.math.nsc.ru)

En esta imagen se puede vislumbrar de una manera eficiente en que consiste este problema. Como vemos hay diferentes etapas y en cada una se elige una planta. De modo que una vez se complete la ruta pasando por las diferentes plantas se da servicio a los clientes.

2.4.1. Variables y formulación

L=$\{1, ..., k\}$: Conjunto de posibles caminos de las plantas.

f_i: Coste de apertura de la planta j. $\forall$ i ϵ I

c_{lj}: Coste de servicio del cliente j por el camino a la planta l. $\forall$ l ϵ L, j ϵ J

Las variables de decisión para el problema de localización **multietapa** son las siguientes.

$$y_i = \begin{cases} 1 & \text{si se ubica la planta en la localización i } \epsilon \text{ I} \\ 0 & \text{en caso contrario} \end{cases}$$

$$x_{lj} = \begin{cases} 1 & \text{si al cliente j } \epsilon \text{ J es servido por la ruta l } \epsilon \text{ L} \\ 0 & \text{en caso contrario} \end{cases}$$

Definida toda la nomenclatura y las variables, el problema de localización de **multietapa** puede formularse de la siguiente manera.

Formulación:

$$min \sum_{j \epsilon J} \sum_{l \epsilon L} c_{lj} x_{lj} + \sum_{i \epsilon I} f_i y_i \quad (4.1)$$

Restricciones:

$$\sum_{l \epsilon L} x_{lj} = 1, \forall \text{ j } \epsilon \text{ J} \quad (4.2)$$

$$y_i \geq \sum_{l \epsilon L : i \epsilon L} x_{lj}, \forall \text{ j } \epsilon \text{ J}, \forall \text{ i } \epsilon \text{ I} \quad (4.3)$$

$$y_i \in \{0,1\}, x_{lj} \in \{0,1\}, \forall \text{ i } \epsilon \text{ I}, \forall \text{ l } \epsilon \text{ L} \quad (4.4)$$

En la función objetivo (4.1) minimizamos los costes de apertura de las plantas así como los costes derivados de dar servicio a los clientes. La restricción (4.2) se asegura que todos los clientes estén destinados a un camino, es igual a la primera restricción de todos los modelos anteriores. La restricción (4.3) nos demuestra que si una planta esta operativa esta debe estar incluida en una ruta que realiza un servicio a los clientes. Para ello establecemos una relación de desigualdad en la que comparamos nuestra variable $\mathbf{y_i}$ con el sumatorio de todas las $\mathbf{x_{lj}}$. Por último las restricciones (4.4) definen que las variables de decisión son binarias.

2.5. Problema de localización de concentradores

Cuando se habla de problemas de localización de concentradores hay que consideran que existen muchas situaciones de aplicación práctica. Todas ellas tienen en común que se debe enviar un producto entre todos los nodos de una red, pero este envío ha de realizarse a través de unos nodos especiales que llamaremos concentradores. Estos nodos tienen la función de recoger el producto y distribuirlo entre los nodos destino y otros concentradores que lo distribuyan a su vez entre los clientes. El problema consiste en localizar aquellos nodos donde se establecerán los concentradores y asignar las rutas de envío a través de los concentradores de modo que el coste total sea mínimo. El caso que vamos a tratar a continuación se trata de localizar *p* concentradores sin capacidades y con asignación múltiple, es decir, que cada nodo puede ser asociado a varios concentradores. Por lo tanto ya no serviremos a clientes por separado sino a grupos de clientes que serán los concentradores.

2.5.1. Variables y formulación

I=$\{1, ..., m\}$: Conjunto de concetradores potenciales, es decir las plantas.

J=$\{1, ..., n\}$: Conjunto de puntos de destino de la ruta, es decir los clientes.

c$_{ijkm}$: Coste por unidad de producto asociado al uso del arco (i,j,k,m). $\forall$ i ϵ I, j ϵ J, k y m ϵ I

Las variables de decisión para el problema de localización de concentradores son las siguientes.

$$x_{ijkm} = \begin{cases} 1 & \text{si} \quad \text{si el nodo i } \epsilon \text{ I está asignado j } \epsilon \text{ J} \\ & \quad\quad \text{y pasa a través de los concentradores } k \text{ y } m \ \epsilon \text{ I} \\ & \\ 0 & \text{si} \quad \text{en caso contrario} \end{cases}$$

$$y_k = \begin{cases} 1 & \text{si} \quad \text{si el nodo } k \text{ es un concentrador k } \epsilon \text{ I} \\ & \\ 0 & \text{si} \quad \text{en caso contrario} \end{cases}$$

Debido a esta nomenclatura en este modelo podemos ver diferentes escenarios. Si *k* = *m* eso significaría que nuestro concentrador de paso sería el mismo, entonces de nuestro concentrador de inicio *i* pasaría a *k* = *m* y de ahí a nuestro punto de destino de ruta *j*. Si *j* = *m* significa que uno de los concentradores de paso es igual que nuestro destino de ruta, entonces empezaríamos en *i* de ahí pasaría a los concentradores de paso *k* y *m* y al ser *j* = *m* llegaría al punto de destino de ruta. Si *k* = *m* = *j*, eso significaría que nuestros concentradores de paso son los mismos que nuestro destino de ruta, entonces de nuestro concentrador de inicio *i* pasaríamos a un concentrador de paso y al ser *k* = *m* = *j* ya habríamos llegado a nuestro punto de destino. Por último si **I = J** entonces x_{iiii} nuestros concentradores serían los mismos.

Definida toda la nomenclatura y las variables, el problema de: Localización de *p* concentradores sin capacidades con asignación múltiple **(Hub Location)** puede formularse de la siguiente manera.

