



**Universidad**  
Zaragoza

1542

**MÁSTER EN MODELIZACIÓN E INVESTIGACIÓN MATEMÁTICA,  
ESTADÍSTICA Y COMPUTACIÓN**

Curso 2015-2016

---

**PROBLEMAS DE LOCALIZACIÓN Y RUTAS EN  
LA GESTIÓN DE RESIDUOS SÓLIDOS**

---

Trabajo Fin de Máster realizado por Lidia Orellana

Dirigido por Herminia I. Calvete Fernández



*Problemas de localización y rutas en la gestión de residuos sólidos*

Autora: Lidia Orellana Lozano

Directora: Herminia I. Calvete Fernández



# Summary

This work is framed within Mathematical Programming, more specifically in Integer Programming.

In general, Mathematical Programming is the branch of mathematics that deals with the treatment of problems that can be formulated as the search for a global optimum (maximum or minimum) of a certain function which represents the objective to be achieved, subject to several constraints which represent the conditions imposed in order to the system works.

In optimization problems, we often found problems where the variables involved have to take integer values. For example, if we have to decide the number of cars manufactured by a company in order to maximize the profit, or the number of people working in a company to minimize costs, we have to handle integer variables. So naturally appear such problems, that are included within what is known as Integer Programming.

The aim of this work is to provide an introduction to Integer Programming in order to be able to solve a practical case of facility location in the Community of Aragón.

The work is organized as follows. It consists of a central part, which includes four chapters that we will detail below, and two annexes.

In Chapter 1 we make an introduction to Integer Programming. The general formulation of these problems is briefly introduced and the main ideas of some of the methods that have been proposed in the literature for solving integer programming problems are showed. There are different techniques to address Integer Programming. We study three of the methods used in solving integer programming problems such as “Branch-and-bound”, “Branch-and-cut” and “Branch-and-cut-and-price”. These techniques are linked to the concept *divide and conquer* rule, i.e, if we are not able to solve a particular integer problem, we divide our problem into subproblems which are easier to be solved. This scheme can be represented by a tree whose root corresponds to the original problem and its branches are the result of the division into parts of the set of feasible solutions. The main references of the chapter are the books of M. Conforti, G. Cornuejols and G. Zambelli, [9], and C. Prins, C. Prodhon and R.W. Calvo, [22].

In Chapter 2 we focus on a type of optimization problems known as location and routing problems. The location and routing problem includes two types of fundamental problems in supply chain management: the problem of facility location and the vehicle routing problem. The different aspects of these two issues as location, allocation and routing have been studied

independently. The location of warehouses and depots plays a critical role, for example in the strategic network design of the supply chain. An example of a location problem with great presence in the literature, because of its simple design and the large number of real situations that can be adjusted to it, is the p-median problem. This problem only takes into account the location of a number of depots or plants. A variant of the problem appears when we take into account the capacity of the depots. The objective remains to minimize the weighted distances between the depots and the customers. This version is known as the capacitated p-median problem (CPMP). We offer a literature review and we give a summary of some selected articles related to location problems ([3], [4], [5], [11], [14], [19], [30] and [33]).

An example of a problem in which the location of the plants and the routes of the vehicles are combined is the Capacitated Location-Routing Problem (CLRP). Among its several versions, the one which is taken into account in this work deals with warehouses with capacity, routes with capacity, and fixed costs to open a depot or a route. In addition, the fleet is homogeneous, i.e, all vehicles have the same capacity. To solve a CLRP means to determine which depots should be opened, assign each client to an open depot and build the routes for vehicles from each depot to its customers while minimizing the total cost. We formulate this problem and we give a detailed list of papers related to the problem CLRP ([2], [7], [10], [12], [13], [20], [21] and [32]).

One of the applications of location problems and location and routing problems is waste management. In Chapter 3 we focus on the study of such problems, which is the topic of this work. We make an study of recent literature and we give a brief classification based on the type of problem or characteristics – type of waste material. We deal with papers which study multiobjective location problems, as [1] and [26]. In these papers, the autors take into account social and environmental factors related to the availability and the quality of human capital or quality of life. We also found differences depending on the type of waste material. Other models arise when the nature of the waste material is hazardous or undesirable, such as the offered in [8] and [29], or when we deal with urban waste ([6] and [15]).

Finally, in Chapter 4 we solve a realistic situation applied to the Community of Aragon. After the search of the specifical data – obtained mostly from the web page of the Aragon Institute of Statistics – we provide an overview of the current situation in the issue related to municipal solid waste in Aragón. We apply different location models for determining the opened landfills in the different provinces. Finally, based on the solution obtained in the location problem, we solve a routing problem. In the last section we give the formulation of the vehicle routing problem and we calculate the routes of the vehicles with the support of a *solver* called ‘VRP Spreadsheet Solver’. The main reference of this section is the book written by P. Toth and D. Vigo [31].

The data that have been collected from various sources cited in the last chapter, and which have been used for solving the problems, are displayed in the first of the annexes, Annex [A](#).

CPLEX® is used for solving the optimization models proposed in the last chapter. Also, we use the Basic User Guide for IBM ILOG CPLEX use V12.6.1 made for the collaboration scholarship with departments in the university departments convened by the Ministry of Education, Culture and Sport (MECD) done during the course 2015-2016, [\[23\]](#). The syntax of the implemented model and data files in CPLEX® are shown in Annex [C](#).

Zaragoza, 14 September 2016



# Resumen

El presente trabajo queda enmarcado dentro de la Optimización Matemática, más específicamente dentro de la Optimización entera.

En general, la Optimización Matemática es la rama de las Matemáticas que aborda el tratamiento de problemas que pueden formularse como la búsqueda de un óptimo global (un máximo o un mínimo) de una cierta función que representa el objetivo que se desea conseguir, sujeto a una serie de restricciones que representan las condiciones impuestas para que el sistema tratado funcione.

En los problemas de optimización, con frecuencia nos encontramos con problemas en los que las variables involucradas han de tomar valores enteros. Por ejemplo, si hemos de decidir el número de coches que fabrica una empresa en busca de maximizar el beneficio o el número de personas que se ha de tener en plantilla para minimizar costes tenemos que manejar variables enteras. Así, aparecen de forma natural este tipo de problemas que quedan englobados dentro de lo que se conoce como Programación Entera.

El propósito principal del trabajo es ofrecer una introducción a la Optimización Entera para resolver un caso práctico de localización de instalaciones en la Comunidad de Aragón.

El trabajo está organizado como sigue. Consiste de una parte central, que incluye cuatro capítulos que se detallan a continuación y de dos anexos.

En el Capítulo 1 se realiza una introducción a la Programación Entera. En él, se introduce brevemente la formulación general de este tipo de problemas y se ofrecen las ideas principales de algunos de los métodos que se han propuesto en la literatura para la resolución de problemas de programación entera. Existen distintas técnicas para abordar los problemas de programación entera. Trataremos tres de los métodos utilizados en la resolución de problemas de programación entera como son los métodos “Branch-and-bound”, “Branch-and-cut” y “Branch-and-cut-and-price”. Estas técnicas están ligadas al concepto *divide y vencerás*, esto es, si no somos capaces de resolver un determinado problema entero, dividimos nuestro problema en subproblemas que sí sepamos resolver. Este esquema se puede representar mediante un árbol cuya raíz corresponde al problema original y sus ramas resultan de la división en partes del conjunto de soluciones factibles. Las referencias principales que se han utilizado en este capítulo han sido los textos de M. Conforti, G. Cornuejols y G. Zambelli, [9], y C. Prins, C. Prodhon y R. W. Calvo, [22].

En el Capítulo 2 nos centramos en un tipo de problemas de optimización conocidos como problemas de localización y rutas. El problema de localización y rutas incluye dos tipos de problemas fundamentales en gestión de la cadena de suministro: el problema de localización de instalaciones y el problema de rutas de vehículos. Los diferentes aspectos de estos dos problemas como la localización, la asignación y las rutas han sido muy estudiados de forma independiente. La localización de almacenes o depósitos juega un papel crítico, por ejemplo, en el diseño estratégico de las redes de la cadena de suministro. Un ejemplo de problema de localización con gran presencia en la literatura debido a su sencilla formulación y a la gran cantidad de situaciones reales que se puedan ajustar a él es el problema de la p-mediana. Se trata de un problema en el que únicamente se tiene en cuenta la localización de una serie de almacenes o plantas. Otra versión del problema aparece cuando se tienen en cuenta las capacidades de los almacenes. El objetivo sigue siendo minimizar las distancias ponderadas entre almacenes y clientes. A esta versión se le conoce como el problema de la p-mediana con capacidades (CPMP). Ofrecemos una revisión bibliográfica en la que se da un resumen sobre una serie artículos seleccionados relacionados con el problema de la p-mediana y la localización de plantas, ([3], [4], [5], [11], [14], [19], [30] y [33]).

Un ejemplo de problema en el que se combina la localización de las plantas y las rutas de vehículos es el problema CLRP (Capacitated Location-Routing Problem). Entre sus diversas versiones, la que aquí tratamos tiene en cuenta almacenes con capacidad, rutas con capacidad y costos fijos para abrir un almacén o una ruta. Además, la flota de vehículos es homogénea, es decir, todos los vehículos tienen la misma capacidad. La solución de un problema CLRP consiste en determinar qué almacenes deben abrirse, asignar a cada cliente un almacén abierto y construir una ruta para los vehículos desde cada almacén hasta sus clientes minimizando el coste total. Formulamos este problema y damos un listado detallado de artículos relacionados con el problema CLRP, ([2], [7], [10], [12], [13], [20], [21] y [32]).

Una de las múltiples aplicaciones de los problemas de localización y de los problemas de localización y rutas introducidos en el capítulo anterior es la gestión de los residuos. En el capítulo 3 nos centraremos en el estudio de este tipo de problemas, que constituye el tema central del trabajo. Realizamos un estudio de la bibliografía reciente y una breve clasificación basado en el tipo de problema o sus características – tipo de residuo tratado. Tratamos artículos de localización multiobjetivo, como [1] y [26], en los que se tienen en cuenta factores sociales y medioambientales, relacionados con la disponibilidad y calidad del capital humano o la calidad de vida, entre otros. Encontramos también diferencias dependiendo del tipo de residuo tratado. Surgen otro tipo de modelos cuando los residuos con los que se trata son peligrosos o indeseables, como los ofrecidos en [8] y [29], o en el tratamiento de los residuos urbanos [6] y [15].

Finalmente, en el capítulo 4 se resuelve una situación realista aplicada a la Comunidad de Aragón. Tras la búsqueda de datos – extraídos en su mayoría de la página web del Instituto Aragonés de Estadística – se ofrece una visión general de la situación actual en el tema relacionado con los residuos sólidos urbanos en Aragón. Aplicamos distintos modelos de localización para determinar la apertura de vertederos en las distintas provincias aragonesas. Por último, partien-

do de la solución obtenida en el problema de localización, resolveremos un problema de rutas. En la última sección se introduce la formulación de un problema de rutas y calcularemos con la ayuda de un *solver* llamado ‘VRP Spreadsheet Solver’ las rutas de los vehículos. La referencia principal de la sección es el libro de P. Toth y D. Vigo [31].

En el primero de los anexos, Anexo A, aparecen reflejados los datos que han sido recopilados de las distintas fuentes citadas en el capítulo 4 y que han sido utilizados para la resolución de los distintos problemas.

Para la resolución computacional de los modelos planteados en el último capítulo utilizamos CPLEX®. Nos apoyamos en el manual de uso básico para IBM ILOG CPLEX V12.6.1 realizado para la Beca de Colaboración en el curso 2015-2016 en departamentos universitarios convocadas por el Ministerio de Educación, Cultura y Deporte (MECD) [23]. La sintaxis de los archivos de modelo y datos implementados en CPLEX® aparecen reflejados en el Anexo C.

Zaragoza, 14 de septiembre de 2016



# Índice general

<b>1. Introducción a la programación entera</b>	<b>1</b>
1.1. Formulación . . . . .	1
1.2. Tipos de algoritmos . . . . .	3
1.2.1. Branch-and-bound . . . . .	3
1.2.2. Branch-and-cut . . . . .	5
1.2.3. Branch-and-cut-and-price . . . . .	7
<b>2. Modelos de localización y rutas</b>	<b>9</b>
2.1. El problema de la p-mediana . . . . .	9
2.1.1. Revisión bibliográfica . . . . .	13
2.2. El problema de localización y rutas con capacidades . . . . .	17
2.2.1. Revisión bibliográfica . . . . .	20
<b>3. Modelos en la gestión de residuos</b>	<b>25</b>
3.1. Características del problema . . . . .	25
3.2. Modelos de localización multiobjetivo . . . . .	26
3.3. Modelos de localización con residuos peligrosos . . . . .	27
3.4. Modelos de localización con residuos urbanos . . . . .	28
<b>4. Resolución de un caso práctico en Aragón</b>	<b>31</b>
4.1. Residuos sólidos urbanos . . . . .	31
4.1.1. Pre-recogida o depósito . . . . .	32
4.1.2. Recogida . . . . .	32
4.1.3. Transporte . . . . .	32
4.1.4. Tratamiento . . . . .	33
4.2. Situación actual en la Comunidad de Aragón . . . . .	33
4.3. Modelo de localización sin Estaciones de Transferencia . . . . .	36
4.3.1. Problema de la p-mediana . . . . .	37
4.3.2. Problema de la p-mediana con capacidades . . . . .	39
4.4. Modelo de localización con Estaciones de Transferencia . . . . .	42
4.4.1. Análisis clúster . . . . .	43
4.4.2. Localización de las ET . . . . .	49
4.4.3. Localización de los vertederos . . . . .	50
4.5. Modelo de rutas . . . . .	53

4.5.1. Problema de Rutas de Vehículos (VRP) . . . . .	53
4.5.2. Resolución . . . . .	55
4.5.3. Resultados . . . . .	56
<b>Anexos</b>	<b>65</b>
<b>A. Datos</b>	<b>67</b>
A.1. Datos de población de la provincia de Zaragoza . . . . .	67
A.2. Datos de población de la provincia de Huesca . . . . .	70
A.3. Datos de población de la provincia de Teruel . . . . .	71
<b>B. Scripts R</b>	<b>75</b>
B.1. Regresión . . . . .	75
B.1.1. Fichero de datos . . . . .	75
B.1.2. Modelo . . . . .	75
B.2. Análisis cluster . . . . .	76
B.2.1. Provincia de Zaragoza . . . . .	76
B.2.2. Provincia de Huesca . . . . .	79
B.2.3. Provincia de Teruel . . . . .	80
<b>C. Scripts CPLEX</b>	<b>83</b>
C.1. Sintaxis del problema de la p-mediana en CPLEX . . . . .	83
C.1.1. Archivo de modelo . . . . .	83
C.1.2. Archivo de datos . . . . .	84
C.2. Sintaxis del problema de la p-mediana con capacidades en CPLEX . . . . .	87
C.2.1. Archivo de modelo . . . . .	87
C.2.2. Archivo de datos . . . . .	87
<b>Bibliografía</b>	<b>89</b>

*Defiende tu derecho a pensar, porque incluso pensar de manera errónea es mejor que no pensar.*

Hipatia de Alejandría

# 1

## Introducción a la programación entera

---

En los problemas de optimización, con frecuencia nos encontramos con problemas en los que las variables involucradas han de tomar valores enteros. Por ejemplo, si hemos de decidir el número de coches que fabrica una empresa en busca de maximizar el beneficio o el número de personas que se ha de tener en plantilla para minimizar costes tenemos que manejar variables enteras. Un tipo particular de estas variables son las variables enteras binarias. Este tipo de variables es muy utilizado en casos en los que se quiere reflejar disyunción o implicación entre restricciones o para linealizar restricciones no lineales presentes en el modelo. Así, aparecen de forma natural este tipo de problemas que quedan englobados dentro de lo que se conoce como programación entera.

Las referencias principales que se han utilizado en este capítulo han sido los textos de M. Conforti, G. Cornuejols y G. Zambelli, [9, Capítulo 1], y C. Prins, C. Prodhon y R. W. Calvo, [22].

El objetivo de este capítulo es introducir brevemente la formulación general de este tipo de problemas y ofrecer las ideas principales de algunos de los métodos que se han propuesto en la literatura para la resolución de problemas de programación entera.

### 1.1. Formulación

Dependiendo de la naturaleza y cantidad de variables enteras se distinguen varios tipos de problemas de programación entera. Un problema de programación entera se conoce como entero puro si todas las variables del problema están restringidas a ser enteras. La formulación general de un problema de programación entera puro viene dada de la siguiente forma

$$\begin{aligned} \max_x \quad & z = c^T x \\ \text{sujeto a:} \quad & \\ & Ax = b \\ & x \geq 0, \text{ entero} \end{aligned}$$

donde los datos del problema se corresponden con el vector de costes  $c = (c_1, \dots, c_n)$ , la matriz de coeficientes  $A = (a_{ij})$  de dimensión  $m \times n$  y el vector  $b^T = (b_1, \dots, b_m)$ . Las variables a optimizar se recogen en el vector  $x^T = (x_1, \dots, x_n)$ . Recordemos que dado un vector  $x \in \mathbb{R}^n$  diremos que es entero cuando  $x \in \mathbb{Z}^n$ . Llamaremos  $S$  al conjunto de todas las soluciones factibles, esto es, el conjunto de vectores que verifican todas las restricciones del problema  $S := \{x \in \mathbb{Z}_+^n : Ax = b\}$ . Para hacer referencia a un problema de este tipo diremos ILP (Integer Linear Programming).

En muchas situaciones reales nos enfrentamos a problemas cuyas variables tienen diferente naturaleza, puede haber un grupo de variables enteras y el resto ser continuas, llamaremos a este problema lineal entero mixto y lo denotaremos en adelante como MILP (Mixed Integer Linear Programming). Se trata de un problema de la forma

$$\begin{aligned} \max_{x,y} \quad & z = cx + hy \\ \text{sujeto a} \quad & \\ & Ax + Gy = b \\ & x \geq 0 \text{ entero} \\ & y \geq 0 \end{aligned} \tag{1.1}$$

donde los datos del problema vienen dados por los vectores de costes  $c = (c_1, \dots, c_n)$  y  $h = (h_1, \dots, h_p)$ , las matrices de coeficientes  $A = (a_{ij})$  de dimensión  $m \times n$  y  $G = (g_{ij})$  de dimensión  $m \times p$  y el vector  $b^T = (b_1, \dots, b_m)$ . Los vectores  $x^T = (x_1, \dots, x_n)$  e  $y^T = (y_1, \dots, y_p)$  contienen las variables a optimizar. Las variables  $x_j$  están restringidas a ser enteras no negativas mientras que las variables  $y_j$  pueden tomar cualquier valor real no negativo. El conjunto  $S$  de soluciones factibles viene dado por  $S := \{(x, y) \in \mathbb{Z}_+^n \times \mathbb{R}_+^p : Ax + Gy = b\}$ .

Otro tipo de problema de programación entera es aquel en el que las variables toman valores en  $\{0, 1\}$ . Este tipo de variables se llaman variables binarias y nos permiten representar, en general, condiciones lógicas. Un problema de programación entera en el cual todas las variables son del tipo anterior se denomina problema de programación entera 0-1 o de programación binaria.

Un concepto importante que aparecerá en lo sucesivo es el de relajación. Denominamos relajación lineal de un problema de programación entera al problema de programación lineal resultante de eliminar las condiciones de integridad de las variables, esto es, se relajan las condiciones de integridad. Por ejemplo, si estamos frente a un MILP, su relajación lineal viene dada por el problema:

$$\max \{cx + hy : (x, y) \in P_0\},$$

donde  $P_0 := \{(x, y) \in \mathbb{R}_+^n \times \mathbb{R}_+^p : Ax + Gy = b\}$ .

De esta manera, se pasa de un problema de programación entera mixta a un problema lineal que puede ser resuelto mediante el algoritmo simplex. Además, si la solución óptima de la relajación lineal es entera, dicha solución es la solución óptima del problema entero. El concepto de relajación lineal aparece muy frecuentemente puesto que su óptimo proporciona una cota superior del problema entero, si estamos en un problema de máximo, o una cota inferior en caso de estar minimizando. La solución del problema lineal relajado es a menudo tomada como solución inicial en algunos métodos iterativos.

## 1.2. Tipos de algoritmos

Existen distintas técnicas para resolver los problemas de programación entera. El objetivo de este apartado es dar una idea general de tres métodos utilizados en la resolución de problemas de programación entera como son los métodos “Branch-and-bound”, “Branch-and-cut” y “Branch-and-cut-and-price”.

La eficacia de un método de resolución de problemas de programación entera reside en la velocidad de convergencia. Cuanto más rápido se alcance la solución óptima entera mejor será el método. Algunos como el llamado “Branch-and-bound” pueden dar lugar a convergencias muy lentas y por ello se ha trabajado en mejoras que buscan ajustar las cotas incorporando lo que se conoce como técnicas de corte. Nacen así técnicas como “Branch-and-cut” o “Branch-and-cut-and-price” siendo éstas dos de las más recientes y con mayor éxito en la resolución de problemas enteros.

Estas técnicas están ligadas al concepto *divide y vencerás*, esto es, si no somos capaces de resolver un determinado problema entero, dividimos nuestro problema en subproblemas que sí sepamos resolver. Este esquema se puede representar mediante un árbol cuya raíz corresponde al problema original y sus ramas resultan de la división en partes del conjunto de soluciones factibles.

Como ya se ha indicado anteriormente, la eficacia de un método de resolución de problemas de programación entera reside en la velocidad de convergencia, esto es, cuanto más rápido se alcance la solución óptima entera mejor será el método. Dada una formulación de un problema de programación entera, se dirá perfecta si el problema puede ser resuelto como un problema de programación lineal, ignorando la integridad de la solución. Esto ocurre cuando el conjunto de soluciones factibles de la relajación lineal del problema entero, al que llamaremos poliedro  $P$ , coincide con el poliedro  $Q$ , envolvente convexa de las soluciones enteras. Esto nos sugiere que cuanto más próximos sean los poliedros  $P$  y  $Q$  mejor convergencia tendrá el método.

### 1.2.1. Branch-and-bound

Antes de describir el algoritmo, damos una idea general sobre el mismo. Sea  $(x^0, y^0)$  la solución de la relajación lineal del problema de programación entera mixto (1.1). Si las componentes del vector  $x$  de la solución son enteras entonces  $(x^0, y^0)$  es de la solución al problema planteado originalmente. Supongamos que existe un índice  $j$ , con  $1 \leq j \leq n$ , tal que  $x_j^0$  es

fraccionario. Consideramos los conjuntos:

$$S_1 := S \cap \{(x, y) : x_j \leq \lfloor x_j^0 \rfloor\}, \quad S_2 := S \cap \{(x, y) : x_j \geq \lceil x_j^0 \rceil\},$$

donde  $\lfloor x_j^0 \rfloor$  denota el mayor entero  $k$  tal que  $k \leq x_j^0$  y  $\lceil x_j^0 \rceil$  denota el menor entero  $l$  tal que  $l \geq x_j^0$ . Notemos que, dado que  $x_j$  ha de ser entero para cada solución factible, se tiene que  $(S_1, S_2)$  es una partición de  $S$ . Si nos restringimos ahora a los nuevos conjuntos  $S_1$  y  $S_2$  obtenemos los siguientes problemas enteros mixtos:

$$\text{MILP}_1 : \max \{cx + hy : (x, y) \in S_1\}, \quad \text{MILP}_2 : \max \{cx + hy : (x, y) \in S_2\}.$$

Ahora, la solución óptima de MILP es la mejor entre las soluciones óptimas de  $\text{MILP}_1$  y  $\text{MILP}_2$ . Por lo tanto, la resolución del problema original se reduce a encontrar la solución óptima de los dos nuevos subproblemas.

El procedimiento que acabamos de describir se basa en buscar la solución óptima por rama y corte, esto es, se divide el conjunto  $S$  de soluciones factibles en subconjuntos  $S_i$  que dan lugar a distintos subproblemas. Cada uno de estos subproblemas se corresponde con un nodo de lo que denominaremos árbol de enumeración. Para cada nodo  $N_i$ , sea  $z_i$  la solución óptima del correspondiente problema lineal  $LP_i$ . Para  $i = 0$  se tiene el problema original relajado  $N_0$ . Denotamos por  $\mathcal{L}$  la lista de nodos (problemas) que todavía no se han resuelto. Sea  $\underline{z}$  una cota inferior del valor óptimo  $z^*$  del problema MILP original. Inicialmente  $\underline{z}$  puede ser fijada a partir de una solución heurística del problema o puede fijarse como  $-\infty$ . Los argumentos de dominancia y factibilidad son los que permitirán descartar ramas del árbol en el proceso de búsqueda.

### Algoritmo Branch-and-bound.

#### 1. Paso inicial

$$\mathcal{L} := \{N_0\},$$

$$\underline{z} := -\infty,$$

$$(x^*, y^*) := \emptyset.$$

#### 2. ¿Termina?

Si  $\mathcal{L} = \emptyset$ , la solución  $(x^*, y^*)$  es óptima.

#### 3. Selección del nodo

Elegimos un nodo  $N_i \in \mathcal{L}$  y lo eliminamos de  $\mathcal{L}$ .

#### 4. Acotación

Resolvemos  $LP_i$ :

- Si  $LP_i$  es no factible, volvemos al Paso 2.
- Si  $LP_i$  es factible, sea  $(x^i, y^i)$  solución óptima de  $LP_i$  y  $z_i$  su valor en la función objetivo.

### 5. Poda del árbol

- Si  $z_i \leq \underline{z}$ , ir al Paso 2.
- Si  $(x^i, y^i)$  es factible para el problema entero, fijar  $\bar{z} := z_i$ ,  $(x^*, y^*) := (x^i, y^i)$  e ir al Paso 2.
- En otro caso, ir al Paso 6.

### 6. Ramificación

Desde  $LP_i$ , construir  $k \geq 2$  problemas lineales  $LP_{i_1}, \dots, LP_{i_k}$  con regiones más pequeñas cuya unión no contenga a  $(x^i, y^i)$ , pero sí contenga todas las soluciones del  $LP_i$  con  $x \in \mathbb{Z}$ . Añadir los correspondientes nodos  $N_{i_1}, \dots, N_{i_k}$  a  $\mathcal{L}$  y volver al Paso 2.

Notemos que en el Paso 3 del algoritmo se ha de tomar una decisión sobre el problema a resolver de la lista  $\mathcal{L}$  de nodos que todavía no se han resuelto. En este sentido, cabe destacar que, dependiendo de la técnica utilizada en la selección, aparecen distintos métodos dentro del algoritmo de ramificación y acotación.

#### 1.2.2. Branch-and-cut

En las técnicas de acotación para la resolución de problemas enteros es crucial el ajuste de las cotas para la poda del árbol. En general, la baja calidad de las cotas obtenidas en la aproximación “Branch-and-bound” en los problemas relajados hacen de éste un procedimiento lento, de modo que se ha trabajado en otros métodos para obtener cotas más ajustadas. Inicialmente, además del algoritmo “Branch-and-bound”, para resolver problemas de programación entera se utilizaban técnicas conocidas como métodos de planos de corte. La idea de estos métodos es similar a la propuesta por el algoritmo “Branch-and-bound”, puesto que resuelve el problema entero mediante una secuencia de resoluciones de problemas lineales resultado de añadir nuevas restricciones válidas al problema, conocidas como planos de corte. Este método fue introducido por Gomory y a los planos de corte se les llama planos de corte de Gomory. El problema de los métodos de plano de corte es que, en la práctica, conforme se va incrementando el número de cortes da lugar a convergencias muy lentas. Surge así la idea de añadir la aproximación del plano de corte a los distintos subproblemas en el algoritmo “Branch-and-bound”. Esto nos conduce a la aproximación “Branch-and-cut”, la cual es considerada una de las más exitosas en la resolución de problemas enteros.

El método “Branch-and-cut” guarda una lista de problemas lineales obtenidos de la relajación lineal de problemas enteros e impone restricciones lineales, como, por ejemplo, cotas sobre las variables  $x_i \leq u_i$  o  $x_i \geq l_i$  con  $i = 1, \dots, n$ .

La notación utilizada en la descripción del algoritmo es la misma que la introducida en el desarrollo del algoritmo “Branch-and-bound”.

#### Algoritmo Branch-and-cut.

##### 1. Paso inicial

$$\mathcal{L} := \{N_0\},$$

$\underline{z} := -\infty$ ,

$(x^*, y^*) := \emptyset$ .

## 2. ¿Termina?

Si  $\mathcal{L} = \emptyset$ , la solución  $(x^*, y^*)$  es óptima.

