

UNIVERSIDAD DE MÁLAGA
E.T.S. de Ingenieros Industriales



Tesis Doctoral

**ANÁLISIS
METAHEURÍSTICO EN LA
LOGÍSTICA INVERSA DE
RESIDUOS**

Antonio Serrano Elena
Universidad de Málaga
2015



Publicaciones y
Divulgación Científica

AUTOR: Antonio Serrano Elena

 <http://orcid.org/0000-0001-6228-0501>

EDITA: Publicaciones y Divulgación Científica. Universidad de Málaga



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-SinObraDerivada 4.0 Internacional:

<http://creativecommons.org/licenses/by-nc-nd/4.0/legalcode>

Cualquier parte de esta obra se puede reproducir sin autorización pero con el reconocimiento y atribución de los autores.

No se puede hacer uso comercial de la obra y no se puede alterar, transformar o hacer obras derivadas.

Esta Tesis Doctoral está depositada en el Repositorio Institucional de la Universidad de Málaga (RIUMA): riuma.uma.es

Departamento de Economía y Administración de Empresas

E.T.S. de Ingenieros Industriales

Universidad de Málaga



ANÁLISIS METAHEURÍSTICO EN LA LOGÍSTICA INVERSA DE RESIDUOS

Autor

Antonio Serrano Elena

Ingeniero en Automática y Electrónica Industrial

Universidad de Málaga

Directores

Elvira Maeso González

Dra. Ingeniera Industrial

Profesora Titular de Universidad

Universidad de Málaga

Enrique Alba Torres

Dr. Ingeniero en Informática

Catedrático de Universidad

Universidad de Málaga

Málaga, 2015

Esta página se ha dejado en blanco intencionadamente

El Dr. **Enrique Alba Torres**, Catedrático de Universidad del Departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga, y la Dra. **Elvira Maeso González**, Profesora Titular de Universidad del área de conocimiento de Organización de Empresas, adscrita al Departamento de Economía y Administración de Empresas de la Universidad de Málaga,

Certifican

que D. **Antonio Serrano Elena**, Ingeniero en Automática y Electrónica Industrial por la Universidad de Málaga, ha realizado en el Departamento de Economía y Administración de Empresas de la Universidad de Málaga, bajo sus direcciones, el trabajo de investigación correspondiente a su Tesis Doctoral titulada

ANÁLISIS METAHEURÍSTICO EN LA LOGÍSTICA INVERSA DE RESIDUOS

Revisado el presente trabajo, estimamos que puede ser presentado al tribunal que ha de juzgarlo, y autorizamos la presentación de esta Tesis Doctoral en la Universidad de Málaga.

En Málaga, Octubre de 2015

Fdo.: Dr. Enrique Alba Torres

Fdo.: Dra. Elvira Maeso González

Esta página se ha dejado en blanco intencionadamente

A Ana y Antonio

Esta página se ha dejado en blanco intencionadamente

Agradecimientos

Son muchas las personas a las que quiero mostrar mi gratitud, pero me gustaría dedicarlo especialmente a mi familia, a Ana y Antonio, por su paciencia y cariño. Quiero manifestar también mi agradecimiento a las siguientes personas:

A mi directora de tesis, Elvira Maeso, por su dedicación y ayuda en todo momento.

A mi también director de tesis, Enrique Alba, sin cuya inestimable ayuda en el conocimiento de los metaheurísticos habría sido imposible la realización de esta tesis. Gracias de nuevo a ambos, por su calidad humana, su desinteresado apoyo, su sabiduría, sus consejos, su paciencia...

A Rafael Arjona, Director Gerente de LIMASA, a Javier Pazos, Jefe de División del CAM de LIMASA y a Ángel Castillo, Jefe de Explotación del Consorcio Provincial De Residuos Sólidos Urbanos de la Diputación de Málaga, por atenderme y facilitar toda la información necesaria para poder completar esta tesis.

Y finalmente, a todas aquellas personas que con su contribución han permitido que esta tesis sea una realidad.

Esta página se ha dejado en blanco intencionadamente

Resumen

El concepto genérico de Logística Inversa incluye el estudio de todos los flujos de materiales o productos en sentido inverso en la cadena logística. La vida de los productos no termina al alcanzar al usuario final, sino que perduran en el tiempo y generan residuos, suponiendo esto no sólo un coste sino un problema social por su posibilidad de contaminar el medioambiente. Es por eso que las empresas consideran un factor de competitividad la protección del medioambiente en el que desarrollan su actividad. Los temas medioambientales deben ser tratados como una cuestión estratégica en las organizaciones actuales, como el control de contaminación, reciclaje, etc., e integradas en la estrategia de operaciones (Handfield et al. (1997); Angell y Klassen (1999)).

Dentro de la logística inversa, esta tesis se centra en la gestión de residuos sólidos urbanos, su reciclado y la optimización en la localización de las instalaciones de transferencia y las plantas de tratamiento. En España y en la mayoría de países europeos, a raíz de la cambiante legislación medioambiental, han surgido numerosos sistemas integrados de gestión (SIG) que se ocupan de la recuperación de una amplia gama de residuos (cartón y papel, envases de plástico, electrónicos, etc.).

La metodología de optimización del problema de diseño de rutas de distribución de paquetería, minimización de costes y localización de hubs, almacenes, etc. que aparecen en la gestión de la cadenas de suministro directa (Dantzing (1955)), es aplicable al área de la logística inversa para resolver el problema de diseño de la red de transporte de residuos y localización de los diferentes centros de recogida, centros de transferencia y plantas de tratamiento.

Es habitual que la cantidad de residuos generados y recuperados en distintos sectores vaya variando a lo largo de los años. En este entorno es muy difícil estimar con precisión la evolución de los valores de todos los datos y parámetros del problema correspondiente. El carácter dinámico (cambian los valores en cada periodo de tiempo) de este tipo de problema dificulta su enfoque cuantitativo. Aunque en la Ingeniería de Organización Industrial existen modelos matemáticos que intentan facilitar la toma de decisiones a ese respecto, en general suelen ser modelos estáticos (consideran un periodo de tiempo) y deterministas (v.g. excluyendo las técnicas estadísticas y no deterministas).

El objetivo de esta tesis es principalmente la utilización de algoritmos metaheurísticos de búsqueda para resolver un problema real, dinámico y determinista, de optimización de costes y localización en logística inversa dentro del ámbito de la gestión de Residuos Sólidos Urbanos (RSU) de la provincia de Málaga.

Para ello, se analiza y valora la metodología utilizada en distintas tesis, revistas y congresos nacionales e internacionales desde el punto de vista de la utilización de técnicas metaheurísticas. En algunas soluciones existentes basadas en metaheurísticas se utiliza como estructura de resolución la compuesta por un algoritmo metaheurístico, un modelador (AIMMS, GAMS) y un *solver* (CPLEX) que resuelva de forma rápida modelos lineales continuos. Éste será el esquema que se seguirá en esta tesis.

En la actualidad existen muchas y novedosas técnicas metaheurísticas de búsqueda, desde las existentes en la literatura convencional como los algoritmos genéticos (GA, Holland (1975)), búsqueda tabú (TS, Glover (1989)), búsqueda dispersa (SS, Glover (1977)), etc., hasta los más novedosos como los algoritmos de optimización por enjambre de partículas (PSO, Kennedy y Eberhart (1995)), otros como la búsqueda local iterada (ILS, Stützle (1999)), etc.

En esta tesis se han seleccionado dos técnicas metaheurísticas de cierta relevancia para probar la validez del enfoque propuesto: una técnica importante

por su implantación internacional como es el algoritmo GA y otra técnica interesante por funcionar con inteligencia de enjambre (*swarm intelligence*) como es el algoritmo PSO. Estas técnicas metaheurísticas se utilizan para resolver los problemas de optimización de costes y localización de instalaciones de recuperación de RSU (centros de transferencia y plantas de tratamiento).

Los pasos que se dan en esta tesis para resolver el problema de localización de centros de transferencia y plantas de tratamiento son los siguientes. Primero se utiliza la programación lineal para encontrar el valor exacto en el proceso de optimización. Y en segundo lugar se utilizan las técnicas metaheurísticas propuestas para obtener el óptimo global aproximado cuando el resultado del cálculo exacto no es viable. Tanto las técnicas exactas como los algoritmos metaheurísticos se han desarrollado con el lenguaje interpretado MATLAB R2014b. Se han valorado algunas otras aplicaciones informáticas para desarrollar dichas técnicas como GAMS 24.4.3 o AIMMS 4.6 como modeladores y CPLEX como *solver* y el desarrollo en C++ de la técnicas metaheurísticas, pero finalmente la potencia y versatilidad de MATLAB como herramienta matemática y su disponibilidad en la UMA determinaron su uso.

La metodología propuesta será aplicada a un caso real de gestión, llevada a cabo por la Diputación de Málaga, de los residuos urbanos generados por 90 municipios. En esta tesis se ha buscado la optimización de los costes de transferencia desde los municipios hasta los centros de transferencia y las plantas de tratamiento, así como su localización.

Los resultados obtenidos con las técnicas metaheurísticas propuestas en la localización y optimización del modelo matemático diseñado para resolver el problema real planteado sugieren la validez de este enfoque, así como propuestas de extensiones de la investigación y del modelo desarrollado mediante el algoritmo metaheurístico utilizado de PSO.

Esta página se ha dejado en blanco intencionadamente

Abstract

The generic concept of Reverse Logistics includes the study of all flows of materials or products in the reverse supply chain. The product life does not end when it reaches the end user, but on the contrary it lasts in time, provoking a cost and also a social problem for its potential to contaminate the environment. That is why companies consider a competitive factor protecting the environment in which they operate. Environmental issues should be treated as a strategic issue in organizations, such as pollution control, recycling, etc., and integrated in operations strategy (Handfield, et al. (1997); Angell and Klassen (1999)).

Within reverse logistics, this thesis focuses on the management of urban solid waste, recycling and optimizing the location of transfer facilities and treatment plants. In Spain and in most European countries, along with an increasing environmental legislation, many integrated management systems (IMS) have arisen dealing with recovery managing of a wide range of waste (cardboard and paper, plastic packaging, electronics, etc.).

One of our hypothesis of work is that the optimization methodology existing for the problem of optimal distribution routes of packages, cost minimization and location of hubs, warehouses, etc. appearing in the chain direct supply management (Dantzing (1955)), is applicable to the area of reverse logistics to solve the problem of network design for waste transport and location of several collection centers, transfer centers, and treatment plants.

It is usual that the amount of waste generated and recovered in different sectors will go varying over the years. In this environment, it is very difficult to accurately estimate the evolution of the values of all data and parameters

according to the models. The dynamic nature (changing the values in each time period) of these types of problems hinders its quantitative approach. Although In the organization engineering there are mathematical models that attempt to facilitate decision-making, it in general tends to be static (considering only a period of time) and deterministic models (e.g. excluding statistics and non-deterministic techniques).

The aim of this thesis focuses mainly on the use of search metaheuristic techniques on a dynamic and deterministic model to analyze and solve cost optimization problems and location in reverse logistics within the field of municipal waste management (MSW) of Malaga.

To do this, we first analyze and evaluate the existing approaches found in PhD theses, magazines and national and international conferences and based on the use of metaheuristic techniques. In some existing solutions based on metaheuristics is used a metaheuristic algorithm, a modeler (AIMMS, GAMS) and a solver (CPLEX) to resolve continuous linear models. This will be the pattern to be followed in this thesis.

At present there are many and novel search metaheuristic techniques, ranging from those in the conventional literature such as genetic algorithms (GA, Holland (1975)), tabu search (TS, Glover (1989)), scatter search (SS, Glover (1977)), etc., to the newest and bio-inspired algorithms based on swarm intelligence such as particles swarm optimization (PSO, Kennedy and Eberhart (1995)), and many others as iterated local search (ILS, Stützle (1999)), etc.

In this work we have selected two metaheuristic techniques having relevance in research to test the validity of the proposed approach: an important technique for its international presence as is the algorithm GA and another interesting technique that works with swarm intelligence as is the algorithm PSO. These metaheuristic techniques will be used to solve cost optimization problems and location of MSW recovery facilities (transfer centers and treatment plants).

The steps we take in this thesis to solve the location of transfer facilities and treatment plants are as follows. First, linear programming is used to find the exact value in the optimization process. Second, we propose several metaheuristic techniques that are used to reach the approximate global optimum when the result of the exact calculation is not feasible. Both the exact techniques and metaheuristic algorithms have been developed with the interpreted language MATLAB R2014b. We have also analyzed other potential approaches, like competing applications to develop these techniques as GAMS 24.4.3 or 4.6 AIMMS as modelers and CPLEX as "solver" and development in C ++ of metaheuristic techniques, but eventually the power and versatility of MATLAB as a mathematical tool and their availability at the UMA, determined its use.

The proposed methodology will be applied to a real case of management, carried out by the Council of Malaga, of urban waste generated by 90 municipalities. In this thesis we have searched the transfer costs optimization from municipalities to transfer centers and treatment plants, as well as its location.

Finally, the results obtained with the proposed metaheuristic techniques in locating and optimization of designed mathematical model to solve the proposed real problem, suggest the validity of the hypothesis done in this thesis. We also propose some research extensions and developed model by metaheuristic used in PSO.

Esta página se ha dejado en blanco intencionadamente

ÍNDICE GENERAL

Agradecimientos.....	I
Resumen.....	III
Abstract.....	VII
1. INTRODUCCIÓN Y OBJETIVO DE LA TESIS	1
1.1. Introducción	1
1.2. Consideraciones previas sobre logística inversa	2
1.3. Objetivos y metodología de la tesis	4
1.4. Estructura de la tesis.....	5
1.5. Aportaciones de esta tesis doctoral	7
2. LA LOGÍSTICA INVERSA DE LOS RSU Y ESTADO DEL ARTE	9
2.1. Introducción.....	9
2.2. Concepto y clasificación de los RSU	10
2.3. Gestión de los RSU	13
2.4. Conclusiones.....	18
3. DEFINICIONES DE LA LOGÍSTICA INVERSA DE RESIDUOS Y ESTADO DEL ARTE.....	21
3.1. Introducción.....	21

3.2. Definiciones de la logística inversa de residuos.....	21
3.3. Estado del arte sobre la logística inversa y localización de instalaciones ..	24
3.3.1. Tesis sobre la logística inversa en España y otros países	26
3.3.2. Artículos, conferencias y congresos de ámbito internacional sobre la logística inversa y heurísticos	32
3.4. Conclusiones.....	35
4. LOCALIZACIÓN DE INSTALACIONES Y METODOLOGÍA METAHEURÍSTICA EN LA LOGÍSTICA INVERSA DE RSU	37
4.1. Introducción.....	37
4.2. Modelo matemático optimización costes y localización de instalaciones...	39
4.3. Problema localización plantas de tratamiento y centros de transferencia ...	40
4.3.1. Antecedentes	41
4.3.2. Diseño del modelo matemático. Función objetivo	42
4.4. Metodología.....	52
4.4.1. Metodos de resolución	52
4.5. Conclusiones.....	55
5. TÉCNICAS METAHEURÍSTICAS UTILIZADAS.....	57
5.1. Introducción.....	57
5.2. Técnicas metaheurísticas	58
5.2.1. Algoritmos genéticos (GA).....	59
5.2.2. Recocido simulado (AS).....	60
5.2.3. Búsqueda Tabú (TS)	60
5.2.4. GRASP.....	61

5.2.5. Colonia de hormigas (ACO)	62
5.2.6. Búsqueda dispersa (SS)	63
5.2.7. Optimización de enjambre de partículas (PSO).....	64
5.2.8. Búsqueda local iterada (ILS)	65
5.3. Selección del algoritmo adecuado al problema	65
5.4. Aplicación de la metodología propuesta mediante GA	67
5.5. Aplicación de la metodología propuesta mediante PSO	75
5.6. Conclusiones.....	83
6. MODELADO Y RESOLUCIÓN: APLICACIÓN A UN CASO REAL ..	85
6.1. Introducción.....	85
6.2. Buscando aplicaciones informáticas de optimización	86
6.2.1. Resultados en Matlab con GA	94
6.2.2. Resultados en Matlab con PSO.....	100
6.3. Aplicación de metaheurísticas al caso real de la provincial de Málaga	104
6.4. Conclusiones.....	122
7. CONCLUSIONES Y FUTUROS DESARROLLOS.....	123
7.1. Conclusiones.....	123
7.2. Futuros desarrollos	125
BIBLIOGRAFÍA	127
ANEXO A.1: Algoritmo PSO básico.....	136
ANEXO A.2: Algoritmo GA	138

ANEXO B: Parámetros algoritmo GA	146
ANEXO C: Parámetros algoritmo PSO	147
ANEXO D: Parámetros y valores caso real.....	148

ÍNDICE DE TABLAS

2.1. Clasificación de los tipos de residuos.....	12
3.1. Resumen revisión bibliográfica de tesis	31
3.2. Resumen de artículos y conferencias sobre logística inversa.....	34
4.1. Índices del modelo matemático	46
4.2. Parámetros del modelo matemático.....	47
4.3. Variables del modelo matemático	47
4.4. Variables y restricciones por nº municipios y periodos	51
5.1. Funciones y valores usados como parámetros GA.....	74
5.2. Valores usados como parámetros PSO.....	81
6.1. Herramientas de resolución indicando problemas que resuelven	87
6.2. Herramientas de modelado indicando solvers compatibles.....	88
6.3. Resultados programación lineal en MATLAB para distintos municipios...92	
6.4. Resultados programación lineal con AIMMS para distintos municipios....93	
6.5. Resultados GA con MATLAB para distintos municipios.....95	
6.6. Resultados algoritmos PSO con MATLAB para distintos municipios	101
6.7. Resumen resultados distintas técnicas y distintos municipios	105
6.8. Costes sin estaciones c. transferencia p. tratamiento Málaga 2013.....	107
6.9. Resultados algoritmos PSO y GA con MATLAB 90 municipios.....	108
6.10. Resultados prueba <i>t-student</i> 40 y 90 municipios GA, PSO y exacto	113
6.11. Coste 2014 para 90 municipios según distribución actual	115

6.12. Coste 2014 aplicando algoritmos PSO y GA a 90 municipios.....	116
D.1. Costes y capacidad transferencia municipio c.transfer.-p.tratam.	148
D.2. Transferencia residuos entre c.transferencia y p.tratamiento	148
D.3. Recogida residuos centros transferencia 2013.....	149
D.4. Transferencia residuos a plantas de tratamiento.....	149
D.5. Producción residuos 90 municipios 5 años.....	150
D.6. Distancia entre los 90 municipios.....	153
D.7. Residuos centro transferencia Ronda 2010	162
D.8. Residuos planta tratamiento Antequera 2010.....	163
D.9. Residuos centro transferencia Axarquía 2010	164
D.10. Residuos centro transferencia Guadalhorce 2010	165
D.11. Residuos centro transferencia Ronda 2011	166
D.12. Residuos planta tratamiento Antequera 2011	167
D.13. Residuos centro transferencia Axarquía 2011	168
D.14. Residuos centro transferencia Guadalhorce 2011	169

ÍNDICE DE FIGURAS

1.1. Procesos de la Logística Inversa.....	2
1.2. Gestión actual de los RR.SS.UU. en Europa.....	3
2.1. Diferentes formas de gestión de RR.SS.UU. en varios países de la UE	10
2.2. Diferentes etapas de la logística inversa.....	14
2.3. Evolución de la generación de RR.SS.UU. en Málaga	15
2.4. Evolución de la generación de RR.SS.UU. en España.....	16
3.1. Estructura de una red de logística inversa	23
4.1. Transferencia residuos municipios, c. transferencia y p. tratamiento	45
4.2. Esquema distintos métodos resolución por técnica y tipo problema.....	54
5.1. Esquema básico de GA.....	68
5.2. Diagrama de flujo de GA utilizado en MATLAB.....	71
5.3. Modificación de posición partícula i	79
6.1. Archivo Excel con datos de distancias entre municipios	90
6.2. Programa AIMMS con resultados óptimo 40 municipios.....	93
6.3. Resultado mejores valores función aptitud GA 20 y 40 municipios	95
6.4. Buscando mejores valores función aptitud GA 40 municipios	96
6.5. Resultado función aptitud GA valores CrossoverFraction, 6 municipios ...	97
6.6. Resultado valores función aptitud PSO 6, 20, 40 y 90 municipios	102
6.7. Resultado mejores valores función aptitud PSO 90 municipios	109

6.8. Resultado mejores valores función aptitud GA 30 gen. 90 municipios	109
6.9. Resultado mejores valores función aptitud GA 100 gen. 90 municipios ..	110
6.10. Resultado valores función aptitud 30 ejec. PSO y GA 90 municipios....	110
6.11. Resultado Diagrama cajas 6, 20, 40, 90 municipios GA, PSO, exacto ...	114
6.12. Centros de transferencia Cártama y planta tratamiento Casarabonela	116
6.13. Centros de transferencia y planta tratamiento Antequera.....	117
6.14. Distribución transporte RR.SS.UU. actualidad provincia Málaga	118
6.15. Propuesta distrib. transporte RR.SS.UU. provincia Málaga GA y PSO .	118
6.16. Nueva distribución centros transferencia en la provincia de Málaga.....	120
6.17. Comparativa actual distribución y propuesta metaheurísticos Málaga ...	121

Capítulo 1

1. INTRODUCCIÓN Y OBJETIVO DE LA TESIS

En este capítulo se verán, en primer lugar algunas consideraciones previas sobre logística inversa y residuos, los objetivos y estructura de esta tesis, qué metodología se utilizará en la optimización de costes y localización de instalaciones y por último, las aportaciones de la tesis.

1.1. Introducción

Actualmente el término de *logística* (CSCMP (2015)) tiene un enfoque más amplio, extendiendo la perspectiva con nuevas actividades logísticas y productivas de las que se está ocupando una rama especializada de la logística que se ha denominado *logística inversa*, completando así el alcance global de la logística para unas condiciones actuales que incluyen aspectos ambientales. Es la denominada *logística total* (Carrasco, 2000).

De las diferentes actividades que se desarrollan en la logística inversa (Figura 1.1), esta tesis se centrará en la gestión de residuos, ya que se está convirtiendo en uno de los principales retos a los que se debe enfrentar la sociedad actual, debido a su creciente impacto ambiental, social y económico (se verá en el capítulo 2).



Figura 1.1: Procesos de Logística Inversa. *Fuente:www.interempresas.net*

1.2. Consideraciones previas sobre logística inversa

El interés por la logística inversa no es nuevo, sino que ha ido creciendo conforme han aparecido motivos de tipo medioambiental, económicos o legislativos. La logística inversa como recuperación de productos al final de su vida útil o su reutilización supone el cierre de la cadena de suministro (Stevens (1989)).

De las diversas definiciones de logística inversa encontradas en la literatura, y como ésta tesis se centra en la logística inversa de RSU, interesan aquellas que incluyan en su definición los términos “desechos y residuos generados en las ciudades”, por lo que se considerarán algunas de las dadas por Stock (1998), sobre el término de logística inversa: “...desde la perspectiva de la logística de los negocios, este término se refiere al rol de la logística en las devoluciones, reducción de las fuentes de aprovisionamiento, reciclaje, sustitución de materiales, reutilización de materiales, desechos, restauración, reparación y refabricación; desde la perspectiva de la ingeniería logística, ésta se refiere a la gestión de la logística inversa y a un modelo sistemático de negocio que aplica, a través de la empresa, las mejores ingenierías logísticas y metodológicas de gestión con el objetivo de obtener beneficios al cerrarse el circuito de la cadena de suministro”.

La producción de residuos urbanos se ha incrementado considerablemente en los últimos años debido a la mayor oferta y capacidad de consumo de productos por parte de la sociedad, productos que al final de su vida se convierten en residuos. En 2011 se generaron en España casi 22 millones de toneladas de RSU.

Hoy en día en España, casi el 60% de estos residuos tiene como destino final el vertedero, opción que supone la vía de gestión menos aconsejable tanto a nivel ambiental como económico y social (Goddard (1995)).

En la Figura 1.2 se muestra como se realiza la gestión de estos RSU en los países de la UE.

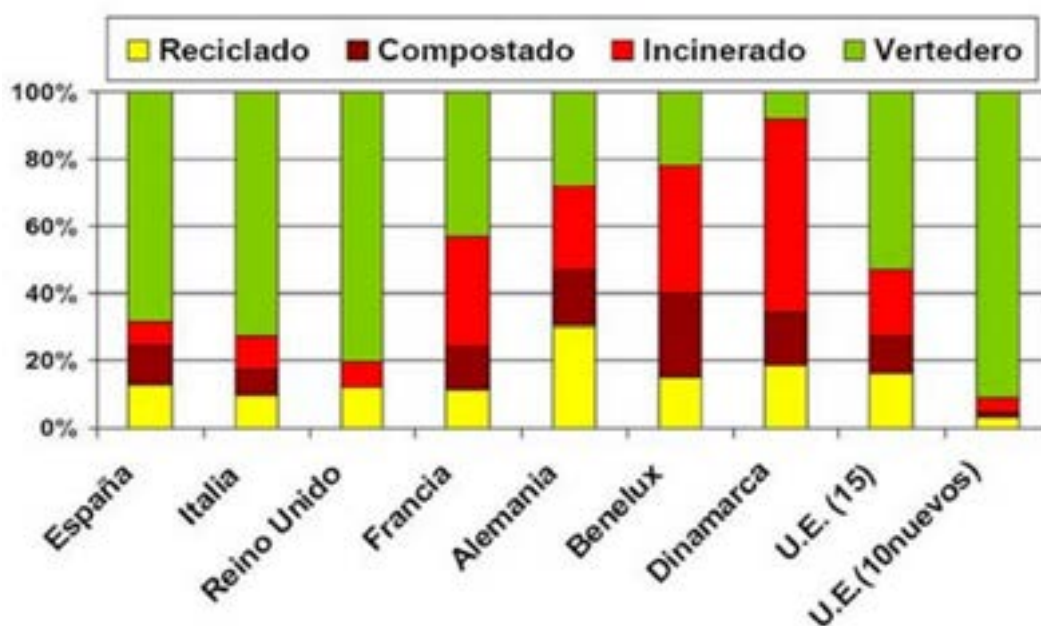


Figura 1.2: Gestión actual de los RSU en Europa. Fuente: Eurostat 2011.

De seguir con esta proporción de vertido se incumplirán los objetivos establecidos en la legislación española (Ley 22/2011, de residuos y suelos contaminados), que contempla reducir notablemente antes de 2020 la cantidad de residuos enviados a vertedero mediante su reutilización, reciclado y valorización de los mismos.

Por otra parte, la Comisión Europea ha revisado en 2014 la normativa sobre gestión de RSU, endureciendo las condiciones de uso de vertederos.

Todo esto hace que al elevado interés generado por la propia logística inversa, se añada el interés por una óptima gestión tanto económica como medioambiental de los vertederos y centros de transferencias. También es importante una idónea ubicación y reutilización de estas instalaciones, ante la cada vez más exigente legislación Nacional y Europea.

En esta tesis se busca resolver los problemas anteriormente citados, minimizando los costes de transferencia de los RSU a los centros de transferencias y a las plantas de tratamiento, así como la localización ideal de dichas instalaciones.

1.3. Objetivos y metodología de la tesis

Como ya se ha indicado en el apartado anterior, el objetivo principal de esta tesis es desarrollar la metodología (capítulo 4) y aplicar las herramientas de modelado y resolución necesarias para la optimización de los costes de transferencia de RSU.

La transferencia se realiza desde cualquier municipio de las distintas provincias españolas hasta los centros de transferencias y entre estos y las plantas de tratamiento. La localización de dichas instalaciones debe cumplir con las restricciones en la optimización de costes.

Este objetivo se concreta de forma más específica en:

1. Evaluar conforme a los propósitos de optimización, para obtener la posible metodología con la que resolver el problema planteado.

2. Modelar el problema de optimización de costes y de localización de centros de transferencia y plantas de tratamiento de RSU, en el lenguaje seleccionado para poder aplicar la metodología elegida.
3. Identificar y desarrollar las técnicas metaheurísticas adecuadas capaces de resolver de forma satisfactoria y en tiempos razonables el problema de optimización.
4. Finalmente, a partir de la función de coste diseñada y sus restricciones, comprobar el interés y la validez de la metodología y de las herramientas informáticas de modelado y resolución anteriores para su aplicación práctica en un problema real de optimización y localización de instalaciones en 90 municipios de la provincia de Málaga.

Se utiliza el método científico para la consecución de dichos objetivos, consistente en una revisión del estado del arte sobre logística inversa, la revisión y selección de técnicas metaheurísticas adecuadas a la resolución del problema planteado y la aplicación de la herramienta informática de modelado y resolución elegida. La hipótesis que se pretende demostrar es que el uso de metaheurísticas permitirá resolver problemas reales ahí donde los algoritmos exactos no pueden, bajo las condiciones de uso finales típicas en un organismo que hace logística inversa (tiempo, recursos, datos disponibles, etc.).

1.4. Estructura de la tesis

Esta tesis se ha articulado en 7 capítulos que se describen a continuación.

El capítulo 1 incluye los objetivos y metodología, la estructura y las principales aportaciones de la tesis. Aunque en el siguiente capítulo se darán algunas definiciones de logística inversa, se ha querido adelantar a este primer capítulo la definición que más se acerca al ámbito de esta tesis: la logística inversa de RSU.

En el capítulo 2 se incluyen los distintos enfoques de la logística inversa de RSU, algunas definiciones sobre gestión de residuos y del concepto de *vertedero*, se analiza el estado del arte de la logística inversa de residuos, se revisa la situación actual del modelo de gestión de RSU y la legislación que regula actualmente en materia de residuos y suelos contaminados.

En el capítulo 3 se revisa el estado del arte de las distintas tesis nacionales e internacionales, así como artículos de reconocidas e importantes revistas, congresos y conferencias de ámbito mundial. La revisión se hará sobre el marco de la logística inversa, enfocada en modelos exactos de optimización de costes y localización de instalaciones, y en el uso de técnicas metaheurísticas.

En el capítulo 4 se analizan los distintos modelos de localización de instalaciones en la literatura, que servirán de referencia para el problema matemático diseñado, así como los métodos de resolución más adecuados para resolver los problemas de optimización y localización de centros de transferencias y plantas de tratamiento de RSU.

En el capítulo 5 se presentan algunas técnicas metaheurísticas usadas actualmente, así como la descripción detallada de las dos técnicas metaheurísticas seleccionadas para resolver el problema de optimización modelado, como son los algoritmos GA y los algoritmos PSO.

En el capítulo 6 se analiza la herramienta informática seleccionada para modelar y resolver el problema de minimización de costes y localización de instalaciones, así como los resultados de la aplicación de la misma en la resolución de un problema real de optimización de la red de recogida y transporte de RSU, de 90 municipios de la provincia de Málaga.

Finalmente, el capítulo 7 incluirá las conclusiones más significativas derivadas de la tesis, así como las líneas de investigación futuras.

1.5. Aportaciones de esta tesis doctoral

Las principales aportaciones de esta tesis son las siguientes:

- Si bien, tras la revisión del estado del arte, la mayoría de los problemas de localización de tipo determinista están contemplados en dicha literatura, esta tesis pretende aportar una nueva aproximación en el ámbito de la logística inversa de gestión de RSU mediante el desarrollo de técnicas novedosas de carácter metaheurístico (como el algoritmo PSO) que permite aproximar el valor óptimo en la localización de instalaciones de transferencia y tratamiento de residuos urbanos a lo largo de diferentes periodos, y con tiempos de ejecución computacional suficientemente breves como para hacer factible la aplicación de dichas técnicas.
- Se introducen consideraciones de carácter más realista que lo encontrado hasta ahora en la literatura, al usar datos reales y resolver un problema de gestión de RSU de 90 municipios de la provincia de Málaga. Esto permite trasladar su estudio a un conjunto de problemas similares de ámbito global.
- Con carácter cualitativo, se realiza un estudio de la situación actual del sistema logístico inverso de gestión de RSU y su necesidad de adaptación conforme a la cada vez más exigente legislación medioambiental.
- La aplicación del lenguaje matemático MATLAB R2014b para modelar y resolver el problema de optimización. MATLAB ha permitido resolver tanto los modelos exactos iniciales, como los metaheurísticos seleccionados para resolver los modelos matemáticos donde el número de variables y restricciones convierten dichos modelos exactos en un problema NP-duro (Garey y Johnson (1979)), poniendo de manifiesto el interés e idoneidad del enfoque propuesto.

En definitiva, este trabajo de investigación supone un avance por el uso de un metaheurístico actual y no utilizado en este problema y que ha constatado su eficiencia al contrastar los datos obtenidos con los de otro metaheurístico ya clásico en la resolución de problemas de localización. También se introducen consideraciones de carácter más realistas de lo que generalmente se ha presentado en la literatura al usar datos reales, lo que dota a nuestros resultados y conclusiones de un mayor interés.

Capítulo 2

2. LA LOGÍSTICA INVERSA DE LOS RSU Y ESTADO DEL ARTE

En este capítulo se van a revisar brevemente las distintas aproximaciones a la Logística Inversa de residuos, centrándonos en un tipo concreto de residuos: los RSU, que son los generados por las actividades propias de las ciudades y que comprende la recogida domiciliaria de la *fracción resto* (recogida no selectiva), la procedente de la recogida de residuos de poda, muebles, etc. y la recogida selectiva de vidrio, papel-cartón, envases y embalajes.

2.1. Introducción

Si bien la Ley de residuos 10/1998 utilizaba una sola definición de residuos urbanos, la Ley 22/2011, de 28 de julio de residuos y suelos contaminados y que deroga a la anterior citada, establece nuevas definiciones de residuos como son los “residuos domésticos, comerciales e industriales” para entender el concepto de aquellos residuos no peligrosos generados por los ciudadanos. Aunque el volumen de los RSU es pequeño si lo comparamos con el resto de residuos (A. Del Val (1994)), en los últimos años ha experimentado un aumento significativo debido principalmente al incremento poblacional y a los hábitos de consumo.

2.2. Concepto y clasificación de los RSU

Un primer concepto general sobre residuos los podría definir como, “aquellas sustancias u objetos abandonados o descartados en forma permanente por quien los produce, por considerarlos ya sin utilidad en su provecho” (Tchobanoglous (1994)). Sin embargo, la legislación actual pretende regular una gestión eficaz de los residuos, para evitar que estos sean descartados de forma permanente buscando una valorización para que al menos algunos de los materiales que contienen puedan ser útiles para otro actor distinto del generador. En la Figura 2.1 se observa el diferente tratamiento dado a los residuos en varios países de la Unión Europea.

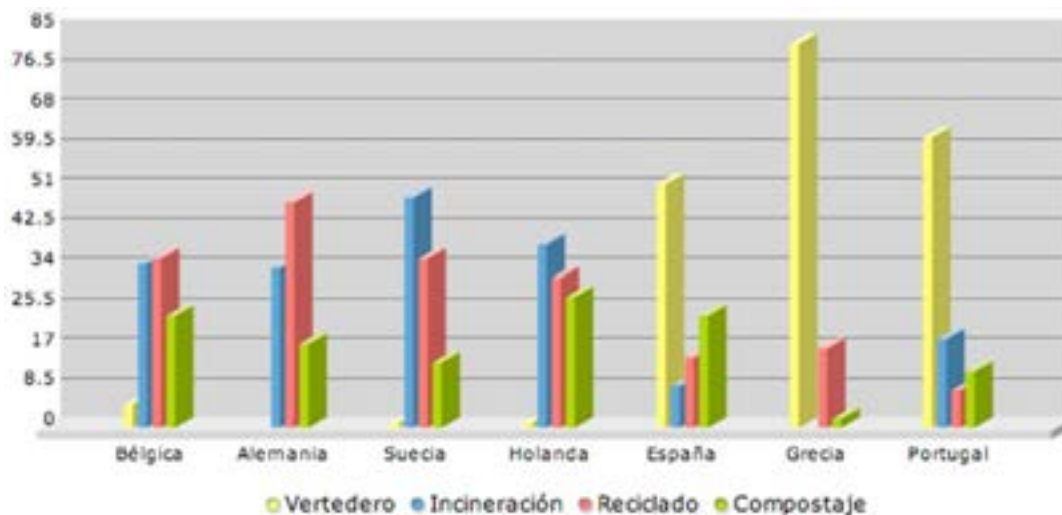


Figura 2.1: Diferentes formas de gestión de RSU en varios países de la UE. Fuente: Eurostat 2011.

Los RSU, por tanto, pueden definirse en una primera aproximación como los desechos generados en la comunidad urbana, provenientes de los procesos de consumo y desarrollo de las actividades humanas, y que normalmente son sólidos a temperatura ambiente. Además de los producidos por los usos residenciales, comerciales e institucionales, y por los trabajos en espacios públicos, los RSU incluyen los residuos originados en las industrias y en centros sanitarios, siempre que no tengan características tóxicas ni peligrosas, en cuyo caso existe otra legislación que establece las normativas específicas de gestión y manejo de este tipo de residuos.

La actual legislación en materia de residuos (Ley 22/2011) establece en su artículo 3 las definiciones sobre los residuos, quedando el concepto de RSU determinado como el conjunto de varios apartados del artículo.

Como definición general, se toma la siguiente:

- a) «Residuo»: cualquier sustancia u objeto que su poseedor deseché o tenga la intención o la obligación de desechar. La definición de RSU quedaría recogida en el conjunto de definiciones dadas en los apartados b, c y d, mostrados a continuación:
- b) «Residuos domésticos»: residuos generados en los hogares como consecuencia de las actividades domésticas. Se consideran también residuos domésticos los similares a los anteriores generados en servicios e industrias. Se incluyen también en esta categoría los residuos que se generan en los hogares de aparatos eléctricos y electrónicos, ropa, pilas, acumuladores, muebles y enseres así como los residuos y escombros procedentes de obras menores de construcción y reparación domiciliaria. Tendrán la consideración de residuos domésticos los residuos procedentes de limpieza de vías públicas, zonas verdes, áreas recreativas y playas, los animales domésticos muertos y los vehículos abandonados.
- c) «Residuos comerciales»: residuos generados por la actividad propia del comercio, al por mayor y al por menor, de los servicios de restauración y bares, de las oficinas y de los mercados, así como del resto del sector servicios.
- d) «Residuos industriales»: residuos resultantes de los procesos de fabricación, de transformación, de utilización, de consumo, de limpieza o de mantenimiento generados por la actividad industrial, excluidas las emisiones a la atmósfera reguladas en la Ley 34/2007, de 15 de noviembre.

Aunque, como se ha visto en las definiciones anteriores, los RSU están constituidos por un conjunto heterogéneo de materiales, estos se pueden dividir por su composición en dos categorías básicas:

- Orgánicos: restos de materiales resultantes de la elaboración de comidas, así como sus restos vegetales y animales.
- Inorgánicos: restos de elementos que no son fruto directo de la naturaleza sino de la industrialización de recursos naturales (plástico, vidrio, papeles, latas, textiles).

Tabla 2.1: Clasificación de los tipos de residuos. *Fuente: Elaboración propia.*

Origen/sector actividad	Tipo residuo	Clases	Ejemplos
Primario: agricultura, ganadería y selvicultura	Agrícolas, Ganaderos, Forestales		Tallos, hojas, etc. Estiércol y purines. Restos de madera, ramas y virutas
Secundario: industria y energía	Industriales Radiactivos	Inertes Asimilables a urbanos Tóxicos y peligrosos	Chatarra, vidrios, arenas, piedras Compuestos con berilio, productos inflamables, explosivos, etc. Restos de minerales de uranio. Restos que contienen radionucleidos
Terciario: servicios	Sólidos Urbanos Sanitarios	Domiciliarios Voluminosos Comerciales Construcción y demolición Asimilables a urbanos Biopeligrosos Químico-sanitarios Peligrosos Residuos radioactivos Restos anatómicos	Papel, materia orgánica, vidrio Restos de muebles, electrodomésticos, coches Envases, bolsas Ladrillo, madera Productos contaminados, jeringuillas, vendas gasas, etc.

En la Tabla 2.1 se da una clasificación completa de los residuos en función de su origen, tipo y clases.

2.3. Gestión de los RSU

La legislación en materia de residuos, Directiva Marco de Residuos Europea (Directiva 2008/98/CE) y la Ley de Residuos y Suelos Contaminados (Ley 22/2011), establece en su artículo 3, la definición de Gestión de residuos como: *“la recogida, el transporte y tratamiento de los residuos, incluida la vigilancia de estas operaciones, así como el mantenimiento posterior al cierre de los vertederos, incluidas las actuaciones realizadas en calidad de negociante o agente”*.

Dicha Ley contempla que el tratamiento de residuos debe seguir un orden de prioridad: prevención, preparación para la reutilización, reciclado, otro tipo de valorización (energética) y eliminación. Esta normativa supone la construcción de infraestructuras de pretratamiento, clasificación y valorización del residuo antes de ir a vertedero para su eliminación.

La gestión de RSU se puede dividir en cuatro fases diferenciadas: pre-recogida, recogida, transporte y tratamiento. La fase de pre-recogida que incluye el almacenamiento, manipulación, clasificación y presentación para su recogida, ha mejorado en los últimos años con la instalación de contenedores y campañas de sensibilización ciudadana. Las fases de recogida y transporte hasta los puntos de tratamiento o a las plantas de transferencia donde se compactan, suelen ser las más costosas. En la Figura 2.2 se pueden ver las distintas etapas de gestión en la logística inversa.

El tratamiento incluye operaciones de aprovechamiento de los materiales contenidos en los residuos o su eliminación. Los sistemas actualmente más utilizados de tratamiento son: el vertido controlado, la incineración, el reciclado y el compostaje. El vertido es el método más empleado en España en cuanto al volumen de residuo tratado.

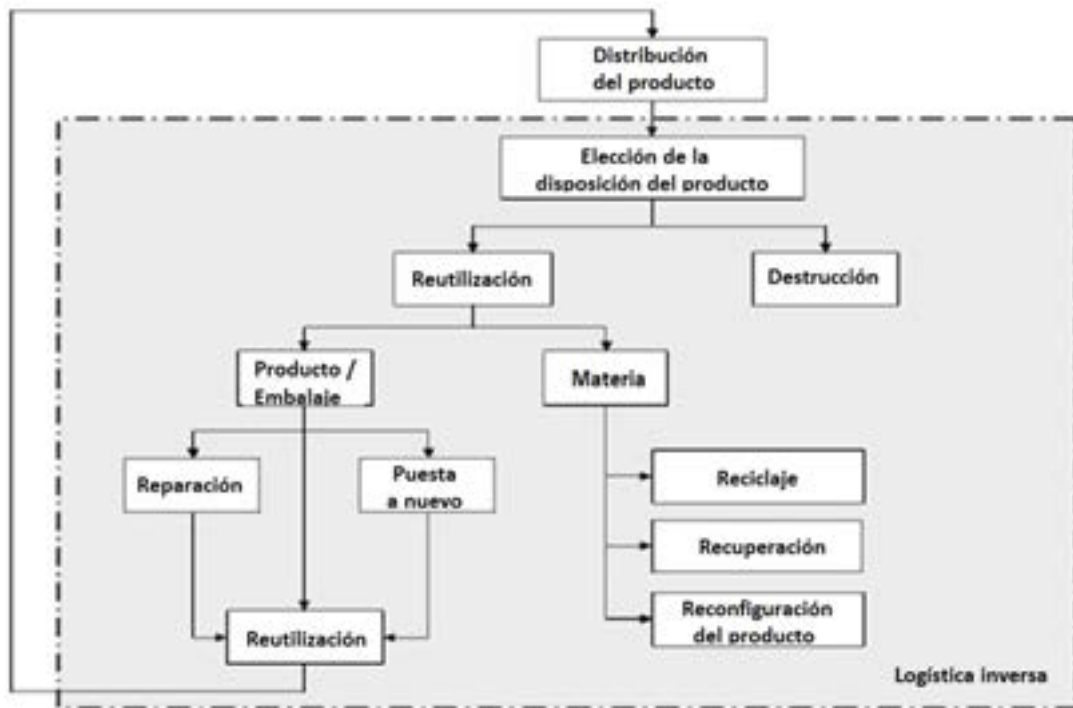


Figura 2.2: Diferentes etapas de la logística inversa.

Fuente: <http://www.free-logistics.com>

Hoy en día, la gestión de los residuos tiende a modelos productivos más eficientes (menor consumo de recursos energéticos, económicos, etc.) para toda su cadena de gestión, desde la recogida selectiva, hasta la salida de los productos recuperados de dichos residuos para conseguir finalmente una mejora ambiental y de protección de la salud. En la literatura se pueden encontrar aplicaciones de este enfoque de gestión de residuos más eficaz desde el punto de vista económico y ambiental en Jacobs y Everett (1992), en Keeler y Renkow (1994) y en Huhtala (1997). Aunque este último autor indica que las acciones para una eficiente gestión de residuos debe empezar en las fases de producción y comercialización de los bienes de consumo (Huhtala (1999)). Otros autores apuntan a la economía, señalando que: “...*el problema de la gestión de residuos sólidos proviene de la falta de conocimiento hacia la naturaleza económica del problema*” (Goddard (1995)).