Formulación:

$$min \sum_{ijkm \in I \cup J} c_{ijkm} x_{ijkm} \quad (5.1)$$

Restricciones:

$$\sum_{k,m \in I \cup J} x_{ijkm} = 1, \forall \text{ i } \epsilon \text{ I}, \forall \text{ j } \epsilon \text{ J} \quad (5.2)$$

$$\sum_{k \in I \cup J} y_k = p \quad (5.3)$$

$$x_{ijkm} \leq y_k, \forall \text{ i } \epsilon \text{ I}, \forall \text{ j } \epsilon \text{ J}, \forall \text{ k,m } \epsilon \text{ I} \cup \text{J} \quad (5.4)$$

$$x_{ijkm} \leq y_m, \forall \text{ i } \epsilon \text{ I}, \forall \text{ j } \epsilon \text{ J}, \forall \text{ k,m } \epsilon \text{ I} \cup \text{J} \quad (5.5)$$

$$y_k \in \{0,1\}, x_{ijkm} \in \{0,1\}, \forall \text{ i } \epsilon \text{ I}, \forall \text{ j } \epsilon \text{ J}, \forall \text{ k,m } \epsilon \text{ I} \cup \text{J} \quad (5.6)$$

En la función objetivo (5.1) tratamos de reducir los costes de envíos de productos por los usos de los arcos (i,j,k,m). La restricción (5.2) obliga que todos los nodos sean asignados para así satisfacer su demanda. Para ello establecemos que nuestra variable $\mathbf{x_{ijkm}}$ sea igual que 1, de esa manera todos estan operativos. En la siguiente (5.3) forzamos a que se ubiquen exactamente *p* concentradores, siendo esta restricción opcional. Para conseguir ese que se cumpla esa restricción hacemos el sumatorio de todas nuestras $\mathbf{y_k}$ y debe dar como resultado **p** concentradores. La restricción (5.4) asegura que no se asignen nodos a concentradores sin abrir. La restricción (5.5) es exactamente igual solo que en vez de ser concentradores de tipo $\mathbf{y_k}$ son de tipo $\mathbf{y_m}$. Establecemos que nuestra variable $\mathbf{x_{ijkm}}$ sea menor o igual que nuestra variable $\mathbf{y_k}$, la cual determina si el nodo esta abierto o no. Las restricción (5.6) establece que nuestras variables de decisión son binarias.

Capítulo 3

Objetivo p-centro

A diferencia que en el anterior apartado, en este nos vamos a centrar específicamente en el problema del **p - centro**. Como hemos podido aprender en los pasos previos, el objetivo principal del tipo **mediana** era minimizar las distancias entre las plantas y todos los nodos de demanda. Ahora con el de la **p - centro** nuestro objetivo principal es minimizar la distancia entre las plantas y el nodo de demanda de más distancia. Por eso aplicamos el problema del **p - centro**, para poder dar servicio a todos los clientes. En los problemas de tipo **mediana** hemos podido observar que existen diversas variantes y aunque en los problemas de tipo **p - centro** solo trataremos el problema de localización del **p - centro** también se podrían modelizar las variantes del tipo anterior cambiando el objetivo mediana por **p - centro**.

3.1. Problema de localización del p-centro

Este problema pertenece a los modelos de máxima distancia. El objetivo es minimizar la máxima distancia entre las plantas y los clientes. En este problema no vamos a tener costes, ni de apertura ni de mantenimiento y transporte. Tampoco tendremos en cuenta la capacidad de las plantas

3.1.1. Variables y formulación

I={1,...,m}: Conjunto de posibles localizaciones de las plantas

J={1,...,n}: Conjunto de nodos de demanda.

h_j: Demanda del nodo j. $\forall$ j ϵ J

d_{ij}: Distancia entre la demanda del nodo j y la planta i. $\forall$ i ϵ I, j ϵ J

También vamos a utilizar la constante **p** que corresponde al número de plantas para localizar.

$$y_i = \begin{cases} 1 & \text{si se ubica la planta en la localización i } \epsilon \text{ I} \\ 0 & \text{en caso contrario} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{si la demanda del nodo j } \epsilon \text{ J es servido por en nodo i } \epsilon \text{ I} \\ 0 & \text{en caso contrario} \end{cases}$$

w= máxima distancia entre un nodo de demanda y su planta a la que es asignada

Definida toda la nomenclatura y las variables, el problema de localización de **p - centro** puede formularse de la siguiente manera.

Formulación:

$$min \quad w \quad (6.1)$$

Restricciones:

$$\sum_{i \epsilon I} y_i = p \quad (6.2)$$

$$\sum_{i \epsilon I} x_{ij} = 1, \forall \text{ j } \epsilon \text{ J} \quad (6.3)$$

$$x_{ij} \leq y_i, \forall \text{ i } \epsilon \text{ I}, \forall \text{ j } \epsilon \text{ J} \quad (6.4)$$

$$w - \sum_{i \epsilon I} h_j d_{ij} x_{ij} \geq 0, \forall \text{ j } \epsilon \text{ J} \quad (6.5)$$

$$y_i \in \{0{,}1\}, x_{ij} \in \{0{,}1\}, \forall \text{ i } \epsilon \text{ I}, \forall \text{ j } \epsilon \text{ J} \quad (6.6)$$

En la función objetivo (6.1) queremos minimizar *w* que es la máxima distancia del nodo de demanda y la planta a la que es asiganda. La restricción (6.2) asegura que existen **p** plantas que pueden ser instaladas, para ello hacemos que el sumatorio de nuestra variable x_j sea *p*, que es el número total de plantas. La restricción (6.3) requiere que cada nodo de demanda sea asignado a solo una planta. Debido a ello igualamos nuestra variable y_{ij} a uno. La restricción (6.4) sólo permite que la demanda de un nodo sea asignada a una planta abierta, con ello establecemos una relación de desigualdad (de menor o igual) entre las variables y_{ij} y x_i. En la siguiente (6.5) estipula que la máxima distancia entre el nodo *i* y la plantas en el sitio *j*, denotado por *w* es más grande que la distancia entre cualquier nodo *i* y la planta localizada en el sitio *j*. Por último las restricciones (6.6) definen que las variables de decisión son binarias.