## 3. Selección del nodo

Elegimos un nodo  $N_i \in \mathcal{L}$  y lo eliminamos de  $\mathcal{L}$ .

## 4. Acotación

Resolvemos  $LP_i$ :

- Si  $LP_i$  es no factible, volvemos al Paso 2.
- Si  $LP_i$  es factible, sea  $(x^i, y^i)$  solución óptima de  $LP_i$  y  $z_i$  su valor en la función objetivo.

## 5. Poda del árbol

- Si  $z_i \leq \underline{z}$ , ir al Paso 2.
- Si  $(x^i, y^i)$  es factible, fijar  $\bar{z} := z_i$ ,  $(x^*, y^*) := (x^i, y^i)$  e ir al Paso 2.
- En otro caso, ir al Paso 6.

## 6. ¿Añadir cortes?

Decidir si:

- Fortalecer la formulación de  $LP_i$ , en cuyo caso añadimos planos de corte y volvemos al Paso 4.
- Ramificar en cuyo caso ir al Paso 7.

## 7. Ramificación

Desde  $LP_i$ , construir  $k \geq 2$  problemas lineales  $LP_{i_1}, \dots, LP_{i_k}$  con regiones más pequeñas cuya unión no contenga a  $(x^i, y^i)$ , pero sí contenga todas las soluciones del  $LP_i$  con  $x \in \mathbb{Z}$ . Añadir los correspondientes nodos  $N_{i_1}, \dots, N_{i_k}$  a  $\mathcal{L}$  y volver al Paso 2.

La decisión de introducir o no nuevos cortes en el sexto paso del algoritmo se toma empíricamente basándose en el éxito de los cortes añadidos anteriormente y las características del nuevo corte, como por ejemplo la fracción de coeficientes no nulos en el corte. Generalmente se añaden varias rondas de cortes en el nodo raíz  $N_0$ , mientras que este número se reduce a medida que se va ramificando el árbol.

La introducción de cada uno de estas desigualdades agregadas –cortes– consigue mejorar la formulación de cada uno de los subproblemas. La mejora se entiende en el sentido anteriormente citado, se consigue una aproximación del poliedro de cada una de las relajaciones lineales al poliedro  $Q$ . De este modo se consigue reducir significativamente el número de nodos del árbol, y por tanto el número de problemas a resolver.

### 1.2.3. Branch-and-cut-and-price

Para introducir el método Branch-and-cut-and-price tenemos que explicar brevemente un método de resolución de problemas de programación lineal: generación de columnas. Encontramos la siguiente explicación en un manual de usuario de CPLEX\*:

En términos coloquiales, la generación de columnas es una forma de comenzar con una parte de un problema pequeña y gestionable (específicamente, unas pocas variables), resolver dicha parte, [...] y, a continuación, resolver el modelo ampliado. La generación de columnas repite este proceso hasta que alcanza una solución satisfactoria a todo el problema. En términos formales, la generación de columnas es una forma de resolver un problema de programación lineal que añade columnas [...] Básicamente, la generación de una columna en la formulación del simplex primal de un problema de programación lineal corresponde a la adición de una restricción en su formulación dual.

Cuando se resuelve mediante generación de columnas cada uno de los problemas relajados del árbol en el método “Branch-and-bound” hablamos de “Branch-and-price”.

Llegados a este punto, del mismo modo que obtuvimos el método “Branch-and-cut” reforzando la formulación del método del “Branch-and-bound” incluyendo planos de corte, podemos ahora fortalecer el método que nos ocupa introduciendo asimismo esos cortes, resultando así el método “Branch-and-cut-and-price”.

La generación de columnas se basa en el hecho de que en el método de simplex, el *solver* no necesita tener acceso a todas las variables del problema de forma simultánea. De esta manera, la ventaja de este método sobre el “Branch-and-bound” es que permite solucionar problemas de programación lineal de grandes dimensiones, en los que la matriz de restricciones es demasiado grande para ser almacenada. Mediante la generación de las columnas de la matriz de restricciones se van incluyendo columnas conforme van siendo necesarias, puesto que el *solver* puede iniciar su trabajo únicamente con la base y después decidir a qué otras variables se debe acceder, según sea necesario.

---

\*<http://www.ibm.com/support/knowledgecenter/>



*Todo saber tiene de ciencia lo que tiene de matemática.*

Poincaré

# 2

## Modelos de localización y rutas

---

Dentro de la programación entera existen diferentes áreas de estudio. Una de ellas, en la cual nos centramos en este trabajo, es la localización de instalaciones, como almacenes o depósitos. La localización de este tipo de instalaciones juega un papel crítico, por ejemplo, en el diseño estratégico de las redes de la cadena de suministro.

En general, en un problema de localización existen un conjunto de nodos o clientes distribuidos en el espacio y un conjunto de instalaciones se han de situar de modo que satisfagan las demandas de los clientes. Además, puede tenerse en cuenta el proceso de reparto entre dichas instalaciones y los distintos clientes. En este caso, hablamos de problemas de localización y rutas.

El objetivo de este capítulo es ofrecer una breve revisión de la gran cantidad de literatura que encontramos en torno a este tema. Para ello, introduciremos dos tipos de situaciones que reflejan la problemática anterior. El primero de los problemas planteados se reducirá a la localización de plantas y en el segundo de ellos se añadirá la dificultad de considerar rutas.

### 2.1. El problema de la p-mediana

La localización de instalaciones ha sido un problema que ha atraído abundante investigación durante los últimos años. Las aplicaciones de este tipo de problemas son muy extensas, abarcan desde problemas del sector privado, cuando se trata de localizar plantas industriales, bancos, u hoteles, hasta el sector público en la localización de centros médicos o vertederos municipales. A pesar de que los contextos en los que se aplican estos problemas pueden diferir mucho entre sí, todos ellos tienen unas características comunes. Existen cuatro componentes a tener en cuenta a la hora de formular un problema de localización: los clientes previamente ubicados, las instalaciones que se han de localizar, un espacio donde están localizados los clientes y las insta-

laciones y una métrica que indique las distancias entre clientes e instalaciones, que a menudo se considera a modo de costo por desplazamiento. Notemos que una de las consideraciones previas a la resolución de nuestro problema será decidir la métrica utilizada que indique las distancias o tiempos entre clientes e instalaciones. Se suelen tomar las distancias de Minkowski, definidas como una familia de distancias dependiente de un parámetro  $p$ . En particular, la distancia  $l_p$  entre dos puntos distintos  $(a_i, b_i)$  y  $(a_j, b_j)$  viene dada por  $d_{ij} = (|a_i - a_j|^p + |b_i - b_j|^p)^{1/p}$ . La mayor parte de la literatura se centra en tres casos especiales, que son la distancia rectangular o Manhattan obtenida para  $p = 1$ , la distancia euclídea obtenida para  $p = 2$  y la distancia de Chebyshev o ‘máx’ resultante de considerar el caso límite en el infinito.

Un ejemplo de problema de localización con gran presencia en la literatura debido a su sencilla formulación y a la gran cantidad de situaciones reales que se puedan ajustar a él es el problema de la  $p$ -mediana. Se trata de un problema en el que únicamente se tiene en cuenta la localización de una serie de almacenes o plantas.

El problema se puede presentar de forma muy simple del siguiente modo: dado un conjunto de clientes, un conjunto de ubicaciones candidatas a almacenes y la distancia entre cada par cliente-almacén, se han de seleccionar  $p$  almacenes para ser abiertos de manera que se minimice la distancia ponderada por la demanda. A pesar de su formulación sencilla, se trata de un problema de programación entera con todo lo que ello implica. Conforme el conjuntos de clientes e instalaciones considerados aumentan, los métodos expuestos anteriormente como los algoritmos Branch-and-bound y Branch-and-cut dan lugar a convergencias muy lentas, por lo que la búsqueda de métodos heurísticos que lleven a buenas soluciones en tiempos razonablemente buenos ha sido tema de investigación con una gran presencia en la literatura.

Existen distintas variantes de este problema dependiendo de si existe un coste adicional por apertura de almacén o si se consideran almacenes con capacidad o no. A continuación se da una formulación para cada uno de estos posibles problemas.

En primer lugar se introduce la notación necesaria para formular el problema de la  $p$ -mediana:

- $p$  : Número de almacenes por localizar.
- $I$  : Conjunto de clientes,  $I = \{1, \dots, n\}$ .
- $J$  : Conjunto de ubicaciones candidatas para almacenes,  $J = \{1, \dots, m\}$ .
- $h_i$  : Demanda del cliente  $i \in I$ .
- $d_{ij}$  : Distancia entre el cliente  $i \in I$  y el almacén candidato  $j \in J$ .

Definimos las siguientes variables para nuestro problema. Para cada  $j \in J$  consideramos las variables binarias  $y_j$  como sigue:

$$y_j = \begin{cases} 1 & \text{si se utiliza el almacén candidato } j \\ 0 & \text{en otro caso} \end{cases}.$$

Por otro lado, definimos las variables de asignación  $x_{ij}$  para cada  $i \in I, j \in J$ :

$$x_{ij} = \begin{cases} 1 & \text{si el cliente } i \text{ es atendido mediante el almacén } j \\ 0 & \text{en otro caso} \end{cases}.$$

A continuación introducimos una formulación del problema de la  $p$ -mediana en su versión más sencilla, es decir, en la que no existe coste adicional por apertura de almacén y los almacenes no tienen límites de capacidad. La formulación que se ofrece a continuación es una variante de la que ofrecieron ReVelle y Swain en 1970 [28], que fueron los primeros en formular el problema de la  $p$ -mediana como un problema programación 0-1.

$$\min_{x_{ij}, y_i} \sum_{i \in I} \sum_{j \in J} h_i d_{ij} x_{ij} \quad (2.1a)$$

sujeto a

$$\sum_{j \in J} x_{ij} = 1, \quad \forall i \in I \quad (2.1b)$$

$$\sum_{j \in J} y_j = P \quad (2.1c)$$

$$x_{ij} - y_j \leq 0, \quad \forall i \in I, \forall j \in J \quad (2.1d)$$

$$y_j \in \{0, 1\}, \quad \forall j \in J \quad (2.1e)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i \in I, \forall j \in J \quad (2.1f)$$

La función objetivo (2.1a) minimiza la distancia total ponderada por la demanda del cliente. Notemos que dado que las demandas son conocidas y la demanda total es fija, es equivalente a minimizar la distancia media ponderada con las demandas.

En el primer grupo de restricciones (2.1b) se asegura que cada cliente reciba servicio exactamente de un almacén. La restricción (2.1c) obliga a que se abran exactamente  $p$  almacenes. Las restricciones (2.1d) aseguran que un cliente no reciba servicio de un almacén cerrado. Finalmente, en (2.1e) y (2.1f) se reflejan las restricciones usuales de integridad. Cabe destacar que las restricciones de integridad de las variables  $x_{ij}$  se pueden relajar a condiciones de no negatividad, puesto que si en la solución óptima resulta que un mismo nodo de demanda ha sido asignado a varios almacenes, entonces el nodo de demanda ha de ser equidistante de cada uno de los almacenes, de modo que si se requiere que la demanda de cada cliente sea satisfecha por un único almacén, puede ser arbitrariamente asignado a cualquiera de los almacenes en los que estaba parcialmente asignado. Asimismo, notemos que las  $|I||J|$  restricciones que imponen que un cliente pueda ser asignado a un determinado almacén únicamente si está abierto pueden ser sustituidas por  $|J|$  restricciones del tipo  $\sum_{i \in I} x_{ij} \leq M y_j$  para cada  $j \in J$  donde  $M$  es un entero lo suficientemente grande. Podemos ajustar esta cota para una mejor convergencia en la rutina de optimización fijándola como  $n - (p - 1)$ .

Notemos que en la formulación anterior, no se tiene en cuenta un posible coste adicional por la apertura de un determinado almacén. En ocasiones esto no es realista y hay que tener en cuenta diferentes costes para cada uno de los almacenes. Para cada  $j \in J$  sea  $f_j$  el coste

por mantener abierto el almacén  $j$ . Introducimos un parámetro  $\alpha$  que convierte las demandas ponderadas en unidades de costo. Con esta notación adicional, el modelo de localización puede formularse como sigue:

$$\min_{x_{ij}, y_i} \quad \sum_{j \in J} f_j y_j + \alpha \sum_{i \in I} \sum_{j \in J} h_i d_{ij} x_{ij} \quad (2.2a)$$

sujeto a

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

$$x_{ij} - y_j \leq 0, \quad \forall i \in I, \forall j \in J \quad (2.2c)$$

$$y_j \in \{0, 1\}, \quad \forall j \in J \quad (2.2d)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i \in I, \forall j \in J \quad (2.2e)$$

La formulación es similar a la anterior. La principal variación reside en la función objetivo (2.2a) que minimiza la suma de los costes fijos de localización de cada almacén y los costes de transporte. La siguiente diferencia que encontramos es que en este modelo se prescinde de la restricción (2.1c) del modelo anterior. Dicha restricción ya no es necesaria puesto que en la función objetivo se penaliza la apertura innecesaria de almacenes. Del mismo modo que en el problema de la p-mediana, las restricciones de integridad sobre las variables  $x_{ij}$  pueden relajarse a condiciones de no negatividad, puesto que cada nodo de demanda será asignado de forma natural al almacén abierto más cercano.

Otra versión del problema aparece cuando se tienen en cuenta las capacidades de los almacenes. El objetivo sigue siendo minimizar las distancias ponderadas entre almacenes y clientes. Conocida la capacidad  $Q_j$  para cada almacén  $j = 1, \dots, m$ , el problema conocido como problema de la p-mediana con capacidades (CPMP) puede formularse como:

$$\min_{x_{ij}, y_i} \quad \sum_{i \in I} \sum_{j \in J} h_i d_{ij} x_{ij} \quad (2.3a)$$

sujeto a

$$\sum_{j \in J} x_{ij} = 1, \quad \forall i \in I \quad (2.3b)$$

$$\sum_{j \in J} y_j = P \quad (2.3c)$$

$$\sum_{i \in I} h_i x_{ij} \leq Q_j y_j, \quad \forall j \in J \quad (2.3d)$$

$$y_j \in \{0, 1\}, \quad \forall j \in J \quad (2.3e)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i \in I, \forall j \in J \quad (2.3f)$$

La única variación respecto a la formulación original del problema de la p-mediana aparece en las restricciones (2.3d). Notemos que en ellas, además de asegurar que un cliente no reciba servicio de un almacén cerrado, se garantiza que la suma de las demandas de los clientes asignados a un almacén no excede su capacidad.

### 2.1.1. Revisión bibliográfica

Los primeros artículos basados en localización de plantas y almacenes aparecen a mediados del siglo XX. El origen del problema de la  $p$ -mediana se debe a S.L. Hakimi [17][18], matemático americano nacido en Irán, que publicó un par de artículos en los años 1964 y 1965 sobre la localización de  $p$  instalaciones en una red de  $n$  nodos de demanda con objetivo de minimizar las distancias ponderadas según demandas. Aunque no dio un método de resolución probó la existencia de al menos una solución óptima y demostró que se trataba de un problema NP-hard.

En lo que sigue, damos un resumen sobre una serie de artículos seleccionados relacionados con el problema de la  $p$ -mediana y la localización de plantas. Los artículos aparecen ordenados por fecha de publicación, comenzando por el más antiguo para ver de forma más clara el avance de la investigación. Cabe destacar que podemos encontrar artículos en los que se recoje la literatura más reciente sobre los problemas de localización. En particular, el artículo de por C. S. RevVelle y H. A. Eiselt, [27], ofrece una revisión de la literatura relacionada con los problemas de localización, presentando la formulación de varios problemas, entre ellos el problema de la  $p$ -mediana.

R. Baldacci, E. Hadjiconstantinou, V. Maniezzo, A. Mingozi. “*A new method for solving capacitated location problems based on a set partitioning approach*”, 2002. [5]

En primer lugar, los autores formulan el problema de la  $p$ -mediana con capacidades (CPMP) de una forma distinta a la dada en el presente trabajo. Veamos las diferencias entre ambos. En el artículo, se consideran  $n$  clientes y la matriz de distancias  $n \times n$  con ceros en la diagonal. Cada cliente tiene asociado un peso no negativo  $q_i$  que puede considerarse como la demanda en nuestro caso. El objetivo de este problema es particionar dicho conjunto de  $n$  clientes en  $p$  clústers –o grupos– disjuntos. En cada clúster, uno de los clientes pertenecientes a él será su media, de modo que la media de cada clúster se corresponde con la instalación a localizar y el resto de clientes pertenecientes a dicho clúster son el conjunto de clientes asignados a dicha instalación. A diferencia de la formulación ofrecida para el problema de la  $p$ -mediana en el apartado 2.1, en este caso únicamente se considera una única variable  $x_{ij}$  que toma valores en  $\{0, 1\}$  dependiendo de si el cliente  $i$  se asocia a un clúster cuya media es el cliente  $j$  o no. A continuación se ofrece otra formulación del problema desde el punto de vista de un modelo de partición de conjuntos. Se introduce también el problema dual asociado a la formulación dada para el CPMP para obtener una cota inferior válida del primal. Dado que ni el primal ni su dual pueden ser resueltos directamente de forma práctica para valores de  $n$  grandes, los autores proponen, por un lado, un método de reducción de variables para el primal y, por otro lado, dos métodos heurísticos para resolver el problema dual y obtener así dicha cota inferior válida para el problema CPMP.

J. Fathali. “*A genetic algorithm for the P-median problem with pos/neg weights*”, 2006.[14]

En este artículo, además de considerar el problema de la  $p$ -mediana, el autor trata otro tipo de problema que no se ha considerado previamente en esta memoria. En

él, se admite la posibilidad de que para un grupo de clientes la cercanía de la instalación es indeseable, de este modo, se introducen pesos negativos y positivos en la función objetivo, de modo que, los clientes o vértices del grafo con pesos no negativos se asignan a la ubicación más cercana y los vértices con pesos negativos se asignan a la instalación más alejada. La principal aportación del artículo es un algoritmo genético para ambos problemas. El tamaño de la población y la población inicial son dos factores importantes que influyen en la convergencia del algoritmo. Mientras que grandes poblaciones ralentizan el algoritmo genético, poblaciones pequeñas pueden no tener suficiente diversidad genética para buscar en la región factible. En su algoritmo se propone  $n + p$  como el tamaño de la población, esto es, la suma del número de clientes con la cantidad de instalaciones a abrir. La condición de parada del algoritmo está basada en el número de iteraciones entre dos mejoras, en particular, el algoritmo termina cuando tras  $n + p$  iteraciones la solución actual no se ha mejorado.

R. Tadei, N. Ricciardi, G. Perboli. “*The stochastic p-median problem with unknown cost probability distribution*”, 2009. [30]

En este artículo, estudia el problema de la p-mediana estocástico. Mientras que en la optimización determinista los parámetros del problema se suponen conocidos con certeza, aunque sean valores estimados como puede ser su valor medio, en optimización estocástica esta condición se ve relajada. Los valores de los parámetros no son conocidos, o bien sólo se conocen sus distribuciones. Los autores buscan determinar la ubicación de las  $p$  instalaciones que minimicen el costo total esperado, cuando el costo para el uso de una instalación es una variable estocástica con distribución de probabilidad desconocida. Los costos están dados por un lado por un coste determinista asociado a cada grupo o clúster de clientes a lo que se añade un término aleatorio, con distribución de probabilidad desconocida, que representa la heterogeneidad de costos dentro de cada grupo. El problema se resuelve en dos etapas. En una primera etapa se suponen conocidas las variables de localización, esto es, los almacenes abiertos y en una segunda etapa se preguntan por el plan óptimo de asignación teniendo en cuenta la incertidumbre descrita. Finalmente se da una solución óptima para las variables de localización.

M. Baïou, F. Barahona. “*On the linear relaxation of the p-median problem*”, 2011. [4]

En este artículo, los autores abordan el problema de la p-mediana utilizando un grafo no dirigido en el que los nodos son las posibles localizaciones y el coste entre cada par de nodos se indica en la arista que los une. Ofrecen una caracterización de los grafos dirigidos para los que el conjunto de soluciones factibles del problema definen un politopo integral. Entendemos por politopo integral aquel politopo cuyos vértices tengan coordenadas enteras. Como consecuencia, el propio algoritmo simplex irá pasando a través de estos puntos extremos enteros y se obtiene que el problema de la p-mediana es polinomial en esa clase de grafos. Es un artículo

principalmente teórico y extenso en el que se enuncian y demuestran las condiciones necesarias y suficientes para que un determinado grafo determine un politopo integral. También se ofrece un algoritmo para reconocer estos gráficos.

P. Avella, M. Boccia, S. Salerno, I. Vasilyev. “*An aggregation heuristic for large scale  $p$ -median problem*”, 2012. [3]

En este artículo, los autores tratan el problema de la  $p$ -mediana de grandes dimensiones. El interés en el problema de la  $p$ -mediana a gran escala surge de aplicaciones en el análisis clúster, donde un conjunto de patrones tiene que ser dividido en subconjuntos o clústers en base a su similitud. Los autores introducen una heurística basada en la relajación de Lagrange. A grandes rasgos, la relajación Lagrangiana consiste en eliminar una restricción o grupo de restricciones e introducirlas en la función objetivo como penalización. Se le asigna un peso a dicha penalización con los multiplicadores de Lagrange. En particular, se ven relajadas las restricciones de asignación (en nuestro caso, en la formulación (2.1) del problema de la  $p$ -mediana, las correspondientes al grupo de restricciones (2.1b)). Dado que los autores tratan con conjuntos de datos que se sitúan en torno a niveles de hasta 90000 clientes e instalaciones, para tratar con estos problemas se han de introducir técnicas como la generación de columnas. Al final del artículo se presentan los resultados computacionales y para valores de  $n$  y  $p$  tan altos como  $n = 89600$  clientes y  $p = 64$  localizaciones se consiguen soluciones de buena calidad en un tiempo razonable.

M. Yaghini, M. Karimi, M. Rahbar. “*A hybrid metaheuristic approach for the capacitated  $p$ -median problem*”, 2013. [33]

En este artículo, los autores proponen, a partir de la formulación del problema CPMP dada en (2.3), un algoritmo híbrido metaheurístico que combina una estructura de planos de corte con una búsqueda tabú. Un primer paso del método es seleccionar una instalación para cerrarla, y para ello se proponen tres estrategias diferentes: seleccionarla aleatoriamente, cerrar aquella que tenga un menor uso de su capacidad o, la que ha resultado más eficiente, cerrar aquella en la que se alcance el máximo en el promedio de capacidad no aprovechada multiplicado por la suma de las distancias de los clientes asignados a dicha instalación. Después, se añaden restricciones a la formulación, como la que impone que se cierre la planta seleccionada y otra que fija las otras  $p - 1$  instalaciones que permanecían abiertas en la solución actual a 1 de modo que en el nuevo problema sólo se ha de seleccionar una planta nueva para ser abierta, distinta a la que se ha decidido cerrar. A continuación se resuelve el problema relajado. Cabe destacar que al añadir dichos planos de corte el problema puede volverse infactible, en ese caso, se elimina la instalación cerrada del espacio de búsqueda y se repite el proceso. En la segunda parte del algoritmo, en cada iteración de la búsqueda tabú, las transformaciones locales que pueden aplicarse a la solución actual definen un conjunto de vecinos en el espacio de búsqueda. Se examina dicha vecindad y se selecciona como nueva solución el candidato que produce el mejor valor de la función objetivo. Para evitar

el ciclado a las soluciones previamente visitadas, se utilizan técnicas de búsqueda tabú para gestionar el historial de búsqueda.

S. Dantrakul, C. Likasiri, R. Pongvuthithum. “*Applied p-median and p-center algorithms for facility location problems*”, 2014. [11]

En el artículo, los autores tratan con dos problema de localización, el problema de la p-mediana y el problema del p-centroide, en el que el objetivo es minimizar la distancia máxima entre las instalaciones y los clientes asignados y proponen tres métodos de resolución: el método del coste de instalación, el método p-mediana modificado y el método p-centroide modificado. Nos centramos aquí en el método modificado que ofrecen para resolver el problema de la p-mediana. La planta con el coste más bajo encontrada en el problema de la p-mediana juega un papel importante en este método modificado. Se demuestra que la solución obtenida con el algoritmo propuesto es un mínimo local. Y que, si se cumplen ciertas condiciones de convexidad sobre las funciones de coste de apertura y de transporte, entonces el mínimo alcanzado es global. Finalmente, se prueban estos métodos utilizando 100 conjuntos de datos aleatorios y aplicándolos a un problema real de las redes de carreteras de la ciudad Chiang Mai y cinco provincias del norte de Tailandia. Los resultados de la simulación muestran que en el caso en que los costes de instalación sean mayores que los de transporte es mejor utilizar el primero de los algoritmos propuestos, mientras que en el caso contrario es más eficiente el método de la p-mediana modificado.

J. Janček, M. Kvet. “*Sequential approximate approach to the p-median problem*”, 2016. [19]

El artículo se centra en una aproximación del problema de la p-mediana basada en una reformulación del modelo, la que denominan formulación radial, en la que se modifican las distancias entre clientes e instalaciones. Se fijan unos radios, generándose así una secuencia ordenada  $d_0 < d_1 < \dots < d_m$ , y se asocia cada distancia entre cliente e instalación al extremo superior del intervalo al que pertenece. Sin pérdida de generalidad se puede suponer  $d_0 = 0$ , si no, se puede reducir cada elemento de la matriz de distancias sustrayendo el valor mínimo. De este modo, se aprovecha el hecho de que existe un único conjunto finito de valores en la matriz de distancias en el problema de la p-mediana asociado. Esta aproximación no da información sobre qué clientes están asignados a qué instalaciones, sino que trata el número de instalaciones localizadas en un determinado radio. Este procedimiento, como cabe esperar, afecta a la exactitud de la solución. Por ello introducen un método secuencial que ofrece una cota inferior de la solución óptima, así como una superior. De este modo, se puede controlar la brecha o diferencia entre dichas cotas, lo cual supone una diferencia entre la aproximación sugerida en el artículo y otros algoritmos heurísticos. Esto se consigue mediante una división adecuada del rango  $[d_0, d_m]$  en  $\mu + 1$  subconjuntos o zonas. Cada cota se obtiene resolviendo un

problema de tipo recubrimiento considerando las divisiones anteriores. El número de divisiones que se realizan afecta al tamaño de los problemas de recubrimiento que dan las cotas superior e inferior. Para valores altos de  $\mu$  tenemos el problema de la dimensión del problema de recubrimiento, mientras que para valores bajos de  $\mu$  se tiene una mayor inexactitud de la solución, es decir, una mayor diferencia entre las cotas.

## 2.2. El problema de localización y rutas con capacidades

El problema de localización y rutas incluye dos tipos de problemas fundamentales en gestión de la cadena de suministro: el problema de localización de instalaciones y el problema de rutas de vehículos. Los diferentes aspectos de estos dos problemas como la localización, la asignación y las rutas han sido muy estudiados de forma independiente. La idea de tratar conjuntamente el problema de localización de instalaciones y las rutas de vehículos entre dichas instalaciones y los clientes nace de forma natural por cuestiones económicas, tras verse que parte del presupuesto de las compañías se consume por los costes logísticos de desplazamiento y tras reconocer que estos costos pueden ser reducidos considerablemente mediante un buen diseño de la cadena de suministro. A pesar de esta clara relación entre ambos problemas, las primeras investigaciones que tratan conjuntamente la localización de instalaciones y las rutas de vehículos nace no más allá de los años 50 del pasado siglo.

Un ejemplo de problema en el que se combina la localización de las plantas y las rutas de vehículos es el problema CLRP (Capacitated Location-Routing Problem). De entre sus diversas versiones, la que aquí tratamos tiene en cuenta almacenes con capacidad, rutas con capacidad y costos fijos para abrir un almacén o una ruta. Además, la flota de vehículos será homogénea, es decir, todos los vehículos tienen la misma capacidad. Variantes de este problema aparecen cuando se considera por ejemplo una flota heterogénea de vehículos o se introducen ventanas de tiempo. La solución de un problema CLRP consiste en determinar qué almacenes deben abrirse, asignar a cada cliente un depósito abierto y construir una ruta para los vehículos desde cada almacén hasta sus clientes minimizando el coste total.

En el problema que aquí tratamos se han de satisfacer las siguientes restricciones:

1. La demanda de los clientes asignada a cada almacén no puede exceder su capacidad.
2. Cada ruta comienza y termina en el mismo almacén.
3. Cada vehículo realiza a lo sumo un viaje.
4. Cada cliente es servido por un único vehículo (no existe entrega dividida).
5. La demanda total de los clientes visitados por un vehículo no excede la capacidad del vehículo.