El modelo actual de gestión integral de los residuos en nuestro país queda reflejado en la evolución de la generación de RSU a lo largo de los últimos años. Así, desde 2004 hasta 2007 la generación de RSU a nivel nacional ha aumentado progresivamente, para luego descender desde 2007 hasta 2013 más de un 10%, si

bien cabe indicar que en las cifras señaladas de descenso ha tenido un fuerte impacto la crisis económica iniciada en 2008. El descenso de la media nacional se ha situado en valores por debajo de los 500 Kg de residuo/habitante/año, teniendo como meta deseable bajar de los 1,30 Kg de RSU/habitante/día.

Hasta hace relativamente poco (año 1995), casi el 100% de los residuos urbanos eran destinados a deposición en vertederos controlados. Estos se pueden entender como recursos naturales agotables y como se indica en Ready & Ready (1995), a diferencia de otros recursos cuyo agotamiento es irreversible, un vertedero al alcanzar su límite de capacidad puede reemplazarse por otro vertedero de nueva construcción. Se trata de un recurso agotable y reemplazable, en los que tanto su capacidad como su ubicación son variables de decisión con transcendencia económica. En Jacobs y Everett (1992) se presenta un modelo para determinar el empleo óptimo de vertederos con reciclaje. En Huhtala (1997) se analiza el instante óptimo en que se debe dejar de utilizar un vertedero y sustituir por uno de nueva construcción. En André, Velasco y González (2004) se analiza el problema conjunto de capacidad y localización.

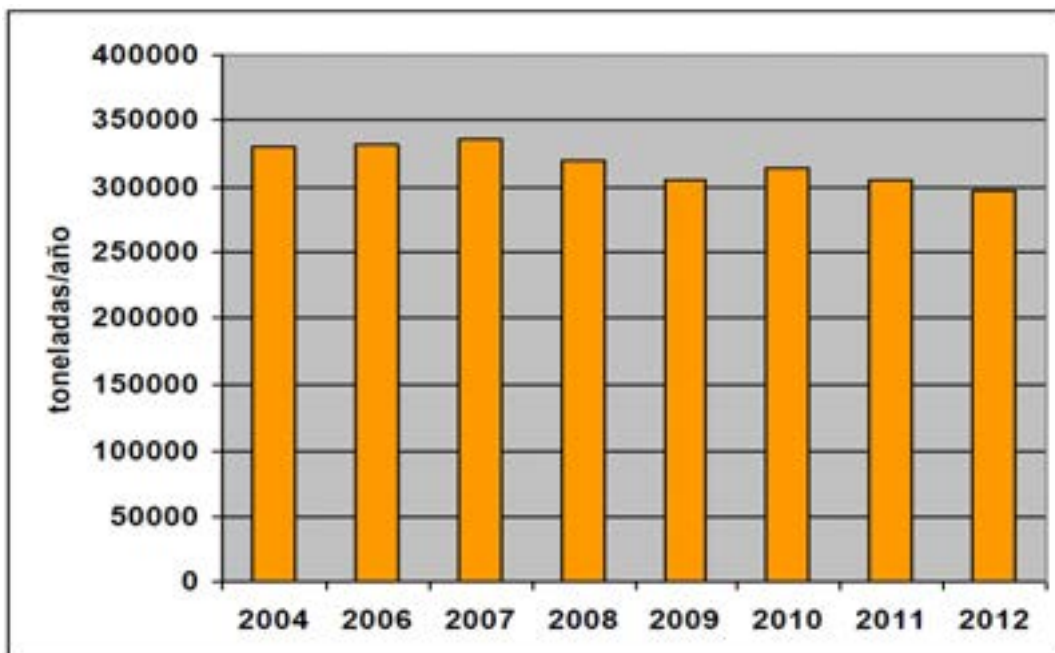


Figura 2.3: Evolución de la generación de RSU en Málaga.
Fuente: LIMASA, Servicios de Limpieza Integral de Málaga III, S.A.

En la actualidad, los RSU generados en los distintos municipios de las provincias españolas (en la Figura 2.3 se pueden observar las toneladas de RSU generadas en los últimos años por el municipio de Málaga) son finalmente llevados a las plantas de recuperación, en el que se integran distintas plantas de tratamiento de los RSU, como plantas de clasificación de envases, planta de reciclaje y compostaje, triturador de voluminosos, etc., con objeto de conseguir el reciclado y valorización de parte de los componentes conseguidos en los RSU.

El porcentaje de RSU destinado a tratamiento está experimentando un crecimiento continuado estando en muchas provincias de España y según datos del Instituto Nacional de Estadística (INE), por encima del 60% y con estimaciones de llegar al 80% óptimo establecido entre los años 2016 y 2017.

Las empresas gestoras de residuos urbanos recogieron 22,4 millones de toneladas de residuos en 2012, un 3,8% menos que en el año anterior. De éstos, 18,3 millones correspondieron a residuos mezclados y 4,1 millones a residuos de recogida separada, según se puede apreciar en la Figura 2.4.

Residuos urbanos recogidos.
Unidad: millones de toneladas

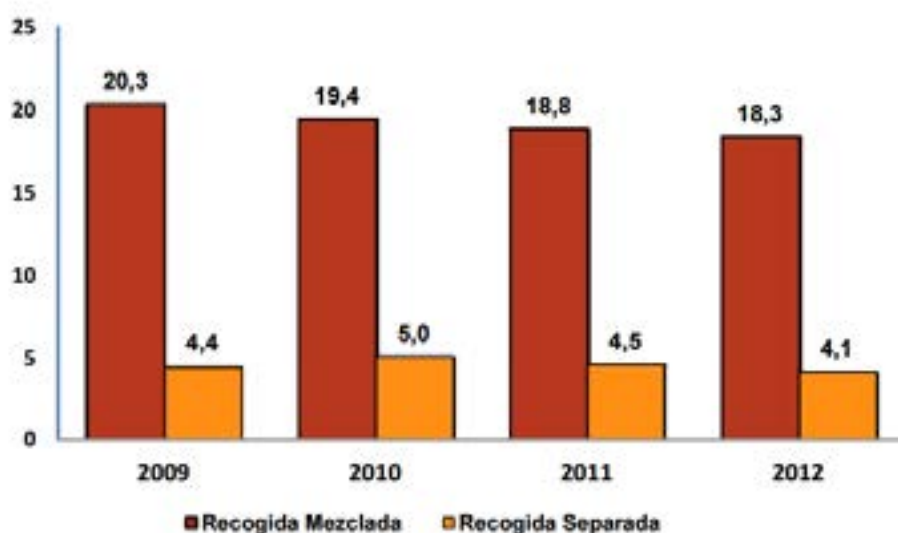


Figura 2.4: Evolución de la generación de RSU en España.
Fuente: INE.

Andalucía fue la comunidad autónoma que más residuos urbanos recogió en 2012, con 4,8 millones de toneladas seguida de Cataluña, con 3,6 millones de toneladas.

Otro objetivo específico respecto al destino de los RSU hace referencia a la cantidad de RSU tratadas para la obtención de *compost*. Sin embargo, desde la entrada en vigor de la Ley de Residuos y Suelos Contaminados de 28 de julio de 2011, solamente será considerado como *compost*, la enmienda orgánica obtenida a partir del tratamiento biológico aerobio y termófilo de residuos biodegradables recogidos separadamente. Por lo tanto, y según establece la nueva normativa no se considerará *compost* el material orgánico obtenido de las plantas de tratamiento, que solo se podrá considerar material bioestabilizado. Esto supone que todo el *compost* obtenido de las plantas de tratamiento tendrá una difícil salida al mercado, al ser considerado un producto de baja calidad.

Respecto a la recogida selectiva específica de vidrio, papel-cartón y envases, ésta supone un porcentaje entre el 6 y el 10% del total de residuos RSU. En el caso del vidrio y según datos del Instituto Nacional de Estadística (INE) entre 2009 y 2010 la recogida de vidrio varió a nivel nacional de los 15 Kg/habitante/año a los 17,1 Kg/habitante/año, sin embargo en autonomías como Andalucía esa media baja alrededor de los 9,4 Kg/habitante/año, si bien cada año incrementa ese porcentaje de recogida. Evidentemente, el número de contenedores de vidrio en cada municipio favorecerá la recogida de los mismos. El Plan Nacional Integrado de Residuos (PNIR) establece que debe haber, al menos 1 contenedor por cada 500 habitantes. En 2011, la media nacional estaba en los 264 habitantes/contenedor y la media andaluza en 337 habitantes/contenedor.

En cuanto a la recogida selectiva de papel-cartón, durante la última década se ha producido un incremento notable a nivel nacional, hasta alcanzar según el INE un valor medio nacional de 31,2 Kg/habitante/año, frente a los 16,2 Kg/habitante/año de la comunidad autónoma andaluza, lo que supone la necesidad de efectuar un gran esfuerzo para conseguir unos resultados de recogida homologables no solo a nivel nacional, sino comparables con los países de primera línea en Europa.

Respecto a la dotación de contenedores para la recogida de papel-cartón, en estos últimos años, España se ha convertido en el país europeo con mayor crecimiento en el reciclado de papel-cartón (14% en 2010), seguido de Alemania, Holanda e Italia. El PNIR establece un ratio de 500 habitantes por contenedor estando España por debajo de los 300 habitantes por contenedor.

Según Ecoembes, la media nacional de recogida selectiva de envases ligeros alcanzó los 11,3 Kg/habitante/año, aunque existen comunidades autónomas que necesitarán hasta 2017 para alcanzar estos niveles de recogida debido principalmente a la dotación de contenedores, que según establece el PNIR, es de 300 habitantes/contenedor, habiendo municipios con rangos superiores a los 450 habitantes/contenedor.

Por último, la recogida de la composición de residuos conocidos como *fracción resto* viene determinada por los residuos orgánicos generados en los domicilios y por otros componentes como vidrio, papel-cartón, envases, etc. que los acompañan. En los últimos años desde 2008, se han establecido objetivos a nivel estatal para aumentar la fracción orgánica por encima del 67% del total y disminuir el resto de fracción, lo que supondría una mejora en la separación de materiales reciclables. Hasta 2012 solo un 25% del vidrio, un 15% del papel-cartón y un 10% de los plásticos contenidos en los residuos son recogidos selectivamente, lo que indica un amplio margen de mejora.

2.4. Conclusiones

El porcentaje de RSU destinado a tratamiento y valorización ha ido aumentando en España de manera continuada desde 2004, pero sin alcanzar el 80%, cifra que se podría alcanzar entre los años 2016 y 2017 y establecida como fundamental para poder plantear a nivel nacional nuevas alternativas que determinen un cambio en el modelo actual de gestión. Este cambio viene definido porque el uso de vertedero es la última alternativa en la jerarquía de gestión de residuos para las administraciones competentes, que se encuentran con problemas

como la falta de espacio, la generación de lixiviados, emisión de gases de efecto invernadero, etc. Solo la valorización energética contemplada en la normativa actual sería una opción aceptable para dar viabilidad a los vertederos, ya que estos en cuestión de años agotarán su capacidad para la deposición de residuos, siendo en algunos caso muy escasa la posibilidad de disponer de nuevos terrenos desde el punto de vista municipal y cada vez más compleja la apertura de un nuevo vertedero por la mayor exigencia legislativa.

Por último, la dotación de un mayor número de contenedores, una ubicación más adecuada, así como la concienciación y sensibilización mediante campañas formativas e informativas a la sociedad permitirán aumentar los niveles de recogida selectiva y cumplimiento de objetivos.

Capítulo 3

3. DEFINICIONES DE LA LOGÍSTICA INVERSA DE RESIDUOS Y ESTADO DEL ARTE

No es el objetivo de esta tesis hacer un recorrido histórico de las últimas décadas de la logística inversa como parte de una logística global. Interesa el enfoque de una logística inversa hacia la gestión de residuos sólidos y urbanos y su tratamiento y eliminación, así como la localización de sus instalaciones de tratamiento.

3.1. Introducción

De la literatura existente se revisarán aquellas definiciones de logística inversa, que se centren en los términos de: reciclaje de residuos, gestión, vertido y vertederos. En el análisis del estado de la cuestión se revisarán todas aquellas tesis, nacionales e internacionales cuya investigación trate sobre logística inversa.

3.2. Definiciones de la logística inversa de residuos

Aunque Luttwak (1971) denominó como “Logística Inversa” al proceso recuperación de productos o materiales de los clientes, este término no se acuñó

hasta los años 90. De las definiciones dadas sobre “logística inversa” y que recojan el término de residuos y su gestión se escoge la que dio Stock (1992) y publicada por el Council of Logistics Management (CLM): “...*el término normalmente usado para referirse al papel de la logística en el reciclado, vertido de residuos y gestión de materiales peligrosos*”.

En el año 1998, Stock publicó lo que ha sido considerado como el libro blanco de la logística inversa de donde se ha sacado una nueva definición sobre logística inversa referente a la gestión de residuos y que se ha incluido en las consideraciones previas del capítulo 1.

Hay otros autores como Kopicki (1993), que al igual que Stock enfatizan la importancia de la gestión de residuos incluida su eliminación, en este nuevo contexto de la logística inversa.

Por otra parte, hay autores como Rogers y Tibben-Lembke (1999) que ya hablaban de eficiencia en la recuperación y eliminación de productos en la logística inversa y la definían como: “... *el proceso de planificación, ejecución y control de la eficiencia y eficacia del flujo de las materias primas, inventario en proceso, productos terminados e información relacionada, desde el punto de consumo hasta el punto de origen, con el fin de recuperar valor o la correcta eliminación*”.

Si se analizan las definiciones anteriores se puede concluir que la logística inversa constituye un importante sector de actividad dentro de la logística, que incluye multitud de actividades.

Algunas de estas actividades están enfocadas a aspectos ambientales, como la recuperación y el reciclaje de los productos. Otras buscan mejorar la eficiencia y eficacia en los procesos productivos y de suministro, para obtener mayores beneficios.

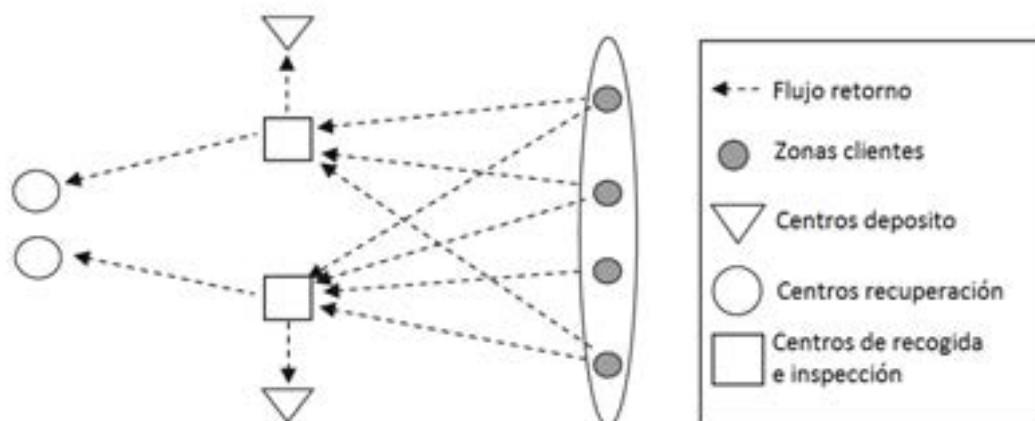


Figura 3.1: Estructura de una red de logística inversa.
Fuente: Elaboración propia.

En la Figura 3.1 se puede ver la estructura de una red de logística inversa. Dentro de las operaciones incluidas en la logística inversa se pueden encontrar: procesos de retorno por excesos de inventario, devoluciones de clientes, productos terminados y obsoletos, inventarios sobrantes del proceso productivo, actividades de retirada, clasificación, reacondicionamiento y reenvío al punto de venta o a otros mercados secundarios.

Y por último, respecto a las instalaciones de tratamiento de los residuos (centros ambientales, incluidos los vertederos) hay autores como André y Cerda (2001) que estudian el problema de determinar óptimamente la capacidad de vertederos. Otros como André, Velasco y González (2004) estudian tanto capacidad como localización y en Gaudet, Moreaux y Salant (1998) analizan el transporte desde varias ciudades a vertederos.

Los vertederos clasificados por algún autor como el destino final en el último proceso de logística inversa se pueden considerar como recursos agotables y reemplazables. Debido a su transcendencia económica, tanto por su capacidad como por su ubicación tienen una consideración especial como variables de decisión en el modelo matemático de minimización de costes en el transporte y en la localización de plantas de tratamiento desarrollado en esta tesis y cuyo diseño se detallará en los siguientes capítulos.

3.3. Estado del arte sobre la logística inversa y localización de instalaciones

En esta sección se hace una revisión en la literatura sobre la investigación en logística inversa y la localización de instalaciones aplicando técnicas metaheurísticas, así como de las tesis en España y en Europa que tratan sobre estas líneas de investigación.

Aunque existen bastantes investigaciones sobre teoría de localización de instalaciones en términos generales, sin embargo son pocas las que se han encontrado que incluyan esta teoría aplicada a la logística inversa. Lo que más ampliamente se ha desarrollado en la literatura de la logística inversa es referente a planificación de la producción, al control de inventario y la refabricación.

Hay varios autores que incluyen la elección óptima de las instalaciones en su modelo de logística inversa. Krikke (1998) propone un modelo de optimización para un efectivo diseño de redes para logística inversa. Son modelos para recuperación de múltiples productos ensamblados en distintos periodos. El modelo trata también problemas de localización de instalaciones con su correspondiente función de costes. Desarrolla un algoritmo heurístico para resolver el problema y lo complementa con una aplicación práctica adicional de rediseño de una cadena de fotocopiadoras.

Barros et al. (1998) proponen un modelo de localización de dos niveles para los problemas de reciclado de arena en Holanda y lo resuelven optimizando mediante técnicas heurísticas.

González y Adenso Díaz (2004) proponen un metaheurístico de búsqueda dispersa para tratar problemas de secuencia de desensamble óptimo para el caso de productos complejos que son reciclados o fragmentados al final de su vida. Evalúan el heurístico sobre un conjunto de 48 productos, considerando el tiempo usado en el desensamble de sus componentes en orden inverso a la secuencia de ensamble.

Lu y Bostel (2005) desarrollan un problema de localización de dos niveles con tres tipos de instalaciones para ser localizado en un sistema de logística inversa. Para resolver el problema diseñan un modelo de programación lineal entera-mixta, considerando simultáneamente flujos de interacción directo e inverso.

Jeung Ko y Evans (2005) proponen un modelo de programación no lineal entera-mixta para el diseño de una red de distribución integrada dinámica para la optimización de sistemas de trabajos directos e inversos.

También hay un creciente interés en la mayoría de las universidades españolas por la logística inversa creando grupos de investigación, muchos de ellos financiados por el ministerio de Educación y Ciencia.

Ciertamente las investigaciones sobre logística inversa están experimentando también un crecimiento notable, debido al aumento de los requisitos legislativos a las empresas en materia medioambiental.

Algunas de las universidades que tienen áreas de investigación en este terreno son: la Universidad Politécnica de Madrid, la Universidad de Sevilla, la Universidad de Extremadura, la Universidad de Oviedo, la Universidad de Navarra, la Universitat Pompeu Fabra, la Universidad Carlos III de Madrid, la Universidad de Salamanca, la Universitat Politècnica de Catalunya y la Universidad del País Vasco.

A nivel internacional se pueden encontrar numerosos artículos sobre logística inversa en prestigiosas revistas académicas como: *European Journal of Operational Research*, *International Journal of Production Economics*, *Interfaces*, en editoriales como Springer, Elsevier, o en Librerías Digitales como IEEE Xplore (Institute of Electrical and Electronics Engineers). También, nuevas líneas de investigación o grupos de investigación específicos sobre esta materia como REVLOG (*European Working Group on Reverse Logistics*) o RELOOP (*Reverse Logistics Executive Council*). En Norteamérica, el más importante grupo

de investigación sobre logística inversa es el formado por el CSCMP (*Council of supply chain management professionals*), antiguo CLM, en el que publicaban los autores norteamericanos de mayor prestigio en el asunto, como Stock, Rogers y Lembke, etc.

3.3.1. Tesis sobre la logística inversa en España y otros países

Aunque como se indicó en el apartado anterior existen grupos de investigación destacados sobre logística inversa en muchas de las universidades españolas, sorprende que en la búsqueda de tesis españolas sobre logística inversa solo se hayan encontrado once. La mayoría de tesis en España sobre logística trata sobre la logística directa o de “aguas abajo”. El resultado de esta revisión indica el amplio horizonte todavía por investigar en este campo.

Las distintas tesis españolas que se tratan a continuación han sido encontradas en la web “TESEO” del Ministerio de Educación del Gobierno de España.

Es evidente que en España el interés investigador en las universidades respecto de la logística inversa surgió más tarde que en el resto de países europeos.

La defensa de la primera tesis sobre logística inversa fue realizada por González Torre (2002), en ella se evalúan los distintos aspectos relacionados con la incorporación de las actividades de logística inversa al ciclo del envase de vidrio hueco. Se desarrolló un modelo de programación lineal multiobjetivo para reasignar eficientemente los contenedores disponibles por una empresa gestora de los residuos, maximizando el vidrio recogido con los mínimos costes logísticos.

En la tesis de Rubio Lacoba (2003) se analizan las posibilidades que presentan los productos desechados por los consumidores o fuera de uso. Se realiza una simulación dinámica de un modelo de gestión de inventario en el que se considera la posibilidad de recuperar los productos fuera de uso.

En la tesis de Fernández Quesada (2004) se quiere contribuir por una parte, al marco teórico de la logística inversa y por otra, a incorporar escenarios reales. En una primera fase utiliza la técnica del Focus Group para recabar información de personas expertas en el tema y en la segunda fase utiliza la metodología del estudio de casos para el análisis de varias empresas.

Soto Zuloaga (2006) en su tesis analiza los problemas de las empresas en los procesos de devoluciones en la industria editorial y propone modelos matemáticos y métodos de solución mediante algoritmos de ramificación y poda, para resolver problemas de casos reales en editoriales de la planificación en las devoluciones, optimizando y reduciendo los costes totales de producción. Utiliza modelos de programación lineal con variables enteras, resueltos por el software LINGO. Utiliza un heurístico de búsqueda local realizado en C++ y conectado con LINGO para resolver problemas donde el tamaño real (mayor número de variables y restricciones) complica la solución del modelo.

La tesis de Ortega Mier (2008) trata sobre la utilización de métodos cuantitativos, en concreto técnicas metaheurísticas para analizar problemas de localización en logística inversa. Utiliza algoritmos GA y SS para los problemas de localización de instalaciones de recuperación de residuo ligero de fragmentación de vehículos fuera de uso. Plantea un problema teórico inspirado en los municipios de la provincia de Guipúzcoa. La técnica menos eficiente y de peores resultados es la GA. Sin embargo, técnicas metaheurísticas como la PSO utilizada en esta tesis resultarían más eficientes, con tiempos computacionales menores para el número de variables y restricciones usadas.

López Parada (2010) en su tesis elabora un modelo de referencia que permite a las empresas una fácil adaptación a las necesidades derivadas de la gestión de logística inversa. En la investigación utiliza el método del caso múltiple, con el fin de encontrar patrones de comportamiento que permitan generalizar los resultados a partir de los datos obtenidos de 20 empresas de prestigio internacional.

Bustos Flores (2010) en su tesis analiza la incorporación de un modelo de logística inversa en el sector artesanal, para la adquisición de los insumos necesarios para la producción.

En la tesis de Mihi Ramírez (2010) se investiga la relación existente entre la creación del conocimiento y la logística inversa como fuente de ventajas competitivas sostenibles, tratando de elaborar instrumentos de medida que logren un acercamiento de ambos contextos. El autor considera que un flujo inverso en la función logística, amplificará las capacidades competitivas de la empresa.

La tesis de Benedito Benet (2010) propone el estudio de la influencia de la logística inversa en las capacidades de fabricación, *re-fabricación* (“devolver a un producto usado al menos su rendimiento original, garantizando que es equivalente o mejor que el del producto fabricado de nuevo”, Johnson et al. (2014)) y almacenaje óptimo de un sistema industrial. Se estudian tres modelos de un sistema en que el producto recuperado es indistinguible del producto nuevo, determinando la política de producción óptima utilizando una cierta función de coste.

En la tesis de Castillo Díaz (2011) se analizan las posibilidades que presenta el tema de la logística inversa para los productos desechados o productos fuera de uso en el sector vitivinícola. Se presentan dos metodologías aplicables a problemas de optimización: la primera trata de la selección de alternativas en empresas con múltiples criterios y atributos. La segunda trata de minimizar o maximizar objetivos en la gestión de productos fuera de uso.

Por último, en la tesis más reciente sobre logística inversa en España de Jiménez Parra (2014) se propone un modelo explicativo del comportamiento de compra del potencial consumidor de portátiles refabricados, que sirva de marco teórico de referencia para orientar a empresas interesadas en el diseño e implantación de estrategias de marketing más adecuadas para la comercialización de sus productos refabricados.

Respecto de las tesis españolas referidas a logística inversa, solo dos han utilizado sistemas metaheurísticos de optimización. En la tesis de Soto Zuloaga (2004) no queda muy claro el tipo de heurístico utilizado, ya que habla de uno de búsqueda local, programado en C++, pero sin citar autor alguno sobre dicho heurístico. La tesis de Ortega Mier (2008) refiere a la poca eficiencia de los GA frente a la TS, aunque para una mejor comparación se habría necesitado un tamaño de muestras mayor de 10 (al menos 30 ejecuciones de cada algoritmo).

Por otra parte, la principal contribución de la tesis presentada, no es solo el uso de un algoritmo metaheurístico más eficiente como es PSO, si no porque hasta ahora no se han encontrado referencias a dicho metaheurístico, en tesis españolas e internacionales (se verá a continuación) referidas a logística inversa.

Referente a las tesis sobre logística inversa a nivel internacional, las primeras tesis doctorales encontradas con referencias a la logística inversa fueron las de: Jahre (1995), Thierry (1997), Krikke (1998), Van der Laan (1997) y Fleischmann (2000).

Jahre (1995) estudió cómo funcionaban los sistemas de recolección y reciclaje de basura urbana, considerando la separación en origen y la recolección conjunta.

Fleischmann (2000) estudia, entre otras cuestiones, cómo pueden describirse las características de los sistemas de logística inversa a través de modelos cuantitativos y, de esta forma, mejorar el proceso de toma de decisiones.

Según la asociación de bibliotecas de investigación (DART-Europe), desde el año 2000 hasta hoy, se han leído 25 tesis doctorales europeas, sin considerar las españolas referentes a logística inversa. De las tesis europeas se comentan aquellas de contenido más afín al de esta tesis, bien por desarrollar metodologías cuantitativas como el uso de metaheurísticos para aproximar soluciones de problemas de optimización combinatoriales complejos, caracterizados como NP-duro, o que tratan de resolver problemas de minimización de costes en el ámbito de la logística inversa.

Kongar (2003) examina diferentes técnicas de decisión multicriterio sobre desensamblaje en la recuperación de productos al final de su vida, resolviendo los problemas NP-completos que se plantean mediante programación multi-objetivo y diversos metaheurísticos como TS y GA.

De Brito (2004) aporta conceptos sobre la logística inversa en la toma de decisiones y establece principios teóricos de la logística como campo de investigación, proponiendo recomendaciones obtenidas de un estudio Delphi realizado con grupos internacionales de académicos expertos en la materia.

Aidonis (2009) examina diferentes aspectos de la gestión de residuos de la construcción y demolición, para desarrollar metodologías cuantitativas y cualitativas innovadoras. En una primera parte, desarrolla un marco de trabajo con metodología holística para gestionar eficientemente los flujos de residuos generados en las actividades de construcción y demolición. En una segunda parte, genera modelos integrados de optimización de esos procesos de logística inversa, para maximizar el beneficio y minimizar el tiempo requerido.

Saibani (2010) propone un marco de trabajo ajustado a la logística de suministros de ciclo cerrado e inversa y realizado en tres niveles, que ofrezca un enfoque práctico para la toma de decisiones y que permita realizar y gestionar la cadena de suministro inversa con mayor eficacia.

Vercraene (2012) modela tres problemas de reciclado y reutilización de residuos con una capacidad de producción limitada, considerando problemas de inventario y producción operativa que se incorporan al flujo de retorno. Valora algunas políticas de heurísticos y los efectos de la capacidad de producción sobre esas políticas.

Finalmente, se menciona la tesis norteamericana de Du (2006) al proponer un modelo de minimización de los costes totales para el diseño de la logística inversa de un servicio post-venta, resuelto mediante métodos simplex dual y metaheurístico SS.

Tabla 3.1: Resumen revisión bibliografía tesis.

Fuente: Elaboración propia.

Tesis de logística inversa nacionales e internacionales			
Autor	Año de publicación	Tipo de problema	Método de solución
Jahre	1995	Problemas reciclaje y recolección basura	Separación origen y recolección conjunta
Fleischmann	2000	Logística inversa y modelos cuantitativos	Mejora procesos toma decisiones
González Torre	2002	Problema minimización costes de recogida vidrio	Programación lineal multi-objetivo
Kongar	2003	Problemas desensamblaje y recuperación productos	Programación multi-objetivo. Búsqueda Tabú y Algoritmos genéticos
Rubio Lacoba	2003	Gestión de inventario	Simulación dinámica
De Brito	2004	Toma decisiones y principios teóricos logística inversa	Estudio Delphi
Fernández Quesada	2004	Análisis empresas	Focus Group-Estudio de casos
Du	2006	Problema minimización costes servicio post-venta	Método simplex dual y búsqueda dispersa
Soto Zuloaga	2006	Devoluciones industria editorial	Algoritmo ramificación y podaLINGO Heurístico búsqueda local
Ortega Mier	2008	Problema localización instalaciones y minimización costes	Algoritmos genéticosbúsqueda dispersa
Aidoris	2009	Gestión de residuos de la construcción y demolición	Programación matemática multi-objetivo, maximizar y minimizar
López Parada	2010	Búsqueda patrones comportamiento generalizar resultados	Método caso múltiple
Eustos Flores	2010	Problema logística inversa adquisición insumos	Modelo gestión logística inversa
Miha Ramirez	2010	Problema relación logística inversa y creación conocimiento	Métodos medida relación
Benedicto Benet	2010	Problema refabricación y almacenaje logística inversa	Método optimización de coste de un modelo con demanda y retorno aleatorio de producto
Saibani	2010	Impacto productos retorno en logística inversa	Modelo medición de 3 niveles en logística inversa
Castillo Diaz	2011	Problema productos desechados secto vitivinícola	Metodología selección múltiples criterios y maximizar o minimizar objetivos
Vercaeren	2012	Problemas de reciclado y reutilización de residuos con una capacidad de producción limitada	Método heurístico
Jiménez Perra	2014	Problema comportamiento comprador portátiles refabricados	Implantación estrategias marketing

En la Tabla 3.1 se presenta un resumen por orden cronológico de todas las tesis tanto nacionales como internacionales, realizadas entre 1995 y 2015 sobre logística inversa. Se especifica tipo de problema y método de resolución.

De todas las tesis internacionales referentes a logística inversa, solo Kongar (2003), Du (2006) y Vercraene (2012) utilizan metaheurísticas para resolver modelos matemáticos de minimización de costes, pero no plantean la optimización en base a metaheurísticos con menor coste computacional como la técnica PSO propuesta en esta tesis.

3.3.2. Artículos, conferencias y congresos de ámbito internacional sobre la logística inversa y metaheurísticos

En cuanto a artículos, se han revisado más de 2500 sobre logística inversa en revistas de investigación como “*European Journal of Operational Research*”, o utilizados para congresos o conferencias a nivel internacional y publicados por editoriales de prestigio mundial como Springer, Elsevier, o la librería digital de IEEE Xplore. A continuación se analizan algunos de ellos que al igual que ésta tesis, utilizan modelos matemáticos lineales y metaheurísticas para resolverlos.

En el artículo de Pishvae (2010) se propone un modelo de programación lineal entera mixta, para minimizar los costes de transporte y la apertura/cierre de instalaciones en una red logística inversa de varias etapas. Resuelve el modelo mediante algoritmos de recocido simulado (SA) con mecanismos de búsqueda local dinámica. El algoritmo SA es programado en Matlab 7.0 y comparado con los resultados obtenidos con el software LINGO 8.0.

En la conferencia de Qin, X. H. (2011) se presentó un modelo de optimización dinámica para un sistema multi-producto, multi-periodo, con capacidad limitada y posibilidad de programación limitada para resolver el problema de diseño de fabricación, con redes de logística inversa/directa integradas. La optimización fue resuelta mediante GA combinados.

En el artículo de Zhou et al. (2012) se propone el uso de un algoritmo de PSO hibridizado, con el uso de operadores de mutación y cruce de un algoritmo GA para actualizar la población. Se pretende minimizar costes en redes de re-fabricación en cadenas de suministro de ciclo cerrado (logística directa e inversa). Los resultados de simulación obtenidos con el algoritmo propuesto presentan una solución óptima global con buen rendimiento convergente y rapidez y con una velocidad de computación, más corta que la de otros algoritmos de optimización.

En la conferencia de Xiwang et al. (2012) se analizan los modelos de desensamble selectivo multi-objetivo de productos de re-fabricación, donde el coste de mano de obra es caro. Mediante algoritmos SS se mejoran los tiempos de desensamble. Los resultados se comparan con algoritmos genéticos multi-objetivos del tipo NSGAI (non-dominated sorting genetic algorithm).

Kassem y Chen (2013) en su artículo estudian el problema de rutas para vehículos con recogida y entrega simultánea, para la optimización de la red logística de ciclo cerrado. Resuelven que el cliente debe ser servido en intervalos de tiempo definidos y aplicados como restricciones, en el modelo de programación lineal entera mixta propuesto. La optimización se alcanza utilizando algoritmos metaheurísticos SA y heurísticos de búsqueda local. Comparan la bondad de los resultados, con los obtenidos con métodos exactos mediante el software LINGO.

En el artículo de Ghezavati y Saadati (2014) se desarrolla un modelo de programación no lineal entera mixta para una red logística inversa de tres etapas, para optimizar el número y la ubicación de los centros de recogida, inspección y recuperación, así como la frecuencia de recogida con el objetivo de minimizar los costos totales, incluyendo los gastos de envío y los costes fijos de la apertura de las instalaciones. Se proponen como solución dos algoritmos: GA y SA. La comparativa de ambos algoritmos indica un mejor comportamiento en general de SA. Se recomienda el uso de nuevos heurísticos en investigaciones futuras.

Finalmente en el artículo de Ismail y Altiparmak (2015) estudian el problema de la búsqueda de localizaciones de depósitos y el diseño de rutas de

vehículos, de forma que la recogida y entrega a cada cliente se debe realizar con el mismo vehículo para optimizar el coste total. Es un problema NP-duro, para el que se propone un *algoritmo memético* para resolverlo. Para evaluar el desempeño del enfoque propuesto se comparan sus resultados con los valores más bajos obtenidos por el algoritmo de ramificación y corte en un conjunto de casos. Los resultados computacionales indican que el enfoque propuesto es capaz de encontrar soluciones óptimas o con muy buena calidad en un tiempo de cálculo razonable.

En la Tabla 3.2 se presenta un resumen por orden cronológico de todos los artículos y conferencias realizadas entre 2010 y 2015 sobre logística inversa y que incluyen problemas de optimización con metaheurísticas. Se especifica tipo de problema, método de resolución y modelo matemático utilizado. La principal relevancia de ésta tesis es que se resuelve un problema real de gestión de RSU para 90 municipios de la provincia de Málaga.

Tabla 3.2: Resumen de artículos y conferencias sobre logística inversa y metaheurísticas. *Fuente: Elaboración propia.*

Artículos y conferencias sobre logística inversa				
Autor	Año de publicación	Tipo de problema	Método de solución	Modelo
Pishvaei	2010	Minimizar los costes de transporte y apertura/cierre de instalaciones	Recocido simulado+heurísticos búsqueda local. LINGO	Programación lineal entera mixta
Qin	2011	Optimización dinámica sistema multi-producto, multi-periodo, capacidad limitada	Algoritmos genéticos combinados	Programación lineal entera mixta
Zhou et al.	2012	Problema minimización costes redes re-fabricación	Partículas inteligentes (PSO) hibridado algoritmo genético	Programación lineal entera mixta
Xiwang et al.	2012	Problemas minimizar tiempos desensamble selectivo multi-objetivo re-fabricación	Algoritmos búsqueda dispersa+algoritmos genéticos multi-objetivo NSGAII	Programación lineal entera mixta
Kassem y Chen	2013	Optimización rutas para vehículos con recogida y entrega simultánea	Recocido simulado (SA) y heurísticos de búsqueda local/LINGO	Programación lineal entera mixta
Ghezavati y Saadati	2014	Optimizar número y ubicación centros recogida, inspección y recuperación	Algoritmo genético (GA) y el recocido simulado (SA)	Programación no lineal entera mixta
Ismail y Altıparmak	2015	Problema de la búsqueda de localizaciones de depósitos y el diseño de rutas de vehículos	Algoritmo memético	Programación lineal entera mixta

3.4. Conclusiones

Según se ha visto en este capítulo existen bastantes tesis, libros, conferencias y artículos de investigación sobre teoría de localización de instalaciones o minimización de costes, pero son pocas las que se han encontrado que incluyan esta teoría aplicada a la logística inversa. Así, en la revisión sobre el estado del arte de las distintas tesis nacionales e internacionales se han encontrado muy pocas que propongan, dentro el marco de la logística inversa, modelos cuantitativos de localización de instalaciones y uso de técnicas metaheurísticas para resolver los problemas NP-duro planteados. Lo que más ampliamente se ha desarrollado en la literatura de la logística inversa hace referencia a la planificación de la producción, al control de inventario y la refabricación.

Por otra parte, en la revisión de los últimos años sobre el estado del arte de artículos publicados sobre logística inversa y metaheurísticas en revistas de investigación como “*European Journal of Operational Research*”, o utilizados para congresos o conferencias a nivel internacional y publicados por editoriales como Springer, Elsevier, ACM o IEEE Xplore, la mayoría tratan de resolver problemas de programación lineal entera mixta utilizando las clásicas técnicas metaheurísticas como AG, TS ó SA. Solo Zhou et al. (2012) proponen una técnica PSO, pero hibridizada con GA.

Todo lo anteriormente expuesto permite concluir la novedosa contribución de esta tesis al usar un algoritmo metaheurístico como es el PSO, que permite resolver un problema real de minimización de costes y localización de instalaciones en la logística inversa de RSU de la provincia de Málaga.

Los resultados son comparados con uno de los algoritmos clásicos y quizás de los más utilizados en la literatura de metaheurísticos, como es el GA. Los resultados demuestran un PSO más eficiente y con menor tiempo de cálculo que GA.

En el siguiente capítulo se proponen los modelos matemáticos diseñados mediante programación lineal entera-mixta para la minimización de coste en la localización de instalaciones, así como distintas técnicas metaheurísticas para resolver dicha programación en un tiempo computacional aceptable.

Capítulo 4

4. LOCALIZACIÓN DE INSTALACIONES Y METODOLOGÍA METAHEURÍSTICA EN LA LOGÍSTICA INVERSA DE RSU

En este capítulo se analizan las distintas soluciones a los problemas de optimización y localización de los centros de transferencias y las plantas de tratamiento para la valorización y reciclado de todos los RSU a nivel municipal.

4.1. Introducción

La mayoría de los problemas de localización de instalaciones vistos en la literatura se refieren a plantas de fabricación, ensamble y almacenes. También es aplicable en la toma de decisiones para determinar la localización de oficinas, hospitales, colegios, etc. En todo caso la calidad de un servicio depende de la localización de una instalación respecto de otras.

Klose y Drexl (2005) establecen tres modelos de localización de instalaciones: los modelos de localización continuos, los modelos de localización de redes de trabajo y los modelos de programación entera-mixta.

Dependiendo del modelo tratan en cada caso los problemas de localización y asignación con formulaciones que varían en complejidad desde la simple lineal, de una sola etapa, modelos deterministas o estocásticos no lineales.

Los modelos de localización continuos han sido investigados por diferentes autores como Hamacher y Nickel (1994), Käfer y Nickel (2001) y Klamroth (2001), o con propuestas de algoritmos heurísticos rápidos como las de Taillard (1996), Hansen et al. (1998) y Brimberg et al. (2000). Son modelos en el plano, cuyo espacio de solución es continuo, es decir requieren calcular las coordenadas $(x, y) \in R^p \times R^p$ para p instalaciones.

En los modelos de localización de redes de trabajo, las distancias se calculan como los caminos más cortos en un grafo. Corresponde a un modelo continuo Weber multi-origen, llamado problema de la *p-mediana* (Christofides y Beasley, (1982), Hanjoul y Peeters, (1985) y Beasley, (1993)).

Klose y Drexl (2000) establecieron, dentro de los modelos que utilizan una programación entera-mixta, la siguiente clasificación aproximada para modelos de localización de instalaciones discretas:

- Modelos de uno o múltiples periodos,
- Modelos con capacidad conocida o no,
- Modelos de uno o múltiples orígenes,
- Modelos de uno o múltiples productos,
- Modelos estáticos o dinámicos y
- Modelos probabilísticos.

Ortega Mier (2008) presenta como extensión de los modelos anteriores, modelos conjuntos para múltiples capacidades definidas y múltiples periodos y soluciones metaheurísticas basadas en GA y SS, a los casos planteados de optimización en procesos de fragmentación de vehículos fuera de uso y la localización de fragmentadoras con múltiples capacidades y en distintos periodos.

Esta tesis ha basado el modelo matemático de optimización y localización de los centros de transferencia y de las plantas de tratamiento de RSU, en varios de los modelos enunciados anteriormente. Es decir, la función de coste diseñada en esta tesis se basa en un modelo dinámico, de múltiples periodos, con capacidad definida y múltiples orígenes.

No se han tenido en cuenta a la hora de asignar una localización, criterios políticos, medioambientales, de instalaciones que por su cercanía o lejanía puedan generar molestias o beneficios (vertederos, centros transferencia). El principal criterio, ha sido la reducción del coste en el traslado de residuos desde distintos orígenes (municipios) hasta los destinos finales (plantas de tratamiento y centros de transferencia), contemplando la apertura o cierre de dichos destinos.

4.2. Modelo matemático de optimización de costes y localización de instalaciones

El modelo matemático planteado pretende optimizar la localización de los centros de transferencia y las plantas de tratamiento en el ámbito municipal de las distintas provincias españolas. Como caso práctico, en el capítulo 6 se aplicó el modelo matemático diseñado para la identificación y localización de las instalaciones de gestión de RSU de 90 municipios de la provincia de Málaga y se contrastarán con los datos facilitados por la Diputación de Málaga y la empresa municipal de limpieza y gestión de residuos de la capital malagueña: LIMASA.

En primer lugar, para definir el problema de optimización combinatoria se debe considerar que la cantidad de residuos generados y transportados desde diferentes municipios a los centros asignados puede variar a lo largo de distintos periodos de tiempo, dificultando este tipo de modelo dinámico la resolución del problema. En este entorno es más difícil estimar con precisión la evolución de los valores de todos los datos y parámetros de los modelos correspondientes. Este es el caso por ejemplo de la variación en el precio de los combustibles o de la distinta cantidad de residuo generado y transportado a lo largo de los años.

Como ya se ha visto del análisis del estado de la cuestión, en la literatura sobre localización de instalaciones existen distintos modelos matemáticos de optimización que facilitan la toma de decisiones a ese respecto, aunque en general están planteados para modelos estáticos y deterministas.

En el siguiente apartado y dentro del sistema de logística inversa de RSU planteado se va a diseñar la función objetivo de tipo determinista, así como las restricciones que delimitan el área factible de búsqueda del óptimo de coste mínimo y la localización de las distintas instalaciones que permitan cumplir con ese mínimo coste.

4.3. Problema de localización de plantas de tratamiento y centros de transferencia

El planteamiento del problema de localización de instalaciones de esta tesis se corresponde con algunos de los modelos incluidos en la clasificación dada por Klose y Drexl (2000) y las aportaciones de Ortega Mier (2008).

Es decir, la función de coste diseñada en esta tesis se basa en un modelo dinámico, de múltiples periodos, con capacidad definida y múltiples orígenes, correspondiente a un problema de optimización combinatoria, en el que se pretende minimizar los costes totales incluyendo los costes operativos de trasladar la producción de RSU generada en los distintos municipios, a los centros de transferencias y las plantas de tratamiento para su valorización, obtención de *compost*, etc., así como los costes fijos derivados de la ubicación o no, de las instalaciones.

Se consideran distintos parámetros y variables en el diseño de la función objetivo y las restricciones, para hacer factible la región convergente de búsqueda donde poder encontrar el óptimo global.

4.3.1. Antecedentes

Previo a la determinación del modelo matemático de función de coste y sus restricciones necesarios para determinar el mínimo coste de gestión de residuos y su ubicación en las distintas instalaciones, se verá el proceso de recogida, transporte y tratamiento del RSU.