Capítulo 4

Implementación en R

Una vez ya se han explicado los modelos, vamos a proceder a realizar la parte práctica del proyecto. En esta parte vamos a explicar como serían los modelos resueltos en el software R. Se seguirá el mismo orden que en la parte teórica. Hay que incidir que el modelo de localización discreta **multietapa** y el modelo de localización de **concentradores** no tendrán una parte práctica debido a su complejidad.

En cada modelo se desarrollará su código en R paso a paso y este será explicado. Se realizarán ejemplos prácticos para cada modelo, una base de datos será real y la otra ficticia. La base de dato que usaremos en cada modelo será detallada previamente y se expondrá de donde se obtuvo. Se destaca que los datos ficiticos han sido obtenidos de dos fuentes distintas, siendo una de ellas la OR-Library.

`http://people.brunel.ac.uk/~mastjjb/jeb/info.html`
`http://www.math.nsc.ru/AP/benchmarks/UFLP/Engl/uflp_eucl_eng.html`

Al final se mostrarán los resultados y estos serán debidamente analizados para su mejor entendimiento.

Para el ejemplo real hemos usado una base de datos de ciudades de la provincia de Alicante, en concreto 10 (Orihuela, Elche, Alicante, Altea, Crevillente, Callosa, Aspe, Torrevieja, Benidorm y Villajoyosa). A cada ciudad le hemos asignado un código para diferenciarlas, en el cuadro 4.1 lo podemos ver.

Cuadro 4.1: Código ciudades

Código	Ciudad

1	Orihuela
2	Elche
3	Alicante
4	Altea
5	Crevillente
6	Callosa
7	Aspe
8	Torrevieja
9	Benidorm
10	Villajoyosa

Para una mejor ubicación del lector, aquí presentamos un mapa de la provincia de Alicante con las 10 ciudades marcadas en él.

(Imagen obtenida de https://es.wikipedia.org/)

4.1. Problema de localización de plantas simple

4.1.1. Ejemplo real

Como el ejemplo real trata sobre ciudades de la provincia de Alicante usaremos las distancias que hay entre cada una de ellas. En el de capacidades ha sido mas sencillo ya que me ponia bien las plantas que se abrian.

```
#Cargar paquetes
library(readr)# Usamos el paquete para la lectura de datos
library(Rglpk)# Paquete encargado de solucionar el problema
```

```
#Lectura de datos
Distancia_entre_ciudades <- read_tsv("Directorio de los datos",
  col_names = TRUE,
  cols(Ciudad_Origen = col_character(),
  Ciudad_Destino = col_character(),
  Distancia = col_integer()
))
Coste_Apertura_Ciudades <- read_tsv("Directorio de los datos",
  col_names = TRUE,
  cols(Ciudades = col_character(),
  Coste_fijo = col_integer()
))
```

En la lectura de datos cargamos dos ficheros de texto. En el primero denominado "Distancia entre ciudades.txt". Como podemos ver en el cuadro 4.2 tengo 3 variables: que serán la ciudad de origen, la ciudad de destino y la distancia que hay entre las dos. En el documento vemos la distancia que hay entre cada ciudad medida en kilómetros y lo podemos visualizar de la siguiente manera.

Cuadro 4.2: Distancia ciudades

Ciudad_Origen	Ciudad_Destino	Distancia
Orihuela	Orihuela	0
Orihuela	Elche	33
Orihuela	Alicante	56
*	*	*
Villajoyosa	Villajoyosa	0

El segundo fichero de texto denominado “Coste Apertura Ciudades.txt". En él aparecen diversas variables aunque para este problema solo necesitamos 3: la ciudad, un primer coste fijo y otro coste fijo distinto. A pesar de que vemos que hay dos costes fijos, en el problema solo se usará uno. La diferencia entre ambos costes fijos es que uno es más económico que el otro. En este caso, estos datos han sido inventados para realizar el modelo. El cuadro 4.3 de datos se puede ver de la siguiente manera.

Cuadro 4.3: Costes de apertura

Ciudades	Coste_fijo	Coste_fijo2
Orihuela	500	50
Elche	500	50
Alicante	500	50
*	*	*
Villajoyosa	500	50

```
I<-10
#Posibles localizaciones
J<-10
#Posibles clientes
```

Aquí definimos que tendremos 10 posibles plantas y 10 clientes a los que servir.

```
obj<- c(Distancia_entre_ciudades$Distancia,Coste_Apertura_Ciudades$Coste_fijo
#coeficientes de la funcion objetivo
rhs<- c(rep (1, J), rep(0,I*J))
#Coeficientes de la derecha de las restricciones
dir<- c(rep("==", J),rep("<=", I*J))
#Son los iguales, o mayores iguales, o menores
types<- c(rep("B",(I*J+I)))
#Tipo de variable
```

En este apartado se define la función objetivo usando las variables de distancia entre ciudades y el primer coste fijo de apertura de cada una de ellas. Se establece también los coeficientes de la derecha de las restricciones al igual que las igualdades. Por último se establece que las variables $\mathbf{x_{ij}}$ y $\mathbf{y_i}$ son de tipo binaria.

```
mat <- matrix (0, J+I*J, I*J+I)
#Matriz
```

Creamos una matriz de ceros que tendrá 110 filas y 110 columnas. A continuación se procederá a rellenar esa matriz con las restricciones.

```
  for(i in 1:I){
      for(j in 1:J){
          mat[j,J*(i-1)+j]<-1  #Primera restriccion
        }
    }

  for(i in 1:(I*J)){ #Segunda restriccion
    mat[J+i,i] <-1

  }

  for(i in 1:I){ #Segunda restriccion
    for(j in 1:J){
        mat[J+(J*(i-1)+j),(I*J)+i]<- -1
```

```
        }
    }
```

```
solucion<- Rglpk_solve_LP(obj, mat,dir,rhs, types, max = FALSE)
solucion$optimum
```

A continuación en los cuadros 4.4 y 4.5 vemos que Elche es la única planta que se ha abierto y que todas las demas ciudades han sido asignadas a ella.