Es habitual considerar este problema basado en una red dirigida  $G=(V,A)$  donde  $A$  denota el conjunto de los arcos de la red y  $V$  es el conjunto de los nodos formado por un conjunto  $J$

de las posibles localizaciones para los almacenes y un conjunto  $I$  de clientes. Notemos así que podemos asociar cada subíndice  $ij$  con el arco  $a$  que va desde el nodo  $i$  hasta el nodo  $j$ . De este modo, hablaremos indistintamente de  $x_a$  y de  $x_{ij}$ .

A continuación introducimos la notación del problema:

- $V$ : conjunto de nodos,  $V = I \cup J$ .
- $I$ : conjunto de clientes a los que servir,  $I = \{1, 2, \dots, n\}$ .
- $J$ : conjunto de los posibles nodos almacén,  $J = \{1, 2, \dots, m\}$ .
- $O_j$ : coste por abrir el almacén  $j \in J$ .
- $W_j$ : capacidad del almacén  $j \in J$ .
- $d_i$ : demanda del cliente  $i \in I$ .
- $M = \{1, \dots, l\}$ : conjunto de vehículos disponibles ( $l$  tamaño de la flota).
- $K$ : capacidad de cada vehículo (flota homogénea).
- $F$ : coste fijo por usar cada vehículo.
- $c_{ij}$ : coste unitario de transporte por el arco  $(i, j)$ .

Las variables decisión de nuestro problema son:

$$y_j = \begin{cases} 1 & \text{si se abre el almacén } j \\ 0 & \text{en otro caso} \end{cases}, \quad 1 \leq j \leq m,$$

$$x_{ijk} = \begin{cases} 1 & \text{si el arco } (i, j) \text{ se atraviesa (de } i \text{ a } j\text{) por el vehículo } k \\ 0 & \text{en otro caso} \end{cases}, \quad 1 \leq i \leq n, 1 \leq j \leq m,$$

$$f_{ij} = \begin{cases} 1 & \text{si el cliente } i \text{ es asignado al almacén } j \\ 0 & \text{en otro caso} \end{cases}, \quad 1 \leq i \leq n, 1 \leq j \leq m.$$

Para cada subconjunto  $S$  de  $V$  definimos los siguientes conjuntos:

$$\delta^+(S) := \{a = (i, j) \in A : i \in S, j \notin S\}, \quad \delta^-(S) := \{a = (j, i) \in A : i \in S, j \notin S\}.$$

Esto es, estamos considerando el conjunto de arcos que salen de  $S$ , denotado por  $\delta^+(S)$ , y que entran en  $S$ , dado por  $\delta^-(S)$ . Si  $S$  tiene un único nodo  $z$  lo denotamos por  $\delta^+(z) \equiv \delta^+(\{z\})$ . Una variante de la formulación para el problema CLRP que ofrecemos a continuación se puede encontrar en [24].

$$\min_{x_{ijk}, y_j, f_{ij}} \sum_{j \in J} O_j y_j + F \sum_{a \in \delta^+(J)} \sum_{k \in M} x_{ak} + \sum_{a \in A} \sum_{k \in M} c_a x_{ak} \quad (2.4a)$$

sujeto a

$$\sum_{k \in M} \sum_{a \in \delta^-(i)} x_{ak} = 1, \quad \forall i \in I \quad (2.4b)$$

$$\sum_{i \in I} \sum_{a \in \delta^-(i)} d_i x_{ak} \leq K, \quad \forall k \in M \quad (2.4c)$$

$$\sum_{a \in \delta^+(j)} x_{ak} - \sum_{a \in \delta^-(j)} x_{ak} = 0, \quad \forall j \in V, \forall k \in M \quad (2.4d)$$

$$\sum_{a \in \delta^+(J)} x_{ak} \leq 1, \quad \forall k \in M \quad (2.4e)$$

$$\sum_{a \in \delta^+(S)} x_{ak} \geq 1, \quad \forall S \subseteq I, \forall k \in M \quad (2.4f)$$

$$\sum_{a \in \delta^+(j) \cap \delta^-(I)} x_{ak} + \sum_{a \in \delta^-(i)} x_{ak} \leq 1 - f_{ij}, \quad \forall i \subseteq I, \forall j \in J, \forall k \in M \quad (2.4g)$$

$$\sum_{i \in I} d_i f_{ij} \leq W_j y_j, \quad \forall j \in J \quad (2.4h)$$

$$x_{ak} \in \{0, 1\}, \quad \forall a \in A, \forall k \in M \quad (2.4i)$$

$$y_j \in \{0, 1\}, \quad \forall j \in J \quad (2.4j)$$

$$f_{ij} \in \{0, 1\}, \quad \forall i \in I, \forall j \in J \quad (2.4k)$$

En la función objetivo (2.4a) aparece reflejado el coste total por la localización y la asignación de las rutas que consta del coste por los almacenes abiertos, los costes fijos de los vehículos utilizados para realizar las entregas a los clientes y el coste de las rutas. El objetivo del problema CLRP es minimizar dicho coste total.

La restricción (2.4b) garantiza que un cliente sea asignado a un único almacén y tiene un único predecesor (un único nodo lleva a él). Las restricciones de capacidad de los vehículos y los almacenes se satisfacen debido a las restricciones (2.4c) y (2.4h) respectivamente. Las restricciones (2.4d), (2.4e) y (2.4f) están relacionadas con las rutas de cada vehículo. La dos primeras garantizan la continuidad de la ruta y que ésta acabe en el mismo depósito del que salió, mientras que la (2.4f) evita subtours, es decir, rutas que no comienzan y terminan en un almacén. La restricción (2.4g) permite asignar a cada cliente a la planta desde la que es servido para poder formular la restricción (2.4h). Por último, las restricciones (2.4i), (2.4j) y (2.4k) establecen el carácter dicotómico de las variables.

En la formulación dada en [24] los autores proponen una restricción diferente a (2.4f) para evitar los subtours.

El problema CLRP es NP-hard, puesto que generaliza dos problemas que es bien conocido que son NP-hard: el problema de localización de plantas con capacidad y el problema de rutas de vehículos (VRP) que se obtiene para  $m = 1$  y que extiende el conocido problema del viajante de comercio (TSP). Así, para la mayoría de los problemas prácticos, que resultan ser de grandes dimensiones, se acude a algoritmos heurísticos que conducen a buenas soluciones en un tiempo razonable.

### 2.2.1. Revisión bibliográfica

A continuación se comentarán algunos artículos relacionados con los problemas de localización y rutas, en particular con el problema CLRP. Del mismo modo que con el problema de localización, los artículos aparecen ordenados cronológicamente, comenzando por el más antiguo.

Destacamos que, del mismo modo que ocurre con los problemas de localización, podemos encontrar artículos, como el de C. Prodhon y C. Prins publicado en 2004, [25], en los que se ha realizado un análisis sobre los artículos previamente publicados sobre los problemas de localización y rutas. En este artículo los autores realizan un estudio exhaustivo de los artículos publicados sobre la materia desde 2007 en adelante. Introducen la notación e ideas principales del problema CLRP y dividen los artículos incluidos en el estudio por secciones, dependiendo de su naturaleza o de la metodología utilizada para su resolución.

T-H. Wu, C. Low, J-W. Bai “*Heuristic solutions to multi-depot location-routing problems*”, 2002. [32]

Este artículo presenta un método para resolver el problema de localización y rutas para múltiples instalaciones. Los autores relajan condiciones como la de flota homogénea, permitiendo capacidades distintas, y consideran un número limitado de vehículos. Dada la complejidad del problema y la imposibilidad de resolverlo a gran escala, se divide el problema original en dos subproblemas, que son el problema de localización de instalaciones (LAP, location-allocation problem) y el problema genérico de rutas de vehículos (VRP, vehicle routing problem). Cada subproblema se resuelve secuencialmente de manera iterativa mediante el algoritmo de recocido simulado(en inglés simulated annealing). El proceso consta de dos partes: construcción de soluciones y mejoras. En primer lugar, se resuelve el LAP y la solución obtenida de esta fase se utiliza como una entrada para el VRP en la siguiente fase.

M. Albareda-Sambola, J. A. Díaz, E. Fernández. “*A compact model and tight bounds for a combined location-routing problem*”, 2005. [2]

Los autores consideran el problema de localización y rutas con capacidades. Para abordar el problema definen una red  $N = (V, A)$ , donde  $V$  es el conjunto de nodos dado por  $V = \{s, t\} \cup J \cup J' \cup I$  donde  $J'$  es una copia del conjunto  $J$ . Tanto  $I$  como  $J$  vienen definidos como en la formulación dada en 2.2. Además, se introducen dos nodos ficticios  $s$  y  $t$ , que son los nodos fuente y destino. A partir de ello, se construye el conjunto de arcos  $A$  y los costes  $c_a$  asociados a cada arco  $a \in A$ . En la red anterior aparece representada la solución, es decir, se ofrece el conjunto de instalaciones a abrir así como las rutas para servir a los clientes desde dichas instalaciones abiertas. Se aplica una heurística búsqueda tabú consistente en una serie de iteraciones, cada una de las cuales realiza una fase de intensificación seguida de una fase de diversificación. Al principio del procedimiento se proponen dos cotas inferiores, por un lado la solución del problema relajado y otra propuesta por los autores basándose en la estructura del problema.

R. B. Lopes, S. Barreto, C. Ferreira, B. S. Santos. “*A decision-support tool for a capacitated location-routing problem*”, 2008. [21]

En este artículo, los autores presentan una herramienta de apoyo a la decisión (DST) que implementa el problema CLRP con dos niveles, almacenes y clientes, y una flota homogénea de vehículos con capacidad. Permite la exploración del proceso de búsqueda de soluciones en una forma fácilmente comprensible por el usuario, y permite el acceso a los datos geográficos en línea a través de servidores web de mapas. El algoritmo utilizado consta principalmente de cuatro pasos. En primer lugar, se construyen grupos de clientes (clústers) conforme a los límites de capacidad de los vehículos, después se determina la distribución en cada grupo de clientes. La última parte del proceso se basa por un lado en mejorar la ruta intercambiando clientes entre las diferentes rutas y finalmente se localizan los depósitos y se les asignan las rutas.

J. M. Belenguer, E. Benavent, C. Prins, C. Prodhon, R. W. Calvo. “*A branch-and-cut method for the capacitated location-routing problem*”, 2010. [7]

En este artículo, los autores tratan con el problema CLRP con capacidades en almacenes y vehículos. Proponen una aproximación exacta basada en el algoritmo Branch-and-cut. Se fortalece la formulación del problema CLRP mediante familias de desigualdades válidas (cortes). Una de las técnicas utilizadas consiste en reforzar restricciones, en particular las relativas a las capacidades de depósitos y vehículos, incluyendo variables que han resultado efectivas a la hora de eliminar soluciones fraccionarias en las que muchas de estas variables añadidas tuviesen valores próximos a cero. También se proponen técnicas para fortalecer las restricciones que impiden los subtours u otras. Se utiliza una técnica de planos de corte que sigue el siguiente esquema. Inicialmente se construye un programa lineal con la función objetivo del CLRP y un subconjunto de restricciones que se especifica en el artículo. Después, en cada iteración se resuelve la relajación lineal y se miran las restricciones válidas que no satisfagan la solución óptima y se añaden. Este algoritmo de plano de corte finaliza cuando no se obtenga no queden restricciones por añadir.

J. W. Escobar, R. Linfati, P. Toth. “*A two-phase hybrid heuristic algorithm for the capacitated location-routing problem*”, 2012. [13]

En el artículo, los autores proponen una heurística para resolver el problema de localización y rutas con capacidades (CLRP). La estructura del algoritmo propuesto se divide en dos partes fundamentales, una primera fase llamada *fase de construcción* y una segunda fase conocida por *fase de mejora*. En la fase de construcción, el objetivo es obtener una solución factible, resolviendo el problema de localización de la p-mediana dado en el apartado 1.1. Dicha solución es mejorada en la segunda fase, en la cual se permiten soluciones no factibles respecto a las limitaciones

de capacidad de almacenes y vehículos. Si tras un número determinado de iteraciones no se produce dicha mejora, el algoritmo trata de salir de ese óptimo local aplicando un procedimiento de perturbación aleatoria. Notemos que dado que las decisiones más importantes de la fase de mejora son aquellas relacionadas con la apertura y cierre de almacenes, una selección adecuada de la localización de estos almacenes durante la fase de construcción permite reducir el espacio de búsqueda.

C. Contardo, V. Hemmelmayr, T. G. Crainic. “*Lower and upper bounds for the two-echelon capacitated location-routing problem*”, 2012. [10]

Hemos seleccionado este artículo para tratar una variante del problema CLRP. Los autores tratan con un problema diferente al tratado en la presente memoria. Se presenta el problema de localización y rutas con capacidades en dos escalones (2E-CLRP en sus siglas en inglés). El problema 2E-CLRP puede formularse como sigue. Dados tres conjuntos de nodos disjuntos que representan las plataformas (instalaciones de primer escalón), satélites (instalaciones de segundo escalón) y clientes, se ha de decidir las localizaciones de un subconjunto de plataformas y de un subconjunto de satélites, así como construir las rutas de vehículos que visiten a cada cliente una única vez utilizando una única ruta desde una plataforma abierta, con el objetivo de minimizar costes. Hay dos flotas de vehículos, una por cada grupo de instalaciones, cada una homogénea pero no necesariamente ambas con la misma capacidad. A lo largo del artículo se proponen dos métodos para abordar dicho problema. Uno de ellos un algoritmo branch-and-cut reforzado con desigualdades válidas y otro en el que se propone una metaheurística de búsqueda de entornos variable (Variable Neighbourhood Search, VNS). Los resultados computacionales muestran que dependiendo del tamaño del problema resulta más eficiente un algoritmo u otro. En concreto, para ejemplos pequeños y medios resulta mejor el primero de los métodos y para problemas de un tamaño grande ofrece mejores resultados la metaheurística.

J. W. Escobar, R. Linfati, M. G. Baldoquin, P. Toth. “*A granular variable tabu neighborhood search for the capacitated location-routing problem*”, 2014. [12]

En este artículo, se propone una heurística para el problema CLRP llamada búsqueda tabú granular de entornos variables (GVTNS). El algoritmo utiliza movimientos internos dentro de una misma ruta y movimientos entre distintas rutas, ya estén asignadas al mismo almacén o no. Para ello los autores utilizan unas estructuras de búsqueda de vecinos. El algoritmo GVTNS combina la potencia del algoritmo de búsqueda de entornos variables (VNS) y sus estructuras de vecinos con la eficiencia de la búsqueda tabú granular (GTS). Con el método VNS se controlan los cambios de vecindades mientras que con la técnica GTS se guía el proceso de búsqueda utilizando las estructuras mencionadas anteriormente. El tamaño de los datos utilizados no es muy elevado. Oscilan entre los 100 y 200 clientes y 10 o 20 depósitos. Al final del artículo se presentan los resultados computacionales obtenidos así como las comparaciones con algunos de los mejores métodos propuestos

en la literatura para la resolución del CLRP. Además, evalúan el efecto de añadir la aproximación granular ejecutando el algoritmo con y sin dicha aproximación. Los resultados muestran que el enfoque de búsqueda granular mejora significativamente el rendimiento de algoritmo GVTNS.

R. B. Lopes, C. Ferreira, B. Sousa Santos. “*A simple and effective evolutionary algorithm for the capacitated location-routing problem*”, 2016. [20]

Este artículo propone un algoritmo genético para resolver el problema CLRP. El algoritmo sigue la línea de un algoritmo genético estándar. En comparación con otros algoritmos genéticos en la literatura, el método aquí propuesto tiene bastantes ventajas. Por ejemplo, tiene una representación de los cromosomas muy intuitiva, ya que representan la solución completa, esto es, la colección de rutas. Además, la descendencia se genera por cruce bajo un marco sencillo, ya que la representación adoptada, asumiendo la factibilidad de la solución, junto con el operador de cruce detallado en el artículo, ofrece soluciones factibles en las soluciones hijo. De esta manera, se evita utilizar métodos de reparación de la factibilidad y permite una rápida evaluación del fitness (el coste de la solución). En cuanto al operador de mutación se presentan dos algoritmos. El primero de ellos lo llaman ‘add’, agregar, está ligado a los almacenes, y evita una convergencia rápida a soluciones con pocos almacenes, diversificando los almacenes abiertos. El segundo de ellos, denominado ‘swap’, intercambia la posición de dos clientes en el trazado de las rutas. Al final del artículo se presentan los resultados computacionales que reflejan que el algoritmo, aunque relativamente sencillo, es efectivo, ofreciendo resultados competitivos en unos tiempos de computación razonables.



*Siempre que enseñas, enseña a la vez a dudar de lo que enseñas.*

José Ortega y Gasset

# 3

## Modelos en la gestión de residuos

---

Una de las múltiples aplicaciones de los problemas de localización y de los problemas de localización y rutas introducidos en el capítulo anterior es la gestión de los residuos. En lo que sigue nos centraremos en el estudio de este tipo de problemas, que constituye el tema central del trabajo.

Todos los productos causan en mayor o menor medida una degradación del medio ambiente ya sea en su fabricación, uso o eliminación. De esta manera, el gran aumento de desechos generados por la sociedad actual y la concienciación alcanzada respecto al cuidado del medio ambiente ha hecho del estudio del tratamiento de residuos y de la localización de plantas de reciclaje una necesidad. Tanto es así que, por ejemplo, la UE ha impulsado un instrumento llamado IPP (Integrated Product Policy) que busca minimizar estos daños examinando todas las fases del ciclo de vida de los productos.

Esta concienciación ha hecho que se produzcan grandes cambios en cuanto al planteamiento de los problemas de localización relativos a la gestión de residuos.

### 3.1. Características del problema

Destacamos que, del mismo modo que ocurre para los problemas de localización y de localización y rutas tratados en el capítulo anterior, encontramos artículos en los que se recoge la literatura más reciente en el caso particular de que estos problemas traten sobre la gestión de residuos.

Un artículo en el que se realiza dicha recopilación es, por ejemplo, el publicado por Ghiani et al. (2006) [16]. Los autores establecen las principales características que han de tenerse en

cuenta a la hora de diseñar un sistema de manejo de residuos sólidos: el tiempo (horizonte de planificación a largo plazo), el tipo de producto (si es tóxico, no deseado, etc.), capacidades, transformación del residuo (una vez llega el producto a la instalación puede ser transformado y aprovechado) y el objetivo (minimización del coste, de la distancia, del número de instalaciones, etc.). Además, realizan una clasificación de los problemas presentes en la literatura más reciente basado en el tipo de problema tratado y las decisiones tomadas en cuanto a las cuestiones tácticas. Respecto a las características del problema encontramos diferencias entre artículos dependiendo de, por ejemplo, si son multiobjetivo o no, si se tienen en cuenta criterios sociales además de económicos, el número de materiales tratados o si es un modelo con incertidumbre o determinista. En cuanto a cuestiones tácticas se ha de tomar decisiones sobre la partición de los territorios en distritos, la elección de los días de recolección para cada distrito y para cada tipo de residuo y la composición de la flota de vehículos, entre otras. A continuación realizaremos una breve clasificación de este tipo de artículos.

### 3.2. Modelos de localización multiobjetivo

Los primeros modelos que resolvían estos problemas de localización únicamente tenían en cuenta criterios económicos. El objetivo de estos problemas se reducía a minimizar los costes fijos por la apertura de las plantas y los costes de transporte. En los últimos años ha aumentado el número de aplicaciones en las que se tienen en cuenta otros factores sociales y medioambientales, relacionados con disponibilidad y calidad del capital humano o la calidad de vida, entre otros. Uno de los métodos de resolución de problemas multiobjetivo más comunes es el método de la suma ponderada, en el que se transforman los múltiples objetivos en una única función en la que aparece reflejado cada uno de los objetivos multiplicado por un peso.

Por ejemplo, en un artículo publicado en 2008 por Queiruga et al. [26] se recogen en 10 criterios una serie de objetivos: económicos, de infraestructura y legales. En cuanto a criterios económicos se tienen en cuenta aspectos como el coste de la tierra, el coste de contratación del personal y el coste energético. Respecto a criterios relacionados con los objetivos de infraestructura se tienen en cuenta los accesos a las instalaciones, los efectos de aglomeración (proximidad a plantas existentes), cercanía a zonas deshabitadas, ausencia de otras plantas de reciclaje o la disponibilidad de trabajo. Finalmente, se tienen en cuenta criterios legales como la disponibilidad de programas de procesamiento de los residuos o subvenciones medioambientales. Para fijar unos pesos a cada uno de los criterios específicos del problema se apoyan en bases de datos estadísticos y encuestas de 30 expertos. Una vez estudiados los criterios, se definen unas funciones para cada uno de éstos que miden la preferencia y se calcula un índice de preferencia para cada una de las alternativas. Finalmente se aplica la metodología anterior en un problema de localización de plantas de tratamiento de residuos de aparatos eléctricos y electrónicos en España. Se valoran 20 localizaciones distintas. Las dos primeras alternativas en el ranking resultan ser Huelva y Sevilla.

Otro ejemplo en el que se tienen en cuenta múltiples criterios para la localización de unidades de tratamiento y reciclaje (UTR) es el artículo publicado en 2010 por Achillas et al. [1]. En él, los autores tienen en cuenta múltiples escenarios dependiendo de los distintos criterios.

Principalmente, para la toma de decisiones, los autores consideran necesario un estudio de los costes, de la existencia de las infraestructuras básicas (red de carreteras, fuerza de trabajo, etc.), la distancia desde una UTR existente, población objetivo o la aceptación social. A continuación se muestra la metodología a seguir. A cada uno de los criterios se le otorga un valor para cada una de las alternativas y se definen una serie de medidas de concordancia y discordancia entre criterios. Los autores aplican la metodología anterior a un caso real en Grecia, en el que se ha de determinar la localización de dos UTR. Finalmente, se tuvieron en cuenta los siguientes criterios: población local, población atendida, distancia desde cada UTR existente, coste del suelo, población desempleada, conexión por carretera, status financiero de la población local, distancia desde la capital de provincia y la distancia desde el puerto más cercano.

### 3.3. Modelos de localización con residuos peligrosos

Otro tipo de problemas surgen cuando los residuos con los que se trata son peligrosos o indeseables. Se considera material peligroso todo aquel que pueda provocar daños personales o en el entorno, tales como materiales inflamables, tóxicos o corrosivos entre otros. El gran desarrollo tecnológico e industrial vivido en los últimos años ha agravado el problema de la gestión de residuos peligrosos.

En la literatura encontramos diversas perspectivas en la modelización matemática de los problemas de localización y rutas de material peligroso. Algunos modelos se centran en minimizar los riesgos que conlleva el transporte del material peligroso, otros tratan de minimizar el coste total del manejo del material. Este tipo de problemas pueden clasificarse dentro del apartado anterior, como problemas de localización multiobjetivo, puesto que habitualmente requieren considerar simultáneamente en el modelo una serie de objetivos que a menudo entran en conflicto unos con otros.

En la literatura encontramos artículos que tratan con la localización y rutas de centros de tratamiento y de eliminación de este tipo de residuos. Por ejemplo, F. Samanlioglu (2013) [29] escribió un artículo en el que se plantea un modelo matemático cuyo objetivo es la determinación de las localizaciones de los centros de reciclado y de eliminación de los residuos así como las rutas de los materiales hasta los centros de reciclado antes del proceso de tratamiento y las rutas de los residuos de desecho a los centros de eliminación después del proceso de reciclado. Además, no sólo se tienen en cuenta los gastos en transporte, sino que también se consideran los riesgos procedentes del tratamiento con residuos peligrosos. De esta manera se plantea un modelo multiobjetivo: se minimiza el coste total que incluye el coste de transporte, el riesgo que conlleva dicho transporte y la exposición a los materiales peligrosos. Además, se aplica el modelo matemático de localización y rutas anterior a un caso real en Turquía. Se consideran 41 de sus 131 distritos como posibles localizaciones de estos centros de tratamiento y de eliminación de residuos peligrosos.

En 2004, Cappanera et al. [8] formulan un modelo de localización y rutas de instalaciones para residuos indeseables (OFLR por sus siglas en inglés). Hacen las siguientes hipótesis: se considera un único material (modelo de producto único), los lugares afectados se representan

como puntos en el plano, los lugares que ni consumen ni generan dicho material pueden ser afectados por el transporte del mismo y se fijan unos umbrales de exposición para cada uno de los lugares afectados.

El problema OFLR consiste en seleccionar un conjunto de localizaciones entre los distintos candidatos para abrir una serie de instalaciones y determinar las rutas que se han de realizar que establecen el flujo del material no deseado entre las zonas afectadas y las instalaciones abiertas de manera que se minimice el coste de apertura de instalaciones y el coste de transporte. Además, lo que diferencia este problema del problema de localización y rutas visto en el Capítulo 2 es que han de satisfacerse una serie de restricciones relativas a la naturaleza del material tratado. Un conjunto de restricciones reflejan que para cada lugar afectado, la exposición total causada por las rutas a lo largo de los arcos de la red no exceda un umbral fijo. Análogamente se imponen otras restricciones que aseguran que, para cada lugar afectado, la exposición total causada por las apertura de las instalaciones no excede un umbral fijo.

### 3.4. Modelos de localización con residuos urbanos

El tratamiento de los residuos urbanos se ha convertido en un asunto de vital importancia en la sociedad actual debido al impacto político, económico y social. La recogida adecuada de residuos es una de las fases más importantes del ciclo de los residuos (generación-transformación-eliminación). La SGAPC (Subdirección General de Análisis, Prospectiva y Coordinación) publicó en 2014 un documento\* en el que se refleja lo siguiente:

El incremento en la generación de residuos a escala global es continuo. La preventión y correcta gestión son esenciales para evitar sus impactos negativos sobre los ecosistemas, la biodiversidad y la salud humana. La concepción del residuo como recurso para incorporarlo al sistema productivo es un reto clave y una obligación normativa para los estados miembros de la UE a cumplir antes de 2020. Desde un punto de vista socioeconómico, el sector de la recogida y del tratamiento (valorización y eliminación) de residuos está en crecimiento y constituye una de las fuentes reconocidas de empleo verde, y en especial el reciclado. Todo ello debe responder a las demandas de la sociedad, que percibe la generación de residuos como una de las mayores preocupaciones ambientales y como una responsabilidad individual y colectiva (...).

Existen múltiples artículos en la literatura que tratan este tema. Un ejemplo de ello es el artículo publicado por Bautista et al. (2006) [6]. En él, los autores consideran una ciudad (o parte de ella) en la que se necesita instalar un nuevo sistema de recogida. El objetivo es localizar un mínimo de puntos de recogida de manera que cada ciudadano tenga al menos uno de estos puntos a un máximo de  $L$  unidades de distancia de su casa. Dicho objetivo reflejará las características del sistema de recogida como las condiciones especiales sobre el impacto visual

\*[http://www.magrama.gob.es/es/ministerio/servicios/analisis-y-prospectiva/AyP\\_RESIDUOS\\_V10\\_tcm7-323997.pdf](http://www.magrama.gob.es/es/ministerio/servicios/analisis-y-prospectiva/AyP_RESIDUOS_V10_tcm7-323997.pdf)

de los contenedores en las calles o el ruido en la fase de recogida, o la calidad del servicio medida en términos de la distancia máxima entre usuarios y puntos de recogida. En el artículo se proponen dos formulaciones para el problema de localización de puntos de recogida visto como un problema MAX-SAT (máxima satisfacción).

Para resolver este problema, que puede ser visto como un problema de conjunto de cobertura (en inglés set covering problem) donde cada ubicación tiene el mismo coste para el establecimiento de una instalación, proponen un algoritmo genético. Además, para asistir a la toma de decisiones, los autores han realizado una aplicación, SIRUS, que constituye un sistema de ayuda al diseño y dirección de operaciones de recogida de residuos sólidos urbanos.

Otro artículo en el que se trata el tema de recogida de residuos urbanos es el publicado en 2012 por Ghiani et al. [15]. En él, los autores formulan un modelo matemático en el que el objetivo es minimizar el número de puntos de recogida. De esta manera, no solo se minimiza el coste de localización de cada uno de los contenedores sino que se reduce el impacto visual debido a la presencia de estos puntos de recogida. Además se tiene en cuenta distintos tipos de contenedores de distintas dimensiones y capacidades. El modelo propuesto por los autores no sólo determina el número de puntos de recogida que han de fijarse para satisfacer la demanda de los clientes, sino que también determina la asignación de los clientes (ciudadanos) a los distintos puntos así como la selección de los distintos tipos de contenedores.