El proceso de recogida de RSU en España y dentro del ámbito provincial, parte de una red de intercambio de residuos entre la amplia mayoría de municipios que normalmente conforman un “Consortio Provincial de Residuos” gestionados principalmente por las Diputaciones. Diferente e independiente tratamiento tiene la gestión de los RSU en las ciudades que suelen estar gestionadas por empresas de carácter mixto, es decir con vocación pública en su gestión de recogida, transporte, tratamiento y eliminación de esos residuos, pero con estructura organizativa con participación de ayuntamientos y accionistas privados.

Los RSU son transportados desde cada uno de los municipios hasta los diferentes centros de transferencia, donde son compactados para su transporte final a las plantas de tratamiento.

Estas plantas pueden ser complejos centros de tratamiento y eliminación, que procesan todos los RSU generados por los distintos municipios y cuyo objetivo principal es maximizar el reciclado de materiales, producir bioestabilizado con la materia orgánica, generar un rechazo con el mínimo de materia orgánica y optimizar la producción de energía eléctrica.

Los residuos que llegan a las plantas, salen transformados en materias primas reciclables (plásticos, metal, madera, etc.), en productos elaborados (bioestabilizados), e incluso en energía eléctrica.

Una planta de tratamiento puede tener unas instalaciones con la siguiente composición: vertedero controlado de residuos no peligrosos, vertedero controlado de residuos inertes, planta de reciclaje y compostaje, planta automática

de clasificación de envases, triturador de voluminosos, horno crematorio de animales muertos, planta de tratamiento de lixiviados, planta de desgasificación y aprovechamiento eléctrico y planta de tratamiento de escombros.

Los residuos tratados pueden ser en algún caso, clasificados y embalados para su entrega a distintos recicladores autorizados por diferentes SIG como Ecoembes, en el caso de envases.

De las plantas de tratamiento, al igual que los centros de transferencia, solo se considerará su capacidad anual de gestión de residuos y los costes fijos, que según datos aportados por la Diputación de Málaga y LIMASA, son estimados en función de las toneladas tratadas al año.

4.3.2. Diseño del modelo matemático. Función objetivo

El planteamiento de esta tesis se basa en establecer un modelo matemático que permita optimizar todo el sistema de gestión de RSU desde que se genera y recoge en cada municipio, pasando por los distintos centros de transferencias, hasta que es tratado y valorizado en la planta de tratamiento de residuos.

Cuando se formula un modelo de optimización se persigue que sea operativo, es decir que represente la realidad y así mismo sea manejable matemáticamente. Con el desarrollo actual de la informática, con ordenadores más potentes y aplicaciones más sofisticadas se ha avanzado enormemente en la resolución de problemas más complejos matemáticamente, es decir, con mayor número de variables y restricciones, aunque como se verá en el caso real desarrollado en el capítulo 6, todavía limitado por el número de dichas variables y restricciones.

Para el diseño de la función objetivo se han utilizado algunos de los modelos incluidos en la clasificación dada por Klose y Drexler (2000), los trabajos de localización de plantas y uso de heurísticos de Marín y Pelegrin (1991) y las aportaciones de Ortega Mier (2008).

Como ya se indicó, la función objetivo se basa en un modelo dinámico, de múltiples periodos, con capacidad definida y múltiples orígenes, correspondiente a un problema de optimización combinatoria, en el que se pretende minimizar los costes totales.

El modelo diseñado parte de un planteamiento general de modelo básico de minimización de costes totales y localización de instalaciones (4.3.2.1) tratado por los tres autores citados anteriormente considerando un solo periodo y sin limitación de capacidad. En este modelo básico, solo se consideran dos restricciones (4.3.2.2) y (4.3.2.3), donde j representa el conjunto de posibles localizaciones para las instalaciones candidatas, i el conjunto de municipios siendo c_{ij} el coste de la transferencia de i a j , f el coste de la instalación y p el número de instalaciones. Las variables x e y toman valores 1 o 0 según se haga la transferencia o no y según se haya seleccionado una instalación o no. El modelo es el siguiente:

$$\text{Minimizar } \sum_j y_j f + \sum_{ij} x_{ij} c_{ij} \quad j=1, \dots, m, \quad i=1, \dots, n \quad (4.3.2.1)$$

Sujeto a:

$$\sum_{ij} x_{ij} = 1 \quad i=1, \dots, n \quad (4.3.2.2)$$

$$\sum_j y_j = p \quad j=1, \dots, m \quad (4.2.2.3)$$

y con los límites:

$$x_{ij} \leq y_j \quad j=1, \dots, m, \quad i=1, \dots, n \quad (4.3.2.4)$$

$$x_{ij}, y_j \in \{0, 1\} \quad j=1, \dots, m, \quad i=1, \dots, n \quad (4.3.2.5)$$

A partir de ese modelo básico y otros de los clasificados en Klose y Drexler (2000), y considerando los parámetros y variables incorporados para definir la red de recogida y tratamiento de residuos se diseña el modelo planteado en esta tesis y que puede ser de utilidad tanto para aquellas provincias que tengan que definir una nueva estrategia de redes de gestión de residuos encontrando los trayectos

óptimos de los transportes en función de las localizaciones idóneas de las instalaciones de transferencias y tratamiento de los residuos, como para aquellas otras provincias, que teniendo establecida su red de recogida de los residuos, quieran mejorar el coste de su gestión, lo que conllevaría un ahorro económico importante por el ahorro de combustible, de mantenimiento del parque de vehículos por un menor kilometraje y finalmente por la considerable disminución de emisión de CO_2 a la atmósfera, contribuyendo a la sostenibilidad medioambiental indicada en la actual legislación de residuos (Ley 22/2011).

Con idea de facilitar el modelo matemático utilizado en esta tesis, así como las restricciones que delimitarán el área convexa de búsqueda del coste óptimo o su valor aproximado (al trabajar con metaheurísticas), solo se tendrán en cuenta los trayectos entre los distintos municipios, sin considerar los recorridos que dentro del municipio tienen que realizar los vehículos de recogida de contenedores, soterrados o no, de envases, orgánicos, papel y cartón o vidrio.

Los centros de transferencias solo se considerarán como destino de municipios y origen en el transporte de residuos a las plantas de tratamiento y cuyos parámetros utilizados para el cálculo de costes, serán solo su capacidad y costes fijos.

La mejor trayectoria se obtendrá resolviendo el problema de identificación y localización de las instalaciones de residuos, según el modelo matemático de minimización de costes que se va a diseñar en esta tesis.

En la Figura 4.1 se muestra un ejemplo de cómo sería el trayecto para el traslado de los residuos desde el municipio de Álora hasta la planta de tratamiento situada en Antequera o el centro de transferencia de Ronda para su compactación y posterior traslado a la planta.

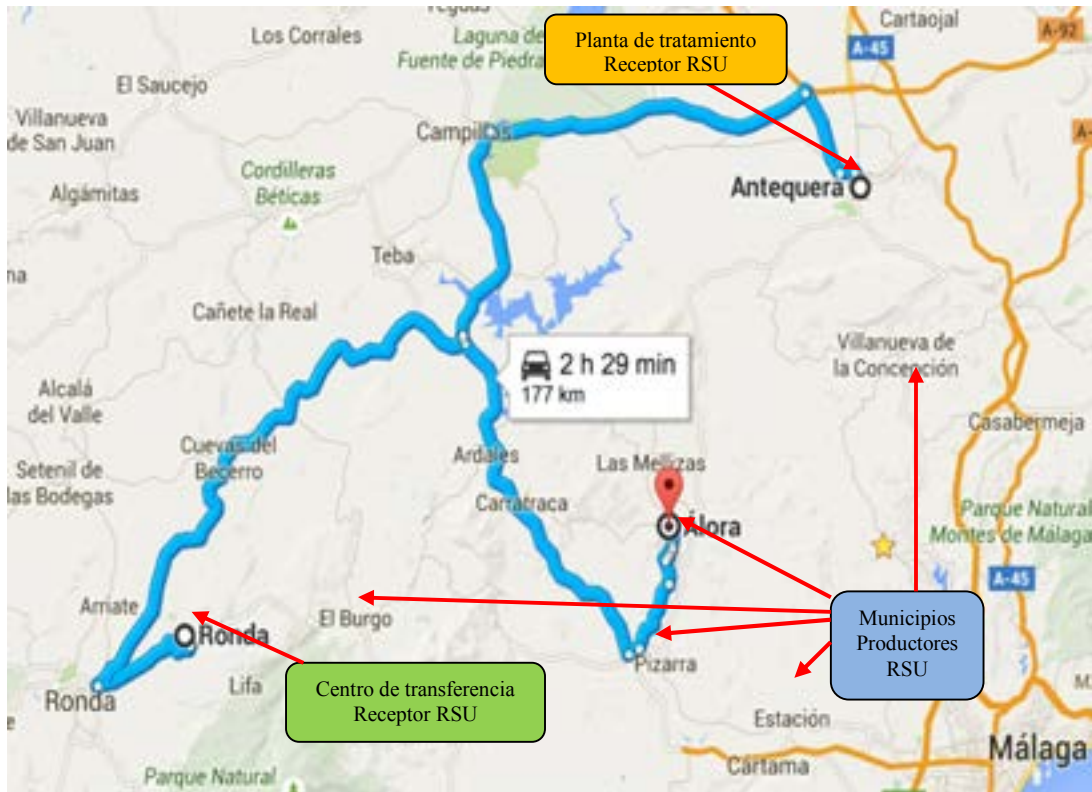


Figura 4.1: Transferencia de residuos desde municipios a centro de transferencia y planta de tratamiento. *Fuente: Elaboración propia.*

El diseño de modelo de la función coste tiene en cuenta las siguientes consideraciones:

5. Se considera el mismo coste de traslado de los residuos entre municipios y centros de transferencia, independientemente de que haya municipios que utilicen vehículos con mayor capacidad, consumo de combustible y coste, respecto de otros que utilicen vehículos más económicos.
6. No se consideran los recorridos para la recogida de los distintos contenedores dentro de cada municipio. Cada municipio, a efectos de cálculo de distancias, es considerado como punto de origen y final de trayectoria.
7. Se tienen en cuenta solo los costes fijos y capacidad de los centros de transferencia y de las plantas de tratamiento. Estos costes se calculan

en función de las toneladas de residuos recibidas. Dichas instalaciones pueden ser localizadas en cualquier municipio, aunque en el caso de las plantas de tratamiento se parte de una primera hipótesis con dos plantas situadas en cualquier municipio, y una segunda opción considerando un número indefinido de plantas de tratamiento.

8. Se consideran cinco periodos (en años) pudiendo variar las localizaciones de los centros de transferencias y las plantas de tratamiento debido a cambios en los costes unitarios de transporte/año y la cantidad de residuo generado en cada periodo.

En las siguientes Tablas 4.1, 4.2 y 4.3 se indican los índices, parámetros y variables utilizadas en el modelo matemático propuesto. El rango M corresponde al número máximo de municipios generadores de residuo, N es el número máximo de centros de transferencia que se pueden instalar (1 por municipio), P las plantas de tratamiento a considerar y T el número máximo de años a analizar.

Tabla 4.1: Índices del modelo matemático.
Fuente: *Elaboración propia.*

Índice	Descripción	Rango
i	Municipios productores de RSU	$1...M$
j	Centros de transferencia receptores RSU	$1...N$
k	Plantas tratamiento receptoras RSU	$1...P$
t	Periodos en años	$1...T$

Tabla 4.2: Parámetros del modelo matemático.
Fuente: Elaboración propia.

Parámetros	Descripción
P_{it}	Producción de residuo del municipio i en el año t (en T)
d_{ij}	Distancia entre los municipios i y j (km)
d_{jk}	Distancia entre los municipios j y k (km)
CTP_t	Coste unitario de transferencia del residuo entre un centro de transferencia y la planta de tratamiento ($\text{€}/\text{km} \cdot T$) en el año t
CMT_t	Coste unitario de transporte del residuo recogido en cada municipio hasta el centro de transferencia ($\text{€}/\text{km} \cdot T$) en el año t
QT_{jt}	Capacidad anual de un centro de transferencia j (en T) en el año t
QP_{kt}	Capacidad anual de la planta de tratamiento k (en T) en el año t
CF_{jt}	Costes fijos centro transferencia j (€) en el año t
CF_{kt}	Costes fijos planta tratamiento k (€) en el año t

Tabla 4.3: Variables del modelo matemático.
Fuente: Elaboración propia.

Variable	Descripción	Tipo
Z_{kt}	1 si la planta de tratamiento está localizada en el municipio k durante el año t , 0 si no	Binaria
V_{jt}	1 si el centro de transferencia del municipio j está abierto durante el año t , 0 si no	Binaria
X_{ijt}	Cantidad de residuo transportado desde el municipio i al centro de transferencia j durante el año t	Entera
Y_{jkt}	Cantidad de residuo transportado desde el centro de transferencia que está en el municipio j a la planta (en el municipio k) durante el año t	Entera

Una vez definidos los índices, parámetros y variables se puede expresar la función objetivo como:

$$F_0 = \min (F_x + F_y + F_v + F_z) \quad (4.3.2.1)$$

siendo:

$$F_x = \sum_{ijt} x_{ijt} d_{ij} C_{MTi} \quad (4.3.2.2)$$

$$F_y = \sum_{jkt} y_{jkt} d_{jk} C_{TPj} \quad (4.3.2.3)$$

$$F_v = \sum_{jt} C_{Fj} v_{jt} \quad (4.3.2.4)$$

$$F_z = \sum_{kt} C_{Fk} z_{kt} \quad (4.3.2.5)$$

Sujeta a las siguientes restricciones (R1...R7):

$$R1: \sum_j -x_{ijt} \leq -P_{it} \quad \forall i, \forall t \quad (4.3.2.6)$$

$$R2: x_{ijt} \leq P_{it} v_{jt} \quad \forall i, \forall j, \forall t \quad (4.3.2.7)$$

$$R3: \sum_{ij} x_{ijt} \leq \sum_j Q_{Tjt} v_{jt} \quad \forall t \quad (4.3.2.8)$$

$$R4: \sum_k y_{jkt} \leq \sum_i x_{ijt} \quad \forall j, \forall t \quad (4.3.2.9)$$

$$R5: y_{jkt} \leq z_k Q_{Pkt} \quad \forall j, \forall k, \forall t \quad (4.3.2.10)$$

$$R6: \sum_k -z_{kt} \leq -1 \text{ a } -90 \quad \forall t \quad (4.3.2.11)$$

$$R7: \sum_{ij} x_{ijt} \leq \sum_k Q_{Pkt} \quad \forall t \quad (4.3.2.12)$$

Con los siguientes límites en las variables:

$$x_{ijt} \geq 0 \quad \forall i, \forall j, \forall t \quad (4.3.2.13)$$

$$y_{jkt} \geq 0 \quad \forall j, \forall k, \forall t \quad (4.3.2.14)$$

$$v_{jt} \in \{0, 1\} \quad \forall j, \forall t \quad (4.3.2.15)$$

$$z_k \in \{0, 1\} \quad \forall k \quad (4.3.2.16)$$

El diseño de la función coste (4.3.2.1) queda definido por la optimización (minimización) de la suma de los costes del transporte de todos los RSU (4.3.2.2) recogidos en cada municipio hasta los centros de transferencia, más los costes de transporte (4.3.2.3) desde cada centro de transferencia hasta las plantas de tratamiento, más los costes fijos de los centros de transferencia y plantas de tratamiento (4.3.2.4 y 4.3.2.5) considerando que dichas instalaciones pueden estar abiertas o cerradas en distintos periodos de tiempo (se han contemplado 5 años).

Los costes de transporte de los RSU (4.3.2.2) desde los municipios a los centros de transferencia se han calculado como la suma de todos los residuos generados en cada municipio considerando las distancias entre municipios, el coste unitario del transporte por tonelada de residuo y el horizonte de periodos considerados.

Los costes de transporte de los RSU (4.3.2.3) desde los centros de transferencia hasta las plantas de tratamiento se han calculado como la suma de todos los residuos recogidos y compactados de cada municipio considerando las distancias entre municipios (puesto que tanto los centros como las plantas se pondrían en cualquier municipio), el coste unitario del transporte por tonelada de residuo entre centro y planta, considerando los distintos periodos de tiempo.

Los costes fijos de los centros de transferencia (4.3.2.4) y de las plantas de tratamiento se han calculado considerando que las instalaciones pueden estar abiertas o cerradas (las variables binarias incluidas en la función definen la apertura o cierre de cada instalación), durante periodos con duración mínima de un año. Es decir, si una instalación está abierta (o cerrada), esta apertura (o cierre) se mantendrá durante el año en curso.

Se han definido siete restricciones que permiten acotar la búsqueda del óptimo coste a un espacio convexo más reducido y con menor coste computacional.

La primera restricción (4.3.2.6) indica que se deben recoger todos los RSU generados en cada municipio y en cada periodo de tiempo. Los signos negativos en variables y parámetros definen la búsqueda del mínimo coste sobre una región convexa.

La restricción (4.3.2.7), que se podría considerar como ampliación de la primera, garantiza que los residuos recogidos en cada municipio no excedan la capacidad de los centros de transferencia abiertos en los periodos de tiempo considerados. Facilita la búsqueda óptima de las instalaciones de transferencia que deben estar abiertas.

Las restricciones (4.3.2.8), (4.3.2.10) y (4.3.2.12) garantizan que los residuos transportados entre municipios, centros de transferencia y plantas de tratamiento, no excedan la capacidad de las instalaciones abiertas durante el periodo considerado.

Con la restricción (4.3.2.9) se pretende garantizar el flujo de vehículos que transportan los residuos entre los distintos municipios, los centros de transferencias y las plantas de tratamiento, sin sobrepasar la capacidad de generación de residuos de cada municipio.

Finalmente, la restricción (4.3.2.11) se establece para garantizar un número mínimo de plantas de tratamiento, entre 1 y 90 (número total de municipios que se utilizarán es el estudio de un caso real). Normalmente los Ayuntamientos y Diputaciones establecen la cantidad de plantas dependiendo del ratio habitantes/municipio y su capacidad generadora de RSU. En el caso real analizado en el capítulo siguiente, la Diputación malagueña estableció 2 plantas de tratamiento y eliminación de residuos para un total de 90 municipios.

Las restricciones (4.3.2.13) y (4.3.2.14) sirven para limitar la búsqueda de los valores de las variables continuas, dentro de valores positivos y por simplificación de cálculos a valores enteros positivos. Las restricciones (4.3.2.15) y (4.3.2.16) solo tomarán los valores binarios 0 o 1 (indicando si la instalación está abierta o cerrada).

El problema planteado por la función objetivo y sus restricciones es un problema determinístico de programación lineal con variables mixtas, es decir, con variables enteras y binarias, en el que la búsqueda de soluciones se hace más complicada conforme aumentan las variables introducidas siendo un problema del tipo NP-duro, donde el mejor criterio de solución pasa por la aplicación de metaheurísticos, ya que los métodos exactos son por su coste computacional, totalmente inviables.

Si se aplica el modelo de función coste diseñado al problema real de 90 municipios y considerando los distintos casos en función del número de municipios y periodos, se comprobará que para un cierto número de variables y restricciones los métodos exactos son inviables en cuanto a tiempo y requisitos computacionales.

Según se puede deducir de la función objetivo y sus restricciones, el número de variables crece según: $(1+Pt+Mu) \cdot Ct \cdot Ti+Pt$, siendo:

Pt: nº. de plantas de tratamiento

Mu: nº. de municipios

Ct: nº. de centros de transferencia

Ti: nº. de periodos (1 a 5 años)

En la Tabla 4.4 se pueden observar el número de variables y restricciones necesarias para problemas con distintos número de municipios y periodos.

Tabla 4.4: Variables y restricciones por nº. de municipios y periodos
Fuente: Elaboración propia.

Municipios	Periodos	Nº. Variables	Nº. Restricciones
6	5	84	92
20	5	4200	4220
40	5	16400	16420
90	5	81900	81920

Este modelo corresponde a un problema NP-duro, sencillo de resolver para problemas pequeños, pero con un gran consumo de tiempo cuando el problema crece como se puede ver en el número de variables y restricciones.

Es un problema de programación lineal con variables mixtas (enteras y binarias) que no es fácil de resolver sobre todo por la cantidad de variables enteras y binarias. Cuando el cálculo de una solución óptima no es posible, se considera suficiente obtener soluciones factibles que sean lo suficientemente buenas mediante el uso de soluciones no exactas (metaheurísticos).

4.4. Metodología

En esta sección se analizan los métodos utilizados en la literatura, que servirán para definir la metodología aplicable al problema planteado en la sección anterior. Esta revisión y considerando la dificultad añadida de resolver problemas lineales con un elevado número de variables, condicionan la propuesta metodológica al uso de diferentes técnicas metaheurísticas.

4.4.1. Metodos de resolución

En este apartado se definen los métodos de resolución más adecuados, para resolver el problema de optimización planteado en la sección anterior.

La Investigación Operativa es una ciencia moderna interdisciplinar que mediante métodos y técnicas especiales busca la solución óptima de problemas complejos. Para hallar la solución se suele representar el problema como un modelo matemático. Aunque algunos historiadores consideran que esta ciencia comenzó con Arquímedes en el siglo III A.C., parece aceptarse entre la comunidad científica que el inicio de la investigación de operaciones comenzó en 1940, con la organización de un grupo de 20 investigadores (“Circo de Blackett”), bajo la dirección de P.M.S. Blackett.

La optimización es una parte importante de la Investigación Operativa. Algunas técnicas como la Programación lineal o la Programación dinámica son anteriores a 1960: el Método Simplex de Dantzig (1947), que desarrollo en otras obras de (1955) y (1963) o el Principio de optimalidad de Bellman (1957).

Actualmente, la Investigación Operativa incluye una gran cantidad de métodos de resolución de problemas de optimización que suelen acometerse mediante dos técnicas: analíticas y numéricas. Para problemas de reducida dimensión suelen utilizarse las primeras, pero en problemas reales son las técnicas numéricas las que se utilizan para encontrar la solución. Estas técnicas están basadas en el empleo de algoritmos.

Son muchos los métodos que pueden aplicarse en la búsqueda de buenos resultados. El estudio de estos ha ido evolucionando desde el uso de algoritmos exactos hacia heurísticos y metaheurísticas finalmente.

Las familias de algoritmos más eficientes y que mejor comportamiento demuestran para problemas de programación lineal son las variantes diversas del Método Simplex. Sin embargo, este tipo de método no es efectivo o viable cuando se pretende encontrar buenas soluciones factibles de manera rápida a los problemas de optimización denominados NP-duro, como es el caso del problema planteado en esta tesis.

Igual ocurre con otros algoritmos exactos. Algunos de estos métodos se han aplicado con éxito, como los algoritmos voraces, algoritmos de divide y vencerás, de ramificación y poda, *backtracking*. También, varios autores han utilizado los métodos exactos en problemas de localización como Ballou (1968), Erlenkotter (1975) o Shulman (1991). Sin embargo, esta técnica no resulta útil en problemas con un elevado número de variables. Es decir, para los denominados problemas NP no se conocen algoritmos exactos con convergencia en tiempo polinómico.

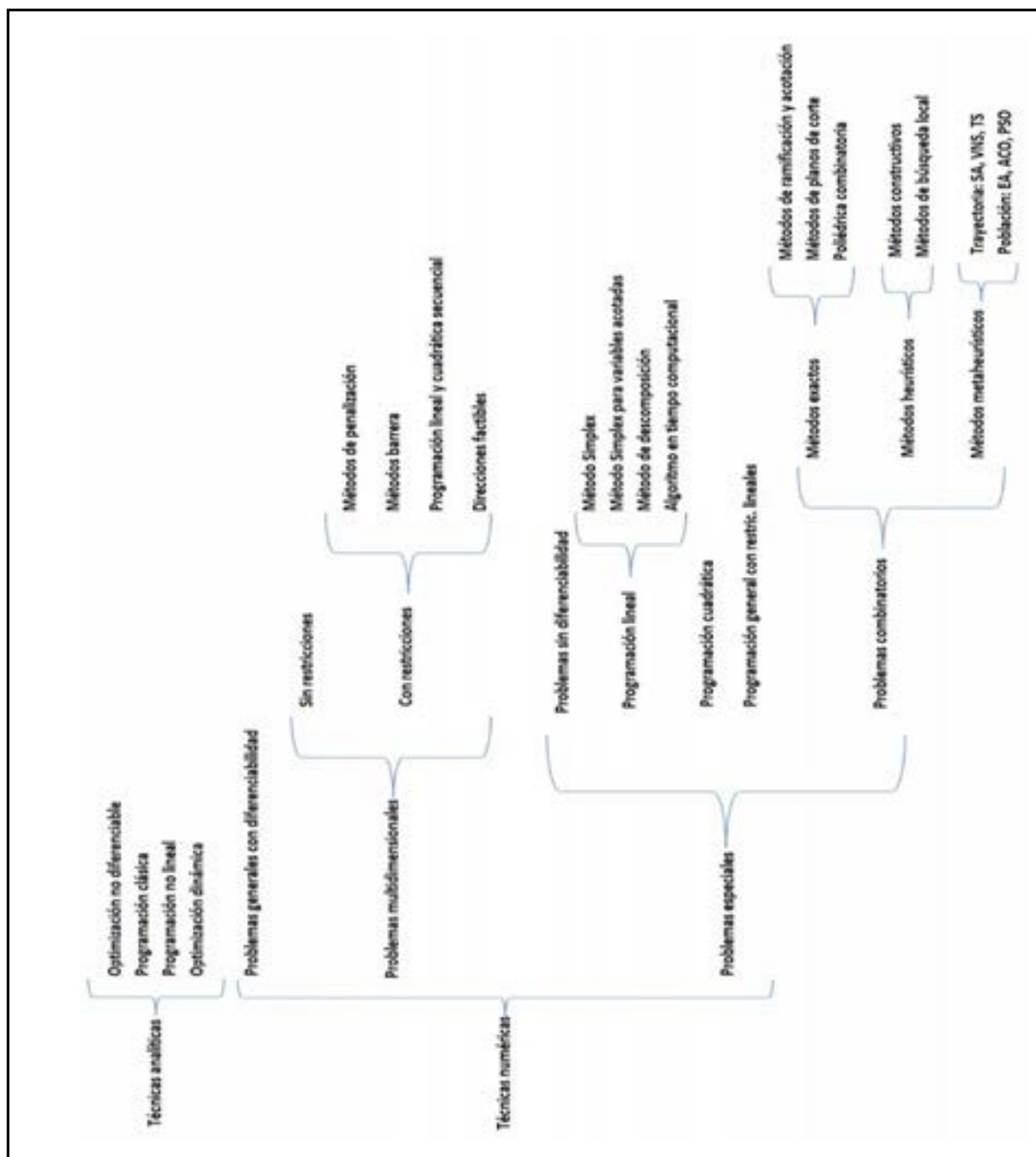


Figura 4.2: Esquema de distintos métodos de resolución, agrupados por técnica y tipo de problema. *Fuente: Elaboración propia.*

En la Figura 4.2 se expone un breve esquema con los distintos métodos de resolución agrupados por técnica y tipo de problema, mostrando lo amplio del campo de los métodos de resolución de problemas de optimización.

Finalmente, se encuentran los algoritmos heurísticos, término empleado por primera vez, desde el punto de vista científico, por el matemático Pólya

(1945) que proporcionan soluciones a un problema próximas a la óptima pero sin garantía de optimalidad, aunque con un tiempo de ejecución mucho menor. Los algoritmos metaheurísticos suelen imitar fenómenos simples observados en la naturaleza, y se consideran asociados con la inteligencia artificial. Los heurísticos clásicos suelen buscar y garantizar un óptimo local, mientras que los métodos metaheurísticos están diseñados para alcanzar, aunque no siempre se puede garantizar, un óptimo global.

Como ya se indicó en el capítulo donde se revisaba el estado de la cuestión, son varios los autores que han utilizado estos métodos metaheurísticos para problemas de localización y en logística inversa como Antunes y Peeters (2000), Bautista y Pereira (2006) o Min et al. (2006).

En el siguiente capítulo se analizan y describen algunos de los algoritmos metaheurísticos más utilizados en la literatura y se proponen dos de ellos como la metodología a utilizar en la búsqueda a los problemas de localización y minimización de costes planteados en esta tesis.

4.5. Conclusiones

En este capítulo y partiendo de una revisión previa de la literatura se ha desarrollado el modelo matemático que servirá para analizar y minimizar los costes en los problemas de localización en la gestión de RSU. Se ha concluido que debido al número de variables tanto enteras como binarias y a las restricciones que se necesitarían aplicar, el modelo diseñado sería muy difícil de resolver o de muy alto coste computacional con métodos exactos para los modelos lineales planteados, convirtiéndose en un problema del tipo NP-duro, cuya resolución solo sería factible con métodos aproximados como son las técnicas metaheurísticas.

En la sección anterior se han indicado los métodos de optimización cuantitativos, y técnicas metaheurísticas utilizadas en la investigación operativa actual.

De estos métodos, se proponen los metaheurísticos como la mejor alternativa para resolver de forma rápida los problemas de localización y minimización de costes en logística inversa de gestión de RSU, y que resultarían irresolubles utilizando otros métodos.

Las técnicas metaheurísticas más utilizadas y sus aplicaciones se describen en el capítulo siguiente. La mayoría de los metaheurísticos han sido usados para resolver problemas estáticos como son: el algoritmo TS, seguido de SA y por último los algoritmos GA.

De las técnicas metaheurísticas existentes en la literatura para probar la validez del enfoque propuesto, se han escogido dos, una de reciente incorporación en su uso por parte de científicos e investigadores y no utilizada hasta ahora para resolver problemas dinámicos en logística inversa según se pudo concluir en el capítulo anterior y la otra, un metaheurístico muy utilizado en distintas aplicaciones y aceptado en la comunidad científica. El hecho de utilizar las dos técnicas indicadas permitirá comparar resultados y acreditar la validez del nuevo metaheurístico propuesto.

Capítulo 5

5. TÉCNICAS METAHEURÍSTICAS UTILIZADAS

En este capítulo se hace una revisión de distintas técnicas metaheurísticas (Alba (2005)) comentando las aplicaciones encontradas en la literatura para problemas de localización. Tras la revisión se seleccionarán dos de ellas, que posteriormente se utilizarán para seguir la metodología propuesta en el capítulo previo y se aplicarán en el problema de localización de instalaciones de transferencia y tratamiento de RSU.

5.1. Introducción

La existencia de una gran cantidad y variedad de problemas difíciles, y que necesitan ser resueltos de forma eficiente, impulsó el desarrollo de procedimientos eficientes para encontrar buenas soluciones aunque no fueran óptimas. Estas técnicas, en las que la rapidez del proceso es tan importante como la calidad de la solución obtenida, se denominan metaheurísticas. Este término fue introducido por primera vez por Glover (1986). En Blum y Roli (2003) se recogen algunas definiciones de metaheurística, de entre las que se destaca la siguiente: “*Un procedimiento de generación iterativo que guía una heurística subordinada, combinando de forma inteligente diferentes conceptos para explorar y explotar el espacio de búsqueda, utilizando estrategias de aprendizaje para estructurar la información a fin de encontrar de manera eficiente soluciones casi óptimas*” (Osman y Laporte (1996)).

Las técnicas metaheurísticas proporcionan una buena solución del problema aunque no necesariamente óptima. Al contrario, los métodos exactos si proporcionan una solución óptima, si esta existe. Por otra parte, un método exacto puede consumir un tiempo excesivamente grande para encontrar la solución óptima de un problema complejo, que en muchos casos puede ser inaplicable. Los tiempos computacionales consumidos por un metaheurístico suelen ser más cortos.

El término científico NP-duro está utilizado en el contexto de la complejidad algorítmica. En esta tesis, el problema de optimización de costes y localización visto en el capítulo anterior se considera NP-duro, por el número de variables y restricciones utilizadas y que como se verá en posteriores capítulos pueden llegar a no tener solución si se aplican métodos exactos o si existen, de un tiempo computacional muy elevado.

Por tanto, aunque existen muchas razones para utilizar metaheurísticas para resolver un problema difícil, para el caso de esta tesis se destacan dos: La primera es que aunque existe un método exacto para resolver el problema, su uso es computacionalmente muy costoso y la segunda es que proporciona una buena solución de partida.

5.2. Técnicas metaheurísticas

En esta sección se revisarán algunas de las técnicas metaheurísticas más utilizadas por la comunidad científica en distintos campos y se seleccionarán dos, para resolver los problemas de optimización de costes y localización dinámica de instalaciones en el sistema de logística inversa establecido para la recogida y transporte de residuos. También se describe la aplicación de la metodología vista en el capítulo anterior a los dos métodos propuestos: los algoritmos GA y PSO.

5.2.1. Algoritmos genéticos

Los Algoritmos Genéticos o *Genetic Algorithm* (GA) son métodos adaptativos utilizados para la resolución de problemas de búsqueda y optimización, inspirados en los procesos de evolución de los seres vivos, en su esfuerzo por adaptarse al entorno en el que habitan. Los principios básicos de los GA fueron establecidos por Holland (1975), donde se propone una forma de estudiar la evolución tanto de los sistemas naturales como de los artificiales (dentro de los cuales se engloban los problemas de optimización combinatoria). Los patrones de evolución de las especies son tales que, por un lado, con el transcurso del tiempo se exploran continuamente nuevas posibilidades y, por otro, en condiciones normales raramente conducen a la obtención de individuos absolutamente desadaptados e incapaces de sobrevivir.

Estos algoritmos están incluidos en el grupo de técnicas evolutivas debido a que disponen en cada iteración de un conjunto de soluciones a partir de las cuales obtienen un nuevo conjunto de soluciones. Los GA operan de forma iterativa, de manera que cada iteración comienza con una población inicial, es decir, un conjunto inicial de soluciones, cada una de ellas representada mediante una cadena de números (típicamente, las cadenas son binarias). A veces, se habla de individuos y no de soluciones. Partiendo de dicha población se realizan manipulaciones a partir de las cuales se obtiene una nueva población. La función objetivo suele denominarse función de adaptación o aptitud y permite evaluar la bondad de las soluciones (individuos) consideradas en cada momento.

Entre las principales operaciones realizadas en cada iteración están: la selección, el cruce, la reducción y la mutación. La aplicación reiterada de los operadores anteriores, permite alcanzar soluciones con mejores funciones de aptitud.

En la literatura existen algunas aplicaciones de GA en la localización de instalaciones: Kongar (2003), Chau (2004), Bautista y Pereira (2006), Min et al. (2006), Ortega Mier (2008) y Ghezavati y Saadati (2014).

5.2.2. Recocido simulado

El recocido simulado o *Simulated Annealing* (SA) es uno de los más antiguos metaheurísticos y seguramente uno de los primeros algoritmos que utilizo una estrategia explícita para evitar los mínimos locales. Esta técnica fue formulada por Kirkpatrick et al. (1983) y cuyos orígenes vienen de la mecánica estadística que estudia los procesos de recocido del metal y el cristal, asumiendo una configuración de baja energía cuando son enfriados con un adecuado programa de enfriamiento.

SA fue presentado como un algoritmo de búsqueda de problemas de optimización combinatoria. Para evitar quedar atrapado en mínimos locales, la idea fundamental es moverse a soluciones con valores de la función objetivo que son peores que los valores actuales. En cada iteración, una solución es aleatoriamente elegida. Así, dado un sistema de N partículas que pueden adoptar diferentes configuraciones, a una determinada temperatura T la probabilidad de que el sistema adopte una determinada configuración sigue una distribución de Boltzmann dada por la expresión $e^{(-\frac{E}{k_b T})}$, donde E representa la energía de esa configuración, k_b es la constante de Boltzmann y T es la temperatura del sistema.

Metropolis et al. (1953) propusieron un modelo para simular el comportamiento del proceso de solidificación a partir de lo anterior.

En las referencias a logística inversa en la literatura utilizando algoritmos SA se ha encontrado a Antunes y Peeters (2000), Pishvaei (2010), Kassem y Chen (2013) y Ghezavati y Saadati (2014).

5.2.3. Búsqueda Tabú

La búsqueda Tabú o *Tabu Search* (TS) es uno de los más exitosos metaheurísticos para la aplicación de problemas de optimización combinatoria.

Las ideas básicas fueron introducidas por Glover (Glover, 1989, Glover, 1990, Glover y Laguna, 1997) que propuso el algoritmo TS como un procedimiento para huir de óptimos locales basado en la utilización de la información del propio proceso de búsqueda.

La idea básica de la búsqueda tabú es el uso explícito de un historial de la búsqueda (una memoria a corto plazo), tanto para escapar de los mínimos locales como para implementar su estrategia de exploración y evitar buscar varias veces en la misma región. Esta memoria a corto plazo se implementa con una lista tabú, donde se mantienen las soluciones visitadas más recientemente para excluirlas de los próximos movimientos. En cada iteración se elige la mejor solución entre las permitidas y la solución es añadida a la lista tabú. Desde el punto de vista de la implementación, mantener una lista de soluciones completas no suele ser práctico debido a su ineficiencia. Por lo tanto, en general, se suelen almacenar los movimientos que ha llevado al algoritmo a generar esa solución o los componentes principales que definen la solución. En cualquier caso, los elementos de esta lista permiten filtrar el vecindario, generando un conjunto reducido de soluciones elegibles denominado $N_a(s)$. El almacenamiento de los movimientos en vez de las soluciones completas es bastante más eficiente, pero introduce una pérdida de información. Para evitar este problema se define un criterio de aspiración que permite incluir una solución en $N_a(s)$ incluso si está prohibida debido a la lista tabú. El criterio de aspiración más ampliamente usado es permitir soluciones cuyo *fitness* sea mejor que el de la mejor solución encontrada hasta el momento.

Referencias a logística inversa en la literatura utilizando TS solo se ha encontrado en Kongar (2003).

5.2.4. GRASP

El procedimiento de búsqueda miope aleatorizado y adaptativo o *Greedy Randomized Adaptive Search Procedure* (GRASP) es un simple metaheurístico que combina heurísticos constructivos y de búsqueda local.

Fue propuesto por Feo y Resende (1995) y es un proceso iterativo compuesto de dos fases: construcción de la solución y proceso de mejora de la solución. El mecanismo de construcción de la solución es un heurístico constructivo aleatorio. Es decir, se genera una solución obtenida tras la ejecución de diferentes etapas, en cada una de las cuales se completa parcialmente la solución hasta llegar a una solución completa. En cada etapa se dispone de un conjunto de elementos candidatos para completar la solución. Estos elementos candidatos están contenidos en lo que se denomina *lista de candidatos restringida*. La lista restringida está formada por elementos que a priori pueden ofrecer buenas soluciones. De entre los elementos candidatos se elige uno de forma aleatoria. Tras la fase constructiva se modifica la lista restringida de candidatos. La segunda fase del algoritmo es un método de búsqueda local, la cual puede ser un algoritmo de búsqueda local básico como una mejora iterativa o técnicas avanzadas como SA o TS.

Es una técnica que debido a su simplicidad (requiere el ajuste de pocos parámetros, lo cual hace más sencillo su diseño) es generalmente muy rápida y para determinados problemas, ofrece buenos resultados en un tiempo computacional muy corto.

En la revisión de la literatura, sólo Bautista y Pereira (2006) resuelven un problema estático de logística inversa, de la localización de contenedores de basura en el área metropolitana de Barcelona mediante un GA combinado con un heurístico GRASP.

5.2.5. Colonia de hormigas

La optimización por colonia de hormigas o *Ant Colony Optimization* (ACO) es una propuesta de metaheurístico inspirado en el comportamiento de búsqueda de alimento de hormigas reales. Estos métodos fueron propuestos por Dorico (1992).

Esta técnica trata de imitar el comportamiento de las hormigas que, cuando se desplazan dejan a su paso un rastro de feromonas que les sirve de medio de transmisión de información entre ellas para encontrar el camino más corto entre fuentes de comida y su hormiguero.

Esta funcionalidad es explotada en colonias de hormigas artificiales con idea de resolver problemas de optimización combinatoria. En cada iteración se construye un conjunto de soluciones, de manera que en la construcción de la solución se emplea una lógica parecida a la de las hormigas al desplazarse empleando el rastro de feromonas. En términos de un problema de optimización, el rastro de feromonas depositadas por las hormigas se traduce en la probabilidad de pasar de un estado a otro determinado. Cada hormiga genera una solución mediante un método constructivo pasando por diferentes estados, donde cada estado representa una solución parcial. El estado final representa una solución completa.

En la revisión de la literatura, no se han encontrado propuestas de este tipo de algoritmo para resolver problemas de localización de instalaciones y minimización de costes en logística inversa.

5.2.6. Búsqueda dispersa

La búsqueda dispersa o Scatter Search (SS) es una metaheurística cuyos principios fueron presentados por Glover (1977) y que ha recibido una gran atención por parte de la comunidad científica (Laguna y Martí (2003)). El algoritmo se basa en mantener un conjunto relativamente pequeño de soluciones tentativas (llamado conjunto de referencia o *RefSet*) que se caracteriza por contener soluciones de calidad y diversas (distantes en el espacio de búsqueda). Para la definición completa de SS hay que concretar cinco componentes: creación de la población inicial, generación del conjunto de referencia, generación de subconjuntos de soluciones, método de combinación de soluciones y método de mejora.

En las referencias a logística inversa en la literatura utilizando la búsqueda dispersa se ha encontrado a González y Adenso Díaz (2004), Ortega Mier (2008), Du (2006) y Xiwang et al. (2012).

5.2.7. Optimización de enjambre de partículas

Es una técnica de inteligencia artificial basada en el estudio del comportamiento colectivo en sistemas descentralizados (auto-organizados). La optimización de enjambre de partículas o *Particle Swarm Optimization* (PSO) propuesta por Kennedy y Eberhart (1995) se encarga de los problemas en los que se pueden representar una solución mejor como un punto a una superficie en un espacio n-dimensional. La principal ventaja es que es muy resistente al problema local de los óptimos.

Es una metaheurística evolutiva inspirada en el comportamiento social de las bandadas de pájaros o bancos de peces. Las soluciones, llamadas partículas se “echan a volar” en el espacio de búsqueda guiadas por la partícula que mejor solución ha encontrado hasta el momento y que hace de líder de la bandada. Cada partícula evoluciona teniendo en cuenta la mejor solución encontrada en su recorrido y al líder.

El procedimiento también tiene en cuenta el mejor valor alcanzado por alguna de las partículas en su entorno. En cada iteración, las partículas modifican su velocidad hacia la mejor solución de su entorno teniendo en cuenta la información del líder.

Pese a que PSO se ha aplicado a numerosos campos, en la literatura solo Zhou et al. (2012) propone el uso de un algoritmo PSO hibridizado con GA, para minimizar costes en redes de re-fabricación en cadenas de suministro de ciclo cerrado (logística directa e inversa).

5.2.8. Búsqueda local iterada

La *Búsqueda Local Iterada* o *Iterated Local Search (ILS)* propuesta por Stützle (1999) y Lourenco et al. (2003) es una metaheurística basada en un concepto simple pero muy efectivo. En cada iteración, la solución actual es perturbada y, a esta nueva solución, se le aplica un método de búsqueda local para mejorarla. El mínimo local obtenido por el método de mejora puede ser aceptado como nueva solución actual si pasa un test de aceptación. La importancia del proceso de perturbación es obvia: si es demasiado pequeña puede que el algoritmo no sea capaz de escapar del mínimo local; por otro lado, si es demasiado grande, la perturbación puede hacer que el algoritmo sea como un método de búsqueda local con un reinicio aleatorio. Por lo tanto, el método de perturbación debe generar una nueva solución que sirva como inicio a la búsqueda local, pero que no debe estar muy lejos del actual para que no sea una solución aleatoria. El criterio de aceptación actúa como contra-balance, ya que filtra la aceptación de nuevas soluciones dependiendo de la historia de búsqueda y de las características del nuevo mínimo local.

No se han encontrado en la literatura aplicaciones ILS para la localización dinámica ni la logística inversa.

5.3. Selección del algoritmo adecuado al problema

Evidentemente, hay muchas más técnicas metaheurísticas de las mencionadas en las anteriores secciones, como las de búsqueda local guiada, de búsqueda reactiva, algoritmos meméticos, redes neuronales, multiobjetivo, e incluso “hiperheurísticas”. Esta tesis se ha extendido en aquellas que más aceptación han tenido en la comunidad científica, incluyendo algunas más recientes como es el algoritmo PSO.

Como ya se indicó en el capítulo anterior, tras estudiar los distintos métodos de resolución de problemas de optimización utilizados en Investigación

Operativa, se concluyó que para la resolución de los problemas deterministas de localización dinámica planteados en la tesis, los métodos de resolución exactos no resultan efectivos o viables cuando se pretende encontrar buenas soluciones factibles de manera rápida. La principal complicación se debe al número tan elevado de variables, que hacen impracticable la evaluación de todas sus soluciones para determinar el óptimo mediante estos métodos.

Estos los problemas complicados de optimización (denominados NP-duro) es una de las razones por la que utilizar métodos metaheurísticos sea la principal opción. Otra razón puede ser que aunque el método exacto se pueda aplicar para resolver el problema, su uso es computacionalmente muy costoso. También el que los métodos metaheurísticos son más flexibles que un método exacto.