Cuadro 4.4: Plantas abiertas

plantas abiertas $\mathbf{y_i}$	2 (Elche)

Cuadro 4.5: Asigancién de clientes

Código	Ciudad	Asignaciones
1	Orihuela	2 (Elche)
2	Elche	2 (Elche)
3	Alicante	2 (Elche)
4	Altea	2 (Elche)
5	Crevillente	2 (Elche)
6	Callosa	2 (Elche)
7	Aspe	2 (Elche)
8	Torrevieja	2 (Elche)
9	Benidorm	2 (Elche)
10	Villajoyosa	2 (Elche)

También vemos que nuestro optimo es “876 uds”. Al establecer unos costes fijos elevados “500 uds", el programa dispone que Elche al ser una ciudad bastante centrica, con respecto a los otros municipios, y de gran densidad de población sea la única planta que se abra.

Al ver estos resultados no nos hemos podido preguntar que hubiese pasado si esos costes fijos se redujeran. Para eso he cambiado la variable de costes fijos por la de menor cantidad. Al hacer la prueba obtenemos los siguientes resultados.

```
obj<- c(Distancia_entre_ciudades$Distancia,Coste_Apertura_Ciudades$Coste_fijo2
#coeficientes de la funcion objetivo
```

En el cuadro 4.6 vemos que son 3 las plantas que se han abierto.

Cuadro 4.6: Plantas abiertas

plantas abiertas y_i	2 (Elche)	6 (Callosa)	9(Benidorm)

Cuadro 4.7: Asiganción de clientes

Código	Ciudad	Asignaciones
1	Orihuela	6 (Callosa)
2	Elche	2 (Elche)
3	Alicante	2 (Elche)
4	Altea	9 (Benidorm)
5	Crevillente	2 (Elche)
6	Callosa	6 (Callosa)
7	Aspe	2 (Elche)
8	Torrevieja	6 (Callosa)
9	Benidorm	9 (Benidorm)
10	Villajoyosa	9 (Benidorm)

El cuadro 4.7 nos muestra que Callosa abastece a los clientes de más al sur de la provincia, Elche por su parte lo hará con los situados en el centro y Benidorm con los que están más al norte. Podemos discernir que a menor coste de apertura mayor número de plantas podrán ser abiertas, hay mayor flexibilidad. Incluso vemos que el óptimo ahora es “271 uds”.

Viendo los resultados de ambos problemas se nos podría plantear la duda de qué modelo sería más eficiente. En nuestra función objetivo nosotros queremos reducir costes que son la suma de los costes de transporte y los de apertura de planta. En nuestro primer problema como solo se ha abierto una planta se entiende que nuestros costes totales de apertura de plantas son “500 uds”. En el segundo problema son 3 las plantas que se establecen por lo tanto nuestros costes son “150 uds”. Si comparamos ambos óptimos, que serían nuestros costes totales, y les restamos los costes fijos de establecimiento de plantas.

Cuadro 4.8: Diferencia de costes

Problema	Óptimo	Coste de establecimiento	Coste de transporte
1	863	- 500	= 363
2	271	- 150	= 121

Al ver el cuadro 4.8 podemos sacar la conclusión de que aunque en el segundo problema aunque se establecen más plantas que en el primero, al tener unos costes de apertura más reducidos las distancias de servicio de las plantas a los clientes se reducen considerablemente. Por lo tanto es más eficiente.

4.1.2. Ejemplo ficticio

Una vez realizado el ejemplo real vamos a proceder a realizar el ejemplo ficticio.

Para este modelo hemos usado la siguiente base de datos:

`http://people.brunel.ac.uk/~mastjjb/jeb/orlib/files/cap41.txt`

Consta de 16 posibles localizaciones donde ubicar las plantas y de 50 clientes. Por cada cliente vamos a tener unos costes derivados de la demanda de producto que solicite y su gasto de transporte. Por cada planta vemos que tenemos un coste fijo de establecimiento. El cuadro 4.9 es un ejemplo de los datos que usaremos

Cuadro 4.9: Costes de apertura y asignación

plantas	Coste apertura	Clientes	Coste envío
y_1	7500	$x_{1,1}$	6739.72500

$\mathbf{y_2}$	7500	$\mathbf{x_{2,1}}$	10355.05000
$\mathbf{y_3}$	7500	$\mathbf{x_{3,1}}$	7650.40000
*	*	*	*
$\mathbf{y_{16}}$	7500	$\mathbf{x_{16,50}}$	7448.10000

```
#Cargar paquetes
library(readr)# Usamos el paquete para la lectura de datos
library(Rglpk)# Paquete encargado de solucionar el problema
```

```
#Lectura de datos

Distancias_SPLP <- read_tsv(
  "Directorio de datos",
  col_names = TRUE,
  cols(capacidad = col_integer(),
       coste_fijo = col_integer()))
Coste_Apertura_SPLP <- read_csv(
  "Directorio de datos",
  col_names = TRUE,
  cols(coste_transporte = col_double()))
```


En un primer instante cargamos las librerias necesarias para resolver nuestro problema y cargamos los datos expuestos con anterioridad. En este caso lo hago mediante dos ficheros. En el primero aparecen los costes fijos de apertura y en el segundo los costes de transporte de los clientes.

```
I<-16
#Posibles localizaciones
J<-50
#Posibles clientes
```