Para abordar el problema se propone una aproximación heurística en dos pasos. En el primero de ellos se trata con el problema de seleccionar qué puntos de recogida son activados dentro del conjunto inicial. Para resolver este problema de localización se utiliza asimismo una aproximación heurística. En una segunda fase, una vez que ya se tienen localizados los puntos de recogida, se trata de seleccionar qué tipos de contenedores se colocan en cada uno de ellos.



*Los modelos del matemático, como los del pintor o los del poeta deben ser hermosos. La belleza es la primera prueba; no hay lugar permanente en el mundo para unas matemáticas feas.*

G. H. Hardy

# 4

## Resolución de un caso práctico en Aragón

---

El objetivo de este último capítulo es aplicar algunos de los modelos y conceptos estudiados en los capítulos anteriores para resolver una situación realista en la Comunidad de Aragón. Tras la búsqueda de datos se ofrece una visión general de la situación actual en el tema relacionado con los residuos sólidos urbanos en Aragón. Por último aplicaremos distintos modelos de localización para determinar la apertura de vertederos en las distintas provincias aragonesas.

Para la resolución computacional de dichos modelos utilizaremos CPLEX®. Nos apoyaremos en el manual de uso básico para IBM ILOG CPLEX V12.6.1 realizado para la Beca de Colaboración en el curso 2015-2016 en departamentos universitarios convocadas por el Ministerio de Educación, Cultura y Deporte [23].

### 4.1. Residuos sólidos urbanos

La Ley de Residuos (Ley 22/2011, de 28 de julio, de residuos y suelos contaminados) define los residuos sólidos urbanos (RSU) como los generados en las actividades desarrolladas en los hogares como consecuencia de las actividades domésticas. Según la citada ley, tienen también la consideración de residuos urbanos los residuos resultantes de la limpieza de lugares públicos como vías, parques y áreas recreativas o playas, así como los desechos resultantes del abandono de enseres voluminosos como vehículos, muebles o escombros procedentes de obras de construcción o reparación en el hogar. Asimismo, se incluyen los generados en comercios, servicios y oficinas, así como todo aquel que no tenga la calificación de residuo peligroso y no sea regulado expresamente por una normativa específica.

Se entiende por gestión de los RSU como la cadena de operaciones realizadas desde que

dichos residuos son generados hasta la última fase en la que son tratados de acuerdo con su legislación, características o sus posibilidades de recuperación y comercialización (en el caso en que éstos puedan ser reciclados). A lo largo de dicho conjunto de operaciones se tienen en cuenta criterios de tipo económico y ambiental.

Así pues, su gestión comprende las fases de depósito, recogida, transporte y transformación o tratamiento.

#### **4.1.1. Pre-recogida o depósito**

Esta fase comprende el periodo entre la generación y el depósito de los residuos sólidos urbanos.

En general, los RSU están compuestos de vidrio, papel y cartón, restos orgánicos, plásticos, textiles, metales, madera y escombros. En la actualidad se dispone en su mayoría de contenedores específicos para su clasificación (papel y cartón, vidrio, envases y materia orgánica) de manera que se facilita la fase de tratamiento. Existen otro tipo de contenedores conocidos como puntos limpios en los que se depositan materiales considerados peligrosos como pueden ser pinturas o pilas.

#### **4.1.2. Recogida**

En esta fase se procede a la carga de los contenedores mencionados en la etapa anterior para su traslado a las plantas de tratamiento. Para ello se utilizan vehículos habilitados que pueden variar dependiendo del tipo de residuo que recogen. Dependiendo de si la recogida es selectiva o no, es decir, si los residuos se depositan mezclados en los contenedores o si el depósito se ha realizado separando los residuos según sus características en los contenedores correspondientes, puede ser adecuado un tipo de procedimiento u otro. Por ejemplo una técnica conocida como compactación es adecuada para los residuos orgánicos pero no es empleada en el caso del vidrio.

También se realizan recogidas de manera excepcional de materiales voluminosos como pueden ser muebles o vehículos.

#### **4.1.3. Transporte**

La fase del transporte es la más delicada, tanto económica como logísticamente ya que se ha de diseñar la red de rutas atendiendo a criterios como ventanas de tiempo (horarios en los que se puede efectuar la recogida y depósito de los residuos) o posibles restricciones en los vehículos en las distintas vías urbanas (como por ejemplo en su tamaño).

Esta etapa comprende las rutas realizadas desde los puntos de recogida hasta los vertederos o centros de tratamiento. Se puede realizar directamente mediante los vehículos de recogida o bien los residuos pueden ser llevados a centros conocidos como Estaciones de Transferencia (ET). El motivo por el que se fijan estas ET es porque el transporte a larga distancia con un vehículo habilitado para la recogida en la ciudad supone altos costes. De esta manera, en estas estaciones se descargan y almacenan de modo temporal los RSU y una vez allí se compactan y

transportan en otro tipo de vehículos con una mayor capacidad de manera que se reducen gastos de transporte.

#### 4.1.4. Tratamiento

El tratamiento es la fase final del proceso. Incluye las operaciones realizadas para la eliminación o el aprovechamiento, siempre que sea posible, de los residuos.

Esta etapa depende de si los residuos vienen ya separados desde el origen (recogida selectiva) o si han sido depositados indistintamente. Por ejemplo, en el caso del papel o el vidrio se dirigen directamente a la planta de reciclado. Si la recogida no es selectiva se han de separar dependiendo de su naturaleza. Una vez separados los residuos hay que realizar su tratamiento dependiendo de las características y del estado de cada tipo de residuo.

En el presente trabajo nos centraremos en la fase previa de localización de los vertederos y las estaciones de transferencia a los cuales se transportarán los RSU.

## 4.2. Situación actual en la Comunidad de Aragón

Antes de plantear el problema de la localización de plantas de reciclaje estudiamos la situación actual. Los datos ofrecidos a lo largo del capítulo han sido extraídos en su mayor parte de la página del Instituto Aragonés de Estadística (IAE) \*.

En Aragón existen un total de 731 municipios, 273 de los cuales forman parte de la provincia de Zaragoza, 215 de la provincia de Huesca, y 243 de la provincia de Teruel.

Según los datos estadísticos a fecha del 1 de enero de 2010 la población de Aragón es de 1.345.419, lo que supone el 2,85 % del total de la población española. La densidad de población es de 26,77 hab/km<sup>2</sup>.

La Tabla 4.1 recoge la población de la Comunidad de Aragón y de sus provincias a fecha de 1 de enero de 2010.

Tabla 4.1: Población de Aragón y sus provincias

Provincia	Población	%
Zaragoza	978.130	72,48
Huesca	227.609	16,87
Teruel	143.728	10,65
Total Aragón	1.349.467	100,0

Aragón produce 597.489 toneladas de RSU por año, lo que constituye el 2,62 % de la producción de residuos en España (22.735.142 toneladas/año) según los datos del año 2004.

En la Figura 4.2 aparecen reflejadas las localizaciones de los distintos vertederos y Estaciones de Transferencia (ET) que se encuentran activos actualmente en la Comunidad de Aragón. Los datos han sido extraídos del IAE y se recogen en la Tabla 4.2.

\*[http://www.aragon.es/DepartamentosOrganismosPublicos/Institutos/InstitutoAragonésEstadística/ÁreasTematicas/14\\_Medio\\_Ambiente\\_Y\\_Energía/11\\_Residuos/ci.03\\_Tratamiento\\_Residuos.detalleDepartamento#](http://www.aragon.es/DepartamentosOrganismosPublicos/Institutos/InstitutoAragonésEstadística/ÁreasTematicas/14_Medio_Ambiente_Y_Energía/11_Residuos/ci.03_Tratamiento_Residuos.detalleDepartamento#)

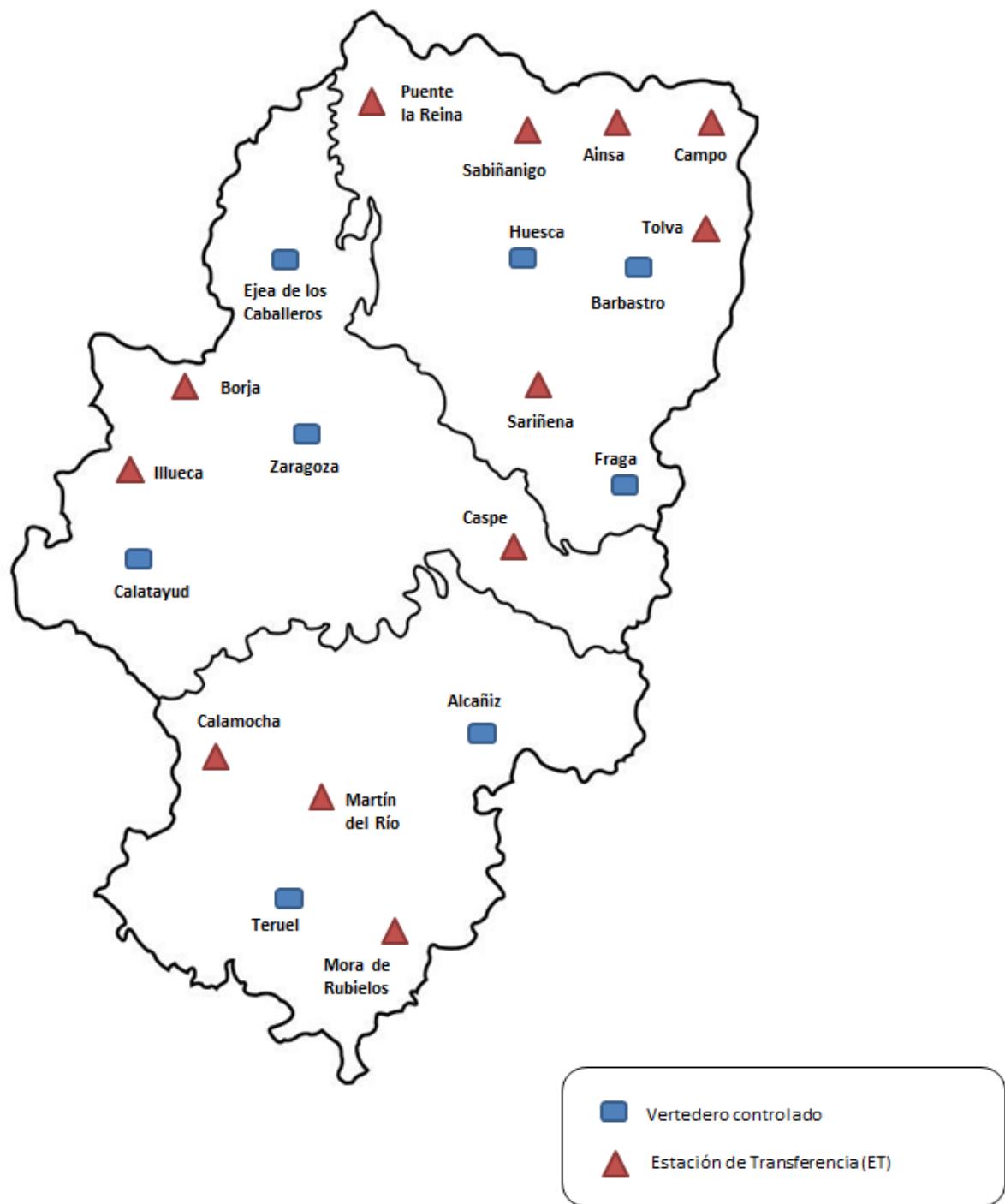


Figura 4.1: Vertederos y estaciones de transferencia en Aragón en 2013

Tabla 4.2: Vertederos de residuos sólidos urbanos considerados controlados en Aragón. Año 2013. (Fuente IAE)

Provincia	Vertedero controlado	Municipios atendidos	RSU tratados (t/año)
Zaragoza	Calatayud	144	38.226
	Vertido directo	131	34.211
	ET de Illueca	13	4.015
	Ejea de los Caballeros	82	32.031
	Vertido directo	48	21.989
	ET de Borja	34	10.042
	Zaragoza	47	176.136
	Vertido directo CTRUZ(*)	47	176.136
Huesca	Barbastro	105	30.034
	Vertido directo	61	22.795
	ET de Ainsa	19	3.733
	ET de Campo	16	1.155
	ET de Tolve	9	2.351
	Fraga	34	12.504
	Vertido directo	18	9.118
	ET Sariñena	16	3.386
	Huesca	76	40.560
	Vertido directo	48	26.475
	ET Puente la Reina	14	1.226
	ET Sabiñánigo	14	12.859
Teruel	Alcañiz	63	21.708
Teruel		180	31.949
	Vertido directo	118	24.808
	ET Martín del Río	30	3.368
	ET Mora de Rubielos	8	503
	ET Calamocha	24	3.270

(\*) CTRUZ (Complejo para el Tratamiento de Residuos Urbanos de Zaragoza)

En la tabla 4.3 aparecen reflejados los datos de producción de residuos en España desde el año 1991 hasta el año 2008. Para estimar la cantidad de RSU generados por cada municipio se hará uso del último dato disponible, resaltado en la tabla.

Tabla 4.3: Fuente: IAE

Año	Kg por habitante y día	Kg por habitante y año	Población	Tn por año
1991	0,92	330	38.885.580	12.821.450
1992	0,99	357	38.885.580	13.827.655
1993	0,99	362	39.433.942	14.255.923
1994	0,99	363	39.433.942	14.296.302
1995	1,04	378	39.433.942	14.914.235
1996	1,06	388	39.433.942	15.307.652
1997	1,19	443	39.668.939	17.178.736
1998	1,20	437	39.669.394	17.418.169
1999	1,26	459	39.669.394	18.376.532
2000	1,30	474	40.202.160	18.925.109
2001	1,29	471	40.847.371	19.124.750
2002	1,34	488	41.837.894	20.465.614
2003	1,38	502	42.717.064	21.444.705
2004	1,44	524	43.197.684	22.735.142
2005	1,39	507	44.108.530	22.353.152
2006	1,45	529	44.708.964	23.648.032
2007	1,43	521	45.200.737	23.562.199
2008	1,43	521	46.157.822	24.049.826

Una vez presentado el panorama actual, nuestro objetivo es, manteniendo algunas de las características del sistema anterior, obtener las localizaciones resultantes de aplicar los problemas estudiados a lo largo del Capítulo 2.

En primer lugar, resolveremos el problema teniendo en cuenta únicamente la apertura de vertederos (sin Estaciones de Transferencia). Mantendremos el número de vertederos controlados por provincia, es decir, impondremos la apertura de tres vertederos en la provincia de Zaragoza, otros tres en la provincia de Huesca y dos más en la provincia de Teruel. A continuación, se resolverá el modelo para el caso en el que también se abran Estaciones de Transferencia.

Por último, introduciremos un problema de rutas para determinar el transporte entre los municipios y los vertederos.

### 4.3. Modelo de localización sin Estaciones de Transferencia

Para resolver con CPLEX® los problemas de localización nos restringiremos en primer lugar a los municipios con más de 1000 habitantes. Del total de 731 municipios, únicamente 117 de éstos tienen más de 1000 habitantes. De ellos, 63 pertenecen a la provincia de Zaragoza, 32 a la provincia de Huesca y 20 a la de Teruel. Los datos han sido extraídos de la página oficial de la Secretaría de Estado de Administraciones Públicas del Gobierno de España\*\*.

\*\*[http://www.seap.minhap.es/dms/es/web/areas/politica\\_local/sistema\\_de\\_informacion\\_local\\_-SIL-/registro\\_eell/estudios/estudios\\_ccaa/parrafo/03/LASENTIDADESLOCALESENARAGON.pdf](http://www.seap.minhap.es/dms/es/web/areas/politica_local/sistema_de_informacion_local_-SIL-/registro_eell/estudios/estudios_ccaa/parrafo/03/LASENTIDADESLOCALESENARAGON.pdf)

Las tablas con los datos relativos a población y residuos generados por cada uno de los municipios se han adjuntado en el Anexo A. Como se ha indicado, el número de toneladas de RSU generadas anualmente por cada uno de los municipios se ha estimado con el dato de producción de residuos sólidos urbanos en España en el año 2008 por habitante y año dado en la Tabla 4.3.

Se ha estimado asimismo la capacidad del vertedero para cada municipio (véase 4.3.2).

### 4.3.1. Problema de la p-mediana

En la formulación del problema no se impone ninguna localización concreta para los vertederos. Así, supondremos que en cada municipio puede establecerse un vertedero. Si recordamos la formulación dada en 2.1, se tiene que el conjunto de clientes  $I$  y el conjunto de posibles nodos depósito  $J$  coinciden. Como ya se ha indicado, se mantiene el número de vertederos actual por provincia en Aragón, es decir, impondremos la apertura de tres vertederos en la provincia de Zaragoza, otros tres en la provincia de Huesca y dos más en la provincia de Teruel.

Podemos añadir a la formulación dada en 2.1 el siguiente grupo de restricciones para cada posible localización  $j \in J$ :

$$y_j \leq x_{jj}.$$

De esta manera, se obliga a que si un vertedero está abierto en un determinado municipio  $j \in J$ , entonces la demanda del propio municipio es satisfecha en dicho vertedero.

Así, podemos formular nuestro problema como sigue

$$\min_{x_{ij}} \quad \sum_{i \in I} \sum_{j \in J} h_i d_{ij} x_{ij} \quad (4.1a)$$

sujeto a

$$\sum_{j \in J} x_{ij} = 1, \quad \forall i \in I \quad (4.1b)$$

$$\sum_{j \in J} y_j = p \quad (4.1c)$$

$$x_{ij} - y_j \leq 0, \quad \forall i \in I, \forall j \in J \quad (4.1d)$$

$$y_j - x_{jj} \leq 0, \quad \forall j \in J \quad (4.1e)$$

$$y_j \in \{0, 1\}, \quad \forall j \in J \quad (4.1f)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i \in I, \forall j \in J \quad (4.1g)$$

El objetivo de nuestro problema es minimizar la distancia total ponderada por el volumen de residuos generados, (4.1a).

Asimismo, los tres primeros grupos de restricciones, (4.1b), (4.1c) y (4.1d), no varían. El primer grupo de restricciones nos asegura que cada cliente reciba servicio exactamente de un vertedero. La segunda restricción obliga a que se abran exactamente  $p$  depósitos. Las restricciones (4.1d) aseguran que un cliente no reciba servicio de un vertedero cerrado.

Introducimos un cuarto grupo de restricciones, (4.1d), como se ha indicado, hacen que si un vertedero está abierto en un determinado lugar, el respectivo lugar es asignado a ese vertedero.

Finalmente, en (4.1f) y (4.1g) se reflejan las restricciones de integridad de las variables.

El modelo ha sido resuelto en CPLEX® en un ordenador de CPU Mobile DualCore Intel Core i5-2415M 2300 MHz, de disco SSD 240Gb y memoria RAM de 16Gb. No se hace mención de los tiempos de ejecución dado que en ningún caso superan los 5 segundos.

### Resolución en CPLEX para la provincia de Zaragoza

Mantenemos el número de vertederos abiertos en la provincia de Zaragoza, de modo que fijamos  $p = 3$ .

CPLEX nos ofrece la siguiente solución:

- Se abren vertederos en la capital, Zaragoza, y en los municipios de Calatayud y Gallur.

Las asignaciones del resto de los municipios a los tres vertederos son:

- Zaragoza = {Lumpiaque, Nonaspe, Sástago, Escatrón, Magallón, Fabara, Gelsa, Morata de Jalón, Muel, Pastriz, Alfamén, Torres de Berrellén, Belchite, Brea de Aragón, Ateca, Alfajarín, Daroca, El Burgo de Ebro, Mequinenza, Villamayor de Gállego, San Mateo de Gállego, Cadrete, Pinseque, Villanueva de Gállego, María de Huerva, Fuentes de Ebro, Épila, La Muela, La Puebla de Alfindén, Alagón, Zuera, Cuarte de Huerva, La Almunia de Doña Godina, Caspe, Ejea de los Caballeros, Utebo, Zaragoza}.
- Calatayud = {Maluenda, Alhama de Aragón, Remolinos, Figueruelas, Sádaba, Maella, Quinto, Pina de Ebro, Calatorao, Illueca, Ricla, Calatayud}.
- Gallur = {Biota, Bujaraloz, Luceni, Ainzón, Ariza, Leciñena, Calatorao, Gallur, Cariñena, Pedrola, Mallén, Borja, Tauste, Tarazona}.

Si comparamos la solución obtenida con la situación actual en la provincia de Zaragoza, dos de los tres vertederos coinciden: se abre un vertedero controlado en Zaragoza y otro en Calatayud. Por otro lado, el restante dejaría de estar en Ejea de los Caballeros para localizarse en el municipio de Gallur.

### Resolución en CPLEX para la provincia de Huesca

Del mismo modo, mantenemos el número de vertederos abiertos en la provincia de Huesca, de modo que fijamos  $p = 3$ .

CPLEX nos ofrece la siguiente solución:

- Se abren vertederos en los municipios de Huesca, Sabiñánigo y Binaced.

Las asignaciones del resto de los municipios a los tres vertederos son:

- Huesca = {Tardienta, La Sotonera, Laluezza, Ayerbe, Lanaja, Gurrea de Gállego, Grañén, Almudévar, Sariñena, Monzón, Huesca}.

- Sabiñánigo = {Boltaña, Sallent de Gállego, Biescas, Benasque, Aínsa-Sobrarbe, Sabiñánigo, Jaca}.
- Binaced = {Alcolea de Cinca, Benabarre, Albalate de Cinca, Torrente de Cinca, Belver de Cinca, Altorigón, Binaced, Zaidín, Graus, Tamarite de Litera, Binéfar, Fraga, Barbastro}.

En este caso, únicamente la localización en el municipio de Huesca para abrir el vertedero controlado coincide con la situación actual en la provincia de Huesca.

### Resolución en CPLEX para la provincia de Teruel

De nuevo fijamos el número de vertederos abiertos en la provincia de Teruel como el actual,  $p = 2$ .

CPLEX nos ofrece la siguiente solución:

- Se abren vertederos en Alcañiz y Teruel.

Las asignaciones del resto de los municipios a los vertederos resultan:

- Teruel = {Albarracín, Santa Eulalia, Sarrión, Mora de Rubielos, Monreal del Campo, Cella, Utrillas, Calamocha, Teruel}.
- Alcañiz = {Calaceite, Montalbán, Mas de las Matas, Híjar, Albalate del Arzobispo, Valderrobres, Alcorisa, Calanda, Andorra, Alcañiz}.

Tras resolver el problema de la  $p$ -mediana sin capacidades para el caso de Teruel obtenemos la misma solución que la situación actual en la provincia, se abren los vertederos controlados en los mismos municipios: Alcañiz y Teruel.

#### 4.3.2. Problema de la $p$ -mediana con capacidades

A continuación, vamos a aplicar el problema de la  $p$ -mediana con capacidades (2.3). Hemos de estimar una capacidad para cada uno de los vertederos (para cada una de las posibles localizaciones o municipios). Para ello nos apoyaremos en los datos obtenidos sobre los vertederos que existen actualmente en Aragón.

Nuestro objetivo es construir un modelo sencillo que permita explicar el total de residuos tratados en los vertederos que se encuentran activos actualmente en Aragón en función de los datos disponibles en cuanto a población y extensión del municipio.

Disponemos del siguiente conjunto de datos:

Localización vertedero	Extension (km <sup>2</sup> )	Población (2014)	Total residuos tratados
Barbastro	107,6	17109	30034
Fraga	437,6	14926	12504
Huesca	161	52555	40560
Calatayud	154,25	20926	38226
Ejea de los Caballeros	609,92	17331	32031
Zaragoza	973,78	674317	176136
Alcañiz	472,12	16424	21708
Teruel	440,41	35841	31949

Ajustamos los datos a un modelo de regresión lineal. La variable explicada será ‘Total residuos tratados’, y el resto de variables, ‘Extensión’ y ‘Población’ serán variables explicativas. Para el análisis estadístico se ha utilizado el programa R Commander. La orden es:

```
lm(formula = TotalRes ~ Extension + Poblacion, data = DatosCapacidad)
```

A continuación mostramos la salida que nos ofrece R Commander:

```
Residuals:
Barbastro      Fraga      Huesca    Calatayud   Ejeacaballeros   Zaragoza
-1898.3   -12778.2     964.2      6226.6      9355.9      -183.2
Alcanniz      Teruel
-3300.0      1612.9

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 2.975e+04 5.937e+03  5.011  0.00407 ***
Extension   -1.854e+01 1.594e+01 -1.163  0.29740
Poblacion    2.441e-01 1.998e-02 12.217  6.5e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7844 on 5 degrees of freedom
Multiple R-squared:  0.9841, Adjusted R-squared:  0.9777
F-statistic: 154.8 on 2 and 5 DF,  p-value: 3.185e-05
```

Observamos que este modelo explica un 97,77% de la varianza (Multiple R-squared: 0.9777).

Para modelos más complejos, R Commander dispone de una herramienta que nos permite refinar el modelo haciendo un modelo paso a paso. La idea general de dicha técnica es quitar y añadir variables explicativas a un modelo ya existente, evaluar los efectos de dichos cambios y elegir qué variables mantener y cuáles omitir.

Existen dos criterios para determinar el modelo paso a paso: criterio Adelante/Atrás y criterio Atrás/Adelante. La diferencia entre estos reside en si se inicia el modelo sin variables y se van introduciendo aquellas que presentan una mayor relación con la variable respuesta (Adelante/Atrás) o si se ajusta un primer modelo con todas las variables y se van eliminando (Atrás/Adelante).

En nuestro caso, tanto el criterio Adelante/Atrás como el criterio Atrás/Adelante nos sugie-

ren un modelo con la variable ‘Población’ como la única variable explicativa. Tras construir dicho modelo, obtenemos un R cuadrado de 0,9764 (ligeramente inferior al modelo con ambas variables). Por esto, y por la simplicidad del modelo, decidimos mantener ambas variables explicativas en el modelo, de manera que obtenemos el siguiente modelo de regresión lineal:

$$\text{Total de residuos generados} = 29750 - 18,54 * \text{Extensión} + 0,2441 * \text{Población}.$$

Aplicamos dicho modelo al resto del conjunto de datos para calcular las capacidades estimadas para el resto de los municipios. Las capacidades aparecen reflejadas en la última columna de las tablas de población adjuntadas en el Anexo [A](#).

El Script de R para la creación del modelo anterior está reflejado en el Anexo [B.1](#).

### **Resolución en CPLEX para la provincia de Zaragoza**

Cabe destacar que para que el problema CPMP sea factible, las demandas de cada uno de los clientes o municipios ha de ser inferior al máximo de las capacidades de los depósitos o vertederos. En el caso de la Provincia de Zaragoza, para la capital se tiene que la cantidad de residuos sólidos urbanos generados por el conjunto de la población es superior a la capacidad asignada al vertedero, que además se trata del vertedero con máxima capacidad (351.319,157 toneladas producidas frente a una capacidad de 176.288,3817 toneladas de capacidad).

De este modo, en un primer intento de resolver el problema CPMP para la provincia de Zaragoza, aumentamos la capacidad del vertedero asignado al municipio de Zaragoza a 400.000 toneladas.

Una vez modificada la capacidad del vertedero asignado al municipio de Zaragoza, ejecutamos el modelo en CPLEX para el caso  $p = 3$ , obteniendo que no existe una solución factible para los parámetros fijados.

Aumentamos el número de vertederos abiertos a  $p = 4$ . CPLEX nos ofrece la siguiente solución:

- Se abren vertederos en Gallur, Caspe, Calatayud y Zaragoza.

Las asignaciones del resto de los municipios a los tres vertederos son:

- Gallur = {Biota, Luceni, Remolinos, Magallón, Figueruelas, Gallur, Cariñena, Pedrola, Mallén, Tauste, Tarazona, Ejea de los Caballeros}.
- Caspe = {Bujaraloz, Nonaspe, Escatrón, Fabara, Gelsa, Maella, Quinto, Mequinenza, Pina de Ebro, Caspe}.
- Calatayud = {Lumpiaque, Maluenda, Alhama de Aragón, Ariza, Morata de Jalón, Muel, Belchite, Brea de Aragón, Ateca, Daroca, Calatorao, Illueca, Ricla, Épila, Borja, La Muela, Calatayud}.
- Zaragoza = {Sástago, Ainzón, Leciñena, Pastriz, Alfamén, Torres de Berrellén, Sádaba, Alfajarín, El Burgo de Ebro, Villamayor de Gállego, San Mateo de Gállego, Cadrete, Pinseque, Villanueva de Gállego, María de Huerva, Fuentes de Ebro, La Puebla de Alfindén, Alagón, Zuera, Cuarte de Huerva, La Almunia de Doña Godina, Utebo, Zaragoza}.