De lo anterior y tras la revisión de la literatura confirmando como mejor opción el uso de técnicas metaheurísticas es por lo que se considera la metodología metaheurística como la más eficiente para resolver los problemas de optimización que se pretenden resolver en la tesis.

De todas las técnicas metaheurísticas descritas en la sección anterior, algunas de ellas como los algoritmos TS y SA, no son metodologías aplicables en principio, ya que son metaheurísticos que pueden caer en búsquedas locales y pueden dejar de explorar parte de la zona de factibilidad del problema.

Por eso, se han propuesto dos que realicen la búsqueda de la manera más amplia y rápida posible en el espacio de soluciones, sin caer en óptimos locales. Primera, la técnica metaheurística GA destaca por tener resultados aceptables en problemas de localización de instalaciones y optimización de costes en logística inversa. La segunda, la técnica metaheurística PSO supone el carácter innovador añadido a esta tesis, al no encontrarse artículos ni literatura en la revisión del estado del arte que usen este metaheurístico aplicado a problemas de localización en logística inversa. En las siguientes secciones se desarrollan las dos técnicas propuestas.

5.4. Aplicación de la metodología propuesta mediante algoritmos genéticos

Técnica metaheurística GA ya ha sido utilizada en problemas de localización de instalaciones y optimización de costes. Ha sido su uso y resultados aceptables en problemas similares lo que ha motivado en parte su elección. Este metaheurístico va a permitir contrastar los datos obtenidos con los del metaheurístico que se describe en la siguiente sección y del que no existen aplicaciones en la literatura para el problema planteado, constituyendo por tanto la principal contribución de esta tesis.

El algoritmo GA ha sido ya descrito en las secciones anteriores existiendo también en la literatura autores como Holland (1975) creador del algoritmo, Whitley (1994) o Michalewicz (1992), que han permitido dar a conocer y entender mejor los fundamentos básicos de dicho algoritmo.

El algoritmo GA establece una analogía entre el conjunto de soluciones de un problema y el conjunto de individuos de una población natural, codificando la información de cada solución en un string (vector binario) a modo de cromosoma. En palabras del propio Holland (1975):

“Se pueden encontrar soluciones aproximadas a problemas de gran complejidad computacional mediante un proceso de evolución simulada”.

A tal efecto se introduce una función de evaluación de los cromosomas (individuo), que será llamada “aptitud” o “calidad” (*fitness*) y que está basada en la función objetivo del problema. Igualmente se introduce un mecanismo de selección de manera que los individuos con mejor evaluación sean escogidos para “reproducirse” más a menudo que los que la tienen peor.

En la Figura 5.1 se muestra el esquema básico que se puede identificar en cualquier código de GA.

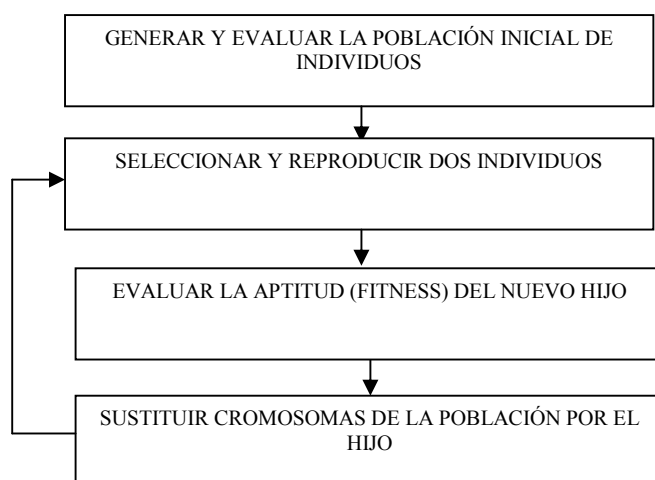


Figura 5.1: Esquema básico de algoritmo genético.
Fuente: Michalewicz (1992).

Aunque GA ha tenido un gran desarrollo en la comunidad científica, con distintas evoluciones respecto del concepto clásico, básicamente mantienen la implementación propuesta por Michalewicz (1992), en donde se combinan los elementos genéticos con la búsqueda local, como se muestra en *Algoritmo Genético*.

Algoritmo Genético

1. **Generar soluciones** — Construir un conjunto de soluciones P con tamaño $PopSize$ mediante generación aleatoria.
2. **Mejorar soluciones** — Aplicar un método de búsqueda local a cada solución del conjunto P .

Mientras (número de evaluaciones $<$ $MaxEval$)

1. **Evaluación** — Evaluar las soluciones en P y actualizar, si es necesario, la mejor solución almacenada.
2. **Supervivencia** — Calcular la probabilidad de supervivencia basada en la calidad de las soluciones. Según dichas probabilidades seleccionar aleatoriamente $PopSize$ soluciones (con reemplazamiento) de P . Sea P el nuevo conjunto formado por las soluciones seleccionadas (algunas pueden aparecer repetidas).
3. **Combinación** — Seleccionar una fracción p_c de soluciones de P para ser combinadas. La selección es aleatoria y

equiprobable para todos los elementos de P . Los elementos seleccionados se emparejan al azar y, por cada pareja, se generan dos descendientes que reemplazaran a los padres en P .

4. **Mutación** — *Una fracción p_m de las soluciones de P se selecciona para aplicar el operador de mutación. La solución resultante reemplaza a la original en P .*

Según Tim Jones (2005), los GA consisten básicamente de cuatro etapas. El primer paso es la inicialización de los individuos de manera aleatoria. Después, se obtiene la aptitud de cada uno de ellos con base en la función de evaluación. Posteriormente, se selecciona un conjunto de individuos a ser recombinados.

La selección, entre otras formas, puede realizarse por el método de la ruleta (basado en la proporción de la aptitud con respecto a la de los demás) o por torneo (eliminación directa entre dos candidatos). Por último, el algoritmo, a partir de la recombinación de los individuos seleccionados, genera nuevos individuos utilizando los operadores de cruce y mutación.

El operador de cruce toma dos individuos y selecciona en ellos un punto de manera aleatoria y, a partir de ese lugar, intercambia las partes de los individuos. La mutación se utiliza para generar material genético nuevo con el objetivo de evitar que el algoritmo se estanque en algún óptimo local. Este proceso, regularmente, se efectúa con una probabilidad pequeña.

El algoritmo termina después de cierto número de iteraciones o por algún otro criterio de parada relacionado con la aptitud.

Para la aplicación de esta metodología se ha utilizado el lenguaje matemático MATLAB, que resuelve mediante la función de algoritmos genéticos (función “*ga*”), la función objetivo (*fun*) y sus restricciones (*Aineq*, *bineq*), diseñadas en el capítulo anterior.

Se parte de la función principal (5.4.1) con la que después de las varias generaciones programadas, se obtendrá una aproximación al óptimo de la función objetivo:

$$[x, fval, exitFlag, output, population, scores] = \mathbf{ga} (fun, nvars, Aineq, bineq, Aeq, eq, lb, ub, nonlcon, intcon, options). \quad (5.4.1)$$

Esta función puede resolver problemas lineales o no lineales. El problema planteado en esta tesis tiene la siguiente forma:

$$\min F(x) \quad (5.4.2)$$

sujeta a las siguientes restricciones:

$$Aineq \cdot x \leq bineq \quad (5.4.3)$$

$$\text{y con los límites: } LB \leq x \leq UB \quad (5.4.4)$$

Es decir, se trata de resolver un problema lineal (5.4.2) con restricciones (5.4.3) y (5.4.4), que no puede ser resuelto con métodos exactos debido al elevado número de variables.

Los operadores incluidos en la función (5.4.1) de MATLAB tienen que ser adecuadamente parametrizados para poder obtener el valor más cercano al óptimo buscado para resolver el problema de minimización de costes de transporte de residuos y localización de instalaciones.

En la Figura 5.2 se puede ver el flujograma con los operadores que utiliza el algoritmo GA.

Dentro de este diagrama, también se nombran algunas de las funciones utilizadas por MATLAB que permiten definir los parámetros utilizados por cada uno de los operadores.

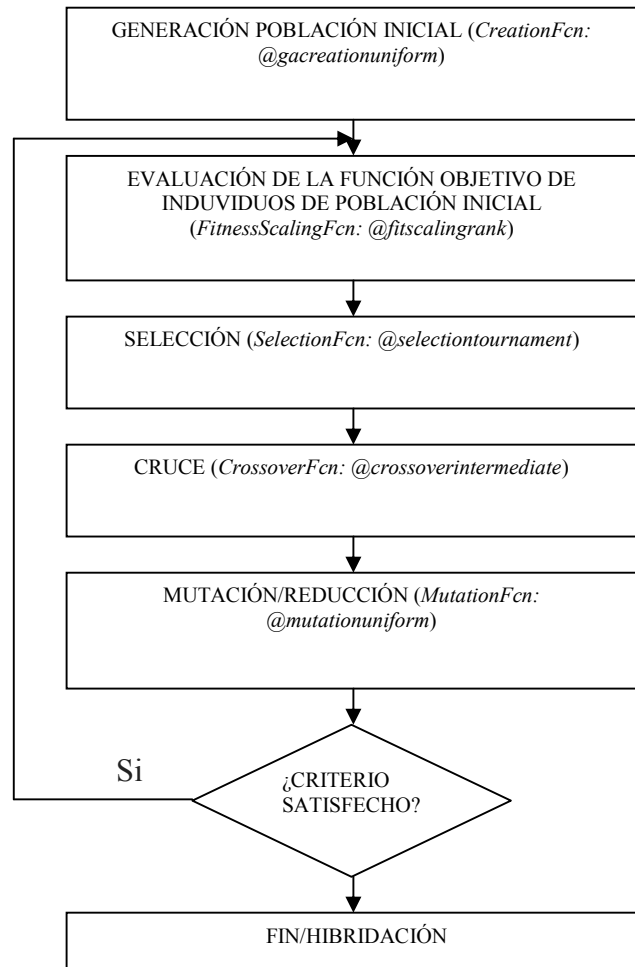


Figura 5.2: Diagrama de flujo de GA utilizado en MATLAB.
Fuente: Elaboración propia.

A continuación se describen los operadores utilizados con la función (5.4.1) de MATLAB y cuales son las mejores funciones a utilizar para buscar el óptimo del problema planteado en el capítulo 4:

Operador “generación de población inicial”: Uno de los factores más importantes que determina el rendimiento de los GA es la “diversidad” de la población. Si la distancia promedio entre los individuos es grande, la “diversidad” será alta; si la distancia media es pequeña, la “diversidad” será baja. El valor correcto de “diversidad de población” se controla con el parámetro de rango inicial de población: “PopIniRange”.

Por defecto, el GA crea una población inicial aleatoria con una distribución uniforme usando la función creación (@gacreationuniform). Se

puede especificar el rango inicial de población, con el parámetro de población. El GA puede encontrar el óptimo, incluso si este no se encuentra en el rango inicial, si la población tiene bastante diversidad.

Operador “Selección”: El GA aplica selecciones de padres de hijos de cruce o mutación. La función “Selección” asigna la mayor probabilidad de selección de individuos como padres, proporcional al valor de su función objetivo. Si los valores escalados varían demasiado ampliamente, los individuos con los valores más altos en la escala, se reproducen demasiado rápido, tomando el relevo la descendencia genética de la población con demasiada rapidez, y evitando que el GA busque otras áreas del espacio de soluciones. Por otra parte, si los valores escalados varían sólo un poco, todos los individuos tienen aproximadamente la misma posibilidad de la reproducción y la búsqueda progresará muy lentamente.

Se puede parametrizar la selección a partir de un muestreo estocástico universal especificado en la función de selección (*@selectionstochunif*) o con la función de selección (*@selectiontournament*, 2) que a partir de un muestreo mediante la técnica de *torneo binario*, se eligen 2 padres aleatorios y se queda el mejor según su función de aptitud.

Operador “Evaluación”: Obtiene los valores de la función fitness para ser utilizados por el operador “Selección”. Existen varias funciones de evaluación como (*@fitscalingprop*) o (*@fitscalingshiflinear*), aunque se usará la función de evaluación (*@fitscalingrank*), que escala las puntuaciones directas basadas en el rango de cada individuo en lugar de su puntuación. El rango de un individuo es su posición en la **escala de** puntuaciones: el rango de la persona más óptimo es 1, el siguiente es $1/\sqrt{2}$, y así sucesivamente.

Operador “Cruce”: El GA aplica hijos de cruces mediante la opción que se especifique en la función de cruce (*@crossoversscattered*), que crea hijos de cruce de la población dada, usando los padres disponibles en un cruce uniforme o con la función (*@crossoverintermediate*), donde se combinan dos padres para formar hijos de cruce para la siguiente generación.

Los hijos se crean como la media ponderada de los padres ($child = parent1 + rand \cdot Ratio \cdot (parent2 - parent1)$). Se elige un 5% de hijos élite (*EliteCount*), que son los individuos que sobrevivirán a la siguiente generación.

El parámetro *CrossoverFraction*, en las opciones de reproducción, especifica la fracción de cada población, con excepción de los niños de élite que componen los hijos de cruce. Una fracción de cruce de 1 significa que todos los hijos, con excepción de los hijos élite, son los hijos de cruce, mientras que una fracción de cruce de 0 significa que todos son hijos de mutación. Ninguno de estos extremos es una estrategia eficaz para la optimización de la función.

Operador “Mutación”: El GA aplica mutaciones mediante la opción que se especifique en la función elegida. Con la función mutación (*@mutationuniform*, 0.01) se genera una mutación aleatoria de hijos en la población, que proporciona una diversidad genética para buscar la solución en un espacio más amplio. Se selecciona la fracción 0,01 de la población que va a ser sustituida por el número aleatorio de una distribución uniforme.

Otra opción de función mutación es Gaussiana (*@mutationgaussian*). Añade un número aleatorio o mutación elegido a partir de una distribución Gaussiana a cada entrada del vector padre. Típicamente, la cantidad de mutación, que es proporcional a la desviación estándar de la distribución, disminuye en cada nueva generación.

Se puede controlar la cantidad promedio de mutación que el algoritmo aplica a cada uno de los padres en cada generación a través de los parámetros “Escala” y “Reducción”.

Después de describir los operadores y las funciones que se pueden usar con el algoritmo GA de MATLAB, en la Tabla 5.1 se indican que funciones y valores son utilizados en esta tesis para parametrizar el algoritmo GA.

Tabla 5.1: Funciones y valores usados como parámetros GA.

Fuente: *Elaboración propia.*

Operadores GA-MATLAB	Función	Valores	Generación	Observaciones
Población inicial	@gacreationuniform	20	100	Distribución uniforme
	@gacreationlinearfeasible			[-9999;10001]
Evaluación	@fitscalingrank	1/√2		
	@fitscalingprop			
Selección	@selectiontournament	2		Torneo binario
	@selectionstochunif			Muestreo estocástico
	@selectionuniform			Distribución uniforme
Cruce	@crossoverintermediate	0,8		Distribución uniforme
	@crossoverscattered			Distribución uniforme
Mutación	@mutationuniform	0,01		Distribución uniforme
	@mutationgaussian			Distribución gaussiana
Fracción cruce (CrossoverFraction)	0:mutación	0,9		Entre [0, 1]
	1:cruce			
Élite	EliteCount	0,05		5% POP cada generación

El esquema general del método propuesto se muestra en el **Pseudocódigo1** del algoritmo genético utilizado en MATLAB.

Pseudocódigo1: *algoritmo genético*

- Inicializa parámetros por defecto
- Comprueba argumentos de entrada
 - Si no es correcta la estructura argumentos → mensaje de error
- Llama a la función “gacommon”
- Cambia las restricciones a un valor correcto si están vacías
- Llama al solver adecuado para resolver la función objetivo:
 - gaminlp (con enteros)
 - galincon (lineal con restricciones)
 - gaunc (sin restricciones)
 - gacon (no lineal con restricciones)
- End ga

Por último, para detener el proceso se definen los criterios que determinan que el algoritmo se pare. Se pueden especificar las siguientes opciones:

- a. Especificando el número máximo de iteraciones que el GA puede realizar antes de parar.

- b. Especificando el tiempo máximo en segundos que el GA se puede ejecutar antes de parar, medido por el tiempo de ordenador (*cputime*).
- c. El algoritmo se detiene si el mejor valor de fitness es menor o igual a un valor límite de *fitness*.
- d. El algoritmo se detiene si el cambio relativo medio en el mejor valor de la función de *fitness* es menor o igual a una función de tolerancia.
- e. El algoritmo se detiene si no hay una mejora en el mejor valor de la aptitud para un intervalo de tiempo (en segundos) especificado por un límite de tiempo, medido por *cputime*.

5.5. Aplicación de la metodología propuesta mediante optimización de enjambre de partículas

Si la anterior técnica metaheurística tenía interés por su aceptación en la comunidad científica, esta segunda metodología propuesta tiene interés por una parte, por ser una técnica más moderna y actual que las utilizadas hasta ahora en aplicaciones de búsqueda y por otra parte, novedosa en cuanto su uso como técnica de localización de instalaciones ya que en la revisión de la literatura sobre PSO no se ha encontrado ningún autor que la aplique en el campo de la logística inversa de residuos ni en la identificación de instalaciones en general.

PSO es un algoritmo bioinspirado en el comportamiento social de individuos dentro de enjambres en la naturaleza. Fue originalmente desarrollado para problemas continuos, pero existen adaptaciones del método a problemas discretos. Esta técnica evolutiva se basa en el paradigma de inteligencia colectiva y trata de emular el comportamiento de los bancos de peces, enjambre de abejas o bandadas de pájaros con el objetivo de solucionar problemas de optimización complejos.

Los agentes de búsqueda de este método son las partículas, las cuales son soluciones potenciales que tienen una posición y velocidad asociada. Estas partículas “vuelan” a través de un espacio multidimensional ayudándose de su propia experiencia, del conocimiento del enjambre y de operadores estocásticos. Así, las partículas modifican sus posiciones a las que tienen la mejor solución del momento, tendiendo a concentrarse en áreas potencialmente buenas.

PSO comparte algunas similitudes con otras técnicas evolutivas, pero a diferencia de otros métodos, el número de parámetros que se deben ajustar es pequeño según Eberhart y Shi (2001). Esto hace que la implementación, comparada con otras técnicas evolutivas, sea más sencilla.

PSO ha sido utilizado con gran éxito para solucionar problemas de tipo continuo. Algunos ejemplos donde se ha demostrado su efectividad son: Entrenamiento de redes neuronales por Eberhart y Shi (2001), y Funciones continuas no lineales por Ching-Jong et al. (2007).

PSO puede ser utilizado para resolver la mayoría de los problemas de optimización, como el problema de localización en logística inversa de esta tesis, que al ser un problema de optimización combinatoria que pertenece a la clase NP-duro, implica que es un problema no tratable con métodos tradicionales.

Previo a resolver el problema de optimización de logística inversa diseñado en el capítulo anterior, se verá el modelo matemático genérico que define al algoritmo PSO y poder aplicarlo, al igual que con GA, con el lenguaje matemático MATLAB.

La solución potencial es representada en PSO por un vector llamado partícula o simplemente solución. Cada una de estas partículas i tiene la misma dimensión n (variables de la función objetivo) y se representan de la siguiente manera:

$$X(i) = (x_{(i1)}, x_{(i2)}, \dots, x_{(in)}) \quad (5.5.1)$$

Las partículas, sobre el espacio de búsqueda *n-dimensional*, “vuelan” tratando de encontrar una solución óptima. Para hacer esto, cada individuo ajusta su posición de acuerdo a una combinación lineal de su inercia, su propia experiencia y del conocimiento del enjambre. Cada agente almacena en una memoria la mejor posición encontrada hasta el instante actual t . La experiencia de la partícula se denota como:

$$P(i) = (p_{(i1)}, p_{(i2)}, \dots, p_{(in)}) \quad (5.5.2)$$

El conocimiento del enjambre es el conjunto de memorias de cada partícula. A diferencia de los GA, en PSO no existe la competencia entre individuos. En cada interacción, cada partícula pone a disposición de los demás su memoria de conocimiento. La asignación de informantes (vecindario) es una forma de compartir la experiencia. Cada partícula recibe información de k agentes seleccionados de forma aleatoria en cada iteración del algoritmo. Después, la partícula determina entre sus informantes aquel que tenga la mejor aptitud previa. Posteriormente, lo selecciona para que sea parte del proceso de actualización de su posición. Normalmente, el valor de k es pequeño, sin embargo, puede ser tan grande como el tamaño del enjambre. Este último criterio hace que las partículas tiendan hacia la mejor partícula encontrada hasta algún instante t . Las características del problema a solucionar determinan cual de los dos métodos es más adecuado para su implementación. La mejor partícula del vecindario se representa por:

$$G(i) = (g_{(i1)}, g_{(i2)}, \dots, g_{(in)}) \quad (5.5.3)$$

Al igual que en otros algoritmos de tipo evolutivo, PSO necesita de una función de evaluación (también llamada función de aptitud). Esta permite determinar la calidad de las soluciones. Su importancia radica en que es la única forma de poder evaluar la posición de cada elemento. Cada coordenada (elemento) de X de cada partícula tiene una velocidad o razón de cambio V , para $d = 1, 2, \dots, n$.

$$V(i) = (v_{(i1)}, v_{(i2)}, \dots, v_{(in)}) \quad (5.5.4)$$

Para realizar un desplazamiento, la partícula determina la velocidad considerando su propia inercia W (busca evitar la convergencia prematura), su memoria de conocimiento y su confianza en el enjambre, para después, sumarla a la posición actual. El grado de confianza lo determinan los operadores aleatorios r_1 y r_2 en el rango $[0-1]$ junto con los coeficientes de confianza c_1 y c_2 . Estos últimos, también llamados constantes de aceleración, son los términos que tiran a cada partícula hacia las posiciones P y G según Eberhart y Shi (2001).

En otras palabras, las partículas hacen un movimiento hacia un punto intermedio tomando en cuenta la mejor posición previa, el mejor informante y un punto accesible desde la posición actual (Figura 5.3).

Las ecuaciones siguientes ajustan la velocidad y posición de cada partícula:

$$V_i^{t+1} = w^t V_i^t + c_1 r_1^t (P_i - X_i^t) + c_2 r_2^t (G_i - X_i^t) \quad (5.5.5)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (5.5.6)$$

Donde:

V_i^{t+1} : Velocidad ajustada

w^t : Coeficiente de inercia

c_1 : Coeficiente de confianza en la experiencia (*CognitiveAttraction*)

c_2 : Coeficiente de confianza en la experiencia del grupo (*SocialAttraction*)

P_i : Mejor posición previa de i

X_i^t : Posición actual de i

G_i : Mejor posición previa encontrada por el grupo

r_1^t y r_2^t : Operadores aleatorios entre 0 y 1

X_i^{t+1} : Posición de la partícula i después del ajuste

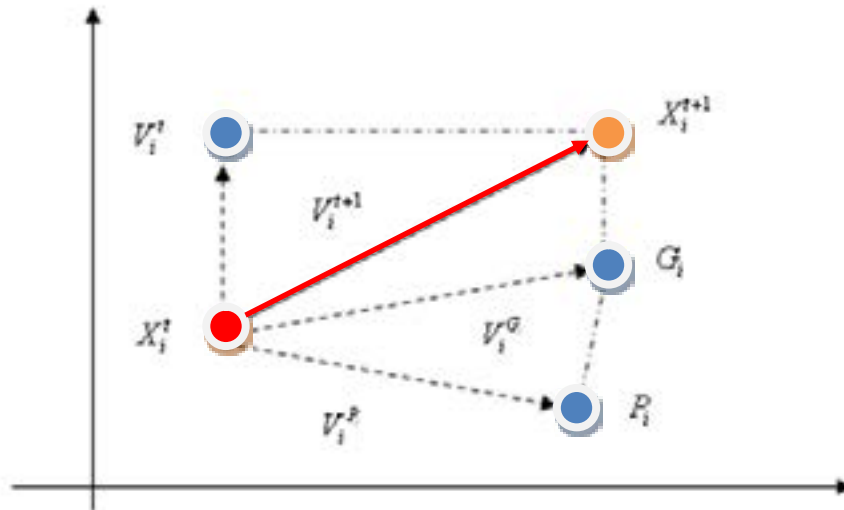


Figura 5.3: Modificación de la posición de la partícula i .
Fuente: Eberhart y Shi (2001).

En el proceso de actualización de la velocidad están implicados operadores aleatorios, por lo que el método de modificación de la posición de las partículas es estocástico.

En términos generales, PSO se puede describir en tres pasos. El primero es evaluar cada elemento para determinar la calidad de la posición actual. Esto permite que se puedan encontrar la mejor posición y las mejores partículas. Después, se deben realizar los ajustes necesarios de las mejores posiciones previas. Por último, se determinan los nuevos desplazamientos para cada partícula con la información ajustada. Por analogía, estos movimientos no son más que una forma de tratar de imitar a otros individuos.

Para detener el proceso se necesita que un criterio de parada se cumpla. Este puede ser determinado por un número fijo de iteraciones, opcionalmente combinado con un umbral de error aceptable. Al terminar la ejecución del algoritmo, la solución mostrada es la mejor posición previa encontrada por alguna partícula.

De manera más detallada, el método de PSO se puede describir como sigue:

1. Inicializar la población. La posición de cada una de las partículas es determinada de manera aleatoria.
2. La mejor posición previa es igualada a la posición actual.
3. Cada posición es evaluada en la función de aptitud para determinar la calidad de la solución.
4. Se compara la aptitud de la posición actual con la mejor previa.
5. Asignar informantes (vecindario) de tamaño k a la partícula.
6. Determinar la mejor partícula del vecindario.
7. Ajustar la velocidad.
8. Ajustar la posición.
9. Verificar si se cumple el criterio de parada.
10. Si no se cumple, regresar al paso 5.

La versión original de PSO presentaba algunas desventajas. Si la mejor solución está estancada en algún óptimo local, todas las partículas tenderán rápidamente a concentrarse en ese punto según Afsahi (2011). Otra importante desventaja era su poco control para tener un balance entre la exploración y la explotación. Para contrarrestar estos problemas se utiliza el coeficiente de inercia W propuesto por Abdel-Kader (2011).

Después de ver el funcionamiento genérico del algoritmo PSO y su modelo matemático se darán algunas consideraciones y se definirá el pseudocódigo necesario para su aplicación con el lenguaje MATLAB.

Hay que tener en cuenta, que se obtienen mejores resultados si la opción de rango de población inicial (*PopInitRange*) se define cerca del dominio esperado de la región factible. Esto se hace de forma automática si el problema tiene dos límites, uno inferior y otro superior, pero en caso de que hubiese restricciones lineales y no lineales habría que definir el rango de población inicial.

Por otra parte hay que considerar según demostraron Pérez y Behdinan (2007), que PSO sólo es estable si se cumplen las siguientes condiciones:

Considerando:

$$w^t : \text{coeficiente de inercia} \quad (5.5.7)$$

$$c_1 : \text{Coeficiente SocialAttraction} \quad (5.5.8)$$

$$c_2 : \text{Coeficiente CognitiveAttraction} \quad (5.5.9)$$

$$\begin{cases} 0 < (c_1 + c_2) < 4 \\ \frac{(c_1 + c_2)}{2} - 1 < w^t < 1 \end{cases} \quad (5.5.10)$$

Si las condiciones de (5.5.10) están satisfechas, se garantiza que el sistema converge a un punto de equilibrio estable. Sin embargo, no se puede garantizar que este punto sea realmente el mínimo global y su aceptación como una solución debe ser verificada.

En la Tabla 5.2 se indican los valores utilizados para parametrizar el algoritmo PSO.

Tabla 5.2: Valores usados como parámetros PSO.

Fuente: Elaboración propia.

Parámetros PSO-MATLAB	Nombre	Valores	Iteración	Observaciones
Población inicial	<i>POP</i>	20	100	Distribución uniforme [LB, UB]
Coeficiente confianza grupo	<i>C₁</i>	1,2		<i>SocialAttraction</i>
Coeficiente confianza individual	<i>C₂</i>	0,012		<i>CognitiveAttraction</i>
Coeficiente Inercia	<i>W</i>	0,0004		-
Operador aleatorio	<i>R₁</i>	[0, 1]		Distribución uniforme [0, 1]
Operador aleatorio	<i>R₂</i>	[0, 1]		Distribución uniforme [0, 1]

A grandes rasgos el algoritmo PSO utilizado en MATLAB presenta el esquema que se muestra en el **Pseudocódigo7**.

Pseudocódigo7: Algoritmo PSO

1. Comprueba argumentos de entrada: “nargin”
2. Si no es correcta la estructura argumentos → mensaje de error
3. Comprueba existencia de variables, opciones
4. Comprueba la estabilidad del enjambre
5. if $options.SocialAttraction + options.CognitiveAttraction \geq 4$
 - mensaje error
6. End if
7. Redimensiona *PopInitRange* si es necesario: “*repmat*”
8. Comprueba la población inicial respecto de las restricciones de los límites *LB* y *UB*
9. Comprueba la validez de “*VelocityLimit*”: debe ser un escalar
10. Genera el estado inicial del enjambre
11. Genera un enjambre de distribución uniforme
12. Inicializa la posición de las partículas:
13. Inicializa el óptimo local, global y la posición para el peor y mejor caso
14. Comprueba la población inicial respecto de las restricciones lineales *Aineq*, *bineq*
15. Comprueba el tipo de restricciones
16. Itera el enjambre de partículas:
 - for 1 hasta n° generaciones:
 1. Resetea puntuación óptimo y penaltis
 2. Comprueba límites antes de proceder
 3. Evalúa los óptimos y actualiza los mejores locales
 4. Penaliza violaciones de restricciones
 5. Actualiza los mejores locales
 6. Actualiza el mejor global y óptimos
 7. Actualiza velocidad de partículas y posiciones
 - End for
17. Asigna variables de salida y mensajes de salida
18. Ir a comienzo

El código del programa PSO desarrollado en MATLAB se puede consultar en el Anexo A, y sus parámetros en el Anexo C. A continuación se muestra la función principal (5.5.11) con la que se llama desde el programa principal de MATLAB, al algoritmo PSO con el que buscar el óptimo de la función objetivo.

$$[xOpt,fval]=pso(fitnessfcn,nvars,Aineq,bineq,[],[],LB,UB,[],options) \quad (5.5.11)$$

Siendo:

- *Fitnessfcn*: Función objetivo.
- *Nvars*: Número de variables.
- *Aineq*: Restricciones derechas de las inecuaciones de desigualdad.
- *Bineq*: Restricciones izquierdas de las inecuaciones de desigualdad.
- *LB, UB*: Límites inferior y superior.
- *Options*: Opciones de configuración de parámetros de PSO.
- *xOpt*: Variables optimizadas
- *fval*: Mejor función aptitud (*fitness*)

5.6. Conclusiones

En este capítulo se han descrito algunas de las técnicas metaheurísticas existentes y se han desarrollado las dos técnicas propuestas para la tesis: los algoritmos GA y los algoritmos PSO. También se ha discutido la aplicación de estos metaheurísticos al problema de localización de instalaciones de RSU.

En el siguiente capítulo se describirá la aplicación informática utilizada para modelar y resolver tanto el modelo exacto, como las técnicas metaheurísticas propuestas. Se discutirán las ventajas e inconvenientes de usar una herramienta informática como es el lenguaje matemático MATLAB, o el uso de otras herramientas como son el lenguaje de simulación y modelación matemática GAMS ó AIMMS.

Capítulo 6

6. MODELADO Y RESOLUCIÓN: APLICACIÓN A UN CASO REAL

En este capítulo se va a describir la aplicación del lenguaje de cálculo numérico seleccionado para modelar y resolver tanto el modelo exacto como las técnicas metaheurísticas propuestas sobre un caso real.

6.1. Introducción

Se analizan las ventajas e inconvenientes de usar una herramienta informática como es el lenguaje matemático MATLAB, o el uso de otras herramientas informáticas como son los lenguajes de simulación y modelado matemático GAMS ó AIMMS.

También se aplican todas las técnicas metaheurísticas presentadas en las secciones anteriores sobre un caso real de localización de instalaciones en 90 municipios de la provincia de Málaga. Los algoritmos se han programado con el lenguaje MATLAB y el código de estos algoritmos (GA y PSO) se presentan en el Apéndice A.

6.2. Buscando aplicaciones informáticas de optimización

Hoy en día es posible adquirir una gran variedad de aplicaciones que resuelvan los problemas de optimización. Una de las ventajas de usar un programa comercial es su velocidad de resolución, ya que gran parte de los cálculos de álgebra de estos programas se hacen desde librerías muy optimizadas, pero también incorpora una serie de recursos que posiblemente nunca se utilizarán.

Como se ha adelantado en la introducción, para modelar y resolver el problema diseñado en el capítulo 4 se utilizarán métodos exactos y las técnicas metaheurísticas propuestas en el capítulo 5.

Se podría desarrollar un programa de resolución en C u otro lenguaje adaptado al problema, y evitar así hacer operaciones internas que un programa comercial dedica a otros aspectos que no se necesitan. Sin embargo, la efectividad y la velocidad de modelado que proporciona un programa comercial es un factor que ha determinado su uso en esta tesis.

Este punto no es menor, es importante realizar un estudio preliminar porque la difusión y herramientas de técnicas y *software* en este campo de la Organización Industrial aún no tiene unos mecanismos de trabajo establecidos para hacer el tipo de investigación que se hace en la tesis.

Dentro de los programas comerciales más usados se encuentran, aparte del lenguaje matemático MATLAB (*MATRIX LABORATORY*) aplicado en ésta tesis, GAMS (*General Algebraic Modeling System*), AMPL (*A Mathematical Programming Language*) ó AIMMS (*Advanced Interactive Multidimensional Modeling System*), que modelarían el problema de optimización lineal mixta y resolverían los *solvers* del tipo: CPLEX, MINOS, GLPK, el comando *linprog* del complemento *Matlab Optimization Toolbox* o incluso los *solvers* de *Microsoft Excel* y *OpenOffice*.

El inconveniente de algunos de estos lenguajes es que tanto GAMS, AMPL o AIMMS necesitan desarrollar en un lenguaje de alto nivel como C++, Java, etc., las técnicas metaheurísticas propuestas, mientras que con MATLAB se pueden programar tanto el algoritmo exacto como los metaheurísticos propuestos.

Así mismo, debido a que MATLAB, GAMS, AMPL o AIMMS son software de carácter propietario, para hacer un uso total de estos sería necesario adquirir una cara licencia comercial. Aunque la comunidad de usuarios de AIMMS, ha puesto de manera pública una distribución de demostración, pero limitada a las siguientes condiciones:

1. Límites del modelo (licencias estudiantes):

- Número de restricciones y variables: 300
- Número de elementos identificadores: 200
- *Solvers*: CPLEX, GUROBI, KNITRO

2. Límites de tiempo (licencias de demostración):

- Para licencias demostración: 30 días de uso, sin limitación de variables ni restricciones, pero con limitación en el uso de *solvers*.

Tabla 6.1: Herramientas de resolución, indicando los problemas que pueden resolver. *Fuente: Elaboración propia.*

Herramienta de resolución cuadrática				
Nombre	Algoritmo	Problemas	Licencia	Limitaciones en demo
CPLEX	Simplex Primal	LP	Comercial	90 días limitada en tamaño
	Simplex Dual	QP, redes		
	P.interior barrera	QCOQP		
	Ramificar y cortar	MILP, MIQP		
Herramienta de resolución lineal				
Nombre	Algoritmo	Problemas	Licencia	Limitaciones en demo
GLPK	Simplex Primal	LP	GPL	
	Simplex Dual			
	P.interior primal-dual			
	Ramificar y cortar	MILP		
MINOS	Simplex	LP	Comercial	3 meses no gratuita
	Gradiente reducido	No lineales		
	Lagrangiano proyectado			

En la Tabla 6.1 se describen algunos de los *solvers* más utilizados, indicando los tipos de problemas que resuelven, los algoritmos usados, tipo de licencia y limitaciones de uso.

Igual ocurre con AMPL o GAMS, de los cuales se puede descargar una versión de evaluación con limitación del número de variables y restricciones (300), o limitación en el número de días de uso. Con estas limitaciones, solo se podrían modelar algunos problemas simples de optimización lineal mixta y comparar con el óptimo alcanzado con el metaheurístico.

En la Tabla 6.2 se pueden ver las limitaciones de estos lenguajes, indicando los solvers que pueden usar y el tipo de licencia.

Tabla 6.2: Herramientas de modelado, indicando los solvers compatibles.
Fuente: *Elaboración propia.*

Herramientas de modelado			
Nombre	Solvers	Licencia	Limitaciones en demo
AMPL	BPMD, CPLX, LAMPS, LOQO, lp_solve, MINOS, Mosek, OSL, SOPT, XA, Xpress-MP, MINTO, CONOPT, DONLP2, FILTER, FSQP, IPOPT, KNITRO, LANCELOT, NPSOL, PENNON, SNOPT, MINLP, Gurobi	Comercial	300 variables y 300 restricciones y objetivos
GAMS	ALPHAECP, BARON, BDMLP, COIN-OR, CPLEX, DECIS, DICOPT, Gurobi, KNITRO, LGO, LINDOGLOBAL, LOGMIP, MILES, MINOS, Mosek, MSNLP, NLPEC, OQNLP, PATH, SBB, SCIP, SNOPT, XA, Xpress	Comercial	300 restricciones y variables, con 2000 elementos distintos de 0, 50 variables discretas. En caso de utilizar un solver global, está limitado a 10 restricciones y variables
MATLAB	MATLAB OPTIMIZATION TOOLBOX	Comercial	30 días solo empresas

De todos los lenguajes citados, solo MATLAB permite modelar y resolver el problema de optimización mediante métodos exactos y técnicas metaheurísticas al mismo tiempo, sin tener que utilizar otro lenguaje de alto nivel. Por otra parte y aunque MATLAB y sus complementos también están sujetos a licencia comercial, en el entorno académico resulta más accesible utilizar esta herramienta informática disponible en la mayoría de las universidades españolas y con un alto grado de aceptación frente al uso de las otras herramientas comerciales, con alto coste económico.

Como conclusión y tras ver algunas ventajas e inconvenientes de los lenguajes comerciales, en esta tesis se utilizará como herramienta informática el lenguaje matemático MATLAB al permitir unificar tanto métodos exactos como metaheurísticos. Las características generales de MATLAB son ampliamente conocidas en el ámbito de la Ingeniería por haberse utilizado en gran cantidad de asignaturas y prácticas de laboratorio, por lo que se entiende que no será necesaria su descripción.

MATLAB incluye el complemento *Matlab Optimization Toolbox*, que permite resolver problemas de optimización global (convexos y no convexos). Mediante funciones del tipo “*intlinprog*” o “*linprog*” se resuelven los problemas lineales con variables mixtas (enteras y binarias) o continuas. En los problemas convexos se podría utilizar el comando “*fmincon*”, en el que se puede especificar el uso de algoritmos de punto interior. También cuenta con dos funciones metaheurísticas, como son “*ga*” que encuentra el mínimo de la función objetivo aplicando el algoritmo GA y “*simulannealbnd*” que encuentra el mínimo de la función aplicando el algoritmo SA.

Como ya se indicó en el capítulo anterior, para resolver el problema de optimización de localización de instalaciones se utilizarán dos metaheurísticos: GA y PSO. Como MATLAB, en la versión utilizada en esta tesis (la R2014b), solo incluye los GA, para trabajar con el algoritmo PSO se ha tenido que desarrollar esta función.

La programación en MATLAB comprenderá varias etapas. En una primera etapa se desarrollará el código en MATLAB para resolver el problema de localización mediante programación lineal y para distintos rangos de variables, hasta llegar al número de variables y restricciones necesarias para resolver el problema de optimización del caso real. En una segunda se desarrollará el código en MATLAB para poder aplicar los algoritmos metaheurísticos y en la tercera etapa se compararán con los resultados obtenidos mediante programación lineal.

El programa principal en MATLAB para la programación lineal tiene las siguientes funciones:

- Lee los datos iniciales del problema (P_{ib} , d_{ij} , d_{jk}), desde un archivo EXCEL (Figura 6.1) mediante el comando “*xlsread*” (por ejemplo: *Pit=xlsread ('Produresimuni(it).xlsx','Produ90')*). Esta acción, para matrices con elevado número de parámetros, puede tomar algunos segundos.

MUNICIPIO	ALMEDA	ALCAON	ALFARITE	ALFARITULO	ALGARRIDO	ALGATOON	ALATORRE	ALA GRANDE	ALBUCA	ALBUZEN	ALBUGA	ALDA	ALDORA	ALPARENE	ANTIGENA
ALMEDA	0	81	85	83	107	122	81	86	101	105	87	81	85	111	101
ALCAON	81	0	21	25	29	101	76	88	21	119	62	102	108	112	101
ALFARITE	85	21	0	4	48	100	80	81	48	81	47	84	100	107	85
ALFARITULO	83	25	4	0	47	107	80	81	48	81	46	84	88	100	83
ALGARRIDO	107	29	48	47	0	117	81	75	28	128	81	85	84	100	107
ALGATOON	122	101	100	107	117	0	128	108	119	119	71	47	102	74	101
ALATORRE	81	76	80	80	81	100	0	12	38	71	27	38	45	112	81
ALA GRANDE	86	88	81	81	75	108	12	0	88	70	38	22	28	108	101
ALBUCA	101	21	48	48	28	110	88	88	0	101	88	81	88	107	81
ALBUZEN	105	119	81	80	108	71	77	70	101	0	82	88	88	88	81
ALBUGA	87	62	47	48	81	47	27	38	81	81	0	38	81	100	81
ALDA	81	102	84	84	88	102	38	22	82	38	38	0	38	88	81
ALDORA	85	108	100	88	84	74	45	28	81	81	81	28	0	82	81
ALPARENE	111	112	107	100	102	112	108	107	81	108	88	88	82	0	108
ANTIGENA	101	101	85	85	107	117	80	78	87	81	85	81	81	88	0
ARCHAZ	100	45	88	27	10	100	74	87	38	100	72	88	100	112	100
ARCHENA	81	21	21	21	88	100	84	21	81	81	81	81	81	100	81
ARAZUL	88	100	118	118	111	70	81	84	108	24	70	44	84	84	81
ARENAS	114	38	48	48	28	102	87	81	28	118	87	87	100	100	81
ARRATE	88	148	134	134	148	27	81	81	134	28	107	71	47	28	81
ATLATE	110	118	107	108	114	10	107	107	88	81	100	81	100	100	107
BENABUL	108	108	100	101	101	81	81	102	114	118	88	81	88	114	101

Figura 6.1: Archivo Excel con datos de distancias entre municipios.
Fuente: Elaboración propia.

- Una vez leídos todos los parámetros del problema, se calcula la función objetivo aplicando distintos bucles. En MATLAB hay que evitar, si es posible, los anidamientos en el bucle para no reducir excesivamente la velocidad de ejecución.
- Un ejemplo de Bucle utilizado para el cálculo de la variable x (cantidad de residuos transportados) del problema se muestra en **Código.1**.

Código.1: *Bucle variable x*

```

    for i=1:length(Pit)
        for j=1:length(Pit)
            for t=1:5
                if cont~=8001
                    fx(cont,1)=CMTt(1,t)*dij(i,j);
                    cont=cont+1;
                end
            end
        end
    end
    fx=round(fx');

```

- Una vez desarrollada la función objetivo, se codifican las restricciones del problema calculando las matrices A y b de la inecuación $A \cdot x \leq b$. Si las matrices son grandes ($20 \times 20 \times 5$, $40 \times 40 \times 5$ o $90 \times 90 \times 5$), se puede aprovechar su estructura utilizando la descomposición de Cholesky, o el complemento de Schur para facilitar la factorización en formas más sencillas. Estas técnicas hacen que MATLAB sea más rápido.
- Con las restricciones del problema ya planteadas se establecen los límites de las variables “ x ”, “ y ”, “ v ” y “ z ”, considerándolas enteras o binarias, según se especificó en el problema de optimización.
- Finalmente, se calcula el óptimo utilizando el comando: “*intlinprog*”. Este comando se incorporó en MATLAB en el segundo semestre del año 2013 calculando problemas de programación lineal mixta (con variables enteras y binarias), que los *solvers* anteriores no podían resolver.

Los resultados obtenidos con MATLAB mediante programación lineal para diferente número de municipios en el problema de localización de instalaciones se pueden ver en la Tabla 6.3.

Tabla 6.3: Resultados de programación lineal en MATLAB para distintos municipios. *Fuente: Elaboración propia.*

Programa-Matlab	Nº. Municipios	Nº. variables	Fval	Tiempo(s)
Tesisresiduos_6	6	84	9,64E+05	0,33
Tesisresiduos_20	20	4200	1,03E+07	0,47
Tesisresiduos_40	40	16400	1,46E+07	1,46
Tesisresiduos_90	90	81900	-	-

Como se puede observar en la Tabla 6.3, para los primeros 40 municipios se han alcanzado los valores óptimos (*Fval*) mediante métodos exactos con MATLAB y con tiempos de ejecución bajos (1,46 s., para una matriz de $40 \times 40 \times 5$). El problema surge cuando se aumenta el número de municipios.