En este paso definimos nuestras variables, el número de localizaciones y clientes.

```
obj<- c(Distancias_SPLP$coste_transporte,Coste_Apertura_SPLP$coste_fijo)
#coeficientes de la función objetivo
rhs<- c(rep (1, J), rep(0,I*J))
#Coeficientes de la derecha de las restricciones
dir<- c(rep("==", J),rep("<=", I*J))
#Son los iguales, o mayores iguales, o menores
types<- c(rep("B",(I*J+I)))
#Tipo de variable
```

En este paso definimos los coeficientes que acompañaran a nuestra función objetivo, los que estén a la derecha de nuestras restricciones así como todas las igualdades que hayan en las mismas. Por último detallamos que las variables $\mathbf{x_{ij}}$ y $\mathbf{y_i}$ són de tipo binarias.

```
mat <- matrix (0, J+I*J, I*J+I)
#Matriz
```

Especificamos las dimensaiones de nuestra matriz, en este caso constará de 850 filas y 816 columnas.

```
  for(i in 1:I){
      for(j in 1:J){
          mat[j,J*(i-1)+j]<-1 #Primera restriccion
        }
    }

  for(i in 1:(I*J)){ #Segunda restriccion
    mat[J+i,i] <-1

  }

  for(i in 1:I){ #Segunda restriccion
    for(j in 1:J){
        mat[J+(J*(i-1)+j),(I*J)+i]<- -1
      }
  }
```

Una vez nuestra matriz está creada procederemos a rellenarla y ejecutamos el comando para ver las solciones.

```
solucion<- Rglpk_solve_LP(obj, mat,dir,rhs, types, max = FALSE)
solucion$optimum
```

Como solución al problema en el cuadro 4.10 podemos ver que 4 son las plantas que se han abierto.

Cuadro 4.10: Plantas abiertas

plantas abiertas $\mathbf{y_i}$	y_2	y_3	y_4	y_{16}

En el **anexo** el cuadro 5.1 podemos ver que la planta $\mathbf{y_3}$ es a la que más clientes se le asocia con un total de 19. Los gastos totales de transporte de de productos y de apertura de plantas asciende a "106677 uds".

4.2. Problema de localización de plantas de la p-mediana

4.2.1. Ejemplo real

```
library(readr)# Usamos el paquete para la lectura de datos
library(Rglpk)# Paquete encargado de solucionar el problema
```

```
Distancia_entre_ciudades <- read_tsv("Directorio de datos",
  col_names = TRUE,
  cols(Ciudad_Origen = col_character(),
  Ciudad_Destino = col_character(),
  Distancia = col_integer()
))
```

Al igual que en el problema anterior se usará un documento de texto en el que tendremos nuestro datos. En esta ocasión solo usaremos el documento "Distancia entre ciudades.txt", cuadro 4.2, ya que en el problema de la **p - mediana** no tenemos costes de establecimiento de plantas.

```
I<-10
#Posibles localizaciones
J<-10
#Posibles clientes
p<-4
#Numero de medianas
```

Establecemos que tendremos 10 posibles plantas y 10 clientes a los que abastecer. En este caso nuestra constante p será 4 que es el número de plantas que quiero abrir.

```
obj<- c(Distancia_entre_ciudades$Distancia,rep(0,I))
#Vector coeficientes de la funcion objetivo
rhs<- c(rep (1, J), p, rep(0,I*J))
#Coeficientes de la derecha de las restricciones
dir<- c(rep("==", J + 1),rep("<=", I*J))
#Son los iguales, o mayores iguales, o menores
types<- c(rep("B",(I*J+I)))
#Tipo de variable
```

Definimos nuestra función objetivo usando la variable de distancia entre las ciudades. Se establece tambén los coeficientes de la derecha de las restricciones al igual que las igualdades. Por último se establece que las variables $\mathbf{x_{ij}}$ y $\mathbf{y_i}$ son de tipo binaria.

```
mat <- matrix (0,J+1+I*J, I*J+I) #Matriz
```

Creamos una matriz de ceros de 111 filas y 110 columnas. Procedemos a rellenarla con las restricciones.

```
for(i in 1:I){
    for(j in 1:J){
      mat[j,J*(i-1)+j]<-1 #Primera restriccion
    }
  }

for(i in 1:I){
```

```
  mat[1+J,I+I*(j-1)+i]<-1 #Segunda restriccion
}

for(i in 1:(I*J)){ #Tercera restriccion
  mat[J+1+i,i] <-1
}

for(i in 1:I){
  for(j in 1:J){
    mat[J+1+(J*(i-1)+j),(I*J)+i]<- -1 #Tercera restriccion
  }
}
```

```
solucion<- Rglpk_solve_LP(obj, mat,dir,rhs, types, max = FALSE)
solucion$optimum
```

Como en nuestro problema establecimos que nuestra constante p sea 4, en el cuadro 4.11 podemos ver cuáles son las plantas que se han establecido.

Cuadro 4.11: Plantas abiertas

plantas abiertas $\mathbf{y_i}$	1 (Orihuela)	2 (Elche)	8 (Torrevieja)	9 (Benidorm)

Cuadro 4.12: Asignación de plantas

Código	Ciudad	Asignaciones
1	Orihuela	1 (Orihuela)
2	Elche	2 (Elche)
3	Alicante	2 (Elche)
4	Altea	9 (Benidorm)
5	Crevillente	2 (Elche)

6	Callosa	1 (Orihuela)
7	Aspe	2 (Elche)
8	Torrevieja	8 (Torrevieja)
9	Benidorm	9 (Benidorm)
10	Villajoyosa	9 (Benidorm)

En nuestra cuadro 4.12 vemos que Orihuela es la planta que se establece más al sur de la provincia. Elche en este caso será la más centrica y la que mayor rango de clientes tenga. En el norte vemos a Benidorm. Por último al estar en la costa y más apartado al resto se abre Torrevieja que en este caso solo se abastecerá así misma. Nuestros costes totales serán de “90 uds”.