### Resolución en CPLEX para la provincia de Huesca

Del mismo modo, mantenemos el número de vertederos abiertos en la provincia de Huesca, de modo que fijamos  $p = 3$ .

CPLEX nos ofrece la siguiente solución:

- Se abren vertederos en los municipios de Huesca, Alcorisa y Utrillas.

Las asignaciones del resto de los municipios a los tres vertederos son:

- Huesca = {Grañén, Sariñena, Monzón, Huesca}.
- Alcorisa = {Tardienta, La Sotonera, Boltanya, Ayerbe, Lanaja, Sallent de Gállego, Biescas, Gurrea de Gállego, Benasque, Aínsa-Sobrarbe, Almudévar, Sabiñánigo, Jaca, Alcorisa}.
- Utrillas = {Lalueza, Alcolea de Cinca, Benabarre, Albalate de Cinca, Torrente de Cinca, Belver de Cinca, Altorricón, Binaced, Zaidín, Graus, Tamarite de Litera, Binéfar, Fraga, Barbastro, Utrillas}.

### Resolución en CPLEX para la provincia de Teruel

De nuevo fijamos el número de vertederos abiertos en la provincia de Teruel como el actual,  $p = 2$ .

CPLEX nos ofrece la siguiente solución:

- Se abren de nuevo los vertederos en Alcañiz y en Teruel.

Las asignaciones del resto de los municipios a los vertederos son:

- Teruel = {Albarracín, Santa Eulalia, Sarrión, Mora de Rubielos, Monreal del Campo, Cella, Utrillas, Calamocha, Teruel}.
- Alcañiz = {Calaceite, Montalbán, Mas de las Matas, Híjar, Albalate del Arzobispo, Valderrobres, Alcorisa, Calanda, Andorra, Alcañiz}.

## 4.4. Modelo de localización con Estaciones de Transferencia

En este segundo planteamiento localizaremos, además de los vertederos, las estaciones de transferencia. Resolveremos el problema en dos fases. En primer lugar, para identificar las posibles localizaciones de las ET, realizaremos un análisis clúster que nos permita agrupar en un número reducido de grupos los municipios. Una vez obtenidos los grupos, seleccionaremos como ET para cada clúster el municipio con un mayor índice de población y se aplicarán los problemas de localización estudiados para la localización de los vertederos entre dichas ET. En este modelo tendremos en cuenta todos aquellos municipios que tengan más de 500 habitantes.

En una primera fase localizaremos las ET. Como ya se ha adelantado, para ello realizaremos un análisis clúster para dividir en grupos los municipios. Del mismo modo que en el problema de localización, mantendremos el número de Estaciones de Transferencia y el número de vertederos por provincia.

- Zaragoza: 3 vertederos y 2 estaciones de transferencia.
- Huesca: 3 vertederos y 6 estaciones de transferencia.
- Teruel: 2 vertederos y 3 estaciones de transferencia.

Dado que en una segunda fase seleccionaremos los vertederos entre el grupo de ET seleccionadas en el primer paso, dividiremos cada clúster en tantos grupos como vertederos y estaciones de transferencia tenga cada provincia:

- Zaragoza:  $k = 5$ .
- Huesca:  $k = 9$ .
- Teruel:  $k = 5$ .

Como variables para realizar las agrupaciones en el análisis clúster se utilizaran las coordenadas geográficas de cada municipio (su latitud y longitud).

Una vez obtenidos los municipios de cada clúster, como criterio seleccionaremos como estación de transferencia aquella con una mayor población, ya que generará una mayor cantidad de residuos y de esta manera se reduce la cantidad de material de desecho a transportar.

#### 4.4.1. Análisis clúster

R Commander nos permite hacer agrupaciones jerárquicas. Los llamados métodos jerárquicos tienen por objetivo la clasificación de observaciones o individuos en grupos homogéneos.

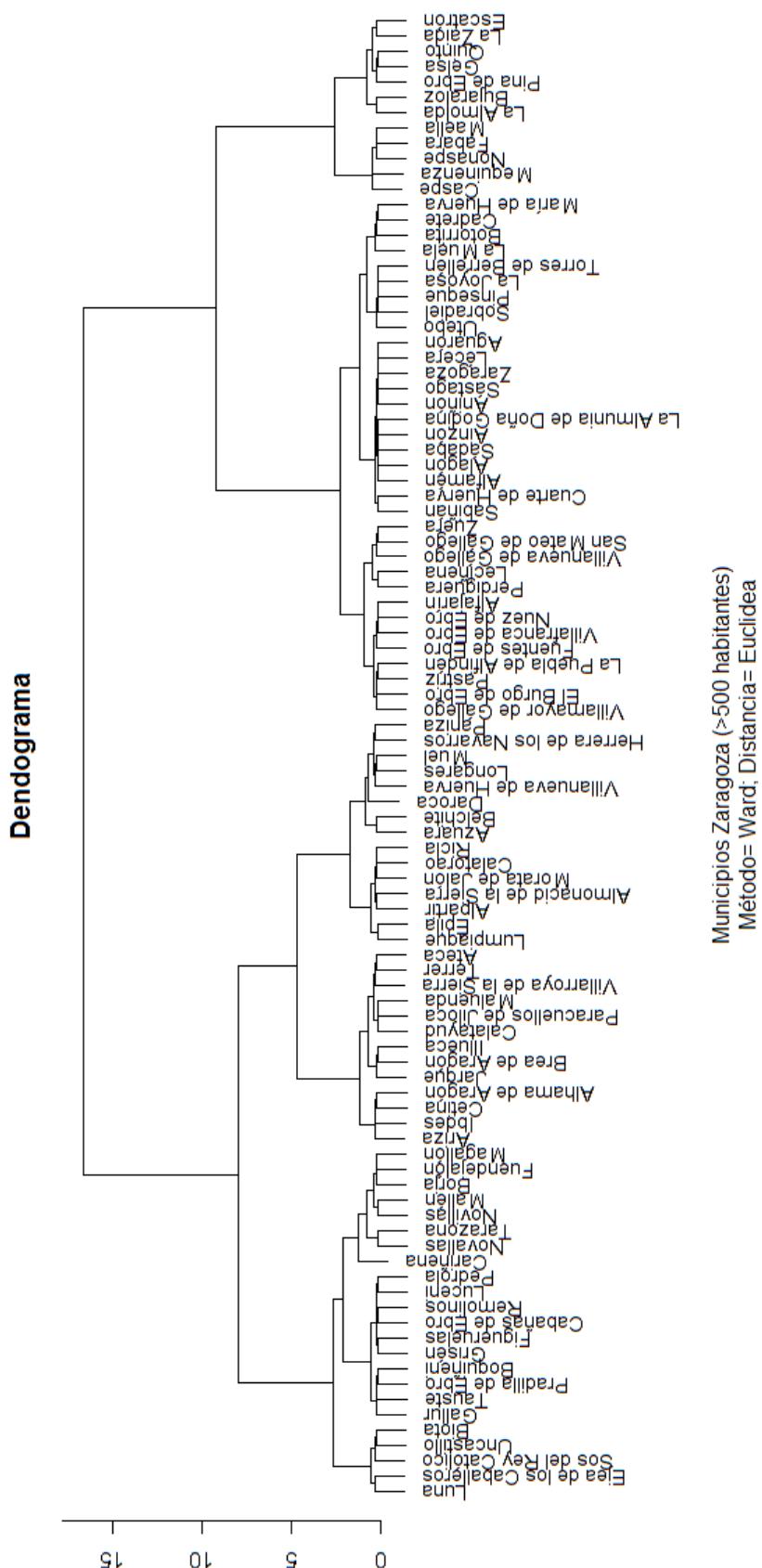
Uno de estos métodos es el método de Ward. Es un procedimiento jerárquico en el cual, en cada etapa, se unen los dos clústers para los cuales se tenga el menor incremento, dentro de cada cluster, en el valor de la suma total de los cuadrados de las diferencias de cada individuo al centroide del clúster. Utilizaremos dicho método en lo que sigue.

Se han adjuntado en el Anexo [B.2](#) los Scripts en R para cada uno de los análisis clúster realizados (uno por cada provincia).

#### Análisis clúster en la provincia de Zaragoza

Realizaremos un análisis clúster utilizando el método de Ward y como medida tomaremos la distancia euclídea.

En la Figura [4.4.1](#) se muestra el dendrograma. R Commander nos permite solicitar el resumen de un número determinado de clústers. Como ya hemos adelantado solicitamos 5 grupos ( $k = 5$ ).



Además de dibujar el dendrograma, una vez hemos determinado el número de grupos, tam-

bién podemos guardar como variable el clúster de pertenencia de cada municipio, de este modo, se creará una nueva variable que es agregada al conjunto de datos activo. Esto nos facilita la localización de los municipios en cada uno de los clúster.

Los 5 grupos son:

- Clúster 1 = {Ibdes, Terrer, Jarque, Paracuellos de Jiloca, Villarroya de la Sierra, Cetina, Maluenda, Alhama de Aragón, Ariza, Brea de Aragón, Ateca, Illueca, Calatayud}.
- Clúster 2 = {Botorrita, Perdiguera, Lécera, Aguarón, Nuez de Ebro, Villafranca de Ebro, Aniñón, Sabiñán, La Joyosa, Sobradiel, Sástago, Ainzón, Leciñena, Pastriz, Alfamén, Torres de Berrellén, Sádaba, Alfajarín, El Burgo de Ebro, Villamayor de Gállego, San Mateo de Gállego, Cadrete, Pinseque, Villanueva de Gállego, María de Huerva, Fuentes de Ebro, La Muela, La Puebla de Alfindén, Alagón, Zuera, Cuarte de Huerva, La Almunia de Doña Godina, Utebo, Zaragoza}.
- Clúster 3 = {La Zaida, La Almolda, Bujaraloz, Nonaspe, Escatrón, Fabara, Gelsa, Maella, Quinto, Mequinenza, Pina de Ebro, Caspe}.
- Clúster 4 = {Cabañas de Ebro, Grisén, Pradilla de Ebro, Novillas, Sos del Rey Católico, Uncastillo, Luna, Novallas, Boquiñeni, Fuendejalón, Biota, Luceni, Remolinos, Magallón, Figueruelas, Gallur, Cariñena, Pedrola, Mallén, Borja, Tauste, Tarazona, Ejea de los Caballeros}.
- Clúster 5 = {Alpartir, Villanueva de Huerva, Herrera de los Navarros, Azuara, Almonacid de la Sierra, Paniza, Longares, Lumpiaque, Morata de Jalón, Muel, Belchite, Daroca, Calatorao, Ricla, Épila}.

En la Figura 4.2, obtenida con Power Bi Desktop, se muestra la localización de los distintos municipios de más de 500 habitantes de la provincia de Zaragoza. Los distintos colores determinan la pertenencia a los distintos grupos.

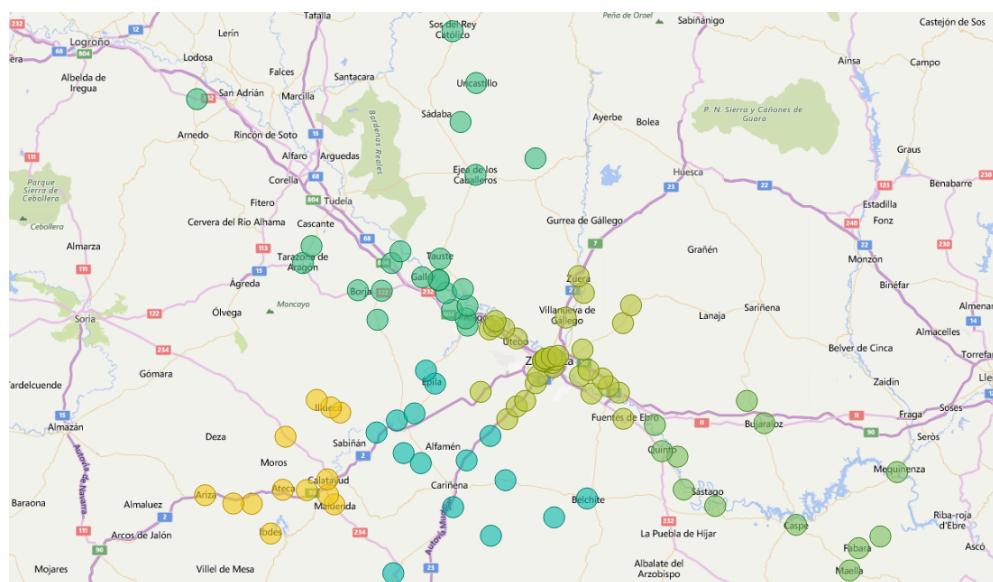


Figura 4.2: Agrupamientos en Zaragoza.

### Análisis clúster en la provincia de Huesca

De nuevo realizaremos un análisis clúster utilizando el método de Ward y como medida tomaremos la distancia euclídea. En la Figura 4.3 se muestra el dendrograma resultante.

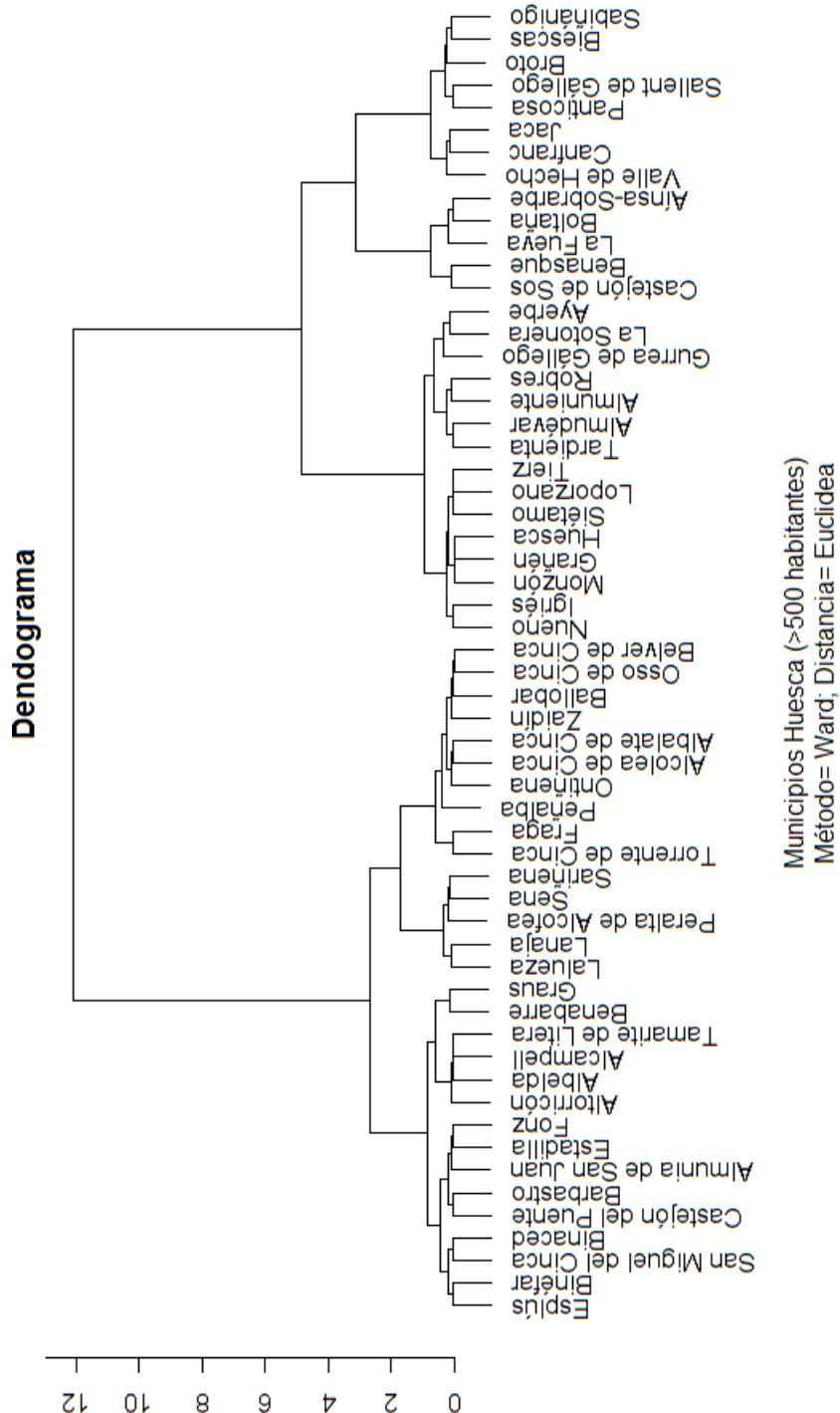


Figura 4.3: Dendrograma de la provincia de Huesca

R Commander nos permite solicitar el resumen de un número determinado de clústeres.

Como se ha indicado, solicitamos 9 grupos ( $k = 9$ ). Asimismo guardamos como variable el clúster de pertenencia de cada municipio.

Los 9 grupos son:

- Clúster 1 = {Sena, Peralta de Alcofea, Laluezza, Lanaja, Sariñena}.
- Clúster 2 = {Broto, Panticosa, Sallent de Gállego, Biescas, Sabiñánigo}.
- Clúster 3 = {Almuniente, Robres, Tardienta, La Sotonera, Ayerbe, Gurrea de Gállego, Almudévar}.
- Clúster 4 = {Loporzano, Nueno, Siétamo, Igriés, Tierz, Grañén, Monzón, Huesca}.
- Clúster 5 = {Canfranc, Valle de Hecho, Jaca}.
- Clúster 6 = {Ontiñena, Peñalba, Osso de Cinca, Ballobar, Alcolea de Cinca, Albalate de Cinca, Torrente de Cinca, Belver de Cinca, Zaidín, Fraga}.
- Clúster 7 = {Castejón del Puente, Esplús, Almunia de San Juan, Estadilla, San Miguel del Cinca, Fonz, Binaced, Binéfar, Barbastro}.
- Clúster 8 = {La Fueva, Castejón de Sos, Boltaña, Benasque, Aínsa-Sobrarbe}.
- Clúster 9 = {Alcampell, Albelda, Benabarre, Altorricón, Graus, Tamarite de Litera}.

En la Figura 4.4, obtenida con Power Bi Desktop, se muestra la localización de los distintos municipios de más de 500 habitantes de la provincia de Huesca. Los distintos colores determinan la pertenencia a los distintos grupos.

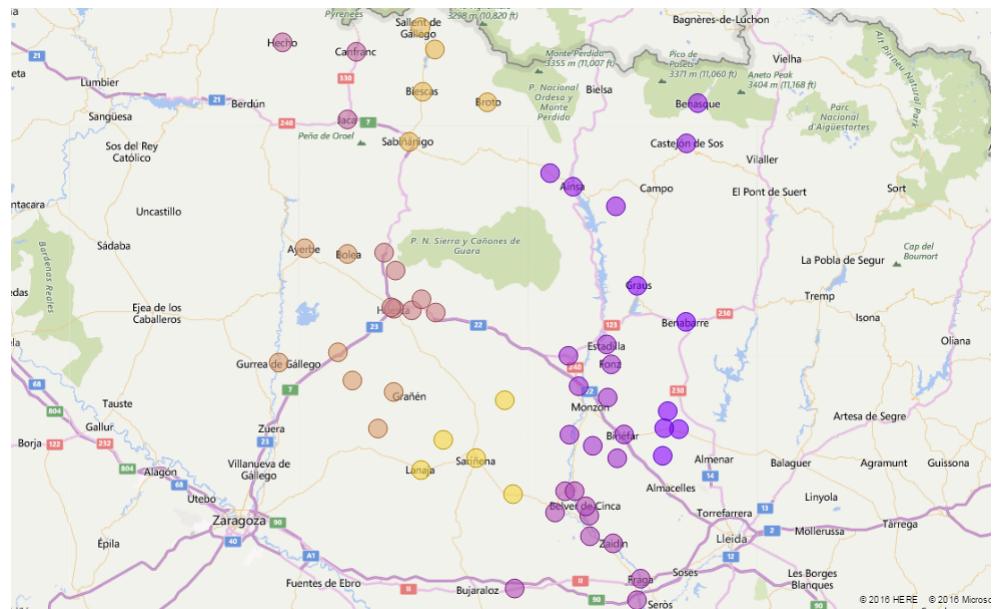


Figura 4.4: Agrupamientos en Huesca.

### Análisis clúster en la provincia de Teruel

Realizaremos un análisis clúster utilizando el método de Ward y como medida tomaremos la distancia euclídea. En la Figura 4.5 se muestra el dendrograma resultante:

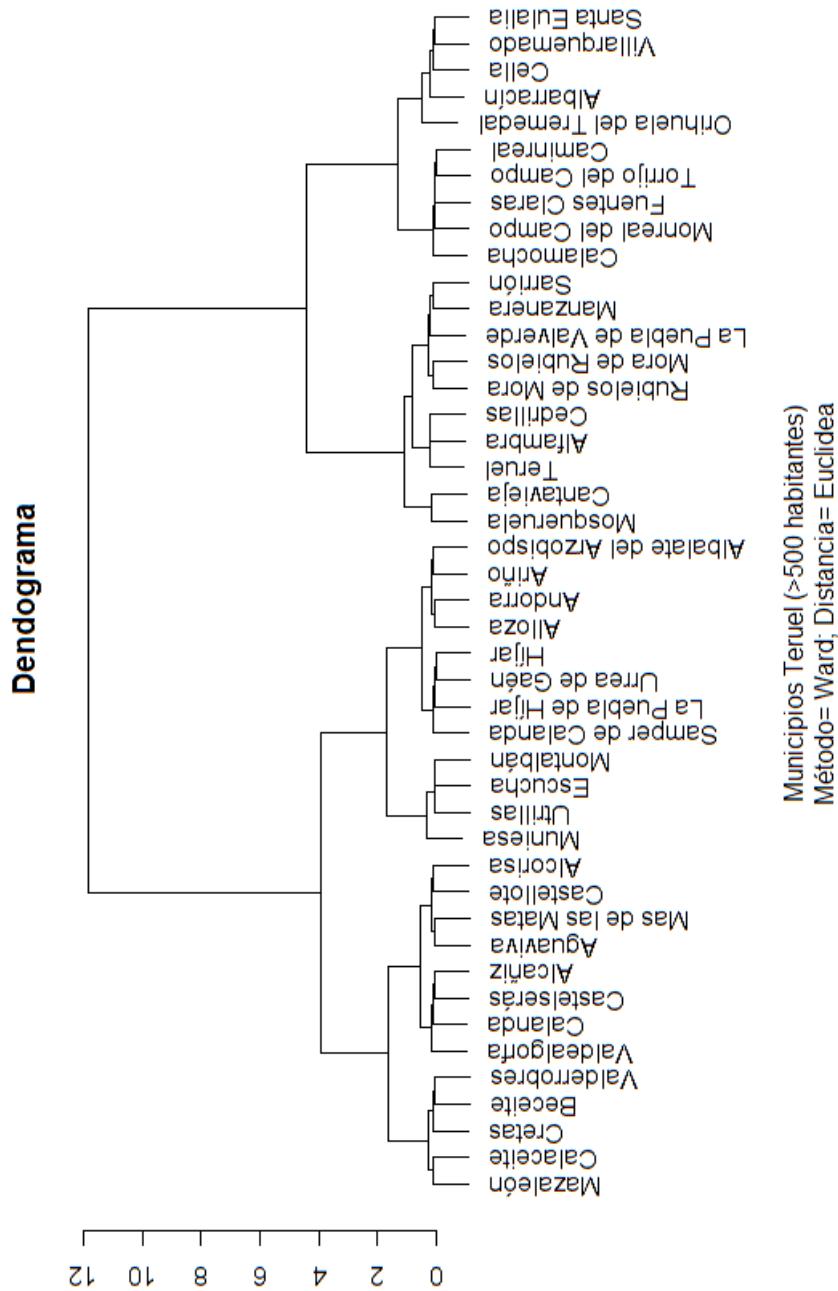


Figura 4.5: Dendrograma de la provincia de Teruel

Como ya se ha indicado solicitamos 5 grupos ( $k = 5$ ) y del mismo modo que en los análisis anteriores guardamos en el conjunto de datos el clúster de pertenencia de cada municipio.

Los 5 grupos son:

- Clúster 1 = {Torrijo del Campo, Orihuela del Tremedal, Fuentes Claras, Caminreal, Villarquemado, Albarracín, Santa Eulalia, Monreal del Campo, Cella, Calamocha}.
- Clúster 2 = {Urrea de Gaén, Alloza, Samper de Calanda, Ariño, La Puebla de Híjar, Híjar, Albalate del Arzobispo, Andorra}.
- Clúster 3 = {La Puebla de Valverde, Manzanera, Mosqueruela, Alfambra, Cedrillas, Cantavieja, Rubielos de Mora, Sarrión, Mora de Rubielos, Teruel}.
- Clúster 4 = {Mazaleón, Cretas, Beceite, Valdealgorfa, Aguaviva, Castellote, Castelserás, Calaceite, Mas de las Matas, Valderrobres, Alcorisa, Calanda, Alcañiz}.
- Clúster 5 = {Muniesa, Escucha, Montalbán, Utrillas}.

En la Figura 4.6, obtenida con Power Bi Desktop, se muestra la localización de los distintos municipios de más de 500 habitantes de la provincia de Teruel. Los distintos colores determinan la pertenencia a los distintos grupos.

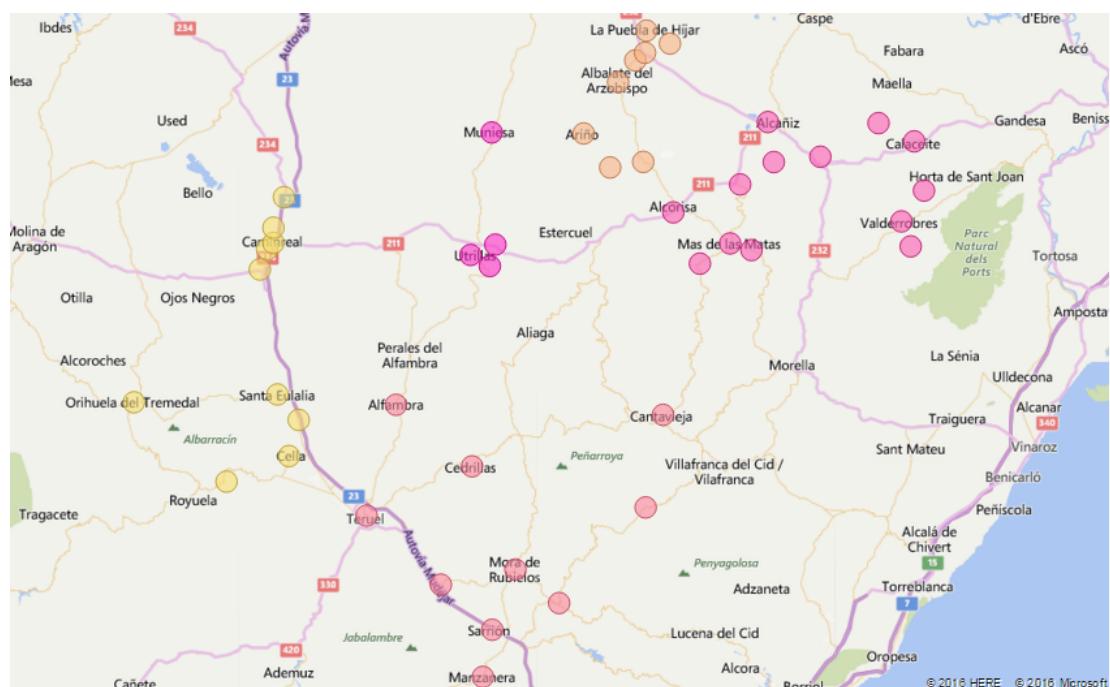


Figura 4.6: Agrupamientos en Teruel.

#### 4.4.2. Localización de las ET

Una vez hemos determinado los grupos para cada una de las provincias hemos de seleccionar en cada uno de los grupos una localización para la estación de transferencia. Como ya se ha indicado, el criterio elegido es seleccionar el municipio con mayor población.

Así, la distribución de los municipios seleccionados como ET para cada una de las provincias es:

- Zaragoza = {Calatayud, Zaragoza, Caspe, Ejea de los Caballeros, Épila}.
- Huesca = {Sariñena, Sabiñánigo, Almudévar, Huesca, Jaca, Fraga, Barbastro, Aínsa-Sobrarbe, Tamarite de Litera}.
- Teruel = {Calamocha, Andorra, Teruel, Alcañiz, Utrillas}.