Así, para el caso de los 90 municipios, donde la función objetivo incluye 81900 variables y 41865 restricciones, MATLAB es incapaz de resolver mediante programación lineal este problema de optimización, al ser un problema NP-duro. Como se indicó en capítulos anteriores, la solución óptima solo podrá ser obtenida de forma aproximada mediante las técnicas metaheurísticas propuestas, de ahí que aparezcan guiones en la Tabla 6.3: no se pueden dar resultados en un tiempo razonable con MATLAB (días o semanas).

Para comparar los resultados obtenidos con MATLAB y presentados en la Tabla 6.3, se han realizado las mismas pruebas realizando la programación con la versión de evaluación del programa comercial AIMMS 4.6 y el solver CPLEX 12.6.1 obteniéndose los resultados mostrados en la Tabla 6.4.

Tabla 6.4: Resultados de programación lineal con AIMMS para distintos municipios. *Fuente: Elaboración propia.*

Programa-AIMMS	Nº. Municipios	Nº. variables	Fval	Tiempo(s)
Tesisresiduos_6	6	84	9,64E+05	0,01
Tesisresiduos_20	20	4200	1,03E+07	0,05
Tesisresiduos_40	40	16400	1,46E+07	0,59
Tesisresiduos_90	90	81900	-	-

Los resultados de la programación con AIMMS son los mismos que los obtenidos con MATLAB, con la única diferencia de que los tiempos de respuesta de este programa dedicado son sensiblemente inferiores a los obtenidos con MATLAB, pero incluso así, tras largos tiempos de espera el óptimo aún se resiste, y AIMMS no devuelve ningún resultado en absoluto para el problema de los 90 municipios.

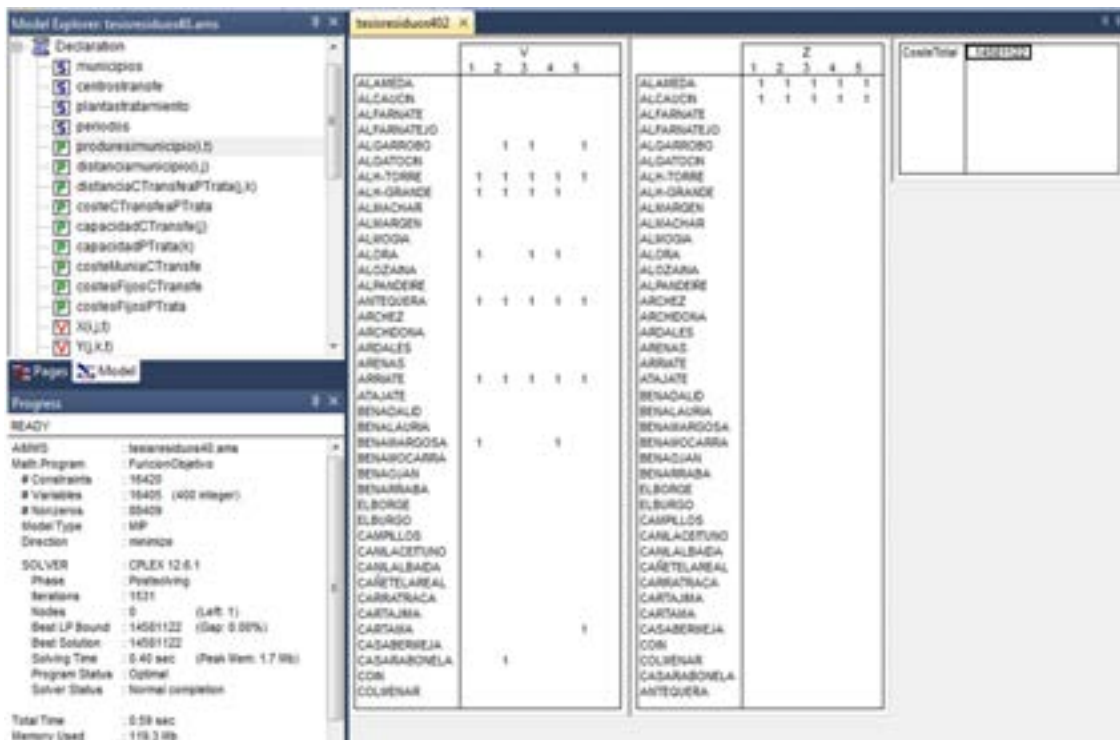


Figura 6.2: Programa AIMMS con los resultados óptimos para 40 municipios. *Fuente: Elaboración propia.*

En la Figura 6.2, se pueden observar los resultados obtenidos con el programa AIMMS para 40 municipios, necesitando 1531 iteraciones para

alcanzar el óptimo de 14581122 en un tiempo de 0,59 segundos. También se pueden ver los resultados de las variables binarias v_{ijt} y z_{jkt} indicando los municipios donde se deberían instalar los centros de transferencia y las plantas de tratamiento.

Como se ha podido comprobar en las Tablas 6.3 y 6.4, los tiempos de ejecución obtenidos en MATLAB son más largos que con los lenguajes compilados, esto se debe a que MATLAB es un lenguaje interpretado. Para compensar en parte esto, en muchas de las funciones programadas tanto en programación lineal como en la programación de los metaheurísticos se ha utilizado el comando “*coder*” (convierte funciones con extensión *.m*, en funciones compiladas con extensión *.mexw64*), que permite compilar parte del programa interpretado de MATLAB, obteniendo mejores tiempos de ejecución.

Según los resultados reflejados en las Tablas 6.3 y 6.4, los métodos exactos usados por MATLAB y AIMMS son insuficientes para resolver los problemas de optimización con matrices de $90 \times 90 \times 5$. Es a partir de este momento, cuando toma interés el uso de las técnicas metaheurísticas propuestas.

6.2.1. Resultados en MATLAB con algoritmos genéticos

Los resultados obtenidos con MATLAB aplicando un GA (con los parámetros descritos en el capítulo 5) para los diferentes ejemplos de problemas de localización de instalaciones se muestran en la Tabla 6.5.

Tabla 6.5: Resultados aplicando algoritmos genéticos con MATLAB para distintos municipios. *Fuente: Elaboración propia.*

GA-Matlab	Nº. Municipios	Nº. variables	Fval	Tiempo(s)
Tesisresiduos_6	6	84	9,98E+05	4,59
Tesisresiduos_20	20	4200	1,16E+07	48,07
Tesisresiduos_40	40	16400	1,58E+07	109,71
Tesisresiduos_90	90	81900	2,37E+07	170,86

Como se puede observar en la Tabla 6.5, los valores promedios obtenidos de la función aptitud ($fval$) aplicando GA mediante MATLAB son soluciones cercanas a las obtenidas con métodos exactos para 6, 20 y 40 municipios, por lo que se pueden considerar válidos los valores alcanzados para 90 municipios, que será la solución cuasi-óptima para el caso real planteado. La principal diferencia se da en unos tiempos de ejecución bastante mayores con GA, si bien existen valores de $fval$ para cualquier número de municipios, hecho que no se da con los métodos exactos.

El resultado indicado en la Tabla 6.5 corresponde al valor promedio de dichas ejecuciones. El tiempo de ejecución de una llamada al GA está en 7,54 segundos, frente a los 185,97 segundos necesario para obtener un valor promedio sobre 30 ejecuciones.

En la siguiente sección, en la Tabla 6.7 se pueden comparar los datos obtenidos con las distintas técnicas utilizadas, incluidos los resultados conseguidos con los métodos exactos, tanto con MATLAB como con AIMMS.

Algoritmos Genéticos

20 municipios

40 municipios

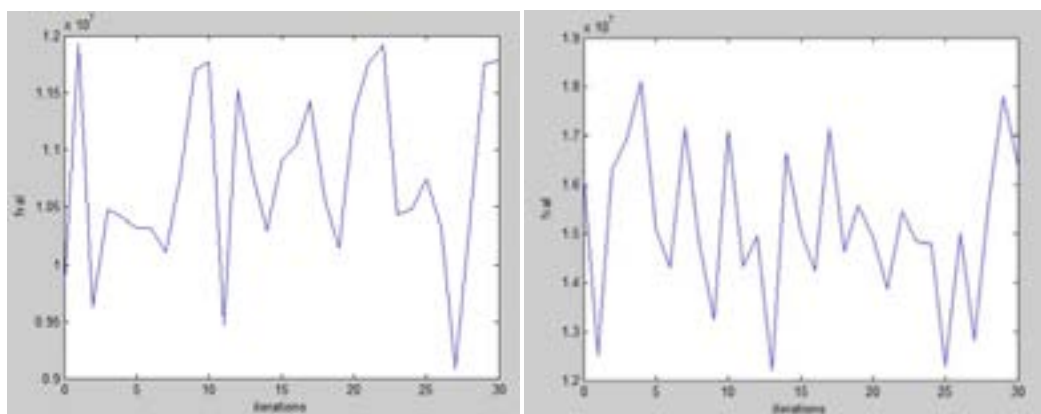


Figura 6.3: Resultado de los mejores valores de la función aptitud para 30 ejecuciones de GA con 20 y 40 municipios. *Fuente: Elaboración propia.*

En la Figura 6.3 se muestran los resultados de 30 ejecuciones del GA para 20 y 40 municipios. Nótese que la escala del eje vertical es distinta en cada problema.

En la Figura 6.4 se muestra el resultado de ejecutar una vez el GA para 40 municipios. Para reducir los tiempos de computación se ha compilado la función objetivo incluida en la función (6.2.1.1) de MATLAB que calcula el GA.

$$[x,fval]=ga(@fotesis40_mex,nvars,A,b,Aeq,beq,lb,ub,[],[],options) \quad (6.2.1.1)$$

Como ya se indicó, MATLAB incluye oficialmente en el complemento “*Matlab Optimization Toolbox*” de las últimas versiones, la función (6.2.1.1) que resuelve los GA utilizando una amplia variedad de parámetros configurables, que permiten ajustar los operadores genéticos según el problema a resolver.

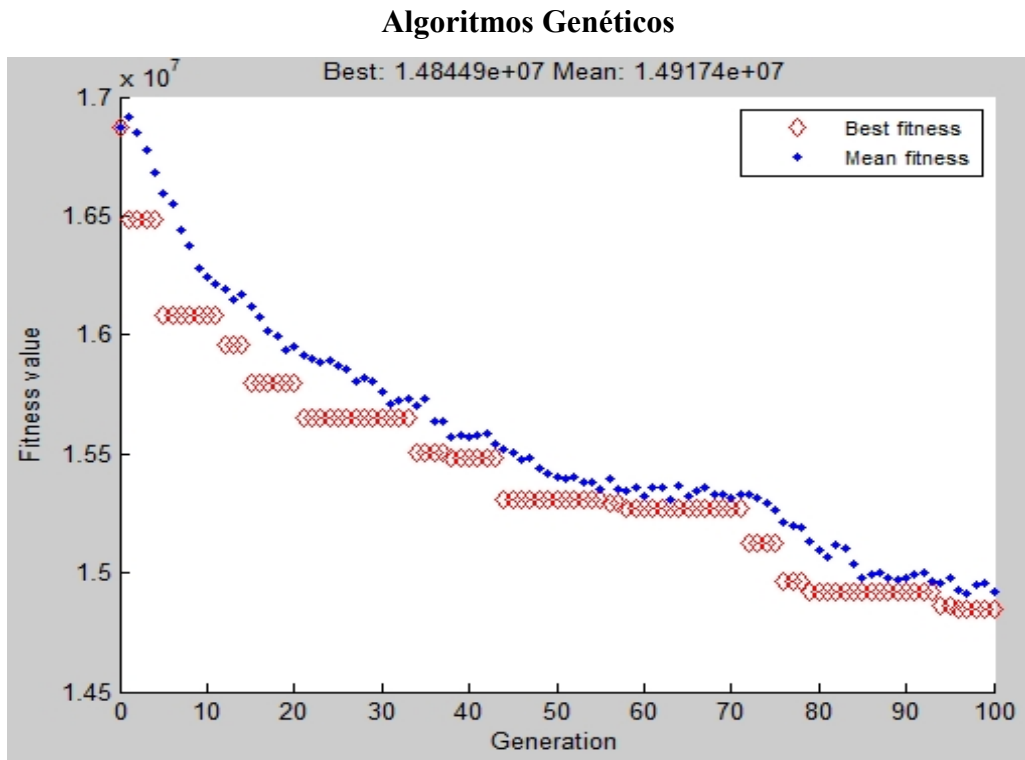


Figura 6.4: Buscando los mejores valores de la función aptitud para 1 ejecución de GA con 40 municipios con POP=20 y 100 generaciones. *Fuente: Elaboración propia.*

A continuación se indican los parámetros utilizados (ya comentados en el capítulo 5) en la programación del GA para un cálculo optimizado:

- 1- Se ha escogido un tamaño de población (POP) de 20, para evitar tiempos de ejecución excesivos, especialmente cuando se trabaja con

los 90 municipios. Un tamaño de población indica cuantos individuos hay en cada generación. Si es grande, permite buscar el espacio de soluciones más exhaustivamente evitando que el algoritmo devuelva un mínimo local en lugar del global. Sin embargo, un tamaño grande, también hace que el algoritmo funcione más lentamente.

- 2- Para crear una población inicial aleatoria se ha utilizado la función creación (*CreationFcn*: [@gacreationuniform](#)). Esta función crea una población inicial con una distribución uniforme. Los valores utilizados en esta distribución están en el rango de [-9999 a 10001] con variables enteras.
- 3- La selección de padres de hijos de cruce o mutación se ha programado a partir de un muestreo mediante la técnica de *torneo binario* según la cual se eligen 2 padres aleatorios y se queda el mejor según su función de aptitud (*fval*). Esto queda especificado con la función de selección ([@selectiontournament, 2](#)).
- 4- Con la opción de reproducción se especifica como el GA crea hijos para la siguiente generación (élite y cruce). Se elige un 5% de *hijos élite* (*EliteCount*), que son los individuos que sobrevivirán a la siguiente generación.

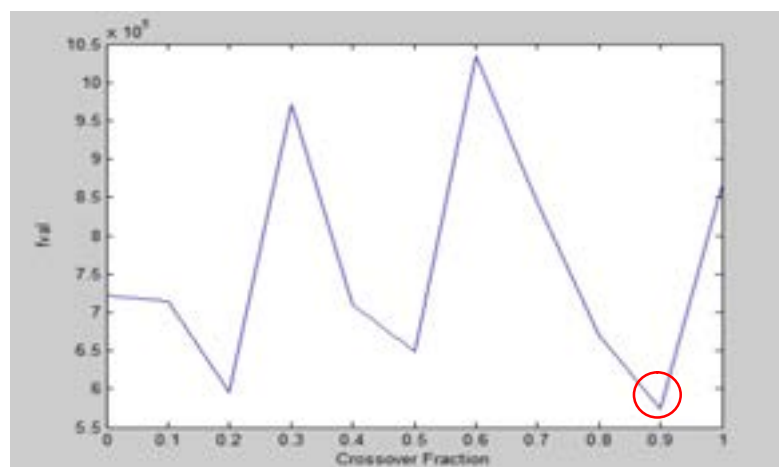


Figura 6.5: Resultado de función aptitud GA para distintos valores de “*CrossoverFraction*” en 6 municipios. *Fuente: Elaboración propia.*

En la Figura 6.5 se pueden ver los resultados obtenidos de la función aptitud (*fitness*) con GA para distintos valores de “*CrossoverFraction*”, sobre 6 municipios y 10 ejecuciones (en rojo se indica el valor usado para el GA). Se elige un *CrossoverFraction*=0.9 (entre 0 y 1), que es la fracción de la siguiente generación, distinta de los hijos élite, que será reproducida por cruce.

- 5- Con la función mutación `{@mutationuniform, 0.01}` se genera una mutación aleatoria de hijos en la población, que proporciona una diversidad genética para buscar la solución en un espacio más amplio. Se selecciona la fracción 0,01 de la población que va a ser sustituida por el número aleatorio de una distribución uniforme.
- 6- Finalmente, con la función `{@crossoverintermediate}` se combinan dos padres, para formar hijos de cruce para la siguiente generación. Los hijos se crean como la media ponderada de los padres.

Aunque hay más parámetros por programar se han mostrado los principales operadores a considerar en la programación del GA. En el Anexo B y C se incluyen todos los parámetros configurables, tanto para los GA como los algoritmos PSO.

Para el cálculo del óptimo con GA se han tratado dos tipos de soluciones:

- 1- Una primera solución, que considera de las restricciones diseñadas en el capítulo 4 solo las (4.3.2.8), (4.3.2.10), (4.3.2.11) y (4.3.2.12) como más relevantes por hacen referencia a la decisión de diseño definida por las variables binarias z_{kt} y v_{jt} , que permitirán buscar la mejor solución para la localización de instalaciones. Es decir, se consideran las inecuaciones de restricción del problema $A=[A3;A5;A6;A7]$ y $b=[A3;A5;A6;A7]$, dentro de la función (6.2.1.1) de MATLAB. Las variables enteras operativas x_{ijt} e y_{jkt} se darán por validas si se alcanza un valor de la función objetivo mínimo, dentro del espacio global óptimo. Esta solución se corresponde con la Figura 6.3.

2- Una segunda solución sería la no inclusión de restricciones en la función (6.2.1.1) que resuelve el GA, e incluirlas en la función objetivo *penalizando* la función para aquellos valores que no cumplan con las restricciones.

$$\begin{aligned} \min F(x, y, v, z) \\ \text{sujeta a: } g_i(x, y, v, z) \leq 0 \end{aligned} \quad (6.2.1.2)$$

La idea es transformar el problema (6.2.1.2) en otro sin restricciones, tal que su solución se aproxime a la ecuación (6.2.1.3).

$$\min V(x, y, v, z, u) = \min F(x, y, v, z) + \sum_i u_i P_i(g_i(x, y, v, z)) \quad (6.2.1.3)$$

Donde V es la función a minimizar que añade a la función original F , las funciones de penalización P_i , u_i son los parámetros que se van ajustando para acercar la minimización de V a la solución del problema. La principal característica de las funciones de penalización P es que tienden a cero cuando se satisfacen las restricciones, pero toman valores muy grandes cuando estas son violadas. Para esta segunda solución se ha optado por una penalización añadida tipo cuadrática asimétrica (6.2.1.4), para evitar discontinuidades en $g_i(x, y, v, z) = 0$.

$$P_i(g_i(x, y, v, z)) \begin{cases} 0 & \text{si } g_i(x, y, v, z) \leq 0 \\ u(g_i(x, y, v, z))^2 & \text{si } g_i(x, y, v, z) > 0 \end{cases} \quad (6.2.1.4)$$

- Los códigos de penalización que se muestran en **Código.2** se programarán dentro en la función objetivo incluida en la función (6.2.1.1) de MATLAB.

Código.2: Código de penalización

```
u = 5;
for i=1:length(fo)
    y=y+x(i)*fo(1,i);
    s=x(i);
    v=[v;s];
end
r=A(j,:)*v;
if r <= b(j,1)
    y;
end
if r > b(j,1)
    y=y+u*(r-b(j,1))^2;
end
```

Los resultados obtenidos con las dos técnicas anteriores son parecidos, si bien los resultados obtenidos con las funciones de penalización necesitan más tiempo de computación al incluir también las restricciones en la función objetivo.

Con esta técnica y en el caso de los 40 municipios se utilizan 968,94 segundos en su ejecución frente a los 109,7 segundos que se consumen con las restricciones.

6.2.2. Resultados en MATLAB con PSO

Los resultados obtenidos con MATLAB aplicando los algoritmos PSO para los diferentes ejemplos de problemas de localización de instalaciones se pueden ver en la Tabla 6.6.

Como se puede observar en la tabla, los valores promedios obtenidos de la función aptitud (*fval*) aplicando PSO mediante MATLAB son similares a los obtenidos con GA.

Tabla 6.6: Resultados aplicando algoritmos PSO con MATLAB para distintos municipios. *Fuente: Elaboración propia.*

PSO-Matlab	Nº. Municipios	Nº. variables	Fval	Tiempo(s)
Tesisresiduos_6	6	84	1,24E+06	0,17
Tesisresiduos_20	20	4200	1,21E+07	1,44
Tesisresiduos_40	40	16400	1,50E+07	4,14
Tesisresiduos_90	90	81900	2,28E+07	54,44

En el caso de PSO, MATLAB no tiene incorporada una toolbox oficial de este tipo de algoritmo en ninguna de sus versiones, por lo que se ha tenido que desarrollar un algoritmo básico cuyo código se incluye en el Anexo A. Este algoritmo se irá ampliando para incluir funciones utilizadas con los GA, que permitan trabajar con algunas de las restricciones que deciden el diseño representado por las variables v_{jt} y z_{kt} .

El hecho de trabajar con restricciones incluidas en la función PSO, en lugar de con penalizaciones, reduce sustancialmente los tiempos de ejecución, al igual que ocurría con los GA.

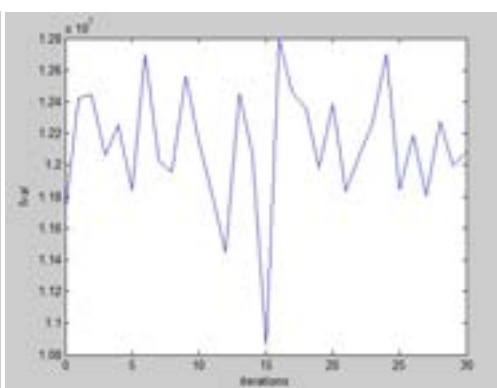
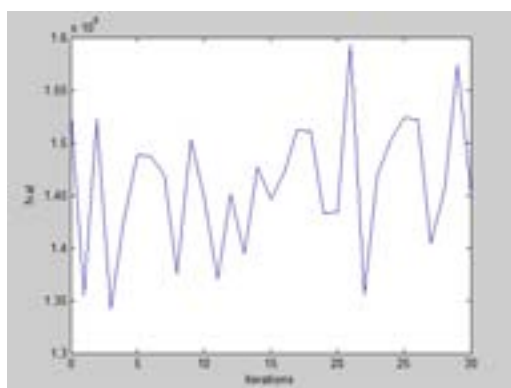
En la Figura 6.6 se pueden ver los resultados de la aplicación de la metaheurística PSO para distintos números de municipios, con resultados cercanos a los obtenidos con el GA, pero con tiempos de ejecución menores. Se consideran distintos números de municipios para el mismo algoritmo, pero realizando 30 ejecuciones de tipo PSO y presentando el resultado promedio como queda indicado en la Tabla 6.4.

Una ventaja adicional de este algoritmo es que el número de parámetros que se deben ajustar es pequeño por lo que su implementación, comparada con GA, es más sencilla.

Optimización por enjambre de partículas (PSO)

6 municipios

20 municipios



40 municipios

90 municipios

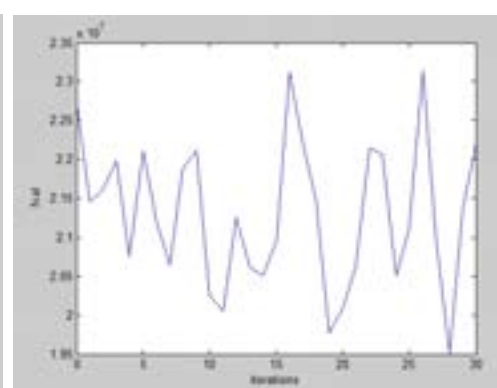
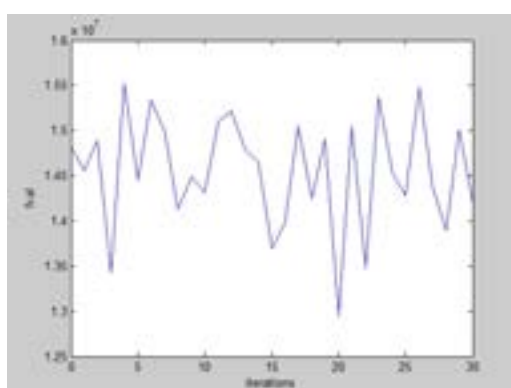


Figura 6.6: Resultado de los mejores valores de la función aptitud después de 30 ejecuciones del algoritmo PSO con restricciones, para 6, 20, 40 y 90 municipios. *Fuente: Elaboración propia.*

Como ya se expuso en la sección 5.5 del capítulo 5, una parte fundamental en la programación de la función PSO en MATLAB es escribir el código con las ecuaciones (5.5.5) y (5.5.6) que ajustan la velocidad y posición de cada partícula, de acuerdo con Eberhart y Shi (2001).

Igualmente, se debería cumplir que las condiciones en (5.5.10) estén satisfechas, para garantizar que el sistema converge a un punto de equilibrio estable (Pérez y Behdian (2007)).

Estas condiciones pueden cumplirse fácilmente asignando valores a los coeficientes de confianza e inercia que los hagan cumplir.

Por tanto, el código en MATLAB para ajustar la velocidad y posición de cada partícula se muestra en **Código.3**.

Código.3: *Ajuste velocidad y posición PSO*

```
% Update Velocity and Position
R1 = rand(N,Nvars); % random Number 1
R2 = rand(N,Nvars); % random Number 2
Velocity = W.*Velocity + C1.*(R1.*(LocalBestPosition-CurrentPosition))
+ C2.*(R2.*(GlobalBestPosition-CurrentPosition));
CurrentPosition = CurrentPosition + Velocity ;
```

Donde la mejor posición local y global corresponde a los valores de la mejor función aptitud (*fitness*) local y global buscada.

Los parámetros *R1* y *R2* son números aleatorios uniformemente distribuidos entre [0, 1], cuyo objetivo es emular el comportamiento estocástico que exhibe la población o enjambre, en cada iteración.

El valor inicial de la posición actual (*CurrentPosition*) se calcula rellenando una matriz con números de columnas igual al número de variables (81900, para el caso de 90 municipios) y 30 filas que son las semillas aleatorias del vector posición. Los números que se utilizan para rellenar esta matriz son obtenidos mediante el comando generador de números de distribución uniforme (*random ('unif')*), entre los límites inferior y superior del problema de localización de instalaciones.

El valor mínimo global se obtiene de los distintos mínimos locales obtenidos en sucesivas iteraciones de la función objetivo.

El pseudocódigo del algoritmo PSO se describe en **Pseudocódigo.3.2.1**.

Pseudódigo.3.2.1: Algoritmo PSO

1. Inicializar la población.
 2. La mejor posición previa es igualada a la posición actual.
 3. Cada posición es evaluada en la función de aptitud para determinar la calidad de la solución.
 4. Se compara la aptitud de la posición actual con la mejor previa.
 5. Determinar la mejor partícula del vecindario.
 6. Ajustar la velocidad.
 7. Ajustar la posición.
 8. Verificar si se cumple el criterio de parada.
-

6.3. Aplicación de metaheurísticas al caso real de la provincia de Málaga

En la sección anterior se obtuvieron resultados para diferente número de municipios comparando los resultados alcanzados con métodos exactos, GA y PSO.

El objetivo de aquella comparación es demostrar que el único método válido para resolver el problema de los 90 municipios (caso real) es mediante las dos técnicas metaheurísticas utilizadas en las secciones anteriores.

En la Tabla 6.7 se muestra un resumen de todos los valores vistos en las secciones anteriores, en la que se pueden comparar datos obtenidos con las distintas técnicas utilizadas. En negrita se indican los mejores tiempos y *fitness* por instancia.

Tabla 6.7: Resumen de resultados obtenidos con distintas técnicas y distintos municipios. *Fuente: Elaboración propia.*

	Programas	Nº. Municipios	Nº. variables	Fval	Tiempo(s)
V.Exacto Matlab	Tesisresiduos_6	6	84	9,64E+05	0,33
	Tesisresiduos_20	20	4200	1,03E+07	0,47
	Tesisresiduos_40	40	16400	1,46E+07	1,46
	Tesisresiduos_90	90	81900	-	-
V.Exacto AIMMS	Tesisresiduos_6	6	84	9,64E+05	0,01
	Tesisresiduos_20	20	4200	1,03E+07	0,05
	Tesisresiduos_40	40	16400	1,46E+07	0,59
	Tesisresiduos_90	90	81900	-	-
GA-Matlab	Tesisresiduos_6	6	84	9,98E+05	4,59
	Tesisresiduos_20	20	4200	1,16E+07	48,07
	Tesisresiduos_40	40	16400	1,58E+07	109,71
	Tesisresiduos_90	90	81900	2,37E+07	170,86
PSO-Matlab	Tesisresiduos_6	6	84	1,24E+06	0,17
	Tesisresiduos_20	20	4200	1,21E+07	1,44
	Tesisresiduos_40	40	16400	1,50E+07	4,14
	Tesisresiduos_90	90	81900	2,28E+07	54,44

En esta sección la función objetivo, restricciones y los dos metaheurísticos utilizados solo se aplicarán para resolver el caso real de gestión de residuos de los 90 municipios diseminados por la provincia de Málaga. Al final de la sección se presentará como quedaría dibujada la nueva red de distribución de RSU entre municipios, centros de transferencia y plantas de tratamiento con las dos técnicas metaheurísticas usadas.

El problema real que se pretende resolver consiste en la optimización de la red de recogida de RSU de 90 municipios de la provincia de Málaga, que junto a la Diputación de esta provincia, constituyen un “Consortio Provincial de Residuos Sólidos Urbanos”. Los RSU recogidos de los distintos municipios son transportados a distintos centros de transferencia y finalmente, valorizados y eliminados (si es el caso) en las plantas de tratamiento o complejos medioambientales.

Actualmente, existen dos plantas de tratamiento que reciben todos los RSU procedentes de los 90 municipios: La planta de tratamiento, clasificación y recuperación de Antequera (Complejo Medioambiental de Valsequillo) y la de Casarabonela.

También, existen cinco Centros de transferencia:

- 1- Centro de Transferencia de RSU de Vélez-Málaga (recibe residuos de 31 municipios) (136.000 T/año)
- 2- Centro de Transferencia de RSU de Ronda (recibe residuos de 21 municipios) (32.000 T/año)
- 3- Centro de Transferencia de RSU de Cártama (recibe residuos de 6 municipios) (74.300 T/año)
- 4- Centro de Transferencia de RSU de Archidona (recibe residuos de 6 municipios) (14.500 T/año)
- 5- Centro de Transferencia de RSU de Campillos (recibe residuos de 8 municipios) (15.500 T/año)

Sobre el escenario real descrito con la ubicación actual de cada instalación se plantea la hipótesis de reestructurar esta red de recogida de residuos, de forma que se minimicen los costes de gestión de estos residuos aplicando las ecuaciones diseñadas en el capítulo 4 y los metaheurísticos propuestos en el capítulo 5.

El número de municipios (90) y lo expuesto en las secciones del capítulo anterior definen la necesidad de aplicar técnicas metaheurísticas al problema. Evidentemente, tanto las fórmulas desarrolladas como los metaheurísticos utilizados son extrapolables a cualquier provincia española con una dimensión municipal parecida (≥ 90 municipios).

Tras diversas reuniones con distintos responsables de la Diputación de Málaga (Jefe de Explotación) y de la Gerencia de LIMASA (Director Gerente y Jefe de División de CAM), se obtuvieron los datos necesarios para poder resolver la hipótesis planteada en esta tesis, sobre el sistema actual de recogida malagueño.

Todos los datos recopilados están incluidos en el Anexo D y se corresponden con los parámetros necesitados de producción (P_{it}), distancias entre los 90 municipios (d_{ij} , d_{jk}), costes unitarios de transporte entre municipios, centros de transferencia y plantas de tratamiento (CMT_i , CTP_i), costes fijos (CF_{jt} , CF_{kt}) y capacidades anuales (QT_{jt} , QP_{kt}). En algún caso, los datos se han ajustado a las necesidades del diseño matemático, como en los siguientes casos:

- Los costes de transferencia entre centros de transferencia y plantas de tratamientos vienen indicados independientes como costes totales (€), toneladas transferidas (T), número de viajes y kilómetros realizados de ida y vuelta. Para el modelo matemático, estos costes deben venir en €/km·T. En la Tabla 6.8 se pueden ver algunos de estos datos.
- Los costes fijos de centros y plantas según LIMASA se estiman en €/T transferida. En el modelo solo se utilizarán €.
- En el modelo matemático no se consideran los costes de recogida de los contenedores dentro de cada municipio, solo el total de residuos transportados. Tampoco se consideran otros conceptos de costes de las instalaciones como pueden ser: costes de amortización de la instalación, de mantenimiento y reparación, iluminación, agua, generales, personal, etc. Solo se consideran los costes en €/T transferida.

Los datos manejados corresponden a un periodo de cinco años, entre el 2010 y el 2014.

Tabla 6.8: Costes sin estaciones, entre centros de transferencia y plantas tratamiento en Málaga 2013. Fuente: Diputación de Málaga.

SIN ESTACIONES	Transferencia de Residuos a estaciones 2013				
	Circuito	COSTE TOTAL	Tn 2013	Nº de Viajes	Km iv
Valsequillo	Archidona-Antequera	14.464,73	8.009,90	548	66
	Campillos-Antequera	30.149,67	10.525,81	671	68
	Ronda-Antequera	453.055,88	23.292,04	1.245	188
	Vélez-Antequera	1.687.335,80	66.578,43	3.619	188
Casarabonela	Cártama-Casarabonela	658.212,38	47.693,53	2.861	70

Todos los resultados obtenidos hasta ahora con los distintos algoritmos se han alcanzado con un ordenador *Intel Core i7-4720-HQ* (doble núcleo) con *CPU 2,60 GHz, 1 TB HD, 16 GB de RAM*, bajo Windows ($\times 64$).

Como inicio para resolver el problema se considerará la restricción de que el número de plantas de tratamiento sean dos a fin de poder comparar los datos obtenidos, al resolver el problema de optimización, con los datos reales. Para la ubicación de los centros de transferencia se considerarán todos los municipios como candidatos.

Las mejores soluciones obtenidas con cada uno de los metaheurísticos se muestran en la Tabla 6.9, y corresponden al promedio de 30 ejecuciones por cada algoritmo.

Tabla 6.9: Resultados aplicando algoritmos PSO y GA con MATLAB para 90 municipios. *Fuente: Elaboración propia.*

PSO-GA	Nº. Municipios	Nº. variables	Fval	Tiempo(s)
GA_tesisresiduos_90	90	81900	2,37E+07	7477,20
PSO_tesisresiduos_90	90	81900	2,28E+07	1424,40
Tesisresiduos_90	90	81900	-	-

Se ha partido con una población de 20 individuos y se han establecido 100 generaciones, antes de detener la ejecución de los algoritmos.

El resto de parámetros de cada técnica quedaron especificados en el capítulo anterior. No se conoce solución óptima, ya que con el método exacto no se ha podido alcanzar.

En las Figuras 6.7 y 6.8 se muestran los resultados obtenidos tras una ejecución con cada uno de los metaheurísticos.

Optimización por enjambre de partículas (PSO)

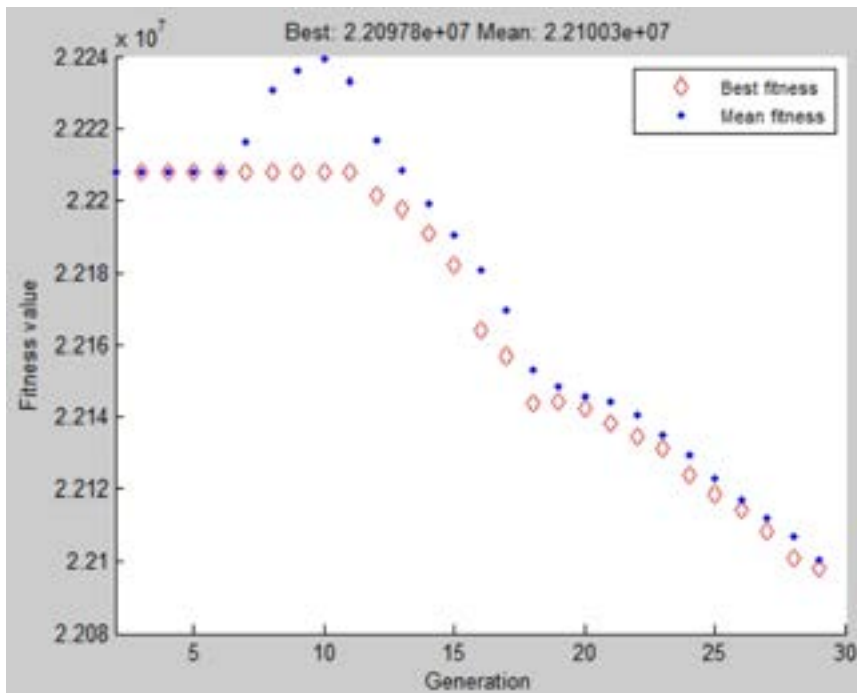


Figura 6.7: Resultado de los mejores valores de la función aptitud para una ejecución con PSO, para 90 municipios. *Fuente: Elaboración propia.*

Optimización GA 30 generaciones

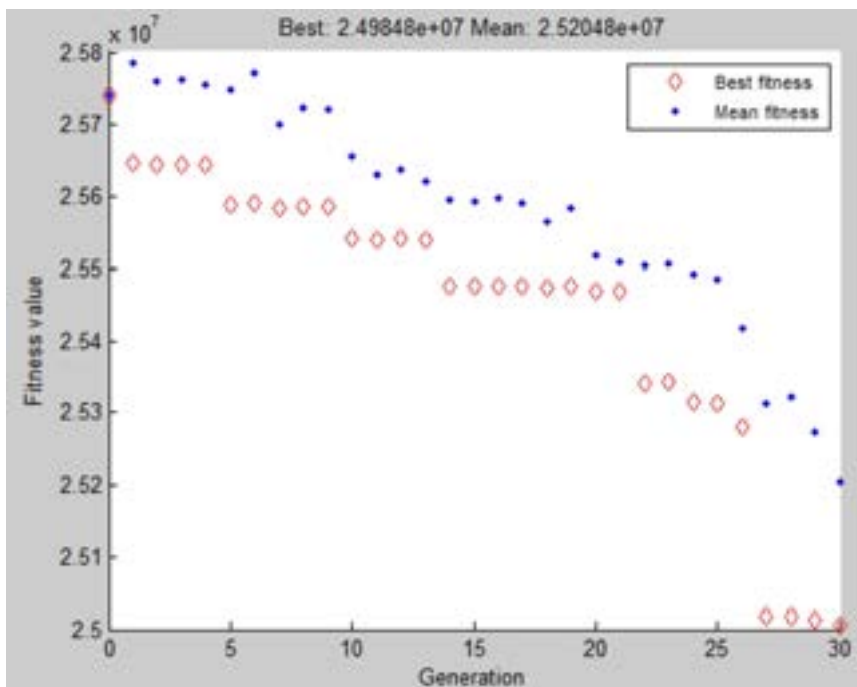


Figura 6.8: Resultado de los mejores valores de la función aptitud para una ejecución de GA y 30 generaciones, para 90 municipios. *Fuente: Elaboración propia.*

Optimización GA 100 generaciones

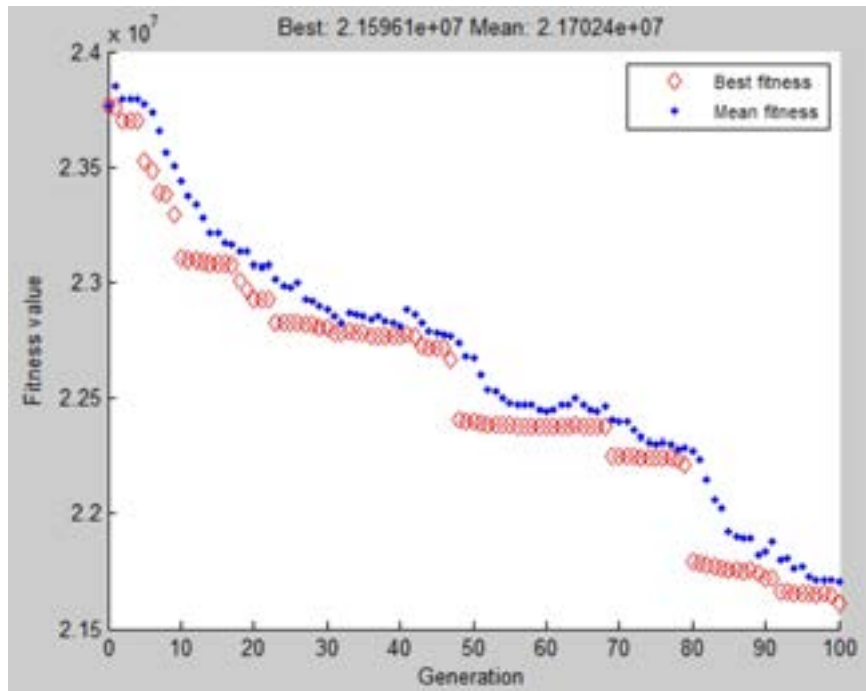
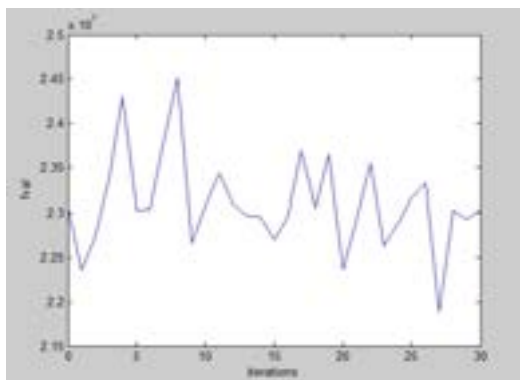


Figura 6.9: Resultado de los mejores valores de la función aptitud para una ejecución de GA y 100 generaciones, para 90 municipios. *Fuente: Elaboración propia.*

En la Figura 6.9 se ve la evolución de los mejores valores de GA tras 100 ejecuciones.

Optimización por PSO



Optimización por GA

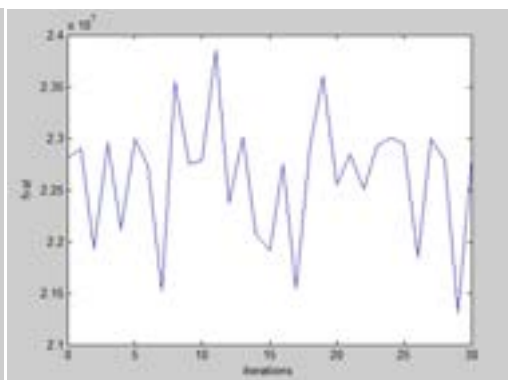


Figura 6.10: Resultado de los mejores valores de la función aptitud para 30 ejecuciones con PSO y GA para 90 municipios. *Fuente: Elaboración propia.*

En la Figura 6.10 se muestra la evolución de los mejores valores tras 30 ejecuciones con cada uno de los metaheurísticos. Los valores promedio de estas últimas ejecuciones se corresponden con los datos mostrados en la Tabla 6.9.

Aunque los resultados obtenidos con las dos técnicas son parecidos, el algoritmo PSO se muestra más eficiente que los GA, al resolver problemas grandes (matriz de $90 \times 90 \times 5$ variables) con bastante menos tiempo de computación y con la facilidad de tener que configurar menos parámetros.

La verificación de los resultados de las figuras anteriores se realiza aplicando la prueba de la “*t de Student*” para una muestra. La prueba se aplica a tamaños de 40 y 90 municipios por ser más significativos. El valor contraste, en el caso de 40 municipios, se realiza entre el óptimo alcanzado con métodos exactos (las 30 muestras iguales) y los dos metaheurísticos. En el caso de 90 municipios, el valor contraste es medido entre las muestras obtenidas con la técnica GA y los valores *fitness* de la muestra PSO.

Se toman como muestra los 30 valores *fitness* de cada técnica metaheurística (GA y PSO), y el *t-test* evalúa si es razonable mantener la *Hipótesis nula* de que las dos muestras de GA y PSO son similares o rechazarla, lo que supondría aceptar la *Hipótesis alternativa* de que las dos muestras son distintas en un 95%, en cuyo caso se elegirá la técnica de mejor *fitness*.

Como se trata de un test paramétrico, las variables analizadas siguen una distribución normal. Es decir, los valores *fitness* obtenidos dibujarán un estadístico que sigue la distribución *t de Student*.

Si cae en una zona central de esa distribución de probabilidad (el 95%, porcentualmente) se mantendrá la hipótesis nula. Si cae fuera de esa zona se rechazará y se optará por la alternativa.

Los resultados son obtenidos aplicando el comando de MATLAB mostrado en (6.2.9).

$$[h,p,ci,stats] = ttest (fval, foptim, 'Alpha', 0.05) \quad (6.2.9)$$

Donde:

h: valor del test de decisión de hipótesis nula. El resultado de *h* es 0 si el test acepta la hipótesis nula mayor del 5% del nivel de significancia; 1 si el test la rechaza (hipótesis alternativa).

p-value: valor escalar en el rango [0,1]. *P-value* es la probabilidad de error.

ci: Intervalo de confianza de que el valor medio sea cierto. Se mueve entre dos límites.

stats: Test estadístico. Tiene la siguiente estructura:

-*tstat* — Valor del test.

-*df* — Grados de libertad del test (nº. muestras-1).

-*sd* — Desviación de la población estándar estimada.