4.2.2. Ejemplo ficticio

En este ejemplo ficticio usaremos una base de datos distinta, obtenida de la siguiente página.

`http://www.math.nsc.ru/AP/benchmarks/UFLP/Euclid/111Eucl.txt`

En ella podemos visualizar que hay 100 posibles plantas y 100 clientes. Los costes fijos no los tendremos en consideración y nuestra p tendrá un valor de 10. Los datos se presentarán como en el cuadro 4.13.

Cuadro 4.13: Costes de envío

plantas	Clientes	Coste envío
$\mathbf{y_1}$	$\mathbf{x_{1,1}}$	0
$\mathbf{y_2}$	$\mathbf{x_{2,1}}$	4838
$\mathbf{y_3}$	$\mathbf{x_{3,1}}$	2724
*	*	*
$\mathbf{y_{100}}$	$\mathbf{x_{100,100}}$	0

```r
library(readr)# Usamos el paquete para la lectura de datos
library(Rglpk)# Paquete encargado de solucionar el problema
```

```
Distancia_Pmediana <- read_tsv(
  "Directorio de datos",
  col_names = TRUE,
  cols(Facility = col_integer(),
  Client = col_integer(),
  coste_transporte = col_integer()))
```

Hacemos una lectura de los datos, en este caso yo los tenía en un documento de texto que creé llamado "Distancia Pmediana.txt".

```
I<-100
#Posibles localizaciones
J<-100
#Posibles clientes
p<-10
#Numero de medianas
```

Establecemos que tendremos 100 posibles plantas y 100 clientes a los que dar servicio. En este problema solo queremos que se abran 10 plantas de las posibles 100.

```
obj<- c(Datos_Math_Pmediana$coste_transporte,rep(0,I))
#Vector coeficientes de la funcion objetivo
rhs<- c(rep (1, J), p, rep(0,I*J))
#Coeficientes de la derecha de las restricciones
dir<- c(rep("==", J + 1),rep("<=", I*J))
#Son los iguales, o mayores iguales, o menores
types<- c(rep("B",(I*J+I)))
#Tipo de variable
```

Ahora definimos nuestra función objetivo usando la variable de costes de transporte de las posibles plantas a los clientes. Se establece también los coeficientes de la derecha de las restricciones al igual que las igualdades. Por último se establece que las variables x_{ij} y y_i son de tipo binaria.

```
mat <- matrix (0, J+1+I*J, I*J+I) #Matriz
```

Nuestra matriz tendrá unas dimensaiones de 10101 filas y 10100 columnas y la formamos entera de ceros para luego ir rellenándola.

```
  for(i in 1:I){
    for(j in 1:J){
      mat[j,J*(i-1)+j]<-1 #Primera restriccion
    }
  }

for(i in 1:I){
  mat[1+J,I+I*(j-1)+i]<-1 #Segunda restriccion
}

for(i in 1:(I*J)){ #Tercera restriccion
  mat[J+1+i,i] <-1
}

for(i in 1:I){ #Tercera restriccion
  for(j in 1:J){
    mat[J+1+(J*(i-1)+j),(I*J)+i]<- -1
  }
}
```

Rellenamos nuestra matriz con las restricciones del problema.

```
solucion<- Rglpk_solve_LP(obj, mat,dir,rhs, types, max = FALSE)
solucion$optimum
```

Como solución en el cuadro 4.14 obtenemos que 10 plantas serán opertativas.

Cuadro 4.14: Plantas abiertas

plantas abiertas $\mathbf{y_i}$	y_4	y_{10}	y_{11}	y_{19}	y_{31}	y_{32}	y_{44}	y_{53}	y_{62}	y_{98}

En la parte del **anexo** el cuadro 5.2 muestra las asignaciones a los 100 clientes. Podemos visualizar que la planta y_{11} es la que más demanda tiene, se le asignan hasta 20 clientes. Como solución al problema vemos que los gastos totales ascienden a “67588 uds”.

4.3. Problema de localización de plantas con capacidades

4.3.1. Ejemplo real

Al igual que en los ejemplos reales de problemas de localización de plantas simple y de problema de localización de p-mediana usaremos la misma base de datos sobre las ciudades, los documentos “Distancia entre ciudades.txt” y “Coste apertura ciudades.txt”. Dentro del documento “Coste apertura ciudades.txt” usaremos dos variables nuevas que serán la demanda de producto de cada ciudad y el valor máximo de producción de cada planta. Las nuevas variables en este caso son inventadas para solucionar el problema. Se pueden visualizar de la siguiente manera en el cuadro 4.15.

Cuadro 4.15: Demanda y producción

Ciudad	Demanda	Valor máximo de producción
Orihuela	1000	1000
Elche	1000	5000
Alicante	1000	1000
*	*	*
Villajoyosa	1000	1000

```r
#Cargar paquetes
library(readr)# Usamos el paquete para la lectura de datos
library(Rglpk)# Paquete encargado de solucionar el problema
```

```r
#Lectura de datos
Distancia_entre_ciudades <- read_tsv("Directorio de datos",
  col_names = TRUE,
```

```
  cols(Ciudad_Origen = col_character(),
  Ciudad_Destino = col_character(),
  Distancia = col_integer()
))
Coste_Apertura_Ciudades <- read_tsv("Directorio de datos",
  col_names = TRUE,
  cols(Ciudades = col_character(),
  Coste_fijo = col_integer()
))
```