El siguiente paso es seleccionar entre dichas estaciones de transferencia las localizaciones para abrir los vertederos.

#### 4.4.3. Localización de los vertederos

Una vez hemos localizado las diecinueve estaciones de transferencia, hemos de seleccionar un total de ocho para localización de vertedero (tres de ellas en la provincia de Zaragoza, otras tres en la provincia de Huesca y dos en la provincia de Teruel). Para ello, necesitamos recalcular las demandas (toneladas de RSU producidas) de las posibles localizaciones o clústers, que vienen dadas por la suma de las demandas de cada uno de los municipios del clúster.

En la Tabla 4.4 se muestran los resultados:

Provincia	Estación de Transferencia	Demanda (Tn/año)
Zaragoza	Calatayud	18220.93
	Zaragoza	406337.80
	Caspe	13527.77
	Ejea de los Caballeros	35705.17
	Épila	12377.92
Huesca	Sariñena	4032.02
	Sabiñánigo	7388.82
	Almudévar	4330.55
	Huesca	38989.04
	Jaca	7595.14
	Fraga	12790.03
	Barbastro	17007.52
	Aínsa-Sobrarbe	3520.92
Teruel	Tamarite de Litera	5839.89
	Calamocha	8170.84
	Andorra	8391.74
	Teruel	22453.53
	Alcañiz	17425.88
	Utrillas	3248.96

Tabla 4.4: Localización de las ET en Aragón

Para localizar los vertederos procedemos de forma análoga a lo realizado en la Sección 4.3.1 y aplicaremos el problema de localización de la p-mediana. Para su resolución utilizaremos CPLEX ®.

### Resolución en CPLEX para la provincia de Zaragoza

Mantenemos el número de vertederos abiertos en la provincia de Zaragoza, de modo que fijamos  $p = 3$ .

CPLEX nos ofrece la siguiente solución:

- Se abren vertederos en Calatayud, en Zaragoza y en Ejea de los Caballeros.

Las asignaciones del resto de los municipios a los tres vertederos son:

- Calatayud = {Calatayud}.
- Zaragoza = {Zaragoza, Caspe, Épila}.
- Ejea de los Caballeros = {Ejea de los Caballeros}.

### Resolución en CPLEX para la provincia de Huesca

Análogamente, mantenemos el número de vertederos abiertos en la provincia de Huesca, de modo que fijamos  $p = 3$ .

CPLEX nos ofrece la siguiente solución:

- Se abren vertederos en los municipios de Huesca, Fraga y Barbastro.

Las asignaciones del resto de los municipios a los tres vertederos son:

- Huesca = {Sabiñánigo, Almudévar, Huesca, Jaca, Aínsa-Sobrarbe}.
- Fraga = {Fraga}.
- Barbastro = {Sariñena, Tamarite de Litera}.

### Resolución en CPLEX para la provincia de Teruel

De nuevo fijamos el número de vertederos abiertos en la provincia de Teruel como el actual,  $p = 2$ .

CPLEX nos ofrece la siguiente solución:

- Se abren vertederos en Teruel y Alcañiz.

Las asignaciones del resto de los municipios a los vertederos son:

- Teruel = {Calamocha, Teruel, Utrillas}.
- Alcañiz = {Andorra, Alcañiz}.

En la Figura 4.7 se muestra la situación resultante de resolver los distintos problemas de localización de vertederos y estaciones de transferencia:

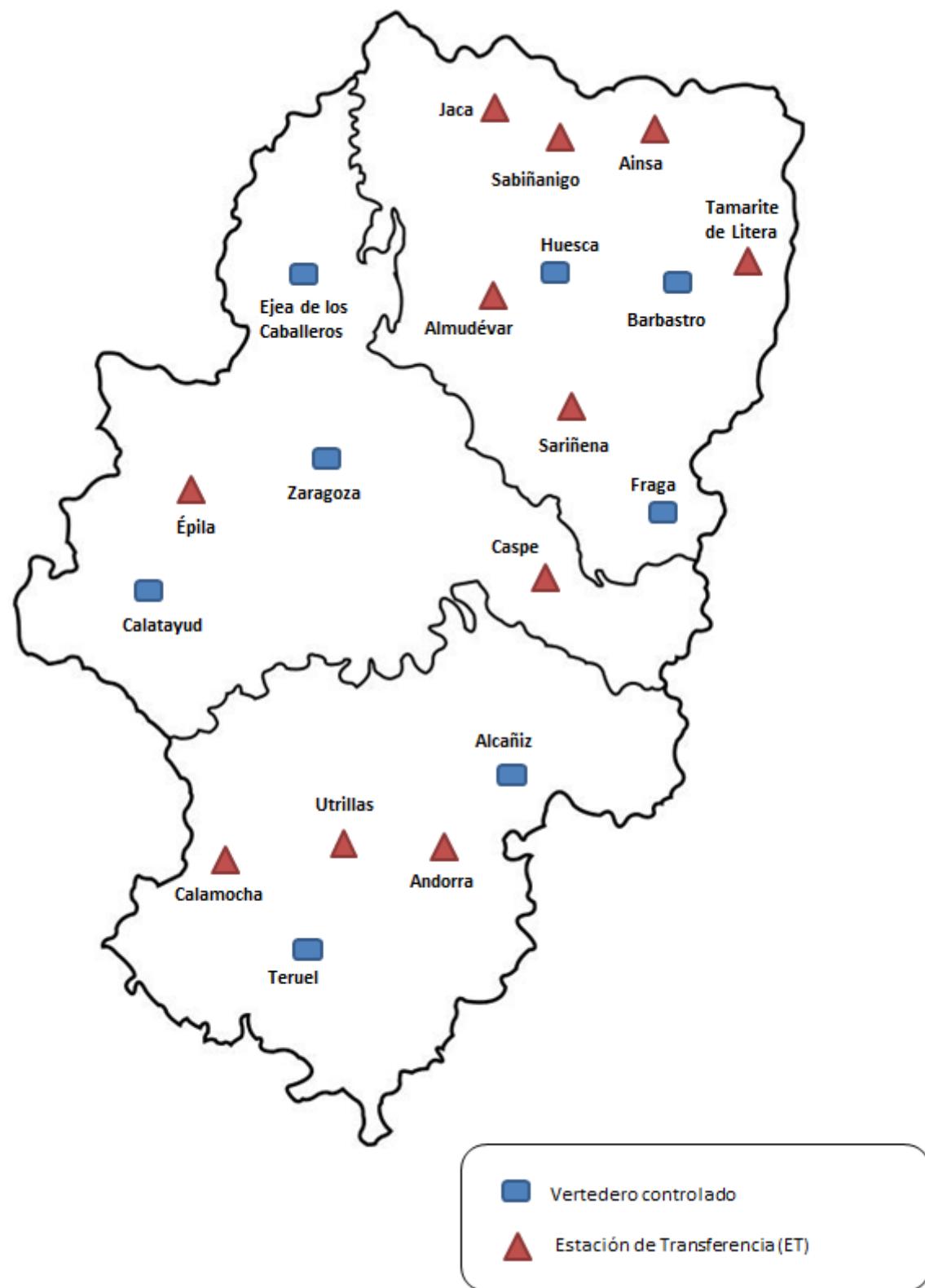


Figura 4.7: Localización de los vertederos y de las ET en Aragón

## 4.5. Modelo de rutas

En esta última sección, introduciremos la formulación de un problema de rutas y calcularemos con la ayuda de un *solver* las rutas de los vehículos para las localizaciones obtenidas tras aplicar el problema de la p-mediana en la Sección 4.3.1. La referencia principal de la sección es el libro de P. Toth y D. Vigo [31, Parte 1].

### 4.5.1. Problema de Rutas de Vehículos (VRP)

El problema de rutas de vehículos (VRP por sus siglas en inglés) consiste en el diseño de rutas de  $m$  vehículos cuyo objetivo es obtener el menor costo total de tal manera que:

- Cada ruta comienza y termina en un depósito.
- Cada cliente está dentro de exactamente una ruta.

Del mismo modo que en el problema de la p-mediana, existe una variante del problema VRP, conocida como CVRP (Capacitated Vehicle Routing Problem) en el que se tienen en cuenta la capacidad de los vehículos utilizados para realizar las rutas. A las condiciones anteriores se ha de añadir que la demanda total de un recorrido no supere la capacidad del vehículo que la realiza. Introduciremos este problema para el caso de tener una flota homogénea (vehículos con la misma capacidad).

El CVRP se define sobre un grafo  $G = (V, A)$ . El conjunto  $V$  denota el conjunto de vértices o clientes,  $V = \{1, \dots, n\}$  con un vértice distinguido, el depósito, situado en el lugar 1. El conjunto  $A$  denota el conjunto de arcos o carreteras que unen los distintos vértices. Para formular el problema CVRP necesitamos de la siguiente notación:

- $c_{ij}$ : coste asociado a recorrer el arco desde  $i$  hasta  $j$ .
- $d_i$ : demanda del cliente  $i$ ,  $i \in V$ .
- $M = \{1, \dots, m\}$ : flota de vehículos.
- $K$ : capacidad de cada vehículo (flota homogénea).

Además, para cada subconjunto de clientes  $S \subseteq V$  denotamos por  $d(S)$  la demanda total del conjunto  $S$ , esto es,

$$d(S) = \sum_{i \in S} d_i.$$

Cabe destacar que los arcos  $(i, i)$  con  $i \in V$  no son posibles de atravesar, por lo que se le impone un coste suficientemente grande. Además, al depósito se le asigna una demanda ficticia  $d_1 = 0$ . Para que el problema sea factible se ha de cumplir que  $d_i \leq K$  para cada  $i \in V$  y que el número de vehículos sea capaz de cubrir la demanda, es decir,

$$|M| \geq \left\lceil \frac{d(V)}{K} \right\rceil.$$

Definimos las siguientes variables:

$$x_{ijk} = \begin{cases} 1 & \text{si el arco } (i, j) \text{ es recorrido por el vehículo } k \\ 0 & \text{en otro caso} \end{cases}, \quad 1 \leq k \leq m,$$

$$y_{ik} = \begin{cases} 1 & \text{si el cliente } i \text{ es visitado por el vehículo } k \\ 0 & \text{en otro caso} \end{cases}, \quad 1 \leq i \leq n, 1 \leq k \leq m.$$

A continuación se refleja una formulación para el CVRP:

$$\min_{x_{ijk}, y_{ik}} \sum_{i \in V} \sum_{j \in V} \sum_{k \in M} c_{ij} x_{ijk} \quad (4.2a)$$

sujeto a

$$\sum_{k \in M} y_{ik} = 1, \quad \forall i \in V \setminus \{1\} \quad (4.2b)$$

$$\sum_{k \in M} y_{1k} = m \quad (4.2c)$$

$$\sum_{j \in V} x_{ijk} = y_{ik}, \quad \forall i \in V, k \in M \quad (4.2d)$$

$$\sum_{j \in V} x_{jik} = y_{ik}, \quad \forall i \in V, k \in M \quad (4.2e)$$

$$\sum_{i \in V} d_{ij} y_{ik} \leq K, \quad \forall k \in M \quad (4.2f)$$

$$\sum_{i \in S} \sum_{j \notin S} x_{ijk} \geq y_{hk}, \quad \forall S \subseteq V \setminus \{1\}, h \in S, k \in M \quad (4.2g)$$

$$x_{ijk} \in \{0, 1\}, \quad \forall i, j \in V, \forall k \in M \quad (4.2h)$$

$$y_{ik} \in \{0, 1\}, \quad \forall i \in J, \forall k \in M \quad (4.2i)$$

El objetivo de nuestro problema (4.2a) es minimizar el coste total de las rutas realizadas por los vehículos.

El grupo de restricciones (4.2b) indican que, a excepción del nodo depósito, cada cliente o vértice debe ser visitado una única vez y de cada cliente sale un único vehículo. También se han de imponer unas condiciones de flujo para el nodo depósito que reflejen que del nodo depósito salen  $m$  vehículos (4.2c). Las restricciones (4.2d)-(4.2e) obligan a que el mismo número de vehículos entren y salgan de cada cliente y del nodo depósito. Las restricciones pertenecientes a estos grupos son conocidas como restricciones de flujo en los nodos. Las restricciones de capacidad vienen dadas en (4.2f), en las que se impone para cada vehículo que la suma de las demandas de los clientes atendidos no sea mayor que la capacidad  $K$  de cada uno de los vehículos. Otra de las restricciones que se han de tener en cuenta son las de eliminación de *subtours*. Estas restricciones aparecen reflejadas en (4.2g). Finalmente en las restricciones (4.2h) y (4.2i) se indica la naturaleza binaria de las variables.

Las restricciones para la eliminación de subtours pueden reformularse de manera que el cardinal sea de orden polinómico. Para ello se ha de introducir una nueva variable  $u_i$  que represente la carga del vehículo  $k$  tras visitar el nodo  $i$ . Así, se puede sustituir (4.2g) por los siguientes grupos de restricciones:

$$u_{ik} - u_{jk} + Cx_{ijk} \leq C - d_j, \quad \forall i, j \in V \setminus \{1\}, i \neq j, k \in M$$

$$d_i \leq u_{ik} \leq C, \quad \forall i \in V \setminus \{1\}, k \in M.$$

Notemos que, cuando  $x_{ijk} = 0$ , el primer grupo de restricciones anterior no restringe, pues  $u_{ik} \leq C$  y  $u_{jk} \geq d_j$  y cuando  $x_{ijk} = 1$  imponen que  $u_{jk} \geq u_{ik} + d_j$ .

#### 4.5.2. Resolución

Para la resolución del problema VRP existen diferentes programas de Software libre a los que se puede acudir. A continuación se muestra una tabla con algunos de ellos.

Tabla 4.5: Fuente: Wikipedia

Nombre	Lenguaje	Información
jsprit	Java	Basado en Java. Está disponible un kit de herramientas de código abierto. Interfaz de usuario compatible con Excel (para mapeo, presentación de informes y edición de rutas).
Open-VRP	Lisp	Open-VRP para Lisp. Alojado en Github
OptaPlanner	Java	Código abierto en JaCoP (Java Constraint solver). Contiene ejemplos.
SYMPHONY	-	Solver de código abierto para la programación lineal entera mixta (MILPs) con soporte para VRP.
VRP Spreadsheet Solver	-	Microsoft Excel. Enlazado con un sistema de información geográfica pública para calcular distancias y duraciones de rutas.

Utilizaremos el último de ellos, disponible en la página web de [VeRoLog](#), el grupo de trabajo en rutas de vehículos y optimización logística dentro de la Asociación de Sociedades Europeas de Investigación Operativa (EURO).

Antes de pasar a dar una solución para el problema de rutas tenemos que determinar el número de vehículos que realizan las distintas rutas y sus capacidades.

Existen diferentes tipos de camiones en cuanto a dimensiones y capacidad para la recogida y transporte integrado de residuos dependiendo del momento del ciclo de recogida que cubran (si se trata de la recogida o del transporte hasta el destino final, vertedero o Estación de Transferencia) o de la zona urbana en la que se muevan (puede ser que el casco antiguo de las ciudades o determinadas zonas rurales requieran de un tipo de vehículo con dimensiones

reducidas). Otro tipo de vehículos es capaz de dar respuesta al mismo tiempo a estos momentos del ciclo de gestión de residuos: la recogida y el transporte integrado hasta el destino final. En nuestro caso el tipo de vehículos considerado será de este último tipo. Gracias a la potencia de prensado, un vehículo con compactador puede reducir el volumen de un residuo hasta una proporción de 5 a 1, convirtiendo 100 metros cúbicos de residuo en tan sólo 20 metros cúbicos, de esta manera, se permite el transporte de una gran cantidad de RSU.

La recogida de residuos urbanos en toda la ciudad de Zaragoza se lleva a cabo durante todos los días del año, incluidos domingos y festivos, a excepción de los días de Navidad y Año Nuevo (información extraída de la página web del Ayuntamiento de Zaragoza \*\*\*).

Una vez realizado este estudio, para tratar una situación realista en la Comunidad de Aragón, podríamos considerar que nuestros camiones tienen una capacidad de 16 toneladas cada uno (flota homogénea) y que la recogida de basuras es diaria. De este modo, las demandas de cada municipio (toneladas producidas que han de ser recogidas por los vehículos) serían las ofrecidas en las Tablas A.1 para la provincia de Zaragoza, A.2 para la provincia de Huesca y A.3 en el caso de la provincia de Teruel, divididas por 365 que son los días que tiene un año.

El inconveniente con el que nos encontramos es que algunas de las demandas de los municipios exceden la capacidad del propio vehículo, por lo que el problema aplicado con la formulación dada en (4.2) no es factible con las capacidades estándar para cada vehículo. Esta situación no es de extrañar puesto que, por ejemplo, en la provincia de Zaragoza, la recogida de basuras nocturna que se realiza en la ciudad se hace por zonas, utilizándose varios vehículos. Por ello, para ser capaces de resolver el problema, supondremos que cada ruta no es realizada por un único vehículo sino que van varios (en bloque) de manera que contamos con vehículos de mayor capacidad.

Se han resuelto tres problemas por separado, uno para cada una de las provincias. Cada problema consta de cuatro fases:

1. En primer lugar se ha de determinar la dirección de cada una de los vertederos y de los distintos municipios. Basta con darlas en el siguiente formato: “Municipio, Provincia”.
2. En una segunda fase se calcula la longitud y la latitud de cada una de las localizaciones.
3. Una vez obtenidas las coordenadas en el paso anterior, se calcula la distancia entre cada par de municipios.
4. En la última fase se resuelve el problema VRP bajo las condiciones impuestas.

#### 4.5.3. Resultados

##### Zaragoza

Resolveremos el problema de rutas de vehículos sobre las localizaciones fijadas tras la resolución del problema de la p-mediana en la Sección 4.3.1. Recordemos la localización de los 3 vertederos y la asociación del resto de municipios:

---

\*\*\*<http://www.zaragoza.es/sedelectronica/>

- Zaragoza = {Lumpiaque, Nonaspe, Sástago, Escatrón, Magallón, Fabara, Gelsa, Morata de Jalón, Muel, Pastriz, Alfamén, Torres de Berrellén, Belchite, Brea de Aragón, Ateca, Alfajarín, Daroca, El Burgo de Ebro, Mequinenza, Villamayor de Gállego, San Mateo de Gállego, Cadrete, Pinseque, Villanueva de Gállego, María de Huerva, Fuentes de Ebro, Épila, La Muela, La Puebla de Alfindén, Alagón, Zuera, Cuarte de Huerva, La Almunia de Doña Godina, Caspe, Ejea de los Caballeros, Utebo, Zaragoza}.
- Calatayud = {Maluenda, Alhama de Aragón, Remolinos, Figueruelas, Sádaba, Maella, Quinto, Pina de Ebro, Calatorao, Illueca, Ricla, Calatayud}.
- Gallur = {Biota, Bujaraloz, Luceni, Ainzón, Ariza, Leciñena, Calatorao, Gallur, Cariñena, Pedrola, Mallén, Borja, Tauste, Tarazona}.

### 1. Vertedero de Zaragoza

Para determinar el número de rutas y la capacidad de éstas hemos de conocer la demanda total diaria de los municipios atendidos por el vertedero situado en el municipio de Zaragoza, así como el máximo de demanda diaria entre los municipios.

- Obtenemos una demanda diaria de 207,9 toneladas.
- El máximo se alcanza en el municipio de Utebo, cuya demanda diaria es de 25,3 toneladas.

Así, si tenemos camiones de 16 toneladas de capacidad, tendremos que considerar que cada ruta es realizada como mínimo por dos de estos camiones, resultando así rutas de 32 toneladas de capacidad. Para gestionar la totalidad demandada de RSU consideraremos que se realizan un total de 7 rutas.

Para identificar las poblaciones se ha utilizado el siguiente código. El resultado obtenido puede verse en la Figura 4.8.

0. Zaragoza	11. Alfamén	Gállego	29. La Puebla de Alfindén
1. Lumpiaque	12. Torres de Berrellén	21. San Mateo de Gállego	30. Alagón
2. Nonaspe	13. Belchite	22. Cadrete	31. Zuera
3. Sástago	14. Brea de Aragón	23. Pinseque	32. Cuarte de Huerva
4. Escatrón	15. Ateca	24. Villanueva de Gállego	33. La Almunia de Doña Godina
5. Magallón	16. Alfajarín	25. María de Huerva	34. Caspe
6. Fabara	17. Daroca	26. Fuentes de Ebro	35. Ejea de los Caballeros
7. Gelsa	18. El Burgo de Ebro	27. Épila	36. Utebo
8. Morata de Jalón	19. Mequinenza	28. La Muela	
9. Muel	20. Villamayor de Gállego		
10. Pastriz			



Figura 4.8: Rutas de recogida de residuos en Zaragoza

A continuación seguimos el mismo esquema para mostrar los resultados para el resto de vertederos.

## 2. Vertedero de Calatayud

Para el caso del vertedero situado en el municipio de Calatayud obtenemos lo siguiente:

- Demanda diaria: 32,82 toneladas.
- Máximo: Ricla (5 toneladas).

En este caso, la cantidad de RSU generados por cada uno de los municipios atendidos es inferior a la capacidad de los vehículos. Por lo tanto, únicamente hemos de asegurarnos de poder gestionar la totalidad demandada de RSU. Se realizan un total de 3 rutas de 16 toneladas de capacidad cada una.

El código utilizado para identificar las poblaciones se muestra a continuación. El resultado obtenido puede verse en la Figura 4.9.

- |                     |                |                 |
|---------------------|----------------|-----------------|
| 0. Calatayud        | 4. Figueruelas | 8. Pina de Ebro |
| 1. Maluenda         | 5. Sádaba      | 9. Calatorao    |
| 2. Alhama de Aragón | 6. Maella      | 10. Illueca     |
| 3. Remolinos        | 7. Quinto      | 11. Ricla       |



Figura 4.9: Rutas de recogida de residuos en Calatayud

### 3. Vertedero de Gallur

Para el caso del vertedero situado en el municipio de Gallur, se tiene:

- Demanda diaria: 59,44 toneladas.
- Máximo: Tarazona (16 toneladas).

Dado que el municipio de Tarazona tiene una demanda diaria que coincide con la capacidad de los vehículos, en la solución final se tendrá que una ruta se realizará únicamente para dar servicio a este municipio. Consideramos 4 rutas para la totalidad de los municipios.

A continuación se muestra el código utilizado para identificar los municipios. El resultado obtenido puede verse en la Figura 4.10.

- |              |              |
|--------------|--------------|
| 0. Gallur    | 7. Cariñena  |
| 1. Biota     | 8. Pedrola   |
| 2. Bujaraloz | 9. Mallén    |
| 3. Luceni    | 10. Borja    |
| 4. Ainzón    | 11. Tauste   |
| 5. Ariza     | 12. Tarazona |
| 6. Leciñena  |              |

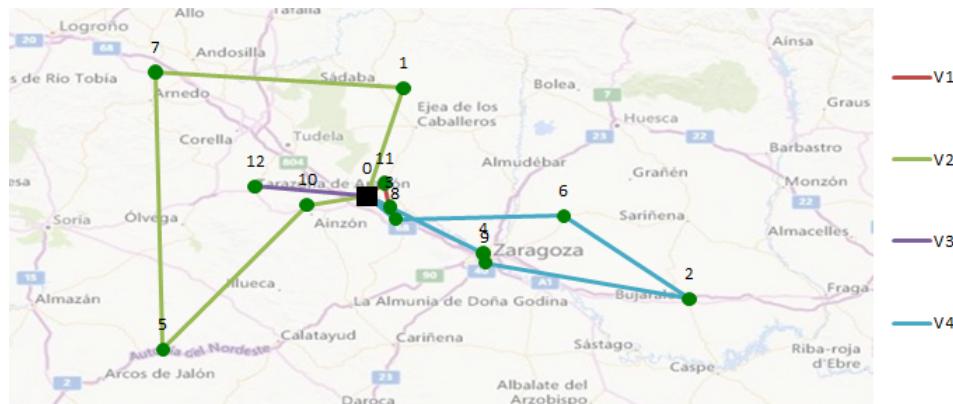


Figura 4.10: Rutas de recogida de residuos en Gallur

En la Tabla 4.6 se muestran los tiempos de ejecución en segundos de cada una de las fases del programa.

Tabla 4.6: Tiempos de ejecución (en segundos)

Provincia	Vertedero	Cálculo lon/lat	Cálculo distancias	Resolución VRP
Zaragoza	Zaragoza	59,2	140,6	60
	Calatayud	19,2	15,6	60
	Gallur	22,4	21	60
	Total	100,8	177,1	180

## Huesca

Igual que hemos hecho sobre la provincia de Zaragoza, resolveremos el problema de rutas de vehículos sobre las localizaciones fijadas tras la resolución del problema de la p-mediana en la Sección 4.3.1. Recordemos la localización de los 3 vertederos y la asociación del resto de municipios:

- Huesca = {Tardienta, La Sotonera, Lalueza, Ayerbe, Lanaja, Gurrea de Gállego, Grañén, Almudévar, Sariñena, Monzón, Huesca}.
- Sabiñánigo = {Boltaña, Sallent de Gállego, Biescas, Benasque, Aínsa-Sobrarbe, Sabiñánigo, Jaca}.
- Binaced = {Alcolea de Cinca, Benabarre, Albalate de Cinca, Torrente de Cinca, Belver de Cinca, Altorricon, Binaced, Zaidín, Graus, Tamarite de Litera, Binéfar, Fraga, Barbastro}.

### 1. Vertedero de Huesca

En la provincia de Huesca se tiene:

- Demanda diaria: 47 toneladas.
- Máximo: Monzón (24,52 toneladas).

Para dar servicio a Monzón consideramos que cada ruta es realizada por dos de estos camiones, resultando así rutas de 32 toneladas de capacidad. Se realizan un total de 2 rutas.

El código utilizado para identificar las poblaciones se muestra a continuación. El resultado obtenido puede verse en la Figura 4.11.

- |                |                      |
|----------------|----------------------|
| 0. Huesca      | 6. Gurrea de Gállego |
| 1. Tardienta   | 7. Grañén            |
| 2. La Sotonera | 8. Almudévar         |
| 3. Lalueza     | 9. Sariñena          |
| 4. Ayerbe      | 10. Monzón           |
| 5. Lanaja      |                      |



Figura 4.11: Rutas de recogida de residuos en Huesca

## 2. Vertedero de Sabiñánigo

En la provincia de Sabiñánigo se tiene:

- Demanda diaria: 30,75 toneladas.
- Máximo: Jaca (18,73 toneladas).

Para gestionar la totalidad de demanda de RSU se realiza una única ruta de 32 toneladas de capacidad.

El código utilizado para identificar las poblaciones se muestra a continuación. El resultado obtenido puede verse en la Figura 4.12.

- |                       |                   |
|-----------------------|-------------------|
| 0. Sabiñánigo         | 4. Benasque       |
| 1. Boltaña            | 5. Aínsa-Sobrarbe |
| 2. Sallent de Gállego | 6. Jaca           |
| 3. Biescas            |                   |

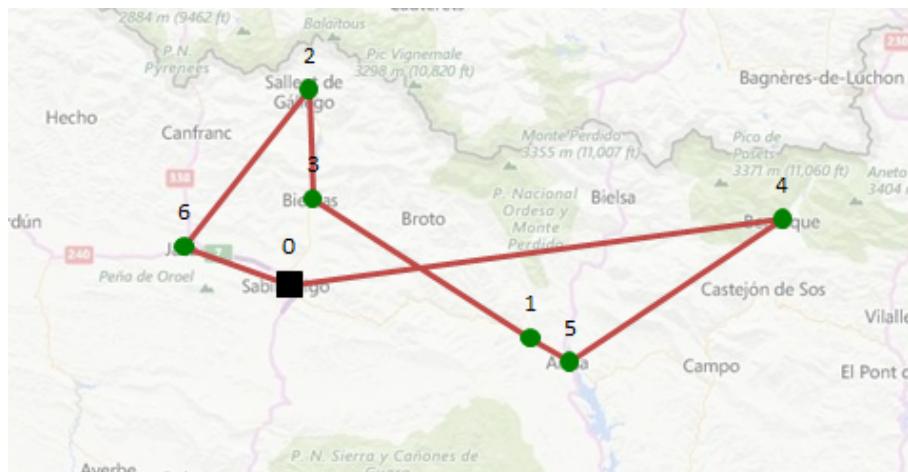


Figura 4.12: Ruta de recogida de residuos en Sabiñánigo

## 3. Vertedero de Binaced

En la provincia de Binaced se tiene:

- Demanda diaria: 84,82 toneladas.
- Máximo: Barbastro (24,43 toneladas).