Una hipótesis nula cierta supone que las muestras de los dos metaheurísticos son estadísticamente equivalentes. La hipótesis alternativa, obviamente es que son diferentes.

Como en todo contraste de hipótesis se declara que el efecto es estadísticamente alcanzado en estos estudios de investigación, si la significación calculada (*'Alpha'*=0.05 en Matlab) es inferior al 5%.

Los contrastes se realizan calculando las diferencias existentes entre los valores *fitness* de GA, PSO y el valor óptimo conseguido con los métodos exactos para 40 municipios y por otro lado, entre las muestras obtenidas con las técnicas metaheurísticas para 90 municipios. Para esto se utiliza la prueba *t-student* para una muestra. Si al calcular los valores *fitness* no son consistentes con el valor óptimo, la hipótesis nula es rechazada.

Los resultados obtenidos con el “*t-test*” de MATLAB se pueden ver en la Tabla 6.10, donde se muestran tanto los resultados para 40 municipios entre las 30 muestras GA y PSO contrastadas con las 30 muestras del método exacto (30 muestras iguales), como los resultados de la prueba *t-student* entre las dos muestras de los algoritmos GA y PSO para 90 municipios. El valor contraste a analizar en el caso de 90 municipios se realiza entre las muestras de 30 ejecuciones del algoritmo PSO, frente a otras 30 del GA.

Tabla 6.10: Resultados prueba *t-student* para 40 y 90 municipios entre método GA, PSO y exacto. *Fuente: Elaboración propia.*

PRUEBA T-STUDENT	V. medio fitness_GA/PSO	Valor hipótesis nula (h)	Probabilidad (p-value)	Intervalo confianza (ci): 1.0e+05*	
90 municipios:GA-PSO	2.3663e+07/2.2822e+07	1	0,008	-14,4550	-2,3630
40 municipios:GA-exacto	1.5190e+07/1.4631e+07	1	0,0399	0,2750	10,8980
40 municipios:PSO-exacto	1.4997e+07/1.4631e+07	1	0,0021	1,4566	5,9662

ESTADÍSTICA (stats)	Valor test	Grados libertad	Desviación
90 municipios:GA-PSO	tstat: -2.8406	df: 30	sd:1.6483e+06
40 municipios:GA-exacto	tstat:2.1482	df: 30	sd:1.4479e+06
40 municipios:PSO-exacto	tstat:3.3616	df: 30	sd:6.1472e+05

El valor obtenido en el nivel de significancia (*p-value*) nos permite decidir si se rechaza o no la hipótesis nula. De los valores mostrados en la Tabla 6.10 interesa analizar los valores mostrados en el área de “Prueba *t-Student*”. El valor de la hipótesis nula en las tres comparaciones es 1, lo que supone que las muestras contrastadas son distintas entre si, con una probabilidad de error menor del 5%. En el caso de contraste entre los dos metaheurísticos para 90 municipios, el valor de su nivel de significancia o *p-value* es de 0.008. La menor dispersión y valor medio mostrados por PSO en este estudio estadístico confirman un mejor comportamiento del algoritmo PSO en comparación con GA.

Por otro lado, los resultados de comparar los dos algoritmos con el valor exacto para 40 municipios supone que aunque se rechaza la hipótesis nula, considerando los resultados que aparecen en el área de datos estadísticos de la

Tabla 6.10, se puede apreciar que el valor de desviación estándar (sd) obtenidos para GA en 40 municipios es mucho mayor que el obtenido con PSO (40 municipios) donde el 50% de valores están más próximos a su media. También, el intervalo de confianza (ci) de GA es bastante más abierto que el de PSO. Esto significa que PSO es más robusto y fiable en la práctica para la toma de decisiones.

En la Figura 6.11 se puede ver gráficamente lo anteriormente expuesto. Se muestra un resumen gráfico de los resultados estadísticos obtenidos mediante el “Diagrama de cajas (*boxplot*)”. La figura presenta una comparativa entre las distintas técnicas metaheurísticas y los valores exactos para los distintos tamaños de municipios (6, 20 40 y 90).

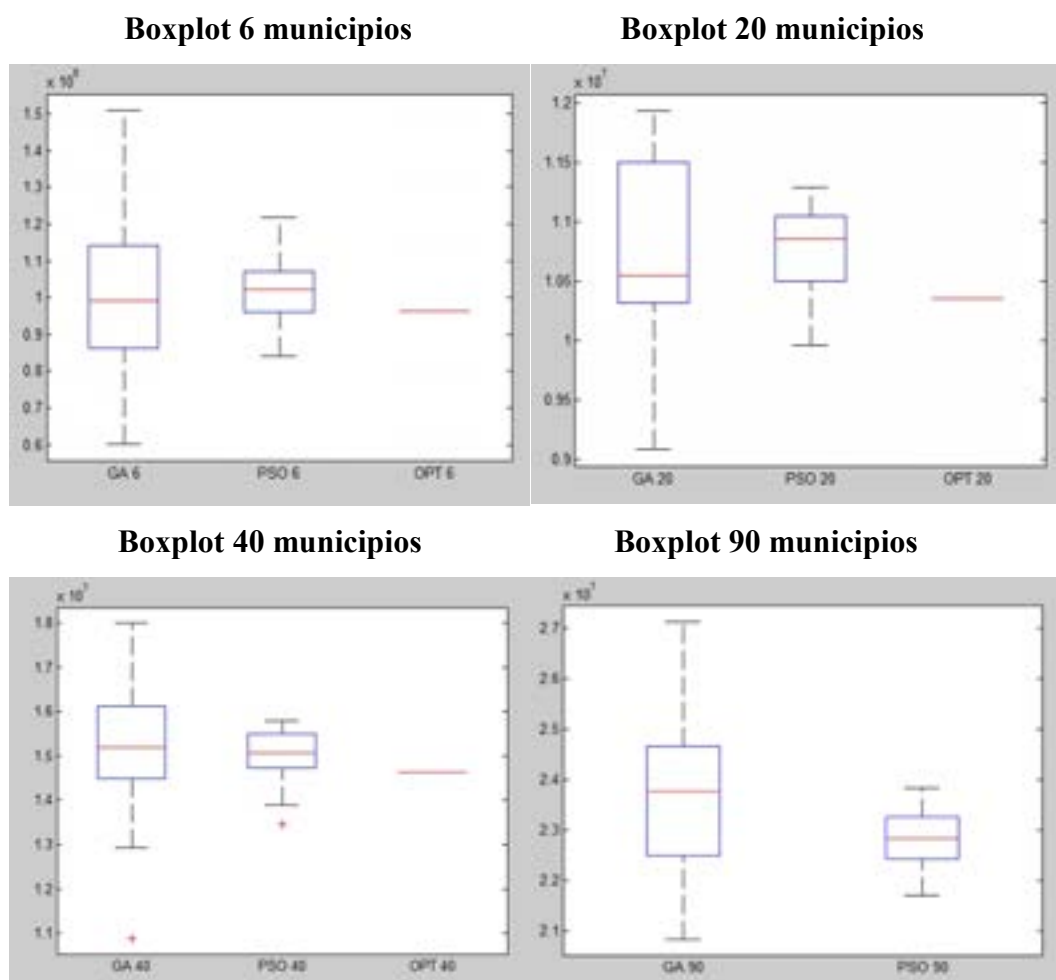


Figura 6.11: Resultados del “Diagrama de cajas” para 6, 20, 40 y 90 municipios entre métodos GA, PSO y exacto (OPT). Fuente: *Elaboración propia*.

Los resultados de los estudios estadísticos confirman la validez de las dos técnicas utilizadas con valores próximos al óptimo. Aunque el valor de la mediana en GA está algo más cerca del valor exacto para el caso de 20 municipios, según se puede observar en el diagrama de cajas de la Figura 6.11 (sin embargo, muestra una asimetría positiva bastante acusada), en general los resultados obtenidos con GA se presentan peores frente a PSO debido a una mayor dispersión (en el diagrama de cajas se ven mayores valores de RIC (recorrido intercuartílico) y de máximos y mínimos, en GA) y también una desviación estándar e intervalos de confianza mayores según el test *t-student*.

Los datos obtenidos revelan que el algoritmo PSO se muestra más eficaz buscando el óptimo global (aproximado) en un espacio factible más reducido que con el método GA. Además, GA presenta un elevado tiempo de computación y mayor complejidad en la parametrización de operadores.

En la Tabla 6.11 se muestran los costes totales actuales y reales del transporte del RSU desde los 90 municipios a los 5 centros de transferencia y las 2 plantas de tratamiento. Estan calculados según el modelo matemático diseñado en el capítulo 2 y los datos aportados por Diputación de Málaga. Los cálculos se han realizado para un periodo, el correspondiente al año 2014 para facilitar la comparación con los datos obtenidos con las técnicas metaheurísticas propuestas.

Tabla 6.11: Coste año 2014 de 90 municipios según distribución actual.
Fuente: *Elaboración propia.*

$X_{qt} \cdot d_{qt} \cdot CMT_t$	$Y_{jt} \cdot d_{jt} \cdot CTP_t$	$V_{jt} \cdot CF_{jt}$	$Z_{kt} \cdot CF_{kt}$	=	Coste Total
3847807	756	985360	430276	=	5264199

Los resultados obtenidos (media de 30 ejecuciones) con los metaheurísticos utilizados en esta tesis y para el periodo (2014) se pueden ver en en la Tabla 6.12.

Tabla 6.12: Coste 2014 aplicando algoritmos PSO y GA para los 90 municipios. *Fuente: Elaboración propia.*

PSO-GA	Nº. Municipios	Periodo	Coste Total
PSOtesisresiduos90_1	90	2014	4,52E+06
GAtesisresiduos90_1	90	2014	4,64E+06

Los resultados obtenidos con los metaheurísticos (Tabla 6.12) demuestran un ahorro de al menos 740.000 euros (20% anual) en el año 2014 y un ahorro final para el periodo analizado entre 2010 y 2014, cercano a los 3,7 millones de euros.

En las Figuras 6.12 y 6.13 se muestran los municipios, los centros de transferencia y las plantas de tratamiento en la actualidad.



Figura 6.12: Centro de transferencia de Cártama y Planta de tratamiento de Casarabonela. *Fuente: Elaboración propia.*

En la Figura 6.12 se quiere indicar que los 6 municipios incluidos en la columna de color verde transportan sus residuos al centro de transferencia de Cártama (rectángulo azul). Los 7 municipios incluidos en la columna de color naranja claro transportan directamente sus RSU a la planta de tratamiento de Casarabonela (rectángulo naranja oscuro). Los residuos recogidos en las instalaciones de Cártama son transportados a la planta de Casarabonela.

En la Figura 6.13 se puede observar como están distribuidos los municipios entre los centros de transferencia y planta de tratamiento. Cada columna está encabezada por un centro de transferencia (azul) quedando debajo, aquellos municipios (en verde) que le transfieren sus residuos. Algunos municipios (columna amarillo claro) transportan directamente sus RSU a la planta de tratamiento en Antequera (amarillo). Es decir, a la planta de Antequera llegan

11 municipios sin pasar por los centros de transferencia. Estos centros de transferencia transportan los residuos recibidos de los distintos municipios a la planta de Antequera.

				VELEZMALAGA (30)	
				ALCALÓN	
				ALFARÑATE	
				ALGARROBO	
				ALMACHAR	
				ARCHEZ	
				ARENAS	
				ARFARÑATEJO	
			RONDA (22)	BENAMARGOSA	
			ALGATOÓN	BENAMOCARRA	
			ALPANDEIRE	BORGE (EL)	
			ARRIATE	CANILLAS ACEITUNO	
			ATAJATE	CANILLAS ALBAIDA	
			BENALADID	COMARES	
			BENALAUPIA	CÓMPETA	
			BENALJÁN	CÚTAR	
			BENARRABÁ	FRIGILIANA	
			CARTAJIMA	IGNATE	
			CORTES FRONTERA	MACHARAVIAYA	ANTEQUERA (11)
			FARAJÁN	MOCLINEJO	ALAMEDA
			GALCÓN	NERJA	ALMOGA
			GENALGUACIL	PERIANA	ANTEQUERA
		CAMPILLOS(8)	IGUALEJA	RINCON DE LA VICTORIA	CASABERMEJA
		ALMORCEN	JIMERA DE LIBAR	FIGORRO	COLMENAR
ARCHIDONA (6)	ARDALES	JUBRIQUE	SALAPES	CUEVAS SAN MARCOS	
ARCHIDONA	CAMPILLOS	JÚZCAR	SAYALONGA	FUENTE PIEDRA	
CUEVAS BAJAS	CAÑETE LA REAL	MONTEJAQUE	SEDELLA	HUMILLADERO	
VILLANUEVA ALGAIDAS	CARRATRACA	PARALUTA	TORROX	MOLLINA	
VVA CONCEPCIÓN	CUEVAS DEL BECERRO	PLUERRA	TOTALÁN	VALLEABDALAIS	
VVA DE TAPA	SIERRA DE YEGUAS	RONDA	VELEZ MÁLAGA	VVA CONCEPCIÓN	
VVA DEL TRABUCO	TEBA	SIERRATO	VIÑUELA	VVA DEL ROSARIO	

Figura 6.13: Centros de transferencia (azul) y Planta de tratamiento de Antequera (amarillo). *Fuente: Elaboración propia.*

En la Figura 6.14 se muestra (con trazo rojo) como están distribuidos geográficamente en la actualidad los centros de transferencias y plantas de tratamiento.

En la Figura 6.15 se muestra como quedarían distribuidos por la geografía malagueña según los resultados obtenidos con los dos metaheurísticos (en trazo rojo las soluciones GA y en azul las PSO).

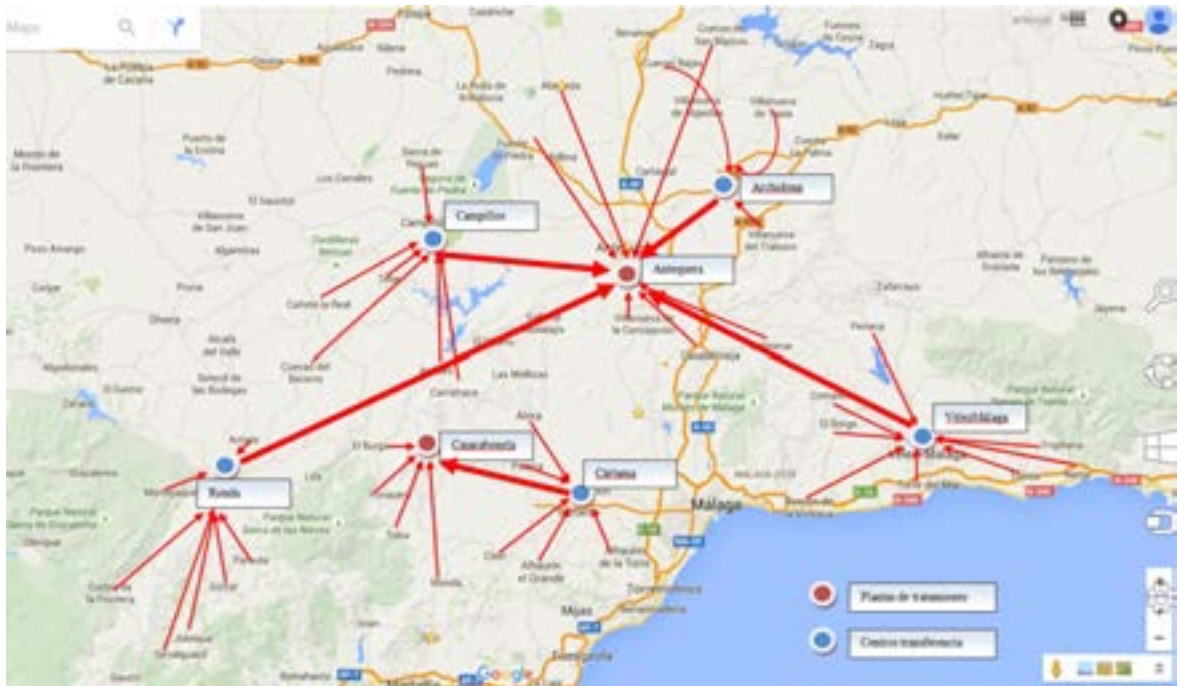


Figura 6.14: Distribución del transporte RSU en la actualidad, en la provincia de Málaga. *Fuente: Elaboración propia.*

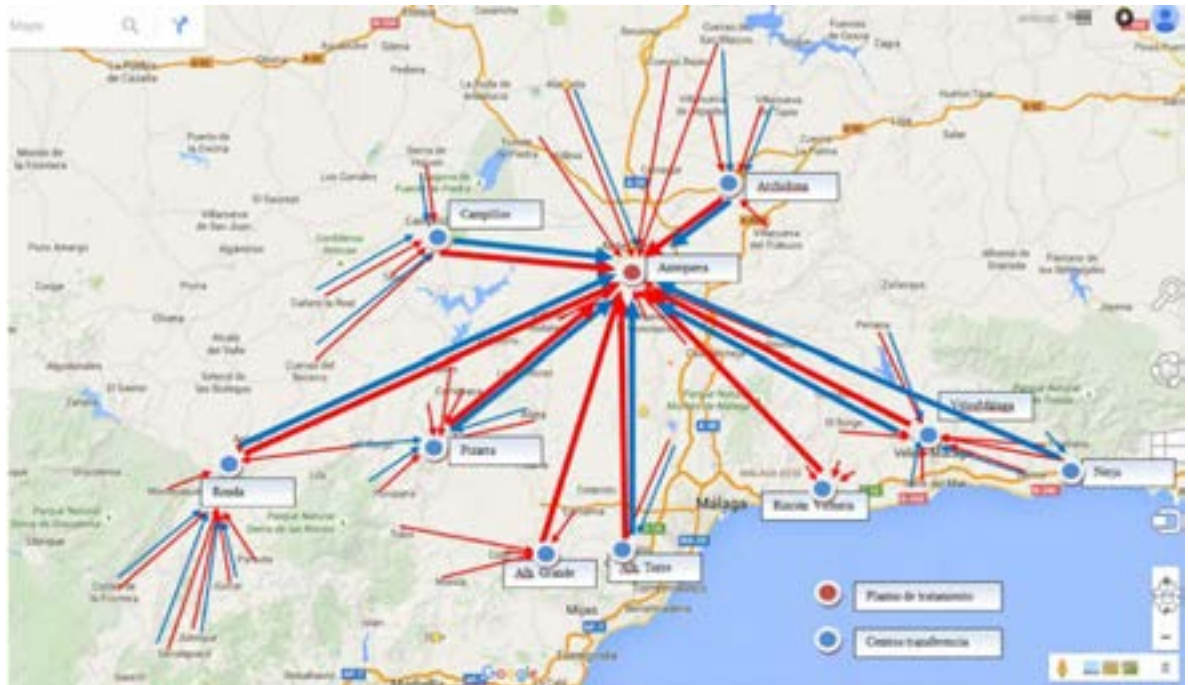


Figura 6.15: Propuesta de distribución del transporte RSU en la provincia de Málaga, según cálculos obtenidos con metaheurísticos GA (rojo) y PSO (azul). *Fuente: Elaboración propia.*

El óptimo en el caso de las plantas de tratamiento y como se podía intuir en el diseño del modelo matemático se corresponderá con el mínimo número de ellas. Es decir, solo será necesaria una planta de tratamiento manteniéndose en este caso, la de Antequera.

Esta nueva distribución considerando que el objetivo último era una minimización de costes supondrá, en base a los datos obtenidos, un ahorro final para el periodo analizado entre 2010 y 2014 cercano al 20% del coste total.

En la Figura 6.16 se puede observar como quedarían distribuidos los municipios entre los nuevos centros de transferencia localizados mediante los métodos metaheurísticos. Cada columna está encabezada por un centro de transferencia (azul), quedando debajo aquellos municipios (verde) que le transfieren sus residuos.

Según se ve en la Figura 6.17 se han abierto 5 nuevos centros de transferencia manteniendo 4 de los 5 existentes y se ha cerrado una planta de tratamiento. Hay que considerar que los Fondos de Cohesión de la Unión Europea financian el 80% de la construcción de nuevas estaciones de transferencia, por lo que el coste total de cada nueva instalación sería inferior a 100.000 € (Diputación (2015)). La nueva organización quedaría como se muestra a continuación:

Se mantienen:

- 1- Centro de Transferencia de RSU de Vélez-Málaga (18 municipios)
- 2- Centro de Transferencia de RSU de Ronda (24 municipios)
- 3- Centro de Transferencia de RSU de Archidona (7 municipios)
- 4- Centro de Transferencia de RSU de Campillos (6 municipios)

Los nuevos centros serían:

- 5- Centro de Transferencia de RSU de Alhaurín Torre (2 municipios)
- 6- Centro de Transferencia de RSU de Alhaurín Grande (6 municipios)
- 7- Centro de Transferencia de RSU de Nerja (6 municipios)
- 8- Centro de Transferencia de RSU de Pizarra (7 municipios)
- 9- Centro de Transferencia de RSU de Rincón de la Victoria (4 municipios)

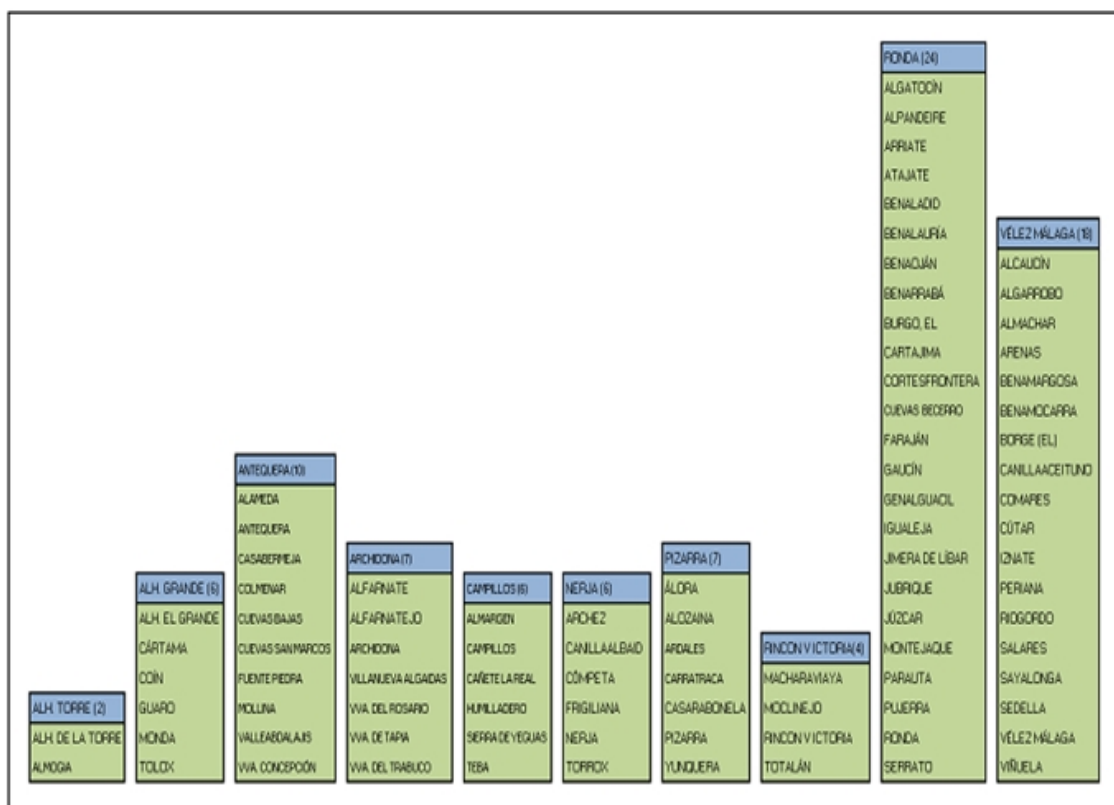


Figura 6.16: Nueva distribución de los centros de transferencia en la provincia de Málaga. *Fuente: Elaboración propia.*

El análisis de optimización de costes sugiere utilizar el menor número de plantas de tratamiento posibles, por lo que la planta de tratamiento de Casarabonela se cerraría, manteniendo solo la de Antequera:

10- Planta de Tratamiento de RSU de Antequera (10 municipios)

En la Figura 6.17 se muestra como quedaría la nueva distribución de centros de transferencia (círculo naranja) y planta de tratamiento (círculo verde) según los resultados obtenidos con los metaheurísticos (en trazo en azul se indican las transferencias de residuos entre centros y planta). También se dibuja la situación actual de los centros de transferencia (círculo azul) y plantas de tratamiento (círculo rojo) y como se realiza la transferencia de residuos (en trazo en rojo se indican las transferencias de residuos entre centros y plantas).

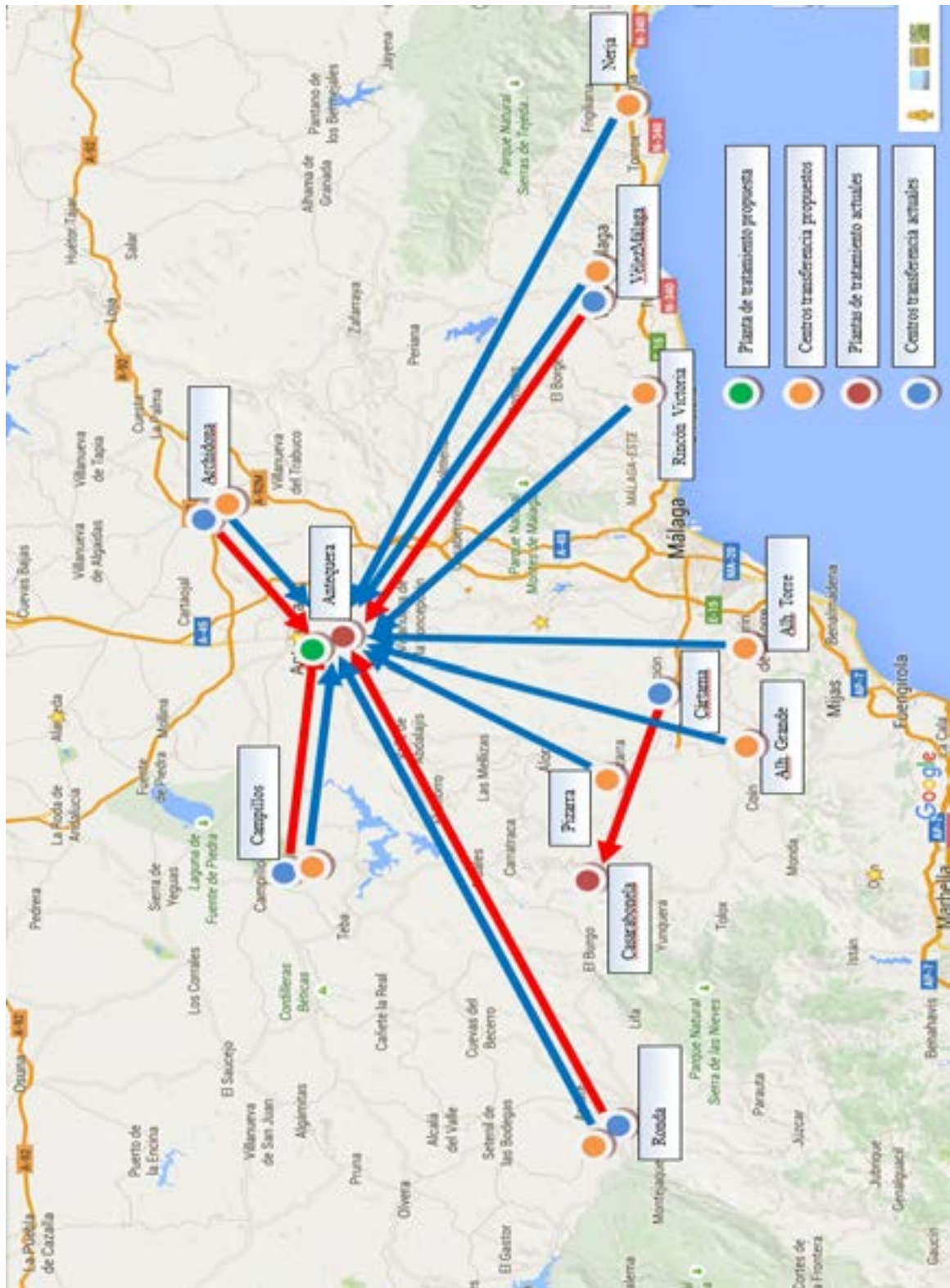


Figura 6.17: Comparativa entre la actual distribución y la propuesta por metaheurísticos, de los centros de transferencia y planta de tratamiento en la provincia de Málaga. *Fuente: Elaboración propia.*

6.4. Conclusiones

En este capítulo se ha descrito la herramienta informática utilizada para modelar y resolver tanto el modelo exacto, como las técnicas metaheurísticas propuestas. Se han discutido las ventajas e inconvenientes de usar una herramienta informática como es el lenguaje matemático MATLAB, o el uso de otras herramientas comerciales como son el lenguaje de simulación y modelación matemática GAMS ó AIMMS.

Los resultados obtenidos con GA se presentan peores frente a PSO debido a una mayor dispersión (en el diagrama de cajas se ven mayores valores de RIC y de máximos y mínimos, en GA) y también una desviación estándar e intervalos de confianza mayores según el test *t-student*. En el caso del algoritmo GA, los tiempos computacionales para resolver el problema de 90 municipios son más elevados que con PSO (7500 segundos frente a los 1470 segundos de PSO).

Los resultados obtenidos con los metaheurísticos demuestran un ahorro de al menos 740.000 euros (20% anual) en el año 2014 y un ahorro final para el periodo analizado entre 2010 y 2014, cercano a los 3,7 millones de euros.

Después de comprobar que los resultados obtenidos con la aplicación de las técnicas metaheurísticas propuestas han permitido resolver el problema determinista del caso real de localización de instalaciones y optimización de costes planteado en el capítulo, se puede confirmar la validez de este enfoque, de las técnicas metaheurísticas utilizadas y la herramienta aplicada.

Capítulo 7

7. CONCLUSIONES Y FUTUROS DESARROLLOS

En este capítulo se describen las conclusiones más importantes respecto de la tesis realizada y otros posibles futuros desarrollos.

7.1. Conclusiones

En resumen, las conclusiones más importantes son las siguientes:

- Este trabajo de investigación supone un avance por el uso de metaheurísticos actuales como los algoritmos de PSO no utilizados hasta ahora, para el que se ha constatado en esta tesis su interesante eficiencia en problemas reales de localización.
- Se introducen consideraciones de carácter más realista de lo que generalmente se ha presentado en la literatura. Este hecho queda plasmado en los resultados obtenidos con los metaheurísticos para el caso real de los 90 municipios de la provincia de Málaga. Los resultados muestran un ahorro para el periodo de 5 años, de unos 3,7 millones de euros, lo que supone un 20% de ahorro anual.

- Aunque existen bastantes investigaciones sobre teoría de localización de instalaciones, son pocas las que se han encontrado que incluyan esta teoría aplicada a la logística inversa. Así, en la revisión sobre el estado del arte de las distintas tesis nacionales e internacionales y de artículos, congresos o conferencias de ámbito mundial, se han encontrado muy pocas que propongan, dentro el marco de la logística inversa, modelos cuantitativos de localización de instalaciones y uso de técnicas metaheurísticas para resolver el problema NP-duro planteado. En este sentido, esta tesis profundiza en la aún tenue línea de trabajo internacional con metaheurísticas en este dominio.
- Así mismo, el algoritmo PSO se ha mostrado más efectivo, rápido y fácil de programar que GA. Este último, con problemas pequeños presenta una respuesta similar a PSO, pero para grandes problemas se ha comportado bastante peor en tiempos de ejecución. También el algoritmo PSO en los estudios estadísticos con *t-student* y diagramas de cajas ha demostrado una menor dispersión para las diferentes muestras de municipios utilizadas (percentiles al 25 y 75% más cercanos a la mediana e igualmente para los valores máximos y mínimos). Esto supone una mayor robustez y confiabilidad, aspectos clave para su uso real.
- Los resultados finales obtenidos con las dos técnicas metaheurísticas establece unos valores cercanos entre si y próximos a los valores exactos. Aunque PSO tiene la ventaja de que es más sencillo de programar y rápido de ejecutar, con valores *fitness* para 30 muestras que presentan mucha menor dispersión. GA tiene desviaciones (*sd*) en la distribución en *t* sobre el valor exacto, de 1.4479e+06 frente a los 6.1472e+05 de PSO.
- También PSO localiza el óptimo en un espacio factible más reducido, con un 50% de los valores *fitness* más cercanos a la mediana que en el caso GA. Por otra parte, GA tiene la dificultad añadida de su complejidad para parametrizar adecuadamente los distintos operadores.

- Finalmente, y después de contrastar que los resultados obtenidos con la aplicación de las técnicas metaheurísticas propuestas, se puede confirmar la validez de este enfoque, de las técnicas metaheurísticas utilizadas y la herramienta diseñada, frente al uso de métodos exactos, ineficaces para grandes problemas.

7.2. Futuros desarrollos

En este trabajo de investigación se han abordado problemas de tipo mono-objetivo exactos junto con la aplicación de dos técnicas metaheurísticas. El método GA, según se ha comprobado en la revisión de la literatura, ha sido utilizado en alguna ocasión en logística inversa para la optimización de costes y localización de instalaciones con resultados aceptables. Sin embargo, de la técnica PSO no hay referencias en el estado del arte sobre su aplicación en logística inversa.

Desde este punto de partida, se podrían sugerir otros posibles futuros desarrollos como:

- Utilizar otras herramientas informáticas, como Java, C++, Python, etc., que mejoren los tiempos de ejecución de los metaheurísticos usados para resolver problemas difíciles, con gran número de municipios (≥ 90). En el caso de la técnica GA, los tiempos computacionales para resolver el problema con todas las restricciones siguen siendo elevados. Por otra parte, el uso de otras aplicaciones informáticas podrían reducir la dispersión numérica mostrada en GA.
- Aplicar los metaheurísticos para la localización de instalaciones utilizando un enfoque estocástico, considerando el carácter dinámico del problema, ya que esta tesis se ha limitado al caso estático.
- Utilizar técnicas híbridas, como algoritmos meméticos y búsqueda dispersa o algoritmos genéticos híbridos (AGHMO).

- Extender las técnicas metaheurísticas al dominio multi-objetivo, lo que permitiría resolver problemas de optimización más complejos, como la metodología denominada MOPSO (Multi-Objective PSO) presentada por Coello et al. (2002).
- Por último, considerando el avance tecnológico en las redes de computadoras, se propone el uso de metaheurísticas paralelas, que reducirían el tiempo de cálculo, la resolución de problemas de mayor tamaño en un tiempo dado y obtener soluciones de mejor calidad.

BIBLIOGRAFÍA

Abdel-kader, R. F. (2011). *An Improved Discrete PSO with GA Operators for Efficient QoS-Multicast Routing*. International Journal of Hybrid Information Technology. Vol. 4, No. 2, pp. 23-38.

Afsahi, Z., Javadzadeh, R., Meybodi, M. (2011). *Hybrid Model of Particle Swarm Optimization and Cellular Learning Automata with New Structure of Neighborhood*. ICMLC 2011 3rd. International Conference on Machine Learning and Computing. Vol 1, 3, pp. 419-423.

Aidonis, D. (2009). *Applied operations research methodologies on the optimal design and operation of construction and demolition materials reuse and recycling networks*. Tesis Doctoral, Aristotle University of Thessaloniki, Grecia. <http://www.didaktorika.gr/eadd/handle/10442/27136?locale=en>

AIMMS, Tutorial by Marcel Roelofs and Johannes Bisschop (2015). *The User's Guide*. LS Haarlem, The Netherlands. <http://www.aimms.com/solutions/supply-chain-optimization-solutions/supply-chain-network-optimization/>

Alba, E. (2005). *Parallel metaheuristics: a new class of algorithms*. John Wiley, pp. 3-62, USA.

André, F. J. y Cerdá, E. (2001). *Optimal Sequence of Landfills in Solid Waste Management*. Optimal Control. Applications and Methods, Vol. 25, 4, pp. 1-25.

André, F. J.; Velasco, F. y González, L. (2004). *Intertemporal and Spatial Location of Disposal Facilities*. Documento de Trabajo, Centro de Estudios Andaluces E2204/74. <http://citec.repec.org/p/a/pan65.html>

Angell, L.C. y Klassen, R.O. (1999). *Integrating environmental issues into mainstream: an agenda for research in operations management*, Journal of Operations Management, Vol. 15, nº 4, pp. 293-315.

Antunes, A. y Peeters, D. (2000). *A dynamic optimization model for school network planning*. Socio-Economic Planning Sciences, 34, pp. 101–120.

Ballou, R. (1968). *Dynamic warehouse location analysis*. Journal of Marketing Research, 5: pp. 271–276.

Barros, A. I., Dekker, R. y Scholten, V. (1998). *A two-level network for recycling sand: a case study*. European Journal of Operational Research, 110, pp. 199–214.

Bautista, B. y Pereira, J. (2006). *Modeling the problema of locating collection areas for urban waste management. An application to the metropolitan area of barcelona*. *Omega*, 34(6), pp. 617– 629.

Beasley, J.E. (1993). *Lagrangean heuristics for location problems*. *European Journal of Operational Research* 65, 383–399.

Bellman, R. (1957). *Dynamic Programming*. Princenton, NJ, USA.

Bellman, R., Zadeh, L. (1970). *Decision-Making in a Fuzzy Environment*. *Management Science*, 17, pp. 141-164.

Benedito Benet, E. (2010). “*Influence of reverse logistics on optimal manufacturing, remanufacturing, and storage capacities*”. Tesis Doctoral, Universidad Politécnica de Cataluña.
<https://www.educacion.gob.es/teseo/mostrarRef.do?ref=873813>

Blum, C. y Roli, A. (2003). *Metaheuristics in combinatorial optimization: Overview and conceptual comparison*. *ACM Computing Surveys*, 35(3), pp. 268-308.

Brimberg, J., Hansen, P., Mladenovic, N., Taillard, E. (2000). *Improvements and comparison of heuristics for solving the multisource Weber problem*. *Operations Research* 48 (3), 444–460.

Bustos Flores, C. E. (2010). “*La gestión de insumos mediante logística inversa en el sector artesanal*”. Tesis Doctoral, Universidad de la Laguna.
<https://www.educacion.gob.es/teseo/mostrarRef.do?ref=889788>

Carrasco, J. (2000). *Evolución de los enfoques y conceptos de la logística: su impacto en la dirección y gestión de las organizaciones*. *Economía industrial*, 331, pp. 17–34.

Castillo Díaz, P. E. (2011). “*Metodología de optimización en problemas de logística inversa. Aplicabilidad al sector vitivinícola*”. Tesis Doctoral, Universidad de Rioja.
<https://www.educacion.gob.es/teseo/mostrarRef.do?ref=935139>

Charon, I. and Hudry, O. (1993). *The noising method: A new method for combinatorial optimization*. *Operations Research Letters*, 14, pp. 133-137.

Chau, K. (2004). *A two-stage dynamic model on allocation of construction facilities with genetic algorithm*. *Automation in Construction*.

Ching-Jong Liao, Chao-Tang Tseng, Pim Luarn, (2007). *A Discrete Version of Particle Swarm Optimization for Flowshop Scheduling Problems*. *Computers & Operations Research*. 34(10), pp. 3099-3111.

Christofides, N., Beasley, J.E. (1982). *A tree search algorithm for the p-median problem*. *European Journal of Operational Research* 10, 196–204.

CLM (actual CSCMP), (2015). *Council of supply chain management professionals*. <https://www.cscmp.org>

Consortio Provincial de Residuos Sólidos Urbanos de Málaga. (2015). <http://www.consorciosumalaga.com/pagina/cod/330/instalaciones-html>

Dantzig, G. B. (1955). Linear programming under uncertainty. *Management Science*, vol.1, nº. 3, pp. 197-206.

Dantzig, G. B. (1963) *Linear Programming and Extensions*. Princeton University Press, Princeton, USA.

DeBrito, M. (2004). *Managing Reverse Logistics or Reversing Logistics Management?* Tesis Doctoral, Erasmus University Rotterdam. <http://hdl.handle.net/1765/1132>

Del Val, A. (1994). *Gestión de Residuos Sólidos*. Material elaborado para el 1.er Curso de Postgrado Sobre Economía de los Recursos Naturales y el Medio Ambiente. Universidad de Alcalá de Henares.

Diaz, B., Glover, F., Ghaziri, H., González Velarde, J., Laguna, M., Moscato, P. y Tsen, F. (1996). *Optimización Heurística y Redes Neuronales*. Editorial Paraninfo.

Diputación de Málaga (2015). *Red provincial de intercambio de residuos*. <http://www.malaga.es/diputacion/2245/consorcio-provincial-residuos-solidos-urbanos>

Dorigo, M. (1992) *Optimization, Learning and Natural Algorithms*, PhD thesis, Politecnico di Milano, Italia.

Du, F. (2006). *A bi-objective reverse logistics network analysis for post-sale service*. Tesis Doctoral, University of Louisville, Louisville, EEUU. <http://www.sciencedirect.com/science/article/pii/S0305054806003200>

Eberhart, R. C., Shi, Y. (2001). *Particle Swarm Optimization: Developments, Applications and Resources*. *Evolutionary Computation* 2001. Vol. 1, pp. 81-86.

Erlenkotter, D. (1975). *Capacity planning for large multilocation systems: Approximate and incomplete dynamic programming approaches*. *Management Science*, 22(3):pp. 274–285.

European Working Group on Reverse Logistics (REVLOG) (2015). <http://www.econbiz.de/Record/revlog-the-european-working-group-on-reverse-logistics/10005851629>

Feller, W. (1968). *An introduction to Probability. Theory and its Applications*. John Wiley.

Feo, T. y Resende, M. (1995). *Greedy randomized adaptive search procedures*. Journal of Global Optimization, 6(2), pp. 109– 113.

Fernández Quesada, I. (2004). “*Análisis de la logística inversa en el entorno empresarial. Una aproximación cualitativa*”. Tesis Doctoral, Universidad de Oviedo. <https://www.educacion.gob.es/teseo/mostrarRef.do?ref=307833>

Fleischmann, M. (2000). *Quantitative Models for Reverse Logistics*. Tesis Doctoral, Erasmus University Rotterdam; Rotterdam, Países Bajos. <http://www.sciencedirect.com/science/article/pii/S0377221797002300>

GAMS, Development Corporation (2015). Tutorial by Richard E. Rosenthal. A User's Guide. Washington, DC, USA. Web, <http://www.gams.com/docs/intro.htm>

Garey, M. R., y Johnson, D.S. (1979). *Computers and Intractability. A Guide to the Theory of NP Completeness*. W.H.Freeman, New York, NY.

Gaudet, G.; Moreaux, M. y Salant, S. W. (1998). *Intertemporal and Spatial Depletion of Landfill*. Fondazione Eni Enrico Mattei Note di Lavoro, 14/98.

Ghezavati, V., Saadati Nia, N. (2014). *Development of an optimization model for product returns using genetic algorithms and simulated annealing*. Methodologies And Application. Soft Computing. Springer, Berlin Heidelberg.

Glover, F. (1977). *Heuristics for integer programming using surrogate constraints*. Decision Sciences, 8, pp. 156–166.

Glover, F. y Laguna, M. (1997). *Tabu Search*. Ed. Kluwer, London.

Glover, F. (1989). *Tabu search - part 1*. ORSA Journal on Computing, 1(3), pp. 190–206.

Glover, F. (1990). *Tabu search - part 2*. ORSA Journal on Computing, 2(1), pp. 4–32.

Goddard, H. C. (1995). *The Benefits and Costs of Alternative Solid Waste Management Policies*. Resources, Conservation and Recycling, 13, pp. 183-213.

González, B. y Adenso-Díaz, B. (2004). *A scatter search approach to the optimum disassembly sequence problem*. Computers & Operations Research, 33(6), pp. 1776–1793.

González Torre, P. L. (2002). “*Análisis del ciclo de logística inversa en el sector del vidrio hueco industrial*”. Tesis Doctoral, Universidad de Oviedo. <https://www.educacion.gob.es/teseo/mostrarRef.do?ref=266316>

Hamacher, H.W. y Nickel, S. (1994). *Combinational algorithms for some 1-facility median problems in the plane*. Europ. J. Oper. Res. 79:340-351.

Handfield, R.B.; Walton, S.V.; Seegers, L.K.; Melnyk, S.A. (1997). *Green Value Chain Practices in the Furniture Industry*, Journal of Operations Management, Vol. 17, nº 5, pp. 575-598.

Hanjoul, P., Peeters, D. (1985). *A comparison of two dual-based procedures for solving the p-median problem*. European Journal of Operational Research 20, 387–396.

Hansen, P., Mladenovic, N., Taillard, E. (1998). *Heuristic solution of the multisource Weber problem as a p-median problem*. Operations Research Letters 22, 55–62.

Holland, J. (1975). *Adaptation in Natural and Artificial Systems*. MIT Press, Cambridge, MA, Estados Unidos.