Hacemos una lectura de los datos, cargamos los archivos “txt”.

```
I<-10
#Posibles localizaciones
J<-10
#Posibles clientes
```

Establecemos que tendremos 10 posibles plantas y 10 clientes a los que abastecer.

```
obj<- c(Distancia_entre_ciudades$Distancia,Coste_Apertura_Ciudades$Coste_fijo2
#Vector coeficientes de la funcion objetivo
rhs<- c(rep (1, J), rep(0,I))
#Coeficientes de la derecha de las restricciones
dir<- c(rep("==", J),rep("<=", I))
#Son los iguales, o mayores iguales, o menores
types<- c(rep("B",(I*J+I)))
#Tipo de variable
```

En este paso definimos los coeficientes que acompañaran a nuestra función objetivo, los que estén a la derecha de nuestras restricciones así como todas las igualdades que hayan en las mismas. Por último detallamos que las variables x_{ij} y y_i són de tipo binarias.

```
mat <- matrix (0, J+I, I*J+I)
#Matriz
```

Definimos que nuestra matriz tendrá 20 filas y 110 columnas y lo vamos rellenando con nuestras restricciones.

```
for(i in 1:I){
  for(j in 1:J){
    mat[j,J*(i-1)+j]<-1  #Primera restriccion
  }
}

for(i in 1:(J)){ #Segunda restriccion
  for(j in 1:J){
  mat[J+i,J*(i-1)+j] <- Coste_Apertura_Ciudades$demanda[i]

  }
}

for(i in 1:I){ #Segunda restriccion
  for(j in 1:J){
    mat[J+i,(I*J)+i]<- - Coste_Apertura_Ciudades$valor_maximo_produccion[i]
  }
}
```

```
solucion<- Rglpk_solve_LP(obj, mat,dir,rhs, types, max = FALSE)
solucion$optimum
```

En el cuadro 4.16 vemos que 3 son las plantas que se han establecido. Hay que inicidir que aunque en este caso sí ha ido bien, en ocasiones este paquete puede no reconocer bien el vector *types* y por lo tanto no se daría la solución deseada. Se recomienda siempre revisar.

Cuadro 4.16: Apertura de plantas

plantas abiertas $\mathbf{y_i}$	2 (Elche)	9 (Benidorm)	10 (Villajoyosa)

Cuadro 4.17: Asignación de plantas

Código	Ciudad	Asignaciones
1	Orihuela	2 (Elche)
2	Elche	2 (Elche)
3	Alicante	2 (Elche)
4	Altea	9 (Benidorm)
5	Crevillente	2 (Elche)
6	Callosa	2 (Elche)
7	Aspe	2 (Elche)
8	Torrevieja	8 (Torrevieja)
9	Benidorm	9 (Benidorm)
10	Villajoyosa	9 (Benidorm)

En nuestro cuadro 4.17 podemos visualizar que Elche es la planta a la que se le han asignado más clientes, un total de 6. En el norte vemos a Benidorm con 3 clientes asociados. Por último Torrevieja es la única planta que se abre para abastecerse a sí misma. Nuestros costes totales son de “300 uds”.

4.4. Problema de localización del p-centro

4.4.1. Ejemplo real

```
#Cargar paquetes
library(readr)# Usamos el paquete para la lectura de datos
library(Rglpk)# Paquete encargado de solucionar el problema
```

```
#Lectura de datos
Distancia_entre_ciudades <- read_tsv("Directorio de datos",
  col_names = TRUE,
  cols(Ciudad_Origen = col_character(),
  Ciudad_Destino = col_character(),
```

```
  Distancia = col_integer()
))
```

Cargamos en el sistema y leemos nuestros ficheros de datos.

```
I<-10
#Posibles localizaciones
J<-10
#Posibles clientes
p<-6
#Numero de plantas para localizar
```

Establecemos que tendremos 10 posibles plantas, 10 clientes a los que abastecer. En este caso la constante p será 6 que es el número de plantas que se desea abrir.

```
obj<- c(rep(0,I*J),rep(0,I),1)
#Vector coeficientes de la funcion objetivo
rhs<- c(p,rep (1, J),rep(0,I*J),rep(0,J))
#Coeficientes de la derecha de las restricciones
dir<- c(rep("==",1 + J),rep("<=", I*J),rep(">=", J))
#Son los iguales, o mayores iguales, o menores
types<- c(rep("B",(I*J+I)),"C")
#Tipo de variable
```

Definimos los coeficientes de nuestra función objetivo, al igual que los coeficientes que estarán a la derecha de las restricciones y las igualdades que hayan en las mismas. Establecemos que nuestras variables x_{ij} e y_i serán de tipo binario y que nuestra variable w será de tipo continuo.

```
mat <- matrix (0,1+J+I*J+J, I*J+I+1) #Matriz
```

Establecemos que nuestra matriz tendra 121 filas y 111 columnas y la completamos con las restricciones.

```
for(i in 1:I){
    mat[1,I*J+i]<-1 #Primera restriccion
  }
```

```
for(i in 1:I){
  for(j in 1:J){
    mat[1+j,J*(i-1)+j]<-1 #Segunda restriccion
  }
}

for(i in 1:(I*J)){ #Tercera restriccion
  mat[1+J+i,i] <-1

}

for(i in 1:I){ #Tercera restriccion
  for(j in 1:J){
    mat[1+J+(J*(i-1)+j),(I*J)+i]<- -1
  }
}

for(i in 1:I){ #Cuarta restriccion
  for(j in 1:J){
    mat[1+J+I*J+j,I*J+I+1]<- 1
  }
}

for(i in 1:J){ #Cuarta restriccion
  for(j in 1:J){
    mat[1+J+I*J+j,J*(i-1)+j]<- - (Distancia_entre_ciudades$Distancia[J*(i-1)+j
  }
}
```

```
solucion<- Rglpk_solve_LP(obj, mat,dir,rhs, types, max = FALSE)
solucion$optimum
```

En el cuadro 4.18 vemos las 6 plantas que se han establecido. Al igual que en el ejercicio anterior hay que inicidir que en ocasiones este paquete puede no reconocer bien el vector *types* y por lo tanto no se daría la solución deseada. Se recomienda siempre revisar.