Para gestionar la totalidad de la demanda de RSU se realizan 3 rutas de 32 toneladas cada una.

El código utilizado para identificar las poblaciones se muestra a continuación. El resultado obtenido puede verse en la Figura 4.13.

- |                     |                      |              |
|---------------------|----------------------|--------------|
| 0. Binaced          | 3. Albalate de Cinca | 6. Alturrión |
| 1. Alcolea de Cinca | 4. Torrente de Cinca | 7. Zaidín    |
| 2. Benabarre        | 5. Belver de Cinca   | 8. Graus     |

- |                       |               |
|-----------------------|---------------|
| 9. Tamarite de Litera | 11. Fraga     |
| 10. Binéfar           | 12. Barbastro |

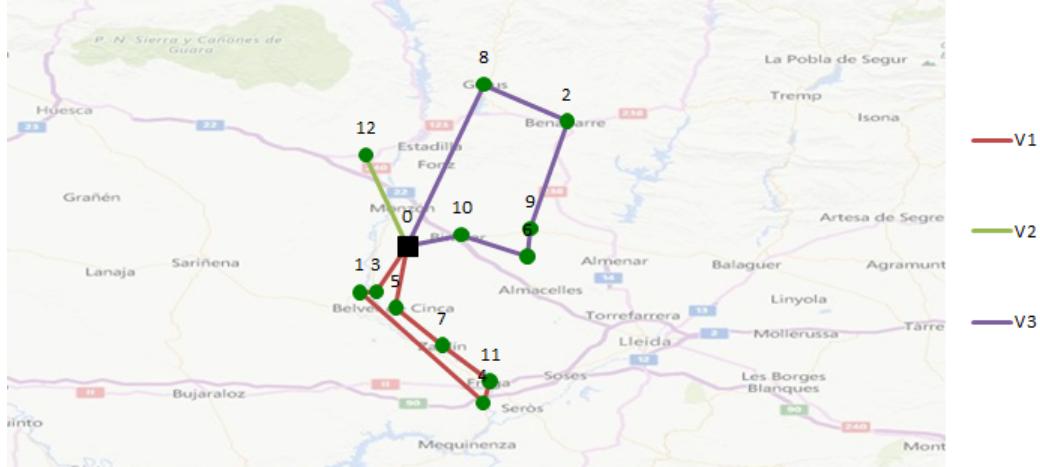


Figura 4.13: Rutas de recogida de residuos en Binaced

En la Tabla 4.7 se muestran los tiempos de ejecución en segundos de cada una de las fases del programa.

Tabla 4.7: Tiempos de ejecución (en segundos)

Provincia	Vertedero	Cálculo lon/lat	Cálculo distancias	Resolución VRP
Huesca	Sabiñánigo	11,2	5,6	60
	Grañén	17,6	13,2	60
	Binaced	20,8	18,2	60
	Total	49,6	37	180

## Teruel

Finalmente, procedemos de la misma manera para la provincia de Teruel. Resolveremos el problema de rutas de vehículos sobre las localizaciones fijadas tras la resolución del problema de la p-mediana en la Sección 4.3.1. Recordemos la localización de los 2 vertederos y la asociación del resto de municipios:

Las asignaciones del resto de los municipios a los vertederos son:

- Teruel = {Albarracín, Santa Eulalia, Sarrión, Mora de Rubielos, Monreal del Campo, Cella, Utrillas, Calamocha, Teruel}.
- Alcañiz = {Calaceite, Montalbán, Mas de las Matas, Híjar, Albalate del Arzobispo, Valderrobres, Alcorisa, Calanda, Andorra, Alcañiz}.

### 1. Vertedero de Teruel

La situación en el municipio de Teruel es la siguiente:

- Demanda diaria: 26,3 toneladas.

- Máximo: Calamocha (6,54 toneladas).

En este caso, la cantidad de RSU generados por cada uno de los municipios atendidos es inferior a la capacidad de los vehículos. Por lo tanto, únicamente hemos de asegurarnos de poder gestionar la totalidad demandada de RSU. Para ellos consideraremos que se realizan un total de 2 rutas de 16 toneladas de capacidad cada una.

El código utilizado para identificar las poblaciones se muestra a continuación. El resultado obtenido puede verse en la Figura 4.14.

- |                     |                      |
|---------------------|----------------------|
| 0. Teruel           | 5. Monreal del Campo |
| 1. Albarracín       | 6. Cella             |
| 2. Santa Eulalia    | 7. Utrillas          |
| 3. Sarrión          | 8. Calamocha         |
| 4. Mora de Rubielos |                      |

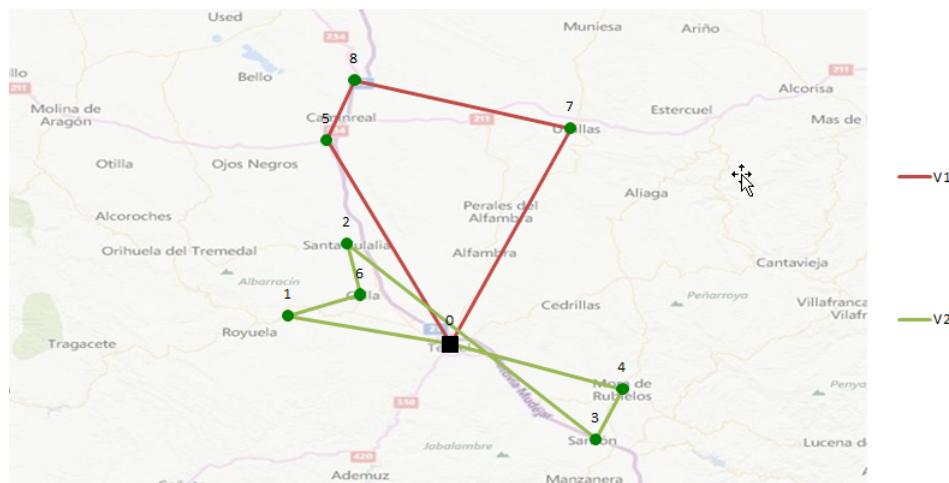


Figura 4.14: Rutas de recogida de residuos en Teruel

## 2. Vertedero de Alcañiz

Para el municipio de Alcañiz se tiene:

- Demanda diaria: 36,86 toneladas.
- Máximo: Andorra (11.8 toneladas).

Consideraremos que se realizan un total de 3 rutas de 16 toneladas de capacidad cada una.

Para identificar las poblaciones se ha utilizado el siguiente código. El resultado obtenido puede verse en la Figura 4.15.

- |                     |                           |
|---------------------|---------------------------|
| 0. Alcañiz          | 5. Albalate del Arzobispo |
| 1. Calaceite        | 6. Valderrobres           |
| 2. Montalbán        | 7. Alcorisa               |
| 3. Mas de las Matas | 8. Calanda                |
| 4. Híjar            | 9. Andorra                |

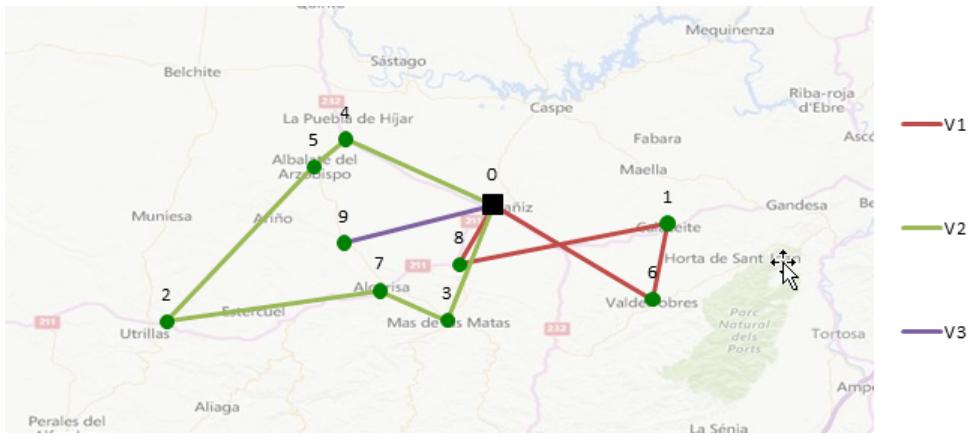


Figura 4.15: Rutas de recogida de residuos en Alcañiz

En la Tabla 4.8 se muestran los tiempos de ejecución en segundos de cada una de las fases del programa.

Tabla 4.8: Tiempos de ejecución (en segundos)

Provincia	Vertedero	Cálculo lon/lat	Cálculo distancias	Resolución VRP
Teruel	Teruel	14,4	9	60
	Alcañiz	16	11	60
	Total	30,4	20	120

## **Anexos**





## Datos

### A.1. Datos de población de la provincia de Zaragoza

La Tabla A.1 muestra los datos de población de los municipios de la provincia de Zaragoza de más de 500 habitantes (ordenados por tamaño de población).

Tabla A.1: Municipios con más de 500 habitantes (Zaragoza)

Municipio	Extensión (km <sup>2</sup> )	Población	(*) RSU (t/año)	(**) Capacidad
Ibdes	56,31	510	265,71	29845,951
Botorrita	19,8	516	268,836	29498,8636
Terrer	33,8	526	274,046	29241,7446
La Zaida	17,36	534	278,214	29548,495
Cabañas de Ebro	8,55	540	281,34	29713,297
Jarque	43	546	284,466	29076,0586
Grisén	4,75	549	286,029	29785,9459
Paracuellos de Jiloca	32,22	554	288,634	29277,8726
Alpartir	26,89	567	295,407	29379,8641
Villarroya de la Sierra	91,52	585	304,785	28186,0177
Villanueva de Huerva	78,35	588	306,348	28430,9218
La Almolda	131,3	608	316,768	27454,1108
Herrera de los Navarros	105,01	630	328,23	27946,8976
Pradilla de Ebro	25,22	638	332,398	29428,157
Novillas	25,27	639	332,919	29427,4741
Azuara	165,8	643	335,003	26823,0243
Perdiguera	110,09	662	344,902	27860,5256
Sos del Rey Católico	216,62	690	359,49	25892,2942

Cetina	80,6	717	373,557	28420,6957
Lécera	109,2	732	381,372	27894,1132
Aguarón	36,62	743	387,103	29242,4315
Almonacid de la Sierra	54,1	744	387,624	28918,5964
Nuez de Ebro	8,2	758	394,918	29772,9998
Paniza	47,34	770	401,17	29050,2734
Villafranca de Ebro	63,6	777	404,817	28750,5217
Aniñón	52,56	789	411,069	28958,1325
Uncastillo	230,6	801	417,321	25660,2001
Sabiñán	15,52	805	419,405	29648,7597
Luna	308,92	861	448,581	24222,7933
Novallas	11,3	891	464,211	29747,9911
Longares	45,96	896	466,816	29106,6152
La Joyosa	6,52	902	469,942	29839,2974
Boquiñeni	19,09	923	480,883	29611,3757
Sobradiel	11,98	967	503,807	29753,9355
Fuendejalón	75,83	969	504,849	28570,6447
Biota	128,8	1036	539,756	27604,9356
Bujaraloz	121,61	1044	543,924	27740,191
Lumpiaque	29,55	1050	547,05	29448,448
Nonaspe	111,4	1083	564,243	27939,0043
Maluenda	40,09	1100	573,1	29265,2414
Luceni	27,08	1101	573,621	29506,6909
Alhama de Aragón	31,11	1106	576,226	29433,1952
Sástago	301,25	1115	580,915	24426,9965
Escatrón	94,64	1163	605,923	28269,2627
Ainzón	40,46	1187	618,427	29279,6183
Remolinos	18,48	1192	621,032	29688,348
Ariza	103,1	1207	628,847	28123,1547
Magallón	78,61	1223	637,183	28581,1049
Fabara	101,63	1225	638,225	28154,8023
Gelsa	72,04	1226	638,746	28703,645
Leciñena	178,59	1299	676,779	26746,0273
Figueruelas	16,99	1351	703,871	29754,7845
Morata de Jalón	45,87	1360	708,56	29221,5462
Muel	79,17	1388	723,148	28610,999
Pastriz	16,21	1392	725,232	29779,2538
Alfamén	102,25	1517	790,357	28214,5847
Torres de Berrellén	53,8	1523	793,483	29114,3123
Belchite	273,7	1633	850,793	25064,2173
Sádaba	129,5	1666	867,986	27745,7406
Brea de Aragón	13,56	1754	913,834	29916,749

Maella	174,88	2027	1056,067	26992,5155
Ateca	84,7	2061	1073,781	28672,7521
Quinto	118,4	2108	1098,268	28059,4268
Alfajarín	137,57	2235	1164,435	27735,0157
Daroca	52,05	2331	1214,451	29343,9901
El Burgo de Ebro	24,86	2366	1232,686	29856,6362
Mequinenza	307,45	2492	1298,332	24648,1742
Pina de Ebro	308,79	2597	1353,037	24648,9611
Villamayor de Gállego	89,36	2868	1494,228	28783,3444
Calatorao	48,02	2998	1561,958	29581,521
San Mateo de Gállego	35,9	3009	1567,689	29808,9109
Gallur	41,71	3026	1576,546	29705,3432
Cadrete	11	3363	1752,123	30356,9683
Illueca	24,86	3381	1761,501	30104,3977
Pinseque	16,12	3423	1783,383	30276,6895
Ricla	90,67	3469	1807,349	28905,7611
Cariñena	82,5	3537	1842,777	29073,8317
Pedrola	113,53	3667	1910,507	28530,2685
Mallén	37,42	3731	1943,851	29956,9703
Villanueva de Gállego	76	4255	2216,855	29369,6055
María de Huerva	108,09	4444	2315,324	28820,7918
Fuentes de Ebro	141,73	4596	2394,516	28234,2094
Épila	194,32	4691	2444,011	27282,3803
Borja	107	4915	2560,715	28955,9715
La Muela	143,49	4928	2567,488	28282,6202
La Puebla de Alfindén	16,95	5033	2622,193	30654,3023
Alagón	24,22	7175	3738,175	31042,3787
Zuera	333,17	7427	3869,467	25375,9589
Cuarte de Huerva	8,9	7687	4004,927	31451,3907
Tauste	405,23	7710	4016,91	24109,0468
La Almunia de Doña Godina	56,65	7766	4046,086	30585,3896
Caspe	503,33	9858	5136,018	22814,5996
Tarazona	244,01	11211	5840,931	27952,6597
Ejea de los Caballeros	609,92	17331	9029,451	22662,5803
Utebo	17,9	17677	9209,717	33723,0897
Calatayud	154,25	20926	10902,446	31988,2416
Zaragoza	973,7	674317	351319,157	176288,3817

(\*) El número de toneladas de RSU generadas anualmente por cada uno de los municipios se ha estimado con el dato de producción de residuos sólidos urbanos en España en el año 2008 por habitante y año (dato ofrecido por el IAE).

(\*\*) La capacidad del vertedero para cada municipio ha sido estimada (véase 4.3.2).

## A.2. Datos de población de la provincia de Huesca

La Tabla A.2 muestra los datos de población de los municipios de la provincia de Zaragoza de más de 500 habitantes (ordenados por tamaño de población).

Tabla A.2: Municipios con más de 500 habitantes (Huesca)

Municipio	Extensión (km <sup>2</sup> )	Población	(*) RSU (t/año)	(**) Capacidad
Sena	104,7	516	268,836	27924,8176
Broto	128	523	272,483	27494,5443
Almuniente	37,6	528	275,088	29171,7808
Loporzano	169,3	535	278,735	26731,7715
Canfranc	71,6	554	288,634	28547,7674
Nueno	147,2	558	290,718	27147,1198
Ontiñena	137	570	296,97	27339,157
Castejón del Puente	25,4	574	299,054	29409,1974
Robres	64,3	575	299,575	28688,2355
Peralta de Alcofea	116,1	582	303,222	27729,5722
La Fueva	218,8	622	324,062	25835,2782
Esplús	73	646	336,566	28544,2686
Siétamo	49	675	351,675	28996,3075
Almunia de San Juan	35,7	681	354,801	29244,3541
Peñalba	156,7	691	360,011	27003,4551
Igriés	19,2	700	364,7	29554,902
Alcampell	58	707	368,347	28837,2587
Castejón de Sos	31,8	733	381,893	29329,3533
Tierz	6,5	734	382,414	29798,6594
Panticosa	95,9	757	394,397	28146,7977
Osso de Cinca	27,7	771	401,691	29414,6431
Estadilla	46,8	822	428,262	29072,9782
Albelda	51,9	839	437,119	28982,5739
San Miguel del Cinca	106,5	852	443,892	27973,4632
Ballobar	127,7	866	451,186	27583,8326
Valle de Hecho	234,4	903	470,463	25614,6463
Fonz	55,5	960	500,16	28945,366
Tardienta	90,6	1005	523,605	28305,5965
La Sotonera	165,5	1013	527,773	26918,9033
Lalueza	88,2	1024	533,504	28354,7304
Boltaña	139,5	1041	542,361	27407,7781
Ayerbe	63,9	1097	571,537	28823,0717
Alcolea de Cinca	83,2	1136	591,856	28474,7696
Benabarre	157,1	1153	600,713	27108,8133
Albalate de Cinca	44,2	1201	625,721	29213,6961

Torrente de Cinca	56,8	1205	627,805	28981,0685
Lanaja	183,7	1324	689,804	26657,3904
Belver de Cinca	82,6	1346	701,266	28537,1546
Altorricón	32,4	1477	769,517	29499,8397
Sallent de Gállego	162,1	1484	773,164	27096,9104
Biescas	189,1	1535	799,735	26608,7795
Binaced	78,5	1539	801,819	28660,2799
Gurrea de Gállego	192	1575	820,575	26564,7775
Zaidín	92,6	1837	957,077	28471,6077
Grañén	124	1902	990,942	27905,3182
Benasque	233,6	2149	1119,629	25933,6269
Aínsa-Sobrarbe	284,8	2213	1152,973	25000,0013
Almudévar	201,5	2519	1312,399	26619,0779
Graus	299,8	3429	1786,509	25018,7269
Tamarite de Litera	110,6	3604	1877,684	28569,2124
Sariñena	275,6	4293	2236,653	25678,2973
Binéfar	25,1	9461	4929,181	31584,0761
Sabiñánigo	586,8	9883	5149,043	21273,1683
Jaca	406,3	13121	6836,041	25410,0341
Fraga	437,6	14926	7776,446	25270,3326
Barbastro	107,6	17109	8913,789	31921,4029
Monzón	155	17176	8948,696	31058,9616
Huesca	161	52555	27381,155	39583,7355

(\*) El número de toneladas de RSU generadas anualmente por cada uno de los municipios se ha estimado con el dato de producción de residuos sólidos urbanos en España en el año 2008 por habitante y año (dato ofrecido por el IAE).

(\*\*) La capacidad del vertedero para cada municipio ha sido estimada (véase 4.3.2).

### A.3. Datos de población de la provincia de Teruel

La Tabla A.3 muestra los datos de población de los municipios de la provincia de Zaragoza de más de 500 habitantes (ordenados por tamaño de población).

Tabla A.3: Municipios con más de 500 habitantes (Teruel)

Municipio	Extensión (km <sup>2</sup> )	Población	(*) RSU (t/año)	(**) Capacidad
Torrijo del Campo	44,04	506	263,626	29047,013
Urrea de Gaén	41,42	516	268,836	29098,0288
La Puebla de Valverde	282,78	550	286,55	24631,5138
Manzanera	168,66	554	288,634	26748,275
Mazaleón	86,25	567	295,407	28279,3297

Orihuela del Tremedal	71,52	572	298,012	28553,6444
Fuentes Claras	36,91	576	300,096	29196,2902
Cretas	52,66	598	311,558	28909,6554
Beceite	96,72	607	316,247	28094,9799
Mosqueruela	265,03	621	323,541	24977,9299
Alfambra	122,44	622	324,062	27621,7926
Cedrillas	73,57	631	328,751	28530,0393
Valdealgorfa	46,93	651	339,171	29028,8269
Muniesa	129,8	653	340,213	27492,9053
Aguaviva	42,15	660	343,86	29119,645
Alloza	81,6	673	350,633	28391,4153
Caminreal	44,39	735	382,935	29096,4229
Cantavieja	124,56	745	388,145	27612,5121
Rubielos de Mora	63,72	755	393,355	28742,9267
Castellote	235,18	794	413,674	25573,5782
Castelserás	31,52	834	434,514	29359,1986
Samper de Calanda	142,8	884	460,564	27308,2724
Ariño	81,93	893	465,253	28438,9991
Villarquemado	56,43	904	470,984	28914,4542
La Puebla de Híjar	60,78	982	511,622	28852,845
Escucha	41,57	995	518,395	29212,1717
Albarracín	452,72	1102	574,142	21615,5694
Calaceite	81,33	1105	575,705	28501,8723
Santa Eulalia	80,97	1122	584,562	28512,6964
Sarrión	140,44	1129	588,209	27411,8313
Montalbán	82,04	1350	703,35	28548,5134
Mas de las Matas	29,99	1360	708,56	29515,9614
Mora de Rubielos	166,23	1649	859,129	27060,6167
Híjar	165,36	1806	940,926	27115,0702
Albalate del Arzobispo	205,69	2087	1087,327	26435,9441
Valderrobres	124,04	2335	1216,535	28010,2719
Monreal del Campo	89,05	2709	1411,389	28750,2799
Cella	124,68	2878	1499,438	28130,9526
Utrillas	39,82	3238	1686,998	29792,133
Alcorisa	121,2	3556	1852,676	28360,9716
Calanda	112,25	3956	2061,076	28624,5446
Calamocha	316,63	4579	2385,659	24987,4137
Andorra	141,36	8266	4306,586	29136,9162
Alcañiz	472,12	16424	8556,904	24995,9936
Teruel	440,41	35841	18673,161	30323,5867

(\*) El número de toneladas de RSU generadas anualmente por cada uno de los municipios se ha es-

timado con el dato de producción de residuos sólidos urbanos en España en el año 2008 por habitante y año (dato ofrecido por el IAE).

(\*\*) La capacidad del vertedero para cada municipio ha sido estimada (véase [4.3.2](#)).



# B

## Scripts R

### B.1. Regresión

#### B.1.1. Fichero de datos

```
LocalizacionVertedero Extension Poblacion TotalRes
Barbastro 107.6 17109 30034
Fraga 437.6 14926 12504
Huesca 161 52555 40560
Calatayud 154.25 20926 38226
Ejeacaballeros 609.92 17331 32031
Zaragoza 973.78 674317 176136
Alcanniz 472.12 16424 21708
Teruel 440.41 35841 31949
```

#### B.1.2. Modelo

```
DatosCapacidad <- read.table("C:/Users/Lidia/Desktop/modelo.txt", header=TRUE,
sep="\t", na.strings="NA", dec=".",
strip.white=TRUE)

row.names(DatosCapacidad) <- as.character(DatosCapacidad$LocalizacionVertedero)
DatosCapacidad$LocalizacionVertedero <- NULL

RegModel.1 <- lm(TotalRes~Extension+Poblacion, data=DatosCapacidad)
summary(RegModel.1)
```

```

stepwise(RegModel.1, direction='backward/forward', criterion='BIC')
stepwise(RegModel.1, direction='forward/backward', criterion='BIC')

RegModel.2 <- lm(TotalRes~Poblacion, data=DatosCapacidad)
summary(RegModel.2)

oldpar <- par(oma=c(0,0,3,0), mfrow=c(2,2))
plot(RegModel.1)
par(oldpar)

```

## B.2. Análisis cluster

### B.2.1. Provincia de Zaragoza

#### Fichero de datos

Municipio	Latitud	Longitud
Ibdes	41.2156982	-1.8337400
Botorrita	41.5078011	-1.0286400
Terrer	41.3266983	-1.7117800
La Zaida	41.3282013	-0.4272700
Cabañas de Ebro	41.7958984	-1.1638600
Jarque	41.5569000	-1.6777800
Grisén	41.7455900	-1.1637200
Paracuellos de Jiloca	41.3143005	-1.6392100
Alpartir	41.4212990	-1.3815600
Villarroya de la Sierra	41.4631004	-1.7840300
Villanueva de Huerva	41.3520012	-1.0346800
La Almolda	41.5544014	-0.2121400
Herrera de los Navarros	41.2099991	-1.0841600
Pradilla de Ebro	41.8604012	-1.2628900
Novillas	41.9338989	-1.3925900
Azuara	41.2570000	-0.8688200
Perdiguera	41.7519989	-0.6342500
Sos del Rey Católico	42.4888000	-1.2151800
Cetina	41.2924004	-1.9617600
Lécera	41.6690800	-0.8551000
Aguarón	41.6589900	-0.8642800
Almonacid de la Sierra	41.3955994	-1.3230200
Nuez de Ebro	41.5915985	-0.6831700
Paniza	41.2830009	-1.2129101
Villafranca de Ebro	41.5751991	-0.6479300
Aniñón	41.6507400	-0.8735100
Uncastillo	42.3587990	-1.1342601
Sabiñán	41.6173600	-0.9244700

Luna 42.1689987 -0.9323700  
Novallas 41.9473991 -1.6953501  
Longares 41.4025993 -1.1669900  
La Joyosa 41.7442017 -1.0745100  
Boquiñeni 41.8616500 -1.2620000  
Sobradiel 41.7392006 -1.0393200  
Fuendejalón 41.7606010 -1.4707400  
Biota 42.2606010 -1.1876200  
Bujaraloz 41.4975014 -0.1527000  
Lumpiaque 41.6314011 -1.3054800  
Nonaspe 41.2080994 0.2425400  
Maluenda 41.2902985 -1.6176800  
Luceni 41.8284988 -1.2369400  
Alhama de Aragón 41.2916985 -1.8989800  
Sástago 41.6401700 -0.8721300  
Escatrón 41.2862015 -0.3201500  
Ainzón 41.6616100 -0.8980900  
Remolinos 41.8384018 -1.1795599  
Ariza 41.3137016 -2.0580599  
Magallón 41.8342018 -1.4560400  
Fabara 41.1782990 0.1684600  
Gelsa 41.4123001 -0.4486600  
Leciñena 41.7972984 -0.6070100  
Figueruelas 41.7640991 -1.1703200  
Morata de Jalón 41.4737015 -1.4738801  
Muel 41.4650002 -1.0860800  
Pastriz 41.6172981 -0.7808200  
Alfamén 41.6564700 -0.9069000  
Torres de Berrellén 41.7575989 -1.0685400  
Belchite 41.3053017 -0.7568400  
Sádaba 41.6544100 -0.8976000  
Brea de Aragón 41.5245018 -1.5974100  
Maella 41.1208000 0.1377400  
Ateca 41.3274994 -1.7930100  
Quinto 41.4261017 -0.5017700  
Alfajarín 41.6111984 -0.7043500  
Daroca 41.1138992 -1.4167200  
El Burgo de Ebro 41.5730019 -0.7416000  
Mequinenza 41.3727989 0.2995900  
Pina de Ebro 41.4930992 -0.5256700  
Villamayor de Gállego 41.6848488 -0.7722000  
Calatorao 41.5223999 -1.3446500  
San Mateo de Gállego 41.8288000 -0.7677000  
Gallur 41.8684998 -1.3183100  
Cadrete 41.5544014 -0.9674700  
Illueca 41.5376015 -1.6293600  
Pinseque 41.7360001 -1.0889500  
Ricla 41.5045013 -1.4046800

```

Cariñena 42.3181800 -2.0854900
Pedrola 41.7854996 -1.2166500
Mallén 41.9043999 -1.4221200
Villanueva de Gállego 41.7658997 -0.8295500
María de Huerva 41.5402985 -0.9962100
Fuentes de Ebro 41.5087013 -0.6330500
Épila 41.5983996 -1.2744599
Borja 41.8354988 -1.5375000
La Muela 41.5773010 -1.1191400
La Puebla de Alfindén 41.6325989 -0.7518000
Alagón 41.6558100 -0.9061200
Zuera 41.8711014 -0.7858600
Cuarте de Huerva 41.5993996 -0.9315700
Tauste 41.9154015 -1.2569700
La Almunia de Doña Godina 41.6617900 -0.8985400
Caspe 41.2378998 -0.0442100
Tarazona 41.9043007 -1.7246500
Ejea de los Caballeros 42.1274986 -1.1366600
Utebo 41.7131004 -0.9969000
Calatayud 41.3541985 -1.6410400
Zaragoza 41.6665001 -0.8773200