Huhtala, A. (1997). *A Post-consumer Waste Management Model for Determining Optimal Levels of Recycling and Landfilling*. Environmental and Resource Economics, 10, pp. 301-314.

Huhtala, A. (1999). *Optimizing Production Technology Choices: Conventional Production vs. Recycling*. Resource and Energy Economics, 21, pp. 1-18.

IBM Corporation, Software Group, *IBM ILOG CPLEX Optimizer*, (2015). *High performance Mathematical Optimization Engines*. Web, <http://www-03.ibm.com/software/products/es/ibmilogcplexoptistud>.

INE: Instituto Nacional de Estadística (2015). <http://www.ine.es/>

Ismail Karaoglan, I., Altiparmak, F. (2015). *A memetic algorithm for the capacitated location-routing problem with mixed backhauls*. Elsevier. Computers & Operations Research, Volume 55, pp. 200–216.

Jacobs, T. L. y Everett, J. W. (1992). *Optimal Scheduling of Consecutive Landfill Operations with Recycling*, Journal of Environmental Engineering, 118, pp. 420-429.

Jahre, M. (1995). *Logistics Systems for recycling: efficient collection of household waste*. Tesis Doctoral, Chalmers University of Technology; Gotenborg, Suecia.

Jeung Ko, H. y Evans, G.W. (2005). *A genetic algorithm-based heuristic for the dynamic integrated forward/reverse logistics network for 3PLs*. Elsevier online database. Computers & Operations Research.

Jiménez Parra, B. (2014). *”Nuevos retos de investigación para la logística inversa. Análisis de la demanda de productos refabricados”*. Tesis Doctoral, Universidad de Extremadura.

<https://www.educacion.gob.es/teseo/mostrarRef.do?ref=1116417>

Johnson, M. R. y McCarthy I. P. (2014). *Product Recovery Decisions within the Context of Extended Producer Responsibility*. Journal of Engineering and Technology Management 34, 9-28.

Käfer, B. y Nickel, S. (2001). *Error bounds for the approximative solution of restricted planar location problems*. European Journal of Operational Research, 135, pp. 67-85.

Kassem, S., Chen, M. (2013). *Solving reverse logistics vehicle routing problems with time windows*. The International Journal of Advanced Manufacturing Technology. Volume 68, Issue 1, pp 57-68. Springer, London.

Keeler, A. G. y Renkow, M. (1994). *Haul Trash or Haul Ash: Energy Recovery as a Component of Local Solid Waste Management*. Journal of Environmental Economics and Management, 27, pp. 205-217.

Kennedy, Y.S.J. and Eberhart, R. (1995). *Particle Swarm Optimization*. Neural Networks, 1995. Proceedings. IEEE International Conference on Vol. 4, pp. 1942-1948.

Kirkpatrick, S., Gelatt, C. y Vecchi, M. (1983). *Optimization by simulated annealing*. Science, 220.

Klamroth, H. (2001). *Planar Weber location problems with line barriers*. Optimization 49, 517-527.

Klose, A. y Drexl, A. (2005). *Lower bounds for the capacitated facility location problem based on column generation*. Management Science, 51, 1689-1705. Doi: 10.1287/mnsc.1050.0410

Kongar, E. (2003). *Multiple Criteria Decision Making Techniques for End-Of-Life Product Recovery*. Tesis Doctoral, Northeastern University, Boston, EEUU.

Kopicki, R. (1993). *Reuse and Recycling*. Council of Logistic Management, 1993.

Krikke, H. (1998). *Recovery strategies and reverse logistic network design*. Tesis Doctoral, Institute for Business Engineering and Technology Application; University of Twente; Twente, Países Bajos, 1998.

Laguna, M. y Martí, R. (2003). *Scatter Search. Methodology and implementations in C*. Kluwer, Boston.

LIMASA, Servicios de Limpieza Integral de Málaga III, S.A. (2015). Memoria 2014.

<http://www.limasa3.es/sites/default/files/MEMORIA%20GESTION%202014.pdf>

López Parada, J. (2010). *”Incorporación de la logística inversa en la cadena de suministros y su influencia en la estructura organizativa de las empresas”*. Tesis Doctoral, Universidad de Barcelona.

<https://www.educacion.gob.es/teseo/mostrarRef.do?ref=904740>

Lourenco, H.R., Martin, O. and Stützle, T. (2003). *Iterated local search*. In F. Glover and G.G. Kochenberger, editors, *Handbook of Metaheuristics*, chapter 11. Kluwer Academic Publishers.

Lu, Z. y Bostel, N. (2007). *A facility location model for logistics systems including reverse flows: The case of remanufacturing activities*. *Computers and Operations Research*, 34:pp. 299–323.

Luttwak, E. (1971). *A Dictionary of Modern War*. Harper & Row, New York, N.Y., 1971

MAE: Ley 22/2011, de 28 de julio de residuos y suelos contaminados. BOE núm. 181, de 29 de julio de 2011. Madrid.

MAE: Ley 10/1998, de 21 de abril, de residuos. BOE núm. 96, de 22 de abril de 1998. Madrid.

Marín, A., Pelegrín, B. (1991). *Heurísticas de decomposición lagrangiana para algunos problemas de localización discreta*. *Trabajos de Investigación Operativa*, Vol. 7. 1, pp. 3–15.

MATLAB, R2014b, (2015). *Language Reference Manual*. The MathWorks, Inc, US. Web, <http://www.mathworks.com>.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. y Teller, E. (1953). *Equation of state calculation by fast computing machines*. *Journal of Chemical Physics*, 21(6), pp. 1087–1092.

Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs*. Springer Verlag.

Mihi Ramírez, A. (2010). “*Un análisis causal de la relación entre la creación del conocimiento y la logística inversa*”. Tesis Doctoral, Universidad de Granada. <https://www.educacion.gob.es/teseo/mostrarRef.do?ref=856608>

Min, H., JeungKo, H. y SeongKo, C. (2006), *A genetic algorithm approach to developing the multi-echelon reverse logistics network for product returns*. *Omega*, 34(1), pp. 56–69, 2006.

MINOS 5.5. Murtagh B. A. y Saunders M. A., (1983). *User Guide*, Tech. Report SOL 83-20R, Systems Optimization Laboratory, Dep. of Operations Research, Stanford University, Web, <http://www.sbsi-sol-optimize.com/manuals>.

Ortega Mier, M.A. (2008). *Utilización de métodos cuantitativos para el análisis de problemas de localización en Logística Inversa*. Tesis Doctoral, Universidad Politécnica de Madrid.

<https://www.educacion.gob.es/teseo/mostrarRef.do?ref=559755>

Osman, I.H. y Laporte, G. (1996). *Metaheuristics: A bibliography*. *Ann. Oper. Res.* 63, pp. 513–623.

Perez, R. E., Behdinan, K. (2007). *Particle swarm approach for structural design optimization*. *Journal Computers and Structures*. Vol. 85. 19-20, pp. 1579-1588.

Pishvae, S., Kianfar, K., Karimi, B. (2010). *Reverse logistics network design using simulated annealing*. *The International Journal of Advanced Manufacturing Technology*. Vol. 47, Issue1, pp. 269-281. Springer, Verlag.

PNIR: Plan Nacional de Integrado de Residuos, (2009-2015). Ministerio de Medioambiente. Dirección General de Calidad y Medioambiente. España. <https://www.boe.es/boe/dias/2009/02/26/pdfs/BOE-A-2009-3243.pdf>

Pólya, G. (1945). *How to Solve It*, Princeton, NJ, USA.

Qiang Lu, Xue-Na Qiu, Shi-Rong Liu. (2009). *Discrete Particle Swarm Optimization Algorithm with Fully Communicated Information*. GEC 2009 Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation.

Qin, X. H. (2011). *Dynamic optimization design for integrated forward/reverse logistics network with stochastic environment*. Computer Science and Service System (CSSS), IEEE Conference Publications. International Conference on, pp. 3205 – 3208.

Ready, M.J. y Ready, R.C. (1995). *Optimal Pricing of Depletable, Replaceable Resources: The case of landfill tipping fees*. *Journal of Environmental Economics and Management*, 28, pp. 307-323.

Reverse Logistics Executive Council (RELOOP), (2015). <http://www.rlec.org>

Rogers, D. y Tibben-Lembke, R. (1999). *Going Backwards: Reverse Logistics, Trends and Practices*. Logistics Executive Council., 1999.

Rubio Lacoba, S. (2003). “*El sistema de logística inversa en la empresa: análisis y aplicaciones*”. Tesis Doctoral, Universidad de Extremadura. <http://dialnet.unirioja.es/servlet/tesis?codigo=305>

Saibani, N. (2010). *Performance measurement for reverse and closed-loop supply chains*. Tesis Doctoral, University of Nottingham. UK. <http://core.ac.uk/download/pdf/387205.pdf>

Shulman, A. (1991). *An algorithm for solving dynamic capacitated plant location problems with discrete expansion sizes*. *Operations Research*, 3, pp. 423–436.

Soto Zuluaga, J.P. (2006). “*Reverse Logistics: Models and applications*”. Tesis Doctoral, Universitat Pompeu Fabra. <http://dialnet.unirioja.es/servlet/tesis?codigo=6324>

- Stevens, G. (1989). *Integrating the supply chain*. International Journal of Physical Distribution and Materials Management, 19(8), pp. 3–8.
- Stock, R. J. (1992). *Reverse Logistics*. Oak Brook, IL. Council of Logistics Management, 1992.
- Stock, R. J. (1998). *The development and implementation the Reverse Logistics programs*. Oak Brook, IL. Council of Logistics Management, 1992.
- Stützle, T. (1999). *Local search algorithms for combinatorial problems analysis, algorithms and news applications*. DISKI Dissertationen zur Künstlichen Intelligenz.
- Taillard, E. (1996). *Heuristic methods for large multi-source Weber problems*. Technical Report IDSIA96-96, IDSIA, Lugano.
- Tchobanoglous, G., Theisen, H., Vigil, S. A. (1994). *Integrated solid waste management*, McGraw Hill, Mexico.
- Thierry, M., Salomon, M. y Van Nunen, J. (1995). Strategic issues in product recovery management. *California Management Review*, 37(2), pp. 114–135.
- Tim Jones, M. (2005). *AI Application Programming*. Second Edition. Charles River Media. Canada. 473 p.
- Van der Laan, E. (1997). *The effects of remanufacturing on inventory control*. Tesis Doctoral, Erasmus University Rotterdam, Rotterdam, Países Bajos.
- Vercraene, S. (2012). *Control of production/inventory systems in reverse logistic context*. Tesis Doctoral, Universidad de Grenoble, Francia.
- Whitley, D. (1994). *A genetic algorithm tutorial*. Statistics and Computing, 4, pp. 65–85.
- Xiwang, G., Shixin, L., Dazhi, W., Chunming, H. (2012). *An improved multi-objective scatter search approach for solving selective disassembly optimization problema*. IEEE Conference Publications, Control Conference (CCC), 31st Chinese, Pages: 7703 – 7708.
- Zhou, X. C., Zhao, Z. X., Zhou, K. J., He, C. H. (2012), *Remanufacturing closed-loop supply chain network design based on genetic particle swarm optimization algorithm*. Journal of Central South University. Vol. 19, Issue 2, pp 482-487.

ANEXO A.1: Algoritmo PSO básico

Programación en MATLAB de algoritmo básico PSO

```
function [X,Fx,StopMessage] = pso(ObjectiveFunction,Nvars,LB,UB)
%% Parámetros por defecto
C1 = 1.2; %% Coeficiente SocialAttraction
C2 = 0.012; %% Coeficiente CognitiveAttraction
MaxIterations = 10000;
N = 30;
W = 0.0004; %% Coeficiente de Inercia
StopMessage = 'Stopping PSO : Reached Maximum Number of
Iterations';

%% Comprobación de entradas

if nargin < 4
    error('PSO : Not Enough Input Arguments');
end
if ~isequal(Nvars,size(LB,2),size(UB,2))
    error('PSO : Invalid Arguments :
isequal(Nvars,size(LB,2),size(UB,2)) should be true ');
end

%% Posición Inicial y Velocidad

CurrentPosition = zeros(N,Nvars); % Posición Inicial
for i = 1:Nvars
    CurrentPosition(:,i) = random('unif',LB(i),UB(i),N,1);
end
Velocity = W.*rand(N,Nvars) ; % Velocidad Inicial

%% Evalua Posición Inicial

CurrentFitness = zeros(N,1); % Valor Fitness
for i = 1:N
    CurrentFitness(i) = ObjectiveFunction(CurrentPosition(i,:));
end

%% Actualiza el mejor local

LocalBestPosition = CurrentPosition; % Mejor Local
LocalBestFitness = CurrentFitness;

%% Actualiza el mejor Global

[GlobalBestFitness,index] = min(LocalBestFitness);
GlobalBestPosition = repmat(LocalBestPosition(index,:),N,1); %
%Mejor Global

%% Actualiza Velocidad y Posición

R1 = rand(N,Nvars); % Operador aleatorio 1
R2 = rand(N,Nvars); % Operador aleatorio 2
Velocity = W.*Velocity + C1.*(R1.*(LocalBestPosition-
CurrentPosition))...
    + C2.*(R2.*(GlobalBestPosition-CurrentPosition));
CurrentPosition = CurrentPosition + Velocity ;
%% Límites: LB UB
```

```

for i = 1:Nvars
    indexes = CurrentPosition(:,i) < LB(i).*ones(N,1);
    CurrentPosition(indexes,i) = LB(i);
    indexes = CurrentPosition(:,i) > UB(i).*ones(N,1);
    CurrentPosition(indexes,i) = UB(i);
end
%% Iteración para obtener el mejor fitness
Iter = 0;
while (Iter < MaxIterations)
    Iter = Iter+1;
    %%Evalua la Posición actual
    for i = 1:N
        CurrentFitness(i)=ObjectiveFunction(CurrentPosition(i,:));
    end
    %% Actualiza el Mejor Local
    indexes = find(CurrentFitness < LocalBestFitness);
    LocalBestFitness(indexes) = CurrentFitness(indexes);
    LocalBestPosition(indexes,:) = CurrentPosition(indexes,:);
    %% Actualiza el Mejor Global
    [GlobalBestFitnessNew,index] = min(LocalBestFitness);
    if GlobalBestFitnessNew < GlobalBestFitness
        GlobalBestFitness = GlobalBestFitnessNew;
        GlobalBestPosition =
repmat(LocalBestPosition(index,:),N,1);
    end
    %% Actualiza la Velocidad y la Posición
    R1 = randn(N,Nvars); % Operador aleatorio 1
    R2 = randn(N,Nvars); % Operador aleatorio 2
    Velocity = W.*Velocity + C1.*(R1.*(LocalBestPosition-
CurrentPosition))...
        + C2.*(R2.*(GlobalBestPosition-CurrentPosition));
    CurrentPosition = CurrentPosition + Velocity ;
    %% Límites: LB UB
    for i = 1:Nvars
        indexes = CurrentPosition(:,i) < LB(i).*ones(N,1);
        CurrentPosition(indexes,i) = LB(i);
        indexes = CurrentPosition(:,i) > UB(i).*ones(N,1);
        CurrentPosition(indexes,i) = UB(i);
    end
    %% Imprime Resultados
    M = mean(LocalBestFitness);
    if (Iter==1) || (rem(Iter,100)==0)
        fprintf('Iteration-%5.0d | Best F(x)=%d | Mean F(x) =
%d\n',Iter,GlobalBestFitness,M);
    end
end
%% Salida
X = GlobalBestPosition(1,:);
Fx = GlobalBestFitness;
end

```

ANEXO A.2: Algoritmo GA

Programación en MATLAB de algoritmo GA

```
function [x,fval,exitFlag,output,population,scores] =
ga(fun,nvars,Aineq,bineq,Aeq,beq,lb,ub,nonlcon,intcon,options)
%GA    Constrained optimization using genetic algorithm.
%    GA attempts to solve problems of the following forms:
%        min F(X) subject to: A*X <= B, Aeq*X = Beq (linear
constraints)
%            X                C(X) <= 0, Ceq(X) = 0 (nonlinear
constraints)
%                                LB <= X <= UB
%                                X(i) integer, where i is in the
index
%                                vector INTCON (integer constraints)
%
%    Note: If INTCON is not empty, then no equality constraints are
allowed.
%    That is:-
%    * Aeq and Beq must be empty
%    * Ceq returned from NONLCON must be empty
%
%    X = GA(FITNESSFCN,NVARS) finds a local unconstrained minimum X
to the
%    FITNESSFCN using GA. NVARS is the dimension (number of design
%    variables) of the FITNESSFCN. FITNESSFCN accepts a vector X of
size
%    1-by-NVARS, and returns a scalar evaluated at X.
%
%    X = GA(FITNESSFCN,NVARS,A,b) finds a local minimum X to the
function
%    FITNESSFCN, subject to the linear inequalities A*X <= B.
Linear
%    constraints are not satisfied when the PopulationType option
is set to
%    'bitString' or 'custom'. See the documentation for details.
%
%    X = GA(FITNESSFCN,NVARS,A,b,Aeq,beq) finds a local minimum X
to the
%    function FITNESSFCN, subject to the linear equalities Aeq*X =
beq as
%    well as A*X <= B. (Set A=[] and B=[] if no inequalities
exist.) Linear
%    constraints are not satisfied when the PopulationType option
is set to
%    'bitString' or 'custom'. See the documentation for details.
%
%    X = GA(FITNESSFCN,NVARS,A,b,Aeq,beq,lb,ub) defines a set of
lower and
%    upper bounds on the design variables, X, so that a solution is
found in
%    the range lb <= X <= ub. Use empty matrices for lb and ub if
no bounds
%    exist. Set lb(i) = -Inf if X(i) is unbounded below; set ub(i)
= Inf if
%    X(i) is unbounded above. Linear constraints are not satisfied
when the
```

```

% PopulationType option is set to 'bitString' or 'custom'. See
the
% documentation for details.
%
% X = GA(FITNESSFCN,NVARS,A,b,Aeq,beq,lb,ub,NONLCON) subjects
the
% minimization to the constraints defined in NONLCON. The
function
% NONLCON accepts X and returns the vectors C and Ceq,
representing the
% nonlinear inequalities and equalities respectively. GA
minimizes
% FITNESSFCN such that C(X)<=0 and Ceq(X)=0. (Set lb=[] and/or
ub=[] if
% no bounds exist.) Nonlinear constraints are not satisfied when
the
% PopulationType option is set to 'bitString' or 'custom'. See
the
% documentation for details.
%
% X = GA(FITNESSFCN,NVARS,A,b,Aeq,beq,lb,ub,NONLCON,options)
minimizes
% with the default optimization parameters replaced by values in
the
% structure OPTIONS. OPTIONS can be created with the GAOPTIMSET
function.
% See GAOPTIMSET for details.
%
% X = GA(FITNESSFCN,NVARS,A,b,[],[],lb,ub,NONLCON,INTCON)
requires that
% the variables listed in INTCON take integer values. Note that
GA does
% not solve problems with integer and equality constraints. Pass
empty
% matrices for the Aeq and beq inputs if INTCON is not empty.
%
% X =
GA(FITNESSFCN,NVARS,A,b,[],[],lb,ub,NONLCON,INTCON,options)
% minimizes with integer constraints and the default
optimization
% parameters replaced by values in the structure OPTIONS.
OPTIONS can be
% created with the GAOPTIMSET function. See GAOPTIMSET for
details.
%
% X = GA(PROBLEM) finds the minimum for PROBLEM. PROBLEM is a
structure
% that has the following fields:
%     fitnessfcn: <Fitness function>
%     nvars: <Number of design variables>
%     Aineq: <A matrix for inequality constraints>
%     bineq: <b vector for inequality constraints>
%     Aeq: <Aeq matrix for equality constraints>
%     beq: <beq vector for equality constraints>
%     lb: <Lower bound on X>
%     ub: <Upper bound on X>
%     nonlcon: <Nonlinear constraint function>
%     intcon: <Index vector for integer variables>
%     options: <Options structure created with GAOPTIMSET>
%     rngstate: <State of the random number generator>
%

```

```

% [X,FVAL] = GA(FITNESSFCN, ...) returns FVAL, the value of the
fitness
% function FITNESSFCN at the solution X.
%
% [X,FVAL,EXITFLAG] = GA(FITNESSFCN, ...) returns EXITFLAG which
% describes the exit condition of GA. Possible values of
EXITFLAG and the
% corresponding exit conditions are
%
% 1 Average change in value of the fitness function over
% options.StallGenLimit generations less than
options.TolFun and
% constraint violation less than options.TolCon.
% 3 The value of the fitness function did not change in
% options.StallGenLimit generations and constraint
violation less
% than options.TolCon.
% 4 Magnitude of step smaller than machine precision and
constraint
% violation less than options.TolCon. This exit condition
applies
% only to nonlinear constraints.
% 5 Fitness limit reached and constraint violation less than
options.TolCon.
% 0 Maximum number of generations exceeded.
% -1 Optimization terminated by the output or plot function.
% -2 No feasible point found.
% -4 Stall time limit exceeded.
% -5 Time limit exceeded.
%
% [X,FVAL,EXITFLAG,OUTPUT] = GA(FITNESSFCN, ...) returns a
% structure OUTPUT with the following information:
% rngstate: <State of the random number generator
before GA started>
% generations: <Total generations, excluding HybridFcn
iterations>
% funcccount: <Total function evaluations>
% maxconstraint: <Maximum constraint violation>, if any
% message: <GA termination message>
%
% [X,FVAL,EXITFLAG,OUTPUT,POPULATION] = GA(FITNESSFCN, ...)
returns the
% final POPULATION at termination.
%
% [X,FVAL,EXITFLAG,OUTPUT,POPULATION,SCORES] = GA(FITNESSFCN,
...) returns
% the SCORES of the final POPULATION.
%
%
% Example:
% Unconstrained minimization of 'rastriginsfcn' fitness
function of
% numberOfVariables = 2
% x = ga(@rastriginsfcn,2)
%
% Display plotting functions while GA minimizes
% options = gaoptimset('PlotFcns',...
% {@gaplotbestf,@gaplotbestindiv,@gaplotexpectation,@gaplotstopping}
% );

```

```

% [x,fval,exitflag,output] =
ga(@rastriginsfcn,2,[],[],[],[],[],[],[],options)
%
% An example with inequality constraints and lower bounds
% A = [1 1; -1 2; 2 1]; b = [2; 2; 3]; lb = zeros(2,1);
% % Use mutation function which can handle constraints
% options = gaoptimset('MutationFcn',@mutationadaptfeasible);
% [x,fval,exitflag] =
ga(@lincontest6,2,A,b,[],[],lb,[],[],options);
%
% FITNESSFCN can also be an anonymous function:
% x = ga(@(x) 3*sin(x(1))+exp(x(2)),2)
%
% If FITNESSFCN or NONLCON are parameterized, you can use
anonymous
% functions to capture the problem-dependent parameters. Suppose
you want
% to minimize the fitness given in the function myfit, subject
to the
% nonlinear constraint myconstr, where these two functions are
% parameterized by their second argument a1 and a2,
respectively. Here
% myfit and myconstr are MATLAB file functions such as
%
% function f = myfit(x,a1)
% f = exp(x(1))*(4*x(1)^2 + 2*x(2)^2 + 4*x(1)*x(2) + 2*x(2)
+ a1);
%
% and
%
% function [c,ceq] = myconstr(x,a2)
% c = [1.5 + x(1)*x(2) - x(1) - x(2);
%      -x(1)*x(2) - a2];
% % No nonlinear equality constraints:
% ceq = [];
%
% To optimize for specific values of a1 and a2, first assign the
values
% to these two parameters. Then create two one-argument
anonymous
% functions that capture the values of a1 and a2, and call myfit
and
% myconstr with two arguments. Finally, pass these anonymous
functions to
% GA:
%
% a1 = 1; a2 = 10; % define parameters first
% % Mutation function for constrained minimization
% options = gaoptimset('MutationFcn',@mutationadaptfeasible);
% x =
ga(@(x)myfit(x,a1),2,[],[],[],[],[],[],[],@(x)myconstr(x,a2),options)
%
% Example: Solving a mixed-integer optimization problem
% An example of optimizing a function where a subset of the
variables are
% required to be integers:
%
% % Define the objective and call GA. Here variables x(2) and
x(3) will
% be integer.
% fun = @(x) (x(1) - 0.2)^2 + (x(2) - 1.7)^2 + (x(3) -5.1)^2;

```

```

% x = ga(fun,3,[],[],[],[],[],[],[],[2 3])
%
% See also GAOPTIMSET, FITNESSFUNCTION, GAOUTPUTFCNTEMPLATE,
PATTERNSEARCH, @.

% Copyright 2003-2013 The MathWorks, Inc.
% $Revision: 1.1.6.13 $ $Date: 2013/10/15 01:33:12 $

% If the first arg is not a gaoptimset, then it's a fitness
function followed by a genome
% length. Here we make a gaoptimset from the args.
defaultopt = struct('PopulationType', 'doubleVector', ...
    'PopInitRange', [], ...
    'PopulationSize', '50 when numberOfVariables <= 5, else 200',
    ...
    'EliteCount', '0.05*PopulationSize', ...
    'CrossoverFraction', 0.8, ...
    'MigrationDirection', 'forward', ...
    'MigrationInterval', 20, ...
    'MigrationFraction', 0.2, ...
    'Generations', '100*numberOfVariables', ...
    'TimeLimit', inf, ...
    'FitnessLimit', -inf, ...
    'StallTest', 'averageChange', ...
    'StallGenLimit', 50, ...
    'StallTimeLimit', inf, ...
    'TolFun', 1e-6, ...
    'TolCon', 1e-6, ...
    'InitialPopulation', [], ...
    'InitialScores', [], ...
    'InitialPenalty', 10, ...
    'PenaltyFactor', 100, ...
    'PlotInterval', 1, ...
    'CreationFcn', @gacreationuniform, ...
    'FitnessScalingFcn', @fitscalingrank, ...
    'SelectionFcn', @selectionstochunif, ...
    'CrossoverFcn', @crossoverscattered, ...
    'MutationFcn', {@mutationgaussian 1 1}, ...
    'HybridFcn', [], ...
    'Display', 'final', ...
    'PlotFcns', [], ...
    'OutputFcns', [], ...
    'Vectorized', 'off', ...
    'UseParallel', false);

% Check number of input arguments
try
    narginchk(1,11);
catch ME
    error(message('globaloptim:ga:numberOfInputs', ME.message));
end

% If just 'defaults' passed in, return the default options in X
if nargin == 1 && nargsout <= 1 && isequal(fun, 'defaults')
    x = defaultopt;
    return
end

if nargin < 11, options = [];
    if nargin < 10, intcon = [];

```

```

    if nargin < 9, nonlcon = [];
        if nargin < 8, ub = [];
            if nargin < 7, lb = [];
                if nargin < 6, beq = [];
                    if nargin < 5, Aeq = [];
                        if nargin < 4, bineq = [];
                            if nargin < 3, Aineq = [];
                                end
                            end
                        end
                    end
                end
            end
        end
    end

% Is third argument a structure
if nargin == 3 && isstruct(Aineq) % Old syntax
    options = Aineq; Aineq = [];
end

% Is tenth argument a structure? If so, integer constraints have
not been
% specified
if nargin == 10 && isstruct(intcon)
    options = intcon;
    intcon = [];
end

% One input argument is for problem structure
if nargin == 1
    if isa(fun, 'struct')

[fun, nvars, Aineq, bineq, Aeq, beq, lb, ub, nonlcon, intcon, rngstate, optio
ns] = separateOptimStruct(fun);
        % Reset the random number generators
        resetDfltRng(rngstate);
    else % Single input and non-structure.
        error(message('globaloptim:ga:invalidStructInput'));
    end
end

% If fun is a cell array with additional arguments get the
function handle
if iscell(fun)
    FitnessFcn = fun{1};
else
    FitnessFcn = fun;
end

% Only function handles or inlines are allowed for FitnessFcn
if isempty(FitnessFcn) || ~(isa(FitnessFcn, 'inline') ||
isa(FitnessFcn, 'function_handle'))
    error(message('globaloptim:ga:needFunctionHandle'));
end

% We need to check the nvars here before we call any solver
valid = isnumeric(nvars) && isscalar(nvars) && (nvars > 0) ...
&& (nvars == floor(nvars));

```

```

if ~valid
    error(message('globaloptim:ga:notValidNvars'));
end

% Set default PopInitRange for non-MINLP problems
defaultopt.PopInitRange = [-10;10];

% Specific checks and modification of options for mixed integer GA
if ~isempty(intcon)
    % Check whether the user has specified options that the mixed
integer
    % solver will either ignore or error.
    gaminlpvalidateoptions(options);
    % Change the default options for PopulationSize and EliteCount
here.
    defaultopt.PopulationSize = max(min(10*nvars, 100), 40);
    defaultopt.EliteCount = floor(0.05*defaultopt.PopulationSize);
    % Adjust PopInitRange for MINLPs
    defaultopt.PopInitRange = [-1e4 + 1; 1e4 + 1];
end

user_options = options;
% Use default options if empty
if ~isempty(options) && ~isa(options, 'struct')
    error(message('globaloptim:ga:optionsNotAstruct'));
elseif isempty(options)
    options = defaultopt;
end
% Take defaults for parameters that are not in options structure
options = gaoptimset(defaultopt, options);

% If a user doesn't specify PopInitRange, we want to set it to the
% bounds when we create the initial population. Need to store a
flag
% that indicates whether the user has specified PopInitRange so we
can
% do this in the creation function.
options.UserSpecPopInitRange = isa(user_options, 'struct') && ...
    isfield(user_options, 'PopInitRange') &&
~isempty(user_options.PopInitRange);

% Check for non-double inputs
msg = isoptimargdbl('GA', {'NVARs', 'A', 'b',
'Aeq', 'beq', 'lb', 'ub'}, ...
    nvars, Aineq, bineq, Aeq, beq, lb,
ub);
if ~isempty(msg)
    error('globaloptim:ga:dataType', msg);
end

[x, fval, exitFlag, output, population, scores, FitnessFcn, nvars, Aineq, b
ineq, Aeq, beq, lb, ub, ...
    NonconFcn, options, Iterate, type] =
gacommon(nvars, fun, Aineq, bineq, Aeq, beq, lb, ub, ...
nonlcon, intcon, options, user_options);

if exitFlag < 0
    return;
end

```

```

% Turn constraints into right size if they are empty.
if isempty(Aineq)
    Aineq = zeros(0,nvars);
end
if isempty(bineq)
    bineq = zeros(0,1);
end
if isempty(Aeq)
    Aeq = zeros(0,nvars);
end
if isempty(beq)
    beq = zeros(0,1);
end

% Call appropriate single objective optimization solver
if ~isempty(intcon)
    [x,fval,exitFlag,output,population,scores] =
    gaminlp(FitnessFcn,nvars, ...

Aineq,bineq,Aeq,beq,lb,ub,NonconFcn,intcon,options,output,Iterate)
;
else
    switch (output.problemtype)
        case 'unconstrained'
            [x,fval,exitFlag,output,population,scores] =
            gaunc(FitnessFcn,nvars, ...
                options,output,Iterate);
        case {'boundconstraints', 'linearconstraints'}
            [x,fval,exitFlag,output,population,scores] =
            galincon(FitnessFcn,nvars, ...
                Aineq,bineq,Aeq,beq,lb,ub,options,output,Iterate);
        case 'nonlinearconstr'
            [x,fval,exitFlag,output,population,scores] =
            gacon(FitnessFcn,nvars, ...

Aineq,bineq,Aeq,beq,lb,ub,NonconFcn,options,output,Iterate,type);
    end
end

```

ANEXO B: Parámetros algoritmo GA

'PopulationType', 'doubleVector', ...
'PopInitRange', [], ...
'PopulationSize', '50 when numberOfVariables <= 5, else 200', ...
'EliteCount', '0.05*PopulationSize', ...
'CrossoverFraction', 0.8, ...
'MigrationDirection', 'forward', ...
'MigrationInterval', 20, ...
'MigrationFraction', 0.2, ...
'Generations', '100*numberOfVariables', ...
'TimeLimit', inf, ...
'FitnessLimit', -inf, ...
'StallTest', 'averageChange', ...
'StallGenLimit', 50, ...
'StallTimeLimit', inf, ...
'TolFun', 1e-6, ...
'TolCon', 1e-6, ...
'InitialPopulation', [], ...
'InitialScores', [], ...
'InitialPenalty', 10, ...
'PenaltyFactor', 100, ...
'PlotInterval', 1, ...
'CreationFcn', @gacreationuniform, ...
'FitnessScalingFcn', @fitscalingrank, ...
'SelectionFcn', @selectionstochunif, ...
'CrossoverFcn', @crossoverScattered, ...
'MutationFcn', {@mutationgaussian 1 1}, ...
'HybridFcn', [], ...
'Display', 'final', ...
'PlotFcns', [], ...
'OutputFcns', [], ...
'Vectorized', 'off', ...
'UseParallel', false);

ANEXO C: Parámetros algoritmo PSO

AccelerationFcn: [Function handle | {@psoiterate}]
CognitiveAttraction: [Positive scalar | {0.5}]
ConstrBoundary: ['soft' | 'penalize' | 'reflect' | 'absorb' | {'penalize'}]
Display: ['off' | 'final' | 'diagnose' | {'final'}]
DemoMode: ['fast' | 'pretty' | 'on' | 'off' | {'off'}]
FitnessLimit: [Scalar | {-Inf}]
Generations: [Positive integer | {200}]
HybridFcn: [@fminsearch | @patternsearch | @fminunc | @fmincon | {}]
InitialPopulation: [empty matrix | nxnvars matrix | {}]
InitialVelocities: [empty matrix | nxnvars matrix | {}]
PlotFcns: [Cell array of fcn handles | {}]
PlotInterval: [Positive integer | {1}]
PopInitRange: [2x1 vector | 2xnvars matrix | {[0;1]}]
PopulationSize: [Positive integer | {40}]
PopulationType: ['bitstring' | 'doubleVector' | {'doubleVector'}]
SocialAttraction: [Positive scalar | {1.25}]
StallGenLimit: [Positive integer | {50}]
StallTimeLimit: [Positive scalar (seconds) | {Inf}]
TimeLimit: [Positive scalar (seconds) | {50}]
TolFun: [Positive scalar | {1e-06}]
TolCon: [Positive scalar | {1e-06}]
UseParallel: ['always' | 'never' | {'never'}]
Vectorized: ['on' | 'off' | {'off'}]
VelocityLimit: [Positive scalar | {}]

ANEXO D: Parámetros y valores caso real

Tabla D.1: Costes y capacidad transferencia municipio-c.transferencia-p.tratamiento

Parámetros	Descripción	Valores
CTPt	Coste unitario de transferencia del residuo entre un centro de transferencia y la planta de tratamiento (€/km · Tm) en el año t	0,00006
CMTt	Coste unitario de transporte del residuo recogido en cada municipio hasta el centro de transferencia (€/km · Tm) en el año t	1,2
QTjt	Capacidad anual de un centro de transferencia j (en Tm) en el año t	50000
QPkt	Capacidad anual de la planta de tratamiento k (en Tm) en el año t	500000
CFjt	Costes fijos centro transferencia j (€) en el año t	197072
CFkt	Costes fijos planta tratamiento k (€) en el año t	215138

Tabla D.2: Transferencia de residuos de centros transferencia a plantas tratamiento 2013

SIN COSTE ESTACIONES

Transferencia de Residuos a estaciones 2013						COSTE Km	COSTE TM
	Circuito	COSTE TOTAL	Tn 2013	Nº de Viajes	Km iv		
Valsequillo	Archidona-Antequera	14.464,73	8.009,90	548	66	0,40	1,81
	Campillos-Antequera	30.149,67	10.525,81	671	68	0,66	2,86
	Ronda-Antequera	453.055,88	23.292,04	1.245	188	1,34	10,45
	Vélez-Antequera	1.687.335,80	66.578,43	3.619	188	2,48	25,34
Casarabonela	Cártama-Casarabonela	658.212,38	47.693,53	2.861	70	3,29	13,80

CON COSTE ESTACIONES

Transferencia de Residuos a estaciones 2013						COSTE Km	COSTE TM
	Circuito	COSTE TOTAL	Tn 2013	Nº de Viajes	Km iv		
Valsequillo	Archidona-Antequera	14.464,73	8.009,90	548	66	0,40	1,81
	Campillos-Antequera	30.149,67	10.525,81	671	68	0,66	2,86
	Ronda-Antequera	732.760,78	23.292,04	1.245	188	3,13	10,46
	Vélez-Antequera	1.902.474,39	66.578,43	3.619	188	1,99	28,57
	Vélez-Antequera (gsa)		28.825,83	1.471	188		
Casarabonela	Cártama-Casarabonela	937.916,67	47.693,53	2.861	70	4,68	18,67

Las capacidades de las plantas de tratamiento son:
 ET Vélez, Ronda y Cártama tienen una capacidad de 120 Tn/hora
 ET Campillos, Archidona: 20 Tn/hora

Tabla D.3: Recogida de residuos de centros transferencia 2013

RECOGIDA ORGANICA 2013

	Tn	Habitantes	Coste €/año	€/Tn	€/hab
AXARQUIA	7.688,33	42.172	796.252,59	103,57	18,88
ANTEQUERA	18.861,93	52.405	1.529.917,63	81,11	29,19
GUADALHORCE	14.527,10	38.528	961.482,45	66,19	24,96
RONDA	5.780,63	13.489	444.152,43	76,83	32,93
			Media Costes	81,92 €	26,49 €

Tabla D.4: Transferencia de residuos a centros tratamiento 2010-2015

TRANSFERENCIAS RSU A CENTROS DE TRATAMIENTOS -- ANUAL

	TRANSF. RSU VÉLEZ ANTEQUER A 102001	TRANSF. RSU G.S.A. VÉLEZ- ANTEQUER A 154000	TRANSF. RSU RONDA ANTEQUER A 102004	TOTAL ENTRADA EN C.T. ANTEQUER A	TRANSF. RSU CÁRTAMA BONELA 102014	TRANSF. RSU VELEZ BONELA 102034	TOTAL ENTRADA EN C.T. CASARABO NELA	TOTAL ENTRADAS
AÑO 2015	16.048,68	6.401,76	5.410,45	27.860,89	11.852,87	421,71	12.274,58	40.135,47
AÑO 2014	64.815,52	30.178,66	22.629,76	117.623,94	46.985,18	706,89	47.692,07	165.316,01
AÑO 2013	66.616,83	28.825,83	23.298,46	118.741,12	47.740,31	177,27	47.917,58	166.658,70
AÑO 2012	65.334,77	26.076,89	23.361,14	114.772,80	55.257,70	456,39	55.714,09	170.486,89
AÑO 2011	31.469,05	9.916,35	25.076,38	66.461,78	48.518,41	38.790,91	87.309,32	153.771,10
AÑO 2010	70.979,48	31.160,19	25.282,32	127.421,99	51.459,10	0,00	51.459,10	178.881,09

Tabla D.5: Producción de residuos (P_{it}) de 90 municipios (T) y 5 años

MUNICIPIO	Total Toneladas 2010	Total Toneladas 2011	Total Toneladas 2012	Total Toneladas 2013	Total Toneladas 2014	TOTAL 2010 - 2014
ALAMEDA	2.361,96	2.138,15	2.022,85	2.015,49	2.168,29	10.706,74
ALCAUCIN	1.073,63	1.263,30	1.252,66	1.208,68	1.187,82	5.986,09
ALFARNATE	563,97	579,84	495,35	831,54	370,44	2.841,14
ALFARNATEJO	225,09	235,32	204,43	147,93	153,85	966,62
ALGARROBO	2.591,96	3.593,88	3.534,46	3.161,72	3.301,08	16.183,10
ALGATOCIN	393,91	437,46	323,78	317,12	279,86	1.752,13
ALH DE LA TORRE	16.271,80	14.652,50	15.729,46	14.013,11	13.576,39	74.243,26
ALH EL GRANDE	10.806,00	9.681,23	10.195,24	9.036,35	8.648,27	48.367,09
ALMACHAR	489,82	321,19	537,32	648,48	614,71	2.611,52
ALMARGEN	866,96	858,62	895,46	887,93	864,30	4.373,27
ALMOGIA	0,00	1.429,40	1.331,67	1.257,60	0,00	4.018,67
ALORA	6.207,23	5.509,82	5.780,89	5.047,17	4.758,35	27.303,46
ALOZAINA	890,50	4.481,51	842,74	848,99	794,28	7.856,02
ALPANDEIRE	160,03	152,68	151,78	152,86	147,46	764,81
ANTEQUERA	0,00	21.195,10	18.982,60	17.588,56	18.157,35	75.923,81
ARCHEZ	136,38	169,46	137,40	131,22	154,60	729,26
ARCHIDONA	4.198,49	3.606,98	3.943,92	3.352,29	3.484,90	18.566,58
ARDALES	1.062,97	1.068,76	1.103,39	1.072,76	1.093,79	5.401,67
ARENAS	431,16	534,12	410,74	388,97	444,06	2.209,05
ARRIATE	1.955,75	1.998,00	1.997,51	2.076,71	2.155,28	10.183,25
ATAJATE	56,33	59,29	61,25	56,49	45,89	279,25
BENADALID	111,37	123,72	93,84	93,67	83,38	505,98
BENALAURIA	217,97	236,24	184,10	182,69	159,00	980,00
BENAMARGOSA	969,00	838,98	1.010,27	1.509,88	999,64	5.327,77
BENAMOCARRA	1.137,07	1.078,16	1.109,64	1.080,42	989,96	5.395,25
BENAOJAN	947,18	905,93	855,17	860,79	833,04	4.402,11
BENARRABA	269,53	273,84	281,72	281,70	287,72	1.394,51
BORGE, EL	266,18	171,67	294,45	349,44	315,86	1.397,60

BURGO, EL	795,96	3.956,27	727,18	742,63	701,02	6.923,06
CAMPILLOS	3.484,75	3.471,86	3.650,17	3.566,13	3.667,25	17.840,16
CANIL, ACEITUNO	720,04	868,20	847,14	700,59	808,51	3.744,48
CANIL, ALBAIDA	284,54	351,08	273,35	265,75	311,19	1.485,91
CAÑETE LA REAL	810,61	799,56	816,23	782,55	765,82	3.974,77
CARRATRACA	360,63	355,99	366,82	354,02	344,87	1.782,33
CARTAJIMA	145,27	150,42	150,69	140,02	136,03	722,43
CARTAMA	9.876,42	9.067,03	9.792,69	8.829,00	8.645,78	46.210,92
CASABERMEJA	2.603,94	2.184,44	2.221,53	2.215,71	3.065,18	12.290,80
CASARABONELA	1.283,37	6.471,50	1.163,07	1.127,92	1.072,39	11.138,25
COIN	10.132,68	9.008,56	9.489,50	8.218,71	8.037,34	44.886,79
COLMENAR	1.311,53	1.165,63	988,25	936,93	915,67	5.318,21
COMARES	663,10	621,79	550,56	531,58	485,72	2.852,75
COMPETA	1.431,65	1.371,94	1.378,77	1.071,34	791,31	6.045,01
CORTES DE LA FRONTERA	1.609,44	1.700,36	1.284,39	1.265,60	1.118,47	6.978,26
CUEVAS BAJAS	0,00	28,90	840,00	568,22	594,67	2.031,79
CUEVAS DE SAN MARCOS	0,00	81,36	1.505,55	1.443,46	1.457,05	4.487,42
CUEVAS DEL BECERRO	724,88	711,57	750,41	719,15	720,18	3.626,19
CUTAR	176,94	31,29	182,12	284,97	212,18	897,50
FARAJAN	162,39	142,51	158,85	160,11	133,85	757,71
FRIGILIANA	1.279,93	1.448,28	1.505,16	1.574,78	1.718,35	7.526,50
FUENTE PIEDRA	1.199,30	1.089,34	1.048,95	1.009,94	1.086,33	5.433,86
GAUCIN	912,15	891,06	852,37	870,29	871,10	4.396,97
GENALGUACIL	217,11	229,24	194,84	186,62	161,26	989,07
GUARO	1.061,07	5.430,54	1.009,57	948,35	910,54	9.360,07
HUMILLADERO	1.170,06	1.364,73	1.260,10	1.368,16	1.439,47	6.602,52
IGUALEJA	565,71	513,47	508,10	485,90	447,81	2.520,99
IZNATE	344,35	327,89	341,45	332,63	302,70	1.649,02
JIMERA DE LIBAR	198,04	220,83	168,70	165,53	148,98	902,08
JUBRIQUE	327,61	350,62	279,73	264,20	230,09	1.452,26
JUZCAR	128,73	124,97	131,65	135,56	130,04	650,95
MACHARAVIAYA	129,98	119,07	147,94	128,07	160,50	683,56
MOCLINEJO	134,65	303,94	371,42	308,28	411,84	1.530,13
MOLLINA	1.831,70	2.165,35	1.982,47	2.118,77	2.175,99	10.274,28