Cuadro 4.18: Apertura de plantas

plantas abiertas y_i	1 (Orihuela)	2 (Elche)	3 (Alicante)	4 (Altea)	8 (Torrevieja)	10 (Villajoyosa)

Cuadro 4.19: Asignación de plantas

Código	Ciudad	Asignaciones
1	Orihuela	1 (Orihuela)
2	Elche	2 (Elche)
3	Alicante	3 (Alicante)
4	Altea	4 (Altea)
5	Crevillente	2 (Elche)
6	Callosa	1 (Orihuela)
7	Aspe	2 (Elche)
8	Torrevieja	8 (Torrevieja)
9	Benidorm	10 (Villajoyosa)
10	Villajoyosa	10 (Villajoyosa)

En el cuadro 4.19 vemos que Elche es la planta que ha sido asignada a más clientes, un total de 3. Orihuela se establece en el sur con 2 clientes y Villajoyosa en el norte con otros 2. Alicante, Altea y Torrevieja solo se establecen a sí mismas. El óptimo en este caso es de “15 uds”.

4.5. Conclusiones

En conclusión en este trabajo final de grado se han realizado 6 ejemplos distintos de 4 modelos diferentes. En el problema de localización de plantas simple se realizaron un ejemplo real y otro ficticio. La matriz del ejemplo real tenía unas dimensiones de 110 filas y 110 columnas y el ficticio su matriz tenía 850 filas y

816 columnas. En el problema de localización de plantas de la p-mediana también se realizaron un ejemplo real y otro ficticio. En el ejemplo real las dimensiones de la matriz eran de 111 filas y 110 columnas y la del ejemplo ficticio eran de 10101 filas y 10100 columnas. Los problemas de localización de plantas con capacidades y de localización del p-centro solo tenían un ejemplo real. Las dimensiones de la matriz del primero son de 20 filas y 110 columnas y las del segundo son de 121 filas y 110 columnas.

Aunque hemos visto que se han realizado diversos ejemplos con dimensiones muy diferentes, he de hacer incapié en que no se observa diferencia alguna entre el tiempo de ejecución de cada problema. Desde mi ordenador no varía de los pocos segundos, entre 1 y 3 segundos.

Capítulo 5

Anexos

Cuadro 5.1: Asignación de plantas

Planta	Asignaciones
2	15, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 30
3	29, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50
4	1, 2, 3, 4, 5, 6, 7, 8, 10
16	9, 11, 12, 13, 14, 16, 31

Cuadro 5.2: Asiganción de plantas

Planta	Asignaciones
4	3, 4, 13, 22, 43, 47, 67, 71
10	10, 20, 23, 28, 39, 66, 81, 82, 91, 94
11	2, 5, 9, 11, 14, 16, 21, 34, 37, 40, 42, 49, 55, 57, 59, 63, 69, 70, 74, 84
19	15, 19, 41, 80, 86, 90
31	31, 46, 60, 72, 83, 85, 88, 96, 99
32	24, 25, 27, 32, 45, 54, 65, 73, 78, 93, 97
44	6, 7, 12, 17, 26, 29, 36, 44, 48, 56, 58, 68, 75, 77, 87
53	30, 33, 38, 52, 53, 64, 76, 89, 100

62	35, 61, 62, 92, 95
98	1, 8, 18, 50, 51, 79, 98

Bibliografía

[1] Araneda Martínez, R.H. y Moraga Suazo, R.J. (2004). “LA DECISIÓN DE LOCALIZACIÓN EN LA CADENA DE SUMINISTRO” en *Revista Ingeniería Industrial - Año 3, N°1*; p. 57-67.

[2] Beresnev, V. (2018). Discrete Location Problems: Benchmark Library. <http://www.math.nsc.ru/AP/benchmarks/english.html>.

[3] Chang, W. y Lee, S. (2005). On solving the discrete location problems when the facilities are prone to failure; p. 817-831.

[4] Charles S. R. y Gilbert L. (1996). The Plant Location Problem: New Models and Research Prospects. Operations Research 44(6):864-874. <http://dx.doi.org/10.1287/opre.44.6.864>

[5] Ferreira Silva, F.J. y Serra, D. (2005). A Capacitated Facility Location Problem with Constrained Backlogging Probabilities; p. 1-30.

[6] García Quiles, S. (2017). “Localización de concentradores.en *Avances en localización de servicios y sus aplicaciones*, B. Pelegrín. Servicio de publicaciones de la universidad de Murcia. p. 35-57.

[7] Goldengorin, B. Tijssen, G. Ghosh, D. y Sierksma, G. Solving the Simple Plant Location Problem using a Data Correcting Approach; p. 377-406.

[8] Hernández Ramírez, C.M. (2004). Diseño de un algoritmo heurístico para el problema de localización p-mediana. Tesis. Puebla: Universidad de las Américas Puebla.

[9] Krarup, J. y Pruzan, P.M. (1981). “The simple plant location problem: Survey and synthesis” en *European Journal of Operational Research*; p. 36-81.

[10] Marín, A. Cánovas, L. y Landete, M. (2004). New formulations for the uncapacitated multiple allocation hub location problem; p. 274-292.

[11] Verter, Vedar. (2017). “Chapter 2: Uncapacitated and Capacitated Facility Location Problems”. <https://www.semanticscholar.org/paper/Chapter-2-Uncapacitated-and-Capacitated-Facility-Verter/787f3d091bd0255532d803d01d79808822867722>