```

## Modelo

```

ClusterZaragoza <- read.table("C:/Users/Lidia/Desktop/ClusterZaragoza.txt",
header=TRUE, sep="\t", na.strings="NA", dec=".",
strip.white=TRUE)

row.names(ClusterZaragoza) <- as.character(ClusterZaragoza$Municipio)
ClusterZaragoza$Municipio <- NULL

HClust.1 <- hclust(dist(model.matrix(~-1 + Latitud+Longitud, ClusterZaragoza)) ,
method= "ward")
plot(HClust.1, main= "Dendograma", ylab= "", xlab= "Municipios Zaragoza
(>500 habitantes)", sub="Método=Ward; Distancia= Euclidea")
summary(as.factor(cutree(HClust.1, k = 5))) # Cluster Sizes

by(model.matrix(~-1 + Latitud + Longitud, ClusterZaragoza), as.factor(cutree(
HClust.1, k = 5)), colMeans) # Cluster Centroids

biplot(princomp(model.matrix(~-1 + Latitud + Longitud, ClusterZaragoza)),
xlab = as.character(cutree(HClust.1, k = 5)))
ClusterZaragoza$hclus.label <- assignCluster(model.matrix(~-1 + Latitud +
Longitud, ClusterZaragoza), ClusterZaragoza, cutree(HClust.1, k = 5))

```

### B.2.2. Provincia de Huesca

#### Fichero de datos

Municipio	Latitud	Longitud
Sena	41.71609879	-0.044399999
Broto	42.6049003	-0.123159997
Almuniente	41.94900131	-0.410459995
Loporzano	42.15980148	-0.325069994
Canfranc	42.71849823	-0.524909973
Nueno	42.26539993	-0.439709991
Ontiñena	41.67409897	0.08444
Castejón del Puente	41.96210098	0.156739995
Robres	41.86569977	-0.458600014
Peralta de Alcofea	41.93019867	-0.069789998
La Fueva	42.37018967	0.270960003
Esplús	41.79790115	0.275160015
Siétamo	42.12960052	-0.281450003
Almunia de San Juan	41.93629837	0.245770007
Peñalba	41.50009918	-0.03946
Igriés	42.22484	-0.40458
Alcampell	41.90530014	0.430059999
Castejón de Sos	42.51229858	0.487749994
Tierz	42.13470078	-0.35473001
Panticosa	42.72380066	-0.283969998
Osso de Cinca	41.66690063	0.189620003
Estadilla	42.05709839	0.242770001
Albelda	41.86429977	0.464489996
San Miguel del Cinca	41.85176849	0.128490001
Ballobar	41.61970139	0.19156
Valle de Hecho	42.73979187	-0.751359999
Fonz	42.01190186	0.257070005
Tardienta	41.97510147	-0.537039995
La Sotonera	42.26247025	-0.552110016
Lalueza	41.83990097	-0.258219987
Boltaña	42.44509888	0.069370002
Ayerbe	42.27510071	-0.683220029
Alcolea de Cinca	41.72269821	0.115270004
Benabarre	42.10839844	0.486070007
Albalate de Cinca	41.72280121	0.144590005
Torrente de Cinca	41.47340012	0.335790008
Lanaja	41.77099991	-0.326510012
Belver de Cinca	41.6882019	0.178900003
Altorricón	41.80360031	0.414680004
Sallent de Gállego	42.77389908	-0.327719986
Biescas	42.62870026	-0.321379989
Binaced	41.82640076	0.200200006
Gurrea de Gállego	42.01599884	-0.762769997
Zaidín	41.60348	0.26254

```

Grañén 42.13809967 -0.409559995
Benasque 42.60300064 0.522159994
Aínsa-Sobrarbe 42.41452026 0.139669999
Almudévar 42.03929901 -0.581439972
Graus 42.19029999 0.336279988
Tamarite de Litera 41.86660004 0.419290006
Sariñena 41.79840088 -0.15772
Binéfar 41.85229874 0.296169996
Sabiñánigo 42.5158 -0.3624299
Jaca 42.56629944 -0.552259982
Fraga 41.52259827 0.347090006
Barbastro 42.03120041 0.125430003
Monzón 42.1414 -0.41767
Huesca 42.13809967 -0.409559995

```

## Modelo

```

ClusterHuesca <- read.table("C:/Users/Lidia/Desktop/ClusterHuesca.txt",
header=TRUE, sep="\t", na.strings="NA", dec=".",
strip.white=TRUE)

row.names(ClusterHuesca) <- as.character(ClusterHuesca$Municipio)
ClusterHuesca$Municipio <- NULL

HClust.2 <- hclust(dist(model.matrix(~-1 + Latitud+Longitud, ClusterHuesca)) ,
method= "ward")
plot(HClust.2, main= "Dendograma", ylab= "", xlab= "Municipios Huesca
(>500 habitantes)", sub="Método=Ward; Distancia= Euclídea")
summary(as.factor(cutree(HClust.2, k = 9))) # Cluster Sizes

by(model.matrix(~-1 + Latitud + Longitud, ClusterHuesca), as.factor(cutree(
HClust.2, k = 9)), colMeans) # Cluster Centroids

biplot(princomp(model.matrix(~-1 + Latitud + Longitud, ClusterHuesca)),
xlab = as.character(cutree(HClust.2, k = 9)))
ClusterHuesca$hclus.label <- assignCluster(model.matrix(~-1 + Latitud + Longitud,
ClusterHuesca, ClusterHuesca, cutree(HClust.2, k = 9)))

```

### B.2.3. Provincia de Teruel

#### Fichero de datos

```

Municipio Latitud Longitud
Torrijo del Campo 40.82490158 -1.339140058
Urrea de Gaén 41.1607121 -0.47090815
La Puebla de Valverde 40.2234993 -0.929340005

```

Manzanera 40.05709839 -0.830340028  
 Mazaleón 41.04930115 0.102619998  
 Orihuela del Tremedal 40.55099869 -1.653049946  
 Fuentes Claras 40.86309814 -1.324219942  
 Cretas 40.92919922 0.209920004  
 Beceite 40.82970047 0.178550005  
 Mosqueruela 40.36230087 -0.446480006  
 Alfambra 40.54650116 -1.035069942  
 Cedrillas 40.43610001 -0.855809987  
 Valdealgorfa 40.98949814 -0.034529999  
 Muniesa 41.03269958 -0.809459984  
 Aguaviva 40.82310104 -0.196930006  
 Alloza 40.97040176 -0.530179977  
 Caminreal 40.83589935 -1.324210048  
 Cantavieja 40.5284996 -0.405519992  
 Rubielos de Mora 40.19020081 -0.650449991  
 Castellote 40.7989006 -0.318430007  
 Castelserás 40.97999954 -0.144260004  
 Samper de Calanda 41.19070053 -0.389120013  
 Ariño 41.03092603 -0.59257365  
 Villarquemado 40.51929855 -1.264660001  
 La Puebla de Híjar 41.21360016 -0.444689989  
 Escucha 40.79460144 -0.813859999  
 Albarracín 40.40869904 -1.434800029  
 Calaceite 41.01689911 0.186920002  
 Santa Eulalia 40.56499863 -1.314700007  
 Sarrión 40.14210129 -0.808309972  
 Montalbán 40.832901 -0.800849974  
 Mas de las Matas 40.83499908 -0.247250006  
 Mora de Rubielos 40.25059891 -0.753620028  
 Híjar 41.17436129 -0.44789535  
 Albalate del Arzobispo 41.12239838 -0.511529982  
 Valderrobres 40.8742981 0.156580001  
 Monreal del Campo 40.78850174 -1.356179953  
 Cella 40.45449829 -1.287529945  
 Utrillas 40.81439972 -0.859380007  
 Alcorisa 40.89039993 -0.381249994  
 Calanda 40.94020081 -0.223609999  
 Calamocha 40.91749954 -1.29934001  
 Andorra 40.9803009 -0.452259988  
 Alcañiz 41.05120087 -0.15828  
 Teruel 40.34759903 -1.104269981

## Modelo

```
ClusterTeruel <- read.table("C:/Users/Lidia/Desktop/ClusterTeruel.txt",
```

```
header=TRUE, sep="\t", na.strings="NA", dec=".",
strip.white=TRUE)

row.names(ClusterTeruel) <- as.character(ClusterTeruel$Municipio)
ClusterTeruel$Municipio <- NULL

HClust.3 <- hclust(dist(model.matrix(~-1 + Latitud+Longitud, ClusterTeruel)) ,
method= "ward")
plot(HClust.3, main= "Dendograma", ylab= "", xlab= "Municipios Teruel
(>500 habitantes)", sub="Método=Ward; Distancia= Euclidea")
summary(as.factor(cutree(HClust.3, k = 5))) # Cluster Sizes

by(model.matrix(~-1 + Latitud + Longitud, ClusterTeruel), as.factor(cutree(
HClust.3, k = 5)), colMeans) # Cluster Centroids

biplot(princomp(model.matrix(~-1 + Latitud + Longitud, ClusterTeruel)),
xlab = as.character(cutree(HClust.3, k = 5)))
ClusterTeruel$hclust.label <- assignCluster(model.matrix(~-1 + Latitud + Longitud,
ClusterTeruel, ClusterTeruel, cutree(HClust.3, k = 5))
```

# C

## Scripts CPLEX

### C.1. Sintaxis del problema de la p-mediana en CPLEX

#### C.1.1. Archivo de modelo

```
//Datos del problema
int P = ...;
{string} Municipios = ...;
{string} Vertederos = ...;
int N = card(Municipios);
float Demanda[Municipios] = ...;
float Distancia[Municipios] [Vertederos] = ...;

//Variables
dvar boolean AperturaVertedero[Vertederos];
dvar boolean EnvioMunicipio[Municipios] [Vertederos];

//Función objetivo
minimize
sum(i in Municipios, j in Vertederos)
Demanda[i]*Distancia[i] [j]*EnvioMunicipio[i] [j];

//Restricciones
subject to {
    forall(i in Municipios)
        sum(j in Vertederos)
            EnvioMunicipio[i] [j] == 1;
```

```

sum(j in Vertederos)
    AperturaVertedero[j] == P;

//Seleccionar una de las dos restricciones siguientes:
forall(i in Municipios , j in Vertederos)
    EnvioMunicipio[i][j] <= AperturaVertedero[j];

//forall(j in Vertederos)
//sum( i in Municipios )
//EnvioMunicipio[i][j] <= (N -P+1)*AperturaVertedero[j];

//Fortalece la formulación
forall(i in Municipios)
    AperturaVertedero[i] <= EnvioMunicipio[i][i];
}

```

### C.1.2. Archivo de datos

#### Provincia de Zaragoza

P=3;

Municipios={"Biota","Bujaraloz","Lumpiaque","Nonaspe","Maluenda","Luceni","Alhama de Aragón","Sástago","Escatrón","Ainzón","Remolinos","Ariza","Magallón","Fabara","Gelsa","Leciñena","Figueruelas","Morata de Jalón","Muel","Pastriz","Alfamén","Torres de Berrellén","Belchite","Sádaba","Brea de Aragón","Maella","Ateca","Quinto","Alfajarín","Daroca","El Burgo de Ebro","Mequinenza","Pina de Ebro","Villamayor de Gállego","Calatorao","San Mateo de Gállego","Gallur","Cadrete","Illueca","Pinseque","Ricla","Cariñena","Pedrola","Mallén","Villanueva de Gállego","María de Huerva","Fuentes de Ebro","Épila","Borja","La Muela","La Puebla de Alfindén","Alagón","Zuera","Cuarte de Huerva","Tauste","La Almunia de Doña Godina","Caspe","Tarazona","Ejea de los Caballeros","Utebo","Calatayud","Zaragoza"};

Vertederos={"Biota","Bujaraloz","Lumpiaque","Nonaspe","Maluenda","Luceni","Alhama de Aragón","Sástago","Escatrón","Ainzón","Remolinos","Ariza","Magallón","Fabara","Gelsa","Leciñena","Figueruelas","Morata de Jalón","Muel","Pastriz","Alfamén","Torres de Berrellén","Belchite","Sádaba","Brea de Aragón","Maella","Ateca","Quinto","Alfajarín","Daroca","El Burgo de Ebro","Mequinenza","Pina de Ebro","Villamayor de Gállego","Calatorao","San Mateo de Gállego","Gallur","Cadrete","Illueca","Pinseque","Ricla","Cariñena","Pedrola","Mallén","Villanueva de Gállego","María de Huerva","Fuentes de Ebro","Épila","Borja","La Muela","La Puebla de Alfindén","Alagón","Zuera","Cuarte de Huerva","Tauste","La Almunia de Doña Godina","Caspe","Tarazona","Ejea de los Caballeros","Utebo","Calatayud","Zaragoza"};

Demanda =[539.756,543.924,547.05,564.243,573.1,573.621,576.226,580.915,605.923,618.427,621.032,628.847,637.183,638.225,638.746,676.779,703.871,708.56,723.148,

```
725.232,790.357,793.483,850.793,867.986,913.834,1056.067,1073.781,1098.268,
1164.435,1214.451,1232.686,1298.332,1353.037,1494.228,1561.958,1567.689,1576.546,
1752.123,1761.501,1783.383,1807.349,1842.777,1910.507,1943.851,2216.855,2315.324,
2394.516,2444.011,2560.715,2567.488,2622.193,3738.175,3869.467,4004.927,4016.91,
4046.086,5136.018,5840.931,9029.451,9209.717,10902.446,351319.157];
```

**Nota C.1.1.** *Se ha omitido en el conjunto de datos la matriz de distancias (matriz 62×62).*

### Provincia de Huesca

P=3;

```
Municipios={"Tardienta","La Sotonera","Lalueza","Boltaña","Ayerbe",
"Alcolea de Cinca","Benabarre","Albalate de Cinca","Torrente de Cinca",
"LANAJA","Belver de Cinca","Altorricón","Sallent de Gállego","Biescas",
"Binaced","Gurrea de Gállego","Zaidín","Grañén","Benasque","Aínsa-Sobrarbe",
"Almudévar","Graus","Tamarite de Litera","Sariñena","Binéfar","Sabiñánigo",
"Jaca","Fraga","Barbastro","Monzón","Huesca"};
```

```
Vertederos={"Tardienta","La Sotonera","Lalueza","Boltaña","Ayerbe",
"Alcolea de Cinca","Benabarre","Albalate de Cinca","Torrente de Cinca",
"LANAJA","Belver de Cinca","Altorricón","Sallent de Gállego","Biescas",
"Binaced","Gurrea de Gállego","Zaidín","Grañén","Benasque","Aínsa-Sobrarbe",
"Almudévar","Graus","Tamarite de Litera","Sariñena","Binéfar","Sabiñánigo",
"Jaca","Fraga","Barbastro","Monzón","Huesca"};
```

```
Demanda=[523.605,527.773,533.504,542.361,571.537,591.856,600.713,625.721,
627.805,689.804,701.266,769.517,773.164,799.735,801.819,820.575,957.077,
990.942,1119.629,1152.973,1312.399,1786.509,1877.684,2236.653,4929.181,
5149.043,6836.041,7776.446,8913.789,8948.696,27381.155];
```

**Nota C.1.2.** *Se ha omitido en el conjunto de datos la matriz de distancias (matriz 31×31).*

### Provincia de Teruel

P=2;

```
Municipios={"Albarracín","Calaceite","Santa Eulalia","Sarrión","Montalbán",
"Mas de las Matas","Mora de Rubielos","Híjar","Albalate del Arzobispo",
"Valderrobres","Monreal del Campo","Cella","Utrillas","Alcorisa","Calanda",
"Calamocha","Andorra","Alcañiz","Teruel"};
```

```
Vertederos={"Albarracín","Calaceite","Santa Eulalia","Sarrión","Montalbán",
"Mas de las Matas","Mora de Rubielos","Híjar","Albalate del Arzobispo",
"Valderrobres","Monreal del Campo","Cella","Utrillas","Alcorisa","Calanda",
```

```

"Calamocha", "Andorra", "Alcañiz", "Teruel"}];

Demanda=[574.142,575.705,584.562,588.209,703.35,708.56,859.129,940.926,
1087.327,1216.535,1411.389,1499.438,1686.998,1852.676,2061.076,2385.659,
4306.586,8556.904,18673.161];

Distancia=
[[0.00,214.11,34.51,76.93,112.07,162.68,81.83,185.52,175.82,214.38,66.29,22.25
,107.03,147.47,163.83,80.45,151.88,177.62,37.76],
[213.81,0.00,192.99,196.48,110.01,64.68,186.40,65.64,76.30,21.65,166.17,207.41,
117.53,67.37,52.46,167.06,71.86,37.87,180.22],
[34.62,192.75,0.00,69.36,83.32,141.32,74.26,156.78,147.08,193.03,30.58,13.81,
84.85,126.12,142.47,44.74,130.53,156.27,35.24],
[77.32,195.80,70.08,0.00,111.82,162.43,19.45,185.27,175.58,174.69,93.66,58.74,
106.79,147.23,163.58,107.82,151.64,177.38,37.74],
[111.40,109.60,83.15,111.95,0.00,58.17,116.85,79.35,69.66,109.87,56.33,97.57,
7.68,42.96,59.32,57.22,47.37,73.11,77.81],
[161.97,70.63,141.16,162.53,58.17,0.00,138.35,62.18,44.84,57.92,114.33,155.58,
65.69,15.53,18.70,115.23,26.58,34.15,128.39],
[81.50,180.77,74.26,18.62,93.76,129.34,0.00,148.33,137.41,164.53,97.84,62.92,
88.72,114.14,130.49,112.00,118.54,144.29,41.92],
[184.70,65.65,156.45,185.25,77.17,60.50,158.82,0.00,10.99,65.93,129.63,170.87,
80.99,43.91,42.12,130.52,30.10,29.37,151.11],
[175.01,76.30,146.76,175.56,65.78,44.48,147.44,10.97,0.00,76.57,119.94,161.18,
71.29,31.94,45.63,120.83,18.72,40.02,141.42],
[213.96,21.52,193.14,175.21,110.16,57.92,165.14,65.79,76.45,0.00,166.32,207.56,
117.68,67.52,52.61,167.21,72.01,38.02,180.37],
[65.61,165.09,29.97,93.18,55.66,113.66,98.08,129.11,119.41,165.37,0.00,44.39,
57.18,98.45,114.81,17.05,102.86,128.60,59.06],
[22.24,207.24,13.77,58.27,97.81,155.81,63.17,171.26,161.56,207.51,45.06,0.00,
88.37,140.60,156.96,59.22,145.01,170.75,24.14],
[106.25,117.73,85.55,106.80,8.30,66.30,111.70,81.75,72.06,118.01,58.72,88.51,
0.00,51.10,67.45,59.62,55.51,81.25,72.66],
[146.94,66.96,126.12,147.49,43.14,15.53,124.79,44.88,33.27,67.24,99.29,140.54,
50.65,0.00,16.68,100.19,15.01,30.48,113.35],
[163.11,52.04,142.30,163.66,59.31,18.50,140.96,43.59,45.95,52.32,115.47,156.71,
66.83,16.67,0.00,116.36,27.70,15.56,129.52],
[80.17,166.87,44.53,107.74,57.44,115.44,112.64,130.89,121.20,167.15,17.67,58.95,
58.96,100.24,116.59,0.00,104.64,130.39,73.62],
[151.25,71.77,130.44,151.81,47.45,26.45,129.11,30.06,19.14,72.05,103.61,144.86,
54.97,13.91,27.59,104.51,0.00,35.49,117.67],
[177.13,38.81,156.31,177.68,73.33,34.16,154.98,29.22,39.88,39.09,129.48,170.73,
80.84,30.68,15.78,130.38,35.44,0.00,143.54],
[38.68,179.79,36.03,37.47,77.75,128.36,42.37,151.20,141.51,180.07,59.61,24.69,
72.72,113.16,129.51,73.77,117.56,143.31,0.00]];

```

## C.2. Sintaxis del problema de la $p$ -mediana con capacidades en CPLEX

### C.2.1. Archivo de modelo

```

//Datos del problema
int P = ...;
{string} Municipios = ...;
{string} Vertederos = ...;
float Demanda[Municipios] = ...;
float Distancia[Municipios][Vertederos] = ...;

//Variables
dvar boolean AperturaVertedero[Vertederos];
dvar boolean EnvioMunicipio[Municipios][Vertederos];

//Función objetivo
minimize
    sum(i in Municipios, j in Vertederos)
        Demanda[i]*Distancia[i][j]*EnvioMunicipio[i][j];

//Restricciones
subject to {
    forall(i in Municipios)
        sum(j in Vertederos)
            EnvioMunicipio[i][j] == 1;

    sum(j in Vertederos)
        AperturaVertedero[j] == P;

    forall(j in Vertederos)
        sum(i in Municipios)
            Demanda[i]*EnvioMunicipio[i][j] <= Capacidad[j]*AperturaVertedero[j];

    //Fortalece la formulación
    forall(i in Municipios)
        AperturaVertedero[i] <= EnvioMunicipio[i][i];
}

```

### C.2.2. Archivo de datos

Basta con añadir a los conjuntos de datos (.dat) dados en la sección anterior (los correspondientes al problema PMP) las capacidades asignadas a cada uno de los posibles vertederos.

### Capacidades Zaragoza

```
Capacidad =[27604.9356,27740.191,29448.448,27939.0043,29265.2414,29506.6909,
29433.1952,24426.9965,28269.2627,29279.6183,29688.348,28123.1547,28581.1049,
28154.8023,28703.645,26746.0273,29754.7845,29221.5462,28610.999,29779.2538,
28214.5847,29114.3123,25064.2173,27745.7406,29916.749,26992.5155,28672.7521,
28059.4268,27735.0157,29343.9901,29856.6362,24648.1742,24648.9611,28783.3444,
29581.521,29808.9109,29705.3432,30356.9683,30104.3977,30276.6895,28905.7611,
29073.8317,28530.2685,29956.9703,29369.6055,28820.7918,28234.2094,27282.3803,
28955.9715,28282.6202,30654.3023,31042.3787,25375.9589,31451.3907,24109.0468,
30585.3896,22814.5996,27952.6597,22662.5803,33723.0897,31988.2416,176286.8985];
```

### Capacidades Huesca

```
Capacidad=[28305.5965,26918.9033,28354.7304,27407.7781,28823.0717,28474.7696,
27108.8133,29213.6961,28981.0685,26657.3904,28537.1546,29499.8397,27096.9104,
26608.7795,28660.2799,26564.7775,28471.6077,27905.3182,25933.6269,25000.0013,
26619.0779,25018.7269,28569.2124,25678.2973,31584.0761,21273.1683,25410.0341,
25270.3326,31921.4029,31058.9616,39583.7355];
```

### Capacidades Teruel

```
Capacidad=[21615.5694,28501.8723,28512.6964,27411.8313,28548.5134,29515.9614,
27060.6167,27115.0702,26435.9441,28010.2719,28750.2799,28130.9526,29792.133,
28360.9716,28624.5446,24987.4137,29136.9162,24995.9936,30323.5867];
```

## Bibliografía

- [1] Ch. Achillas, Ch. Vlachokostas, N. Moussiopoulos, G. Banias. Decision support system for the optimal location of electrical and electronic waste treatment plants: A case study in Greece. *Waste Management* 30 (2010) 870-879
- [2] M. Albareda-Sambola, J. A. Díaz, E. Fernández. A compact model and tight bounds for a combined location-routing problem. *Computers & Operations Research* 32 (2005) 407-428
- [3] P. Avella, M. Boccia, S. Salerno, I. Vasilyev. An aggregation heuristic for large scale p-median problem. *Computers & Operations Research* 39 (2012) 1625-1632.
- [4] M. Baiou, F. Barahona. On the linear relaxation of the p-median problem. *Discrete Optimization* 8 (2011) 344-375.
- [5] R. Baldacci, E. Hadjiconstantinou, V. Maniezzo, A. Mingozzi. A New Method for Solving Capacitated Location Problems Based on a Set Partitioning Approach. *Computers & Operations Research* 29 (2002) 365-386.
- [6] J. Bautista, J. Pereira. Modeling the problem of locating collection areas for urbanwaste management. *Omega* 34 (2006) 617 - 629.
- [7] J.M. Belenguer, E. Benavent, C. Prins, Caroline Prodhon, Roberto Wolfler Calvo. A Branch-and-cut method for the capacitated location-routing problem. *Computers & Operations Research* 38 (2011) 931-941.
- [8] P. Cappanera, G. Gallo, F. Maffioli. Discrete facility location and routing of obnoxious activities. *Discrete Applied Mathematics* 133 (2004) 3 - 28.
- [9] M. Conforti, G. Cornuejols, G. Zambelli. *Integer Programming*. Springer-Verlag, New York, 2014.
- [10] C. Contardo, V. Hemmelmayr, T. G. Crainic. Lower and upper bounds for the two-echelon capacitated location-routing problem. *Computers & Operations Research* 39 (2012) 3185-3199.
- [11] S. Dantrakul, C. Likasiri, R. Pongvuthithum. Applied p-median and p-center algorithms for facility location problems. *Expert Systems with Applications* 41 (2014) 3596-3604.
- [12] J. W. Escobar, R. Linfati, M. G. Baldoquin, Paolo Toth. A granular variable tabu neighborhood search for the capacitated location-routing problem. *Transportation Research Part B* 67 (2014) 344-356
- [13] J. W. Escobar, R. Linfati, P. Toth. A two-phase hybrid heuristic algorithm for the capacitated location-routing problem. *Computers & Operations Research* 40 (2013) 70-79.
- [14] J. Fathali. A genetic algorithm for the p-median problem with pos/neg weights. *Applied Mathematics and Computation* 183 (2006) 1071-1083.
- [15] G. Ghiani, D. Laganà, E. Manni, C. Triki. Capacitated location of collection sites in an urban waste management system. *Waste Management* 32 (2012) 1291-1296

- [16] G. Ghiani, D. Laganà, E. Manni, R. Musmanno, D. Vigo. Operations research in solid waste management: A survey of strategic and tactical issues. *Computers & Operations Research* 44 (2014) 22-32.
- [17] S. L. Hakimi. Optimum location of switching centers and the absolute centers and medians of a graph. *Operations Research*, 12:450-459, 1964.
- [18] S. L. Hakimi. Optimum Distribution of switching centers in a communication network and some related graph theoretic problems. *Operations Research*, Vol 13 Issue 3, p462, 14p, 1965.
- [19] J. Janéek, M. Kvet. Sequential approximate approach to the p-median problem. *Computers & Industrial Engineering* 94 (2016) 83-92.
- [20] R. B. Lopes, C. Ferreira, B. S. Santos. A simple and effective evolutionary algorithm for the capacitated location-routing problem. *Computers & Operations Research* 70 (2016) 155-162.
- [21] R. B. Lopes, S. Barreto, C. Ferreira, B. Sousa Santos. A decision-support tool for a capacitated location-routing problem. *Decision Support Systems* 46 (2008) 366-375.
- [22] G. Nemhauser, L. Wolsey. *Integer and combinatorial optimization*. Wiley-Interscience in Discrete Mathematics and Optimization, New York, 1999.
- [23] L. Orellana. *IBM ILOG CPLEX V12.6.1. Manual de uso básico*, Zaragoza, 2016.
- [24] C. Prins, C. Prodhon, R. W. Calvo. Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking. *4OR* 4 (2006) 221-238.
- [25] C. Prodhon, C. Prins. A survey of recent research on location-routing problems. *European Journal of Operational Research* 165 (2005) 1-19.
- [26] D. Queiruga, G. Walther, J. González-Benito, T. Spengler. Evaluation of sites for the location of WEEE recycling plants in Spain. *Waste Management* 28 (2008) 181-190
- [27] C.S. ReVelle, H.A. Eiselt. Location analysis: A synthesis and survey. *European Journal of Operational Research* 238 (2014) 1-17.
- [28] C.S. ReVelle, R. W. Swain. Central Facilities Location. *Geographical Analysis* 2 (1970) 30-42.
- [29] F. Samanlioglu. A multi-objective mathematical model for the industrial hazardous waste location-routing problem. *European Journal of Operational Research* 226 (2013) 332-340.
- [30] R. Tadei, N. Ricciardi, G. Perboli. The stochastic p-median problem with unknown cost probability distribution. *Operations Research Letters* 37 (2009) 135-141.
- [31] P. Toth y D. Vigo. *The vehicle routing problem*. Society for Industrial and Applied Mathematic, Siam. Monographs on Discrete Mathematics and Applications. 2001.
- [32] T-H. Wu, C. Low, J-W. Bai. Heuristic solutions to multi-depot location-routing problems. *Computers & Operations Research* 29 (2002) 1393-1415.
- [33] M. Yaghini, M. Karimi, M. Rahbar. A hybrid metaheuristic approach for the capacitated p-median problem. *Applied Soft Computing* 13 (2013) 3922-3930.