MONDA	1.125,52	5.870,21	1.086,49	1.041,57	950,01	10.073,80
MONTEJAQUE	595,63	575,11	570,11	567,35	549,56	2.857,96
NERJA	9.090,40	13.549,58	13.308,96	12.978,46	13.440,41	62.367,81
PARAJUTA	136,41	143,64	144,16	135,00	131,13	690,34
PERIANA	1.510,75	1.560,97	1.336,65	1.041,22	1.058,15	6.507,74
PIZARRA	4.070,96	3.676,21	3.912,88	3.474,10	3.316,08	18.460,23
PUJERRA	199,00	196,23	186,59	181,86	170,85	934,53
RINCON DE LA VICTORIA	16.115,24	19.325,29	18.199,31	18.022,20	18.258,63	89.920,67
RIOGORDO	1.038,14	1.076,68	1.062,34	413,59	921,03	4.511,78
RONDA	21.746,76	20.871,94	20.440,22	20.472,58	19.949,99	103.481,49
SALARES	66,33	77,89	66,97	65,06	72,79	349,04
SAYALONGA	580,98	552,40	560,22	433,97	498,41	2.625,98
SEDELLA	216,66	261,03	211,58	197,38	227,27	1.113,92
SERRATO (RONDA)	0,00	0,00	0,00	0,00	0,00	0,00
SIERRA YEGUAS	1.435,28	1.424,34	1.489,08	1.447,78	1.474,17	7.270,65
TEBA	1.690,86	1.681,09	1.738,93	1.699,29	1.709,16	8.519,33
TOLOX	0,00	1.030,84	1.124,28	901,75	899,22	3.956,09
TORROX	7.039,42	1.030,84	9.011,30	8.193,50	8.869,92	34.145,06
TOTALAN	187,32	124,65	218,92	254,08	236,25	1.821,22
VALLE ABDALAJIS	0,00	59,93	1.187,42	1.002,84	997,34	3.247,53
VELEZ MALAGA	30.920,95	38.422,69	36.273,31	33.638,14	37.177,70	176.432,79
Vv ALGAIDAS	0,00	87,53	1.551,38	1.546,04	1.561,94	4.746,89
Vv CONCEPCION	1.239,88	1.008,87	1.252,19	1.187,04	2.020,59	6.708,57
Vv ROSARIO	1.759,40	1.509,40	1.623,52	1.384,81	1.428,15	7.705,28
Vv TAPIA	790,12	683,68	740,52	619,57	638,05	3.471,94
Vv TRABUCO	2.563,27	2.223,29	2.415,25	2.085,01	2.166,91	11.453,73
VIÑUELA	1.285,39	1.458,73	1.244,19	1.023,90	905,48	5.917,69
YUNQUERA	1.285,70	6.400,73	1.181,99	1.193,27	1.112,92	11.174,61
TOTAL:	207.431,33	259.598,74	242.672,65	227.619,39	230.971,35	1.168.293,66

MUNICIPIO	ALAMEDA	ALCAZON	ALFARRATE	ALFARATEJ	ALGARROBO	ALGATON	ALIS-TORRE	ALH-GARBE	ALHORNOS	ALPUJOLA	ALDIA	ALDANA	ALPANDERE	ARTEQUENA	ARCHEZ	ARCHONA	ARDALES	ARENAS	ARRABE	ARAUJE	BOMBALE
CUEVAS DE SAN MARCEL	28	89	75	67	107	69	30	107	102	71	109	124	104	103	125	101	101	101	101	101	101
CUEVAS DEL SECURDO	78	84	101	104	101	101	101	101	101	101	101	101	101	101	101	101	101	101	101	101	101
CUTAR	78	77	36	27	24	61	51	69	7	38	54	27	30	160	24	51	107	77	107	107	107
FARAJAN	109	101	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
FRIELIANA	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
FUENTE PEDRA	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
GAUCON	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
GENERALIZAL	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
GUARDO	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
HUMILADERO	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
IBRALEA	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
IONATE	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
LAPEDE DE LIBRAS	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
LAZQUEJE	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
LAZCAR	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
MACANABARRIA	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
MOLINO	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
MOLINA	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
MORONA	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
MONTAÑUE	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
NEFILA	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
PARAUTA	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
PICHANA	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
PICARRA	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
PLAZA	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
PRACION DE LA VICTOR	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
RECORDERO	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
ROMERA	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
SALARES	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
SAYALONGA	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
SEDELLA	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
SERRANO ROMAS	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
SIERRA VIEJAS	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
TIBA	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
TOLDE	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
TORREK	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
TOTALAN	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
VALLE ARDALES	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
VELEZ-MÁLAGA	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
VN ALGARRO	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
VN CONCEPCION	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
VN RICARDO	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
VN TAPPA	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
VN TRABUCO	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
VIRUELA	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
VIZQUEIRA	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107

MUNICIPIO	CIE-5	MAR	CUBOS-BECOTAS	FAMILIA	FRIGORÍFICA	FRUITE-PREC/GAUCON	GENA/GUAC/GUARD	HUELLAS/EGUALTA	IONATE	APEDA-LEA/ABORQUE	ASOCAN	MACHAMAYO/POCIBOJO	PELLINA	ROMBA	MONTAQUEMERA	PARADITA
CUEVAS DE SAN PABLO	1	89	107	82	289	42	289	107	89	107	82	289	107	89	107	82
CUEVAS DEL REBORDO	87	7	7	42	67	42	67	67	37	67	42	67	67	37	67	42
CUTAR	89	29	45	74	89	89	209	84	109	89	84	89	89	84	89	89
FAPALAN	85	42	74	7	239	87	47	77	82	8	89	82	87	77	87	8
FRIGORÍFICA	89	74	45	239	202	89	202	89	89	202	89	89	89	89	89	8
FUENTE PEDRA	42	87	87	87	89	7	89	89	8	89	87	89	87	89	87	89
GAUCON	89	87	89	47	202	89	7	24	89	27	87	89	87	89	202	89
GENERAL GONZALEZ	89	87	202	45	202	89	24	7	89	8	89	89	87	89	47	202
GUARD	87	87	77	69	87	87	24	7	89	8	89	89	87	89	47	202
HUELLAS	37	37	102	102	102	4	102	102	102	102	102	102	102	102	102	102
HUELLAS/EGUALTA	89	74	74	89	89	89	37	24	84	8	89	89	89	89	89	89
IONATE	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107	107
APESA DE LIBRAS	87	42	89	25	87	89	25	24	89	87	89	87	89	87	89	87
ABORQUE	82	37	89	7	202	89	22	7	89	87	89	87	89	87	89	87
ASOCAN	89	45	89	4	87	89	57	24	89	8	89	89	89	89	89	89
MACHAMAYO	87	87	87	89	89	89	89	89	89	89	89	89	89	89	89	89
POCIBOJO	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89
PELLINA	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89
MONTAQUEMERA	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89
NECIA	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89
PARADITA	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89
PERAMA	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89
PIERRA	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89
PLUERRA	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89
FRONCO DE LA VICTORIA	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89
RECORDADO	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89
ROMBA	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89
SALARES	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89
SAYALONGA	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89
SEDELLA	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89
SERRATO BORDA	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89
SERRA VEDRAS	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89
TIBA	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89
TILDEK	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89
TORROK	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89
TOTALAN	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89
VALLE ROSALES	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89
VILEZ-MALAGA	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89
Va ALGABAS	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89
Va CONCEPCION	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89
Va RECORDADO	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89
Va TAPPA	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89
Va TRASECO	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89
VIBELA	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89
YOMBREA	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89	89

MUNICIPIO	PARAGUATA	PERAMA	POBARRA	PUERRA	REBÓN VIEJO	ROMA	SALARES	SAYALONGA	SEDELLA	SERRATO	SIERRA YUGUETA	TOLDOX	TORREX	TOTALAN	VALLE-ARZOBISPADO	VALLE-ARZOBISPADO Y ALCAZAR Y CONCEPCIÓN	V. ROSARIO	V. TAPPA
CUEVAS DE SAN INACIO	27	64	21	27	38	18	162	127	18	8	72	74	122	102	102	8	162	162
CUEVAS DEL RECORRIDO	35	87	84	45	62	10	102	102	102	102	102	102	102	102	102	102	102	102
CUTAR	167	75	2	27	24	62	45	27	27	102	102	102	102	102	102	102	102	102
FARAJAN	8	167	87	1	167	87	167	167	167	167	167	167	167	167	167	167	167	167
FUENTE PIEDRA	87	45	45	27	27	87	87	87	87	87	87	87	87	87	87	87	87	87
GAUCON	42	264	36	47	167	84	27	102	102	102	102	102	102	102	102	102	102	102
GENERALCIBAL	29	87	35	22	102	102	42	102	102	102	102	102	102	102	102	102	102	102
GUINDO	167	167	167	167	167	167	167	167	167	167	167	167	167	167	167	167	167	167
HUENILAFERRO	167	167	167	167	167	167	167	167	167	167	167	167	167	167	167	167	167	167
IGUALZAR	8	167	75	8	102	102	27	102	102	102	102	102	102	102	102	102	102	102
IONATE	167	25	75	84	8	102	102	102	102	102	102	102	102	102	102	102	102	102
JAROSA DE LIBROS	37	167	87	42	167	167	167	167	167	167	167	167	167	167	167	167	167	167
ARROQUE	28	87	167	167	167	167	167	167	167	167	167	167	167	167	167	167	167	167
AYTOCAR	8	167	167	167	167	167	167	167	167	167	167	167	167	167	167	167	167	167
MACHOGARAYTA	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102
MOJONERO	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102
MOLLINA	87	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102
MONDA	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102
MONTEJUNQUE	7	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102
NEPIJA	84	48	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102
NEPIJA	84	48	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102
PARAGUATA	8	102	72	7	102	102	102	102	102	102	102	102	102	102	102	102	102	102
PERAMA	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102
POBARRA	72	72	7	167	167	167	167	167	167	167	167	167	167	167	167	167	167	167
PUERRA	8	102	167	167	167	167	167	167	167	167	167	167	167	167	167	167	167	167
REBÓN DE LA VICTORIA	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102
RECORRIDO	167	8	75	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102
ROMA	8	167	72	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102
SALARES	87	47	167	167	167	167	167	167	167	167	167	167	167	167	167	167	167	167
SAYALONGA	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102
SEDELLA	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102
SERRATO BONDAL	42	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102
SIERRA VEGUAS	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102
TAPPA	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102
TOLDOX	167	167	167	167	167	167	167	167	167	167	167	167	167	167	167	167	167	167
TORREX	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102
TOTALAN	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102
VALLE ROSALES	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102
VILEZ-MALAGA	167	167	167	167	167	167	167	167	167	167	167	167	167	167	167	167	167	167
V. ALCAZAR	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102
V. CONCEPCIÓN	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102
V. ROSARIO	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102
V. TAPPA	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102
V. TRASECO	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102
V. VIEJA	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102
YANQUEIRA	42	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102	102

MUNICIPIO	V-TRABUCCO	VIBUELA	YUNQUERA	MUNICIPIO	V-TRABUCCO	VIBUELA	YUNQUERA
CUEVAS BAJAS	54	67	37	ALAMEDA	57	66	67
CUEVAS DE SAN MARCO	47	36	66	ALCAZIN	36	7	109
CUEVAS DEL BECERRO	35	100	36	ALFARNATE	6	30	629
CUTAR	64	49	107	ALFARNATEJO	30	27	100
FARAJAN	103	177	50	ALGABROBO	62	22	74
FRIGLIANA	39	39	17	ALGATOCIN	117	179	64
FUENTE PIEDRA	48	24	70	ALH-TORRE	56	60	55
GAUCIN	80	108	73	ALH-GUAMBE	70	67	38
GENAL GUACIL	32	107	29	ALMACHAR	76	27	30
GUARNO	107	34	26	ALMARGEN	74	102	64
HUELVADEIRO	47	107	67	ALMOGIA	41	57	107
IGUALLEJA	115	102	54	ALORA	64	36	30
ISONTE	74	37	36	ALOCAINA	30	107	10
JAMEA DE LIBAR	106	174	60	ALPANDERE	100	100	54
JARRIQUE	107	119	70	ANTEQUEBIA	33	66	79
JAYCAR	102	107	10	ARCHEZ	34	34	106
MACHARAVAYA	60	36	30	ARCHONA	7	70	36
MOLINEJO	68	36	30	ARDALES	77	100	38
MOLLINA	47	79	37	ARENAS	60	27	107
MUNDA	62	30	30	ARRATE	100	100	40
MUNTEJANQUE	109	105	57	ATAJATE	107	109	55
NEBULA	36	36	10	BENADALID	106	174	60
PARAUTA	104	107	49	BENALABRA	100	170	64
PERBANA	33	17	16	BENAMARCOJA	50	10	107
POZARRA	77	62	37	BENAMOCARRA	71	10	100
PUEBLO	110	104	50	BENALJAN	104	102	49
RINCÓN DE LA VICTORIA	50	37	107	BENARRABIA	110	107	77
ROGORDO	40	11	100	BORQUE, EL	10	27	30
ROMA	107	100	36	BURGO, EL	106	106	9
SALARES	107	107	10	CAMPILLOS	57	36	49
SAYALONGA	100	27	100	CANIL ACETILINO	77	11	106
SEDELLA	70	10	10	CANIL ALBAIDA	36	36	100
SERRATO BOMBAI	100	106	27	CANIL LA REAL	10	10	37
SERRA YEGUAS	100	104	100	CARRATRACA	62	100	37
TEDA	77	100	49	CARTAJIMA	107	100	52
TOLIX	37	102	17	CARTAMA	50	70	40
TORREIX	10	37	100	CASABONERA	24	39	62
TOTALAN	64	42	114	CASABONELA	107	34	20
VALLE ARDILARES	53	77	107	COIN	73	64	39
VELEZ-MÁLAGA	70	10	107	COLMENAR	37	37	37
Vv-ALGARAS	107	100	36	COMARES	50	27	100
Vv-CONCEPCION	25	53	107	COMPETA	36	36	50
Vv-ROSARIO	5	50	10	CORTES DE LA FRONTE	11	107	107
Vv-TAPPA	36	37	100	CUEVAS BAJAS	54	67	37
Vv-TRABUCCO	10	107	30				
VIBUELA	67	10	107				
YUNQUERA	36	107	10				

Tabla D.7: Residuos centro transferencia Ronda 2010

MUNICIPIO	Hab.2009	MESES												Total Kg. Municipio / Anual	Total Toneladas Municipio / Anual	
		Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre			
Circuito nº 1	JAMERA DE LEJAR	457	14.444	12.395	18.028	16.206	14.519	14.333	11.657	17.918	38.180	14.575	19.496	15.988	190.038	192,04
	ATAJATE	130	4.109	3.528	5.128	4.638	4.130	4.077	3.316	5.007	10.861	4.146	2.986	4.320	56.335	56,33
	SEVALJADO	257	8.123	6.971	10.138	9.170	8.165	8.061	6.555	10.676	21.471	8.196	5.902	8.541	111.370	111,37
	SEVALJURA	503	15.887	13.643	19.842	17.947	15.961	15.776	12.830	19.721	42.025	16.042	11.552	16.717	217.872	217,87
	ALGATÓN	909	28.729	24.655	35.868	32.433	28.686	28.516	23.186	35.639	75.942	28.991	26.877	30.208	383.910	383,91
	JUBROQUE	756	23.893	20.585	29.923	26.974	24.019	23.711	19.283	29.641	63.960	24.111	17.363	25.129	327.698	327,69
	SEVALJUNCL	507	15.834	13.589	19.763	17.876	15.917	15.714	12.779	19.643	41.866	15.978	11.506	16.650	217.106	217,11
	CORTES DE LA FRONTERA	3.714	117.381	100.736	146.510	132.516	117.998	116.487	94.733	145.615	310.286	118.450	85.298	123.438	1.609.441	1.609,44
	Totales :	7.227	228.410	196.020	285.096	257.860	229.618	226.678	184.340	283.350	603.760	230.430	165.966	240.180	3.131.780	3.131,78
	Circuito nº 2	SAUCÓN	1.829	82.665	71.915	67.476	75.837	73.625	82.073	96.674	86.183	74.381	66.752	67.013	67.845	912.148
SEVALFRASA		570	24.424	21.250	19.938	22.350	21.755	24.252	28.566	25.443	21.979	19.724	19.802	20.048	269.537	269,53
AFRIATE		4.136	177.222	154.195	144.676	182.174	157.860	175.975	207.280	184.615	159.481	143.124	143.685	145.468	1.650.751	1.650,75
Totales :		6.635	284.300	247.360	232.090	260.160	253.248	282.300	332.520	296.160	255.840	229.600	239.500	233.360	3.137.430	3.137,43
ALPARRICEPE		271	13.536	10.744	13.778	13.096	14.029	13.484	13.223	15.225	14.035	9.788	12.575	15.396	180.029	180,03
BEVALJÁN		1.664	82.487	63.589	81.550	77.514	86.688	79.810	78.267	90.112	83.068	57.458	74.427	92.310	947.180	947,18
CARTAJUNA		246	12.651	9.752	12.507	11.888	13.286	12.240	12.004	13.820	12.740	8.852	11.415	14.151	145.286	145,27
FARAJÁN		275	14.142	10.902	13.982	13.289	14.845	13.683	13.419	15.449	14.242	9.851	12.760	15.826	182.391	182,39
SEVALJEA		698	49.266	37.979	48.707	46.296	51.715	47.667	46.746	53.820	49.613	34.317	44.462	55.133	565.710	565,71
LECCAR		218	11.211	8.642	11.084	10.536	11.768	10.847	10.637	12.347	11.290	7.809	10.115	12.546	128.731	128,73
MONTEJADE	1.009	51.889	40.001	51.296	48.780	54.469	50.205	49.234	56.685	52.254	36.144	48.818	58.068	585.826	585,83	
PARAJUA	237	11.879	9.158	11.744	11.953	12.470	11.494	11.272	12.977	11.963	8.275	10.719	13.294	136.408	136,41	
PUEBERRA	307	17.331	13.360	17.134	16.286	18.192	16.768	16.444	18.922	17.453	12.872	15.637	19.394	199.002	199,00	
RONDA	36.827	1.893.867	1.459.973	1.872.365	1.779.673	1.988.023	1.832.407	1.798.974	2.068.913	1.907.194	1.319.195	1.708.802	2.119.396	21.746.758	21.746,76	
Totales :	47.876	2.158.660	1.664.190	2.134.140	2.028.500	2.265.986	2.088.600	2.048.220	2.358.180	2.173.850	1.503.640	1.947.720	2.415.710	24.787.300	24.787,30	
Total Anual - Circuito RONDA :												31.866.510	31.866,51			

Tabla D.8: Residuos Planta tratamiento Antequera 2010

MUNICIPIO	Hab 2009	MESES												Total Kg Mensuales/Anual	Total Toneladas Mensuales/Anual	MEDIA ANUAL			
		Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre			Kg/Habitado	Ton/Habitado		
Circulo Lateral 1	ALAMEDA	5.481	261.727	167.420	153.945	209.244	247.653	177.244	164.606	108.290	287.925	296.241	298.536	295.035	2.381.963	2.381,96	430,84	0,431	
	FUENTE PIEDRA	2.782	132.893	86.012	86.834	105.736	125.747	89.996	53.134	54.960	169.535	194.719	105.885	149.006	1.199.267	1.199,26	430,84	0,431	
	Totales :		8.264	394.620	252.440	254.880	313.980	373.400	267.240	163.260	313.960	310.960	314.420	444.040	3.581.230	3.581,26	430,84	0,431	
Circulo Lateral 2	MOLINA	5.152	62.461	122.237	173.943	149.990	113.829	169.996	273.118	279.452	158.764	133.734	121.322	82.789	1.831.703	1.831,70	355,53	0,356	
	MUJALLADERO	3.291	39.899	78.083	116.537	96.810	72.711	102.834	174.462	178.508	161.416	86.426	77.498	52.871	1.170.057	1.170,06	355,53	0,356	
	Totales :		8.442	102.360	200.320	293.580	245.800	199.540	283.829	447.980	457.960	269.160	198.820	135.646	3.001.760	3.001,76	355,53	0,356	
Circulo Archidona y Villanueva	ARCHIDONA	8.698	365.878	329.629	383.227	364.222	365.455	377.645	329.313	371.812	331.707	316.814	308.540	344.445	4.198.489	4.198,49	473,98	0,474	
	VIA DEL ROSARIO	3.712	153.240	128.206	160.594	152.829	161.527	158.256	126.325	155.726	129.084	120.260	129.296	144.342	1.759.403	1.759,40	473,98	0,474	
	VIA DEL TRABUCCO	5.498	223.254	198.437	223.969	222.365	225.329	230.641	198.811	228.877	202.614	192.688	188.370	219.296	2.903.268	2.903,27	473,98	0,474	
	VIA DE TAPPA	1.667	68.818	61.168	72.120	68.543	72.538	71.078	81.221	89.934	42.424	59.396	58.965	64.822	790.120	790,12	473,98	0,474	
	Totales :		19.545	810.590	720.840	845.910	807.790	854.858	837.536	721.470	834.150	698.960	694.270	753.906	9.311.280	9.311,28	473,98	0,474	
Circulo El Calmar	CASABERNEJA	3.520	34.420	88.175	124.602	89.366	97.610	96.456	107.636	107.606	172.180	97.225	81.580	118.292	1.274.949	1.274,94	362,20	0,362	
	COLMENAR	3.621	97.139	96.785	128.178	91.724	100.410	99.234	110.724	110.694	177.120	100.015	83.920	121.687	1.311.571	1.311,57	362,20	0,362	
	Totales :		7.141	191.559	178.880	252.780	180.890	198.020	195.680	218.360	349.300	197.240	165.500	239.979	2.586.520	2.586,52	362,20	0,362	
Circulo El Tossal	CASABERNEJA	3.520	89.789	48.868	96.400	97.623	111.177	101.792	114.229	119.961	106.676	109.320	174.531	164.639	1.268.998	1.268,99	377,55	0,378	
	VIA DE LA CONCEPCION	3.284	83.751	45.932	84.340	91.071	103.723	94.968	106.571	111.919	99.534	101.960	162.829	153.601	1.228.864	1.228,86	377,55	0,378	
	Totales :		6.804	173.520	94.800	174.740	188.790	214.900	216.190	220.800	219.880	211.310	337.360	318.200	2.497.862	2.497,87	377,55	0,378	
CONSORCIO GUADAL TEBA	ALVARIGÓN	2.154	71.814	60.319	73.873	66.136	74.928	79.282	77.666	79.163	72.885	89.479	79.311	71.095	866.962	866,96	402,49	0,402	
	APDALES	2.641	88.651	73.966	96.979	81.889	91.868	86.184	95.226	97.061	89.364	95.188	97.243	87.169	1.082.874	1.082,87	402,49	0,402	
	CAMPILLOS	8.698	288.638	242.461	296.932	265.835	301.172	282.538	313.160	318.197	282.962	279.272	318.791	285.796	3.684.752	3.684,75	402,49	0,402	
	CARTE LA REAL	2.014	67.147	56.398	69.071	61.837	78.058	65.723	72.818	74.018	68.148	64.964	74.156	66.476	810.613	810,61	402,49	0,402	
	CARRATRACA	896	29.873	25.091	36.729	27.511	31.168	29.279	32.307	32.930	36.318	28.961	32.991	29.579	360.630	360,63	402,49	0,402	
	CUEVAS DEL BECERRO	1.801	60.045	50.424	61.766	55.296	62.648	58.772	64.938	66.190	69.541	58.093	66.314	59.444	724.883	724,88	402,49	0,402	
	SIERRA DE VEGUAS	3.596	118.891	99.839	122.298	109.490	124.045	116.370	128.579	131.057	129.683	115.825	131.302	117.700	1.426.277	1.426,27	402,49	0,402	
	TEBA	4.207	140.061	117.641	144.076	128.987	146.133	137.092	151.475	154.394	142.159	136.507	154.925	138.658	1.690.858	1.690,86	402,49	0,402	
		Totales :		25.071	864.540	726.105	889.320	796.190	902.026	846.210	934.990	953.010	877.430	836.430	855.080	10.426.930	10.426,93	402,49	0,402
	NORTE - ANTEQUERA	ANTEQUERA	45.168												0	0,00	0,00	0,000	
		CUEVAS BAJAS	1.454												0	0,00	0,00	0,000	
		CUEVAS DE SAN MARCOS	4.131												0	0,00	0,00	0,000	
VILLE DE ACHALAS		2.842												0	0,00	0,00	0,000		
	VALLEVALE ALBUCAJAS	4.560												0	0,00	0,00	0,000		
	Totales :		58.155	0	0	0	0	0	0	0	0	0	0	0	0,00	0,00	0,000		
Total Anual - Circuito ANTEQUERA :												31.468.608	31.468,60						

Tabla D.9: Residuos centro transferencia Axarquia 2010

MUNICIPIO	Habr. 2009	MESES												Total Residuos Municipio/ Añal	RESIDUA ANUAL			
		Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre		Ag/Municipio	Total/Municipio		
Axarquia																		
Circuito 1																		
ALFARABATE	7.667	43.297	39.677	50.717	54.303	46.534	42.226	54.494	48.610	52.976	37.782	40.536	563.669	563.669	48.37	0.448		
ALFARABATELO	539	17.380	15.636	20.342	21.673	18.564	16.653	21.745	18.684	21.944	18.850	16.178	225.065	225.065	48.37	0.448		
PERIANA	3.677	15.363	106.267	125.367	145.465	124.602	103.715	145.951	143.563	125.403	99.476	108.566	1.510.747	1.510.747	48.37	0.448		
Total:	5.687	176.960	161.600	206.820	221.440	193.660	172.200	222.800	200.900	226.940	161.340	165.300	2.269.600	2.269.600	48.37	0.448		
Circuito 2																		
BENASQUERA	3.687	66.002	74.602	91.446	94.180	98.165	114.376	114.376	100.968	91.342	93.724	91.676	1.037.074	1.037.074	371.47	0.371		
CÓMPETA	3.654	107.024	93.328	116.107	123.634	126.033	144.007	127.125	116.005	116.005	116.677	116.436	1.431.651	1.431.651	371.47	0.371		
SAYALONGA	1.564	43.432	38.111	46.724	48.121	50.172	54.736	58.440	46.671	47.688	46.343	48.097	580.367	580.367	371.47	0.371		
Total: p. CANTABRIA	3.27	25.742	22.533	27.634	28.522	29.726	32.478	34.638	30.577	27.662	28.384	28.503	344.354	344.354	371.47	0.371		
Circuito 3																		
ARCHEZ	467	11.100	9.665	10.100	10.300	11.070	12.320	12.650	14.370	12.837	10.074	10.962	126.307	126.307	309.96	0.310		
ARENAS	1.897	35.363	31.257	32.247	34.747	34.936	38.363	39.950	45.250	40.581	31.647	33.390	32.534	431.681	431.681	309.96	0.310	
CABILLAS DE ACITURIO	2.327	59.066	52.160	53.843	58.016	58.443	65.070	68.787	75.568	67.771	53.165	55.765	720.040	720.040	309.96	0.310		
CABILLAS DE ALBARRAS	316	23.342	20.624	21.278	22.928	23.096	25.714	26.393	29.863	26.782	21.096	22.007	21.471	264.544	264.544	309.96	0.310	
SALARES	274	5.441	4.806	4.960	5.345	5.394	5.994	6.163	6.962	6.243	4.900	5.107	5.009	66.337	66.337	309.96	0.310	
SEDELLA	609	11.773	10.704	10.202	17.456	17.506	19.900	20.097	22.779	20.390	16.004	16.790	16.349	276.667	276.667	309.96	0.310	
VIGÜELA	1.594	50.701	44.736	46.276	43.800	50.166	55.654	57.326	64.866	58.173	45.653	47.667	670.667	670.667	309.96	0.310		
Total:	7.979	202.880	175.260	184.540	199.280	200.740	223.500	229.400	259.560	232.780	182.680	186.630	2.473.800	2.473.800	309.96	0.310		
Circuito 4																		
MACHARAVIA	507	10.194	9.100	9.600	9.974	10.603	12.406	12.700	12.482	10.534	10.107	10.329	129.379	129.379	293.44	0.293		
TOTALÁN	722	14.633	13.855	13.846	14.207	15.096	17.676	18.374	17.360	15.267	14.965	15.745	16.846	187.105	187.105	293.44	0.293	
BORQUE (EL)	1.038	20.734	18.637	19.676	20.302	21.878	25.403	26.024	25.522	21.636	20.637	22.381	22.975	268.885	268.885	293.44	0.293	
BENAGÜJE	287	4.070	3.635	3.737	3.916	4.222	4.902	5.022	4.305	4.167	3.954	4.378	4.424	57.309	57.309	293.44	0.293	
CÚTAR	687	13.822	12.527	13.079	13.495	14.343	16.066	17.299	16.365	14.427	13.756	14.677	15.272	176.537	176.537	293.44	0.293	
ALMACHAR	1.689	36.265	34.662	36.206	37.359	40.259	46.746	47.669	46.364	39.322	38.087	41.164	42.276	469.627	469.627	293.44	0.293	
MOCLEJAO	579	10.576	9.526	9.953	10.270	11.087	12.660	13.164	12.910	10.934	10.476	11.321	11.623	124.649	124.649	293.44	0.293	
EL VALDES	464	9.404	8.576	8.898	9.161	9.694	10.488	11.769	11.542	9.611	9.360	10.322	10.384	130.380	130.380	293.44	0.293	
BENAMARROSA	1.627	33.656	29.944	31.276	32.274	34.775	40.363	41.370	40.571	34.488	32.302	35.576	38.523	423.445	423.445	293.44	0.293	
Total:	7.677	160.340	146.100	149.340	161.000	162.720	188.940	193.660	189.620	161.360	153.940	160.460	1.866.060	1.866.060	309.96	0.260		
Circuito 5																		
BENAMARROSA	1.627	44.331	38.715	48.244	44.452	47.235	46.391	48.763	54.464	46.363	35.670	34.972	45.643	545.646	545.646	334.67	0.335	
PROSORO	3.162	84.312	73.633	81.735	84.543	89.837	88.232	92.742	122.604	89.329	67.841	66.910	1.038.144	1.038.144	334.67	0.335		
Total:	1.594	54.197	47.332	58.981	54.345	57.746	58.710	59.635	57.422	43.609	42.755	55.796	667.330	667.330	334.67	0.335		
Circuito 6																		
ALCAICÓN	6.277	162.640	163.660	169.960	163.340	194.630	201.100	205.660	193.720	147.120	144.240	166.340	2.251.120	2.251.120	334.67	0.335		
ALCAICÓN	2.578	66.330	75.329	93.031	95.476	90.160	95.700	85.143	89.162	89.162	89.162	85.835	32.126	1.073.627	1.073.627	416.79	0.417	
ALGARROBO	6.129	208.563	184.273	224.596	220.360	217.640	221.057	205.517	223.696	216.304	216.304	207.223	222.411	2.591.969	2.591.969	416.79	0.417	
COMARES	1.597	53.956	47.142	57.458	58.933	55.675	59.111	53.577	58.763	55.061	55.061	53.094	56.856	663.089	663.089	416.79	0.417	
FRIGILIANA	3.077	102.390	90.996	110.300	113.754	107.473	114.096	101.496	110.426	106.379	106.379	102.329	103.629	1.279.333	1.279.333	416.79	0.417	
MERLA	27.677	731.463	646.276	707.634	607.909	763.296	810.360	720.760	805.577	705.104	705.104	726.367	810.000	9.090.400	9.090.400	416.79	0.417	
FINCA DE LA VICTORIA	36.665	1.296.719	1.145.701	1.396.404	1.432.241	1.363.163	1.436.571	1.277.762	1.428.107	1.338.724	1.338.724	1.288.367	1.362.166	16.165.234	16.165.234	416.79	0.417	
TORROX	8.680	566.430	500.462	609.974	625.638	591.081	623.520	558.568	623.823	584.737	584.737	562.751	604.943	7.039.429	7.039.429	416.79	0.417	
VELIZ MALAGA	74.189	2.468.068	2.198.307	2.675.336	2.748.031	2.596.345	2.796.406	2.451.731	2.740.966	2.566.463	2.566.463	2.472.060	2.653.266	30.320.948	30.320.948	416.79	0.417	
Total:	85.074	5.533.360	4.883.480	5.959.400	6.182.340	6.174.620	6.100.620	5.453.160	6.094.700	5.712.840	5.712.840	5.438.420	6.174.620	68.774.620	68.774.620	416.79	0.417	
Total Anual - Circuito AXARQUIA																		
81.229.640																		

Tabla D.10: Residuos centro transferencia Gualhorce 2010

MUNICIPIO	Hab.2009	MESES												Total Trazadas Municipio /Anual	MEDIA ANUAL		
		Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre		Kg/Habitado	Ton/Habitado	
Circuito nº 1																	
CASABUQUELA	2.748	118.212	118.792	96.317	99.927	113.929	104.380	121.158	110.689	108.248	111.999	68.658	112.065	1.283.371	1.283,37	487,02	0,487
GUARO	2.272	97.796	96.562	79.633	82.618	94.196	96.299	100.170	91.516	89.496	92.599	57.582	92.654	1.081.070	1.081,07	487,02	0,487
MOYDA	2.410	103.672	102.427	84.470	87.636	98.916	91.541	106.254	97.075	94.934	98.223	61.090	98.291	1.125.519	1.125,52	487,02	0,487
Totales :	7.430	319.620	315.780	290.420	270.180	308.040	292.220	327.580	299.280	292.680	302.820	188.340	303.000	3.489.960	3.489,96	487,02	0,487
Circuito nº 2																	
ALONDA	2.242	83.148	86.968	87.866	69.061	70.792	85.304	87.373	96.905	53.457	66.562	100.972	59.298	890.498	890,50	397,19	0,397
BURGO, EL	2.094	74.321	59.789	60.653	61.700	63.277	58.371	78.086	86.528	47.782	59.496	82.834	53.004	795.965	795,96	397,19	0,397
YUNQUERA	3.237	120.049	96.543	97.971	99.710	102.210	94.285	126.149	139.767	77.181	96.102	150.114	85.616	1.285.698	1.285,70	397,19	0,397
Totales :	7.483	277.518	223.180	228.486	209.500	236.280	217.960	291.620	323.100	178.420	222.160	347.020	197.930	2.972.158	2.972,16	397,19	0,397
Transf. Cartama - Casarabonela																	
CARTAMA	21.313	735.811	639.294	604.712	725.908	765.771	784.938	2.100.089	758.379	710.306	783.677	678.224	688.116	9.878.425	9.878,42	463,40	0,463
ALH. DE LA TORRE	35.144	1.212.277	1.053.281	996.267	1.195.961	1.182.795	1.280.260	3.459.978	1.249.458	1.170.257	1.291.467	1.117.401	1.192.396	16.271.795	16.271,80	463,40	0,463
ALH. EL GRANDE	23.319	805.066	699.465	661.628	794.231	772.199	838.935	2.297.751	829.738	777.161	857.656	742.056	732.094	10.806.003	10.806,00	463,40	0,463
ALORA	13.365	462.450	401.789	388.055	465.226	443.570	480.759	1.319.884	476.633	446.420	482.658	428.257	420.533	6.207.231	6.207,23	463,40	0,463
COB	21.866	754.903	655.861	628.402	744.743	724.083	784.786	2.154.579	778.058	728.736	864.216	685.822	688.478	10.132.684	10.132,68	463,40	0,463
POARRA	8.785	303.294	283.510	249.256	299.212	290.912	315.300	865.635	312.996	282.791	323.106	279.557	275.803	4.070.961	4.070,96	463,40	0,463
Totales :	123.792	4.273.800	3.713.200	3.573.340	4.216.280	4.099.320	4.442.982	12.187.917	4.604.880	4.125.660	4.552.980	3.938.320	3.886.428	57.365.099	57.365,10	463,40	0,463
TOLOX	2.373													0	0,00	0,00	0,000
Totales :	2.373	0	0	0	0	0	0	0	0	0	0	0	0	0	0,00	0,00	0,000
Total Anual - Circuito GUADALHORCE :												63.807.217	63.807,22				

Tabla D.11: Residuos centro transferencia Ronda 2011

RECOGIDA R.S.U. - CONSORCIO PROVINCIAL DE RESIDUOS SÓLIDOS DE MÁLAGA ---- AÑO 2.011

MUNICIPIO	Hab.2010	MESES												Total Kg. Municipio / Anual	Total Tn. Municipio / Anual	MEDIA ANUAL	
																Kg/Habitante	Tn/Habitante
		Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre				
Circuito nº 1																	
CASAPASELA	2.770	45.968	90.815	5.424.848	76.825	112.512	84.623	126.794	113.697	118.890	100.363	100.363	81.683	6.471.496	6.471.501	2.376,61	2,377
GUARO	2.285	38.574	76.207	4.552.244	64.468	94.498	71.011	101.364	95.409	99.766	84.219	84.238	68.544	5.430.543	5.430.547	2.376,61	2,377
RONDA	2.470	41.697	82.377	4.320.806	68.687	102.149	76.760	109.571	103.134	107.844	91.038	91.058	74.093	5.670.215	5.670.219	2.376,61	2,377
Totales :	7.478	125.240	249.400	14.897.896	210.980	309.290	232.394	331.730	312.249	328.500	275.620	275.600	224.320	17.772.254	17.772.257	2.376,61	2,377
Circuito nº 2																	
ALDONA	2.244	78.344	65.000	3.604.583	93.789	79.262	87.030	86.621	98.996	83.955	75.982	68.625	65.321	4.481.507	4.481.511	1.997,11	1,997
BURGO,EL	1.981	69.162	57.362	3.182.121	82.797	69.972	76.830	71.172	87.393	74.116	67.077	60.582	57.666	3.966.268	3.966.271	1.997,11	1,997
YUNQUERA	3.205	111.895	92.837	5.148.257	133.954	113.295	124.301	115.147	141.391	119.909	108.521	98.013	93.254	6.400.726	6.400.731	1.997,11	1,997
Totales :	7.430	259.400	215.226	11.934.960	310.540	262.440	266.940	266.940	327.780	277.980	251.500	227.220	216.200	14.838.500	14.838.505	1.997,11	1,997
Transf. Cárrama - Casarabonela																	
CÁRTAMA	22.173	735.793	499.182	1.076.094	569.296	798.327	799.292	803.469	807.747	771.537	734.721	734.619	736.457	9.067.022	9.067.027	408,92	0,409
ALH DE LA TORRE	35.632	1.189.055	806.687	1.738.988	919.933	1.290.319	1.291.672	1.298.422	1.305.335	1.246.819	1.187.323	1.187.568	1.190.128	14.682.500	14.682.505	408,92	0,409
ALH EL GRANDE	23.679	785.635	532.996	1.148.988	607.860	852.338	853.436	857.896	862.464	823.881	784.491	784.382	786.345	9.681.230	9.681.235	408,92	0,409
ALORA	13.474	447.124	303.341	653.916	345.947	485.428	485.711	488.249	498.848	468.844	446.472	446.410	447.527	5.509.816	5.509.821	408,92	0,409
ODÉN	22.030	731.047	495.962	1.069.154	565.624	793.675	794.137	798.288	802.537	766.561	729.982	729.881	731.767	9.008.556	9.008.561	408,92	0,409
POZARRA	8.990	298.326	202.362	436.300	230.820	323.083	324.071	325.765	327.499	312.818	297.891	297.850	298.595	3.676.211	3.676.216	408,92	0,409
Totales :	126.174	4.195.900	2.840.560	6.123.448	3.239.540	4.545.670	4.548.320	4.572.090	4.596.430	4.396.380	4.180.880	4.180.300	4.190.760	51.595.350	51.595.355	408,92	0,409
Circuito nº 3																	
TELDOX	2.346	82.125	78.548	56.620	58.745	65.040	91.320	104.120	154.800	92.880	92.546	77.060	77.840	1.030.844	1.030.849	439,40	0,439
Totales :	2.346	82.125	78.548	56.620	58.745	65.040	91.320	104.120	154.800	92.880	92.546	77.060	77.840	1.030.844	1.030.849	439,40	0,439
Total Anual - Circuito GUADALHORCE :														85.238.948	85.238,95		

RECOGIDA R.S.U. - CONSORCIO PROVINCIAL DE RESIDUOS SÓLIDOS DE MÁLAGA ---- AÑO 2.011

MUNICIPIO	Hab. 2010	MESES												Total Kg Municipio/Anual	Total Tn Municipio/Anual	MEDIA ANUAL	
		Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre			Kg/Habitante	Tn/Habitante
Circuito Lateral 1	5.406	144.116	167.488	167.941	196.704	189.986	175.094	198.492	173.718	184.302	230.904	204.706	99.094	2.138.145	2.138,15	389,75	0,380
	2.795	73.424	85.332	85.359	100.216	96.794	89.206	101.128	91.582	93.898	117.641	104.294	50.486	1.069.340	1.069,34	389,75	0,380
Totales :		8.297	217.540	252.820	296.920	286.780	264.300	299.620	271.280	278.200	348.545	309.000	149.580	3.227.485	3.227,49	389,75	0,380
Circuito Lateral 2	5.274	132.936	139.499	181.946	226.234	220.394	193.871	204.801	259.271	182.767	203.894	144.517	65.217	2.863.346	2.863,35	492,57	0,411
	3.324	83.784	87.921	114.674	142.596	138.906	122.189	129.079	163.409	121.493	128.506	91.083	41.103	1.364.734	1.364,73	492,57	0,411
Totales :	8.598	216.720	227.420	296.620	368.820	359.300	316.080	333.880	422.680	314.260	332.400	235.600	106.320	3.530.680	3.530,68	492,57	0,411
Circuito Archidona y Villanueva	8.637	317.920	275.594	323.247	330.680	334.787	334.005	346.887	367.232	316.782	314.332	245.109	100.441	3.606.977	3.606,98	408,17	0,408
	3.698	133.039	115.327	135.269	138.371	140.098	139.770	145.153	153.675	132.563	131.538	102.570	42.031	1.509.404	1.509,40	408,17	0,408
Totales :	5.447	195.961	169.872	199.245	203.814	206.358	205.876	213.804	226.357	195.260	193.750	151.082	61.910	2.223.289	2.223,29	408,17	0,408
Circuito El Colmenar	1.675	60.260	52.237	61.270	62.675	63.457	63.309	65.747	69.607	60.044	59.580	46.459	19.038	683.687	683,68	408,17	0,408
	8.657	707.180	619.090	719.090	735.520	744.700	742.960	771.570	816.870	704.650	699.200	545.220	223.420	8.023.350	8.023,35	408,17	0,408
Totales :	3.554	171.673	77.869	96.309	89.530	130.724	87.526	87.448	84.166	89.098	88.843	91.348	30.986	1.425.510	1.425,51	316,69	0,317
Circuito El Torcal	3.697	177.807	80.651	99.751	92.730	135.396	90.654	90.674	87.164	92.282	92.017	94.612	32.094	1.865.637	1.865,63	316,72	0,317
	7.235	349.480	158.520	196.060	182.260	286.120	176.180	178.122	171.320	181.380	180.860	185.960	63.080	2.291.942	2.291,94	316,70	0,317
Totales :	3.554	165.717	87.006	102.175	72.289	51.210	118.275	113.841	114.005	113.851	43.775	65.488	11.297	1.659.329	1.659,33	297,95	0,298
CONSORCIO GUADALTEBA	3.386	157.883	82.894	97.345	68.871	48.790	112.695	108.459	108.615	108.469	41.705	62.392	10.763	1.008.872	1.008,87	297,95	0,298
	6.940	323.600	169.900	199.520	141.160	100.000	230.960	222.300	222.620	222.320	85.480	127.880	22.060	2.067.800	2.067,80	297,95	0,298
Totales :	2.137	66.393	64.620	71.719	75.235	78.016	75.444	80.304	88.435	76.898	77.236	74.553	29.170	659.624	659,62	401,79	0,402
NORTE ANTEQUERA	2.660	83.388	80.435	89.271	93.648	97.109	93.908	99.958	110.079	95.717	96.139	92.799	36.309	1.068.780	1.068,78	401,79	0,402
	8.641	270.886	261.292	289.996	304.216	315.460	305.059	324.712	357.590	310.937	312.307	301.458	117.949	3.471.662	3.471,66	401,79	0,402
Totales :	1.990	62.384	60.175	66.785	70.080	72.650	70.254	74.780	82.352	71.608	71.923	69.425	27.163	799.567	799,56	401,79	0,402
Circuitos ANTEQUERA	886	27.775	26.791	29.735	31.193	32.345	31.279	33.294	36.665	31.882	32.022	30.910	12.094	355.985	355,99	401,79	0,402
	1.777	55.519	53.553	59.436	62.350	64.654	62.523	66.551	73.289	63.728	64.008	61.785	24.174	711.569	711,57	401,79	0,402
Totales :	3.545	111.132	107.196	118.972	124.806	129.418	125.162	133.214	146.703	127.563	128.125	123.674	48.389	1.424.343	1.424,34	401,79	0,402
Circuitos ANTEQUERA	4.894	131.164	126.518	140.417	147.302	152.747	147.711	157.227	173.146	150.557	151.220	145.367	57.111	1.681.087	1.681,09	401,79	0,402
	25.814	809.240	780.580	866.330	908.810	942.400	911.330	970.040	1.068.260	928.890	932.980	900.570	352.360	10.371.790	10.371,79	401,79	0,402
Totales :	45.294	1.898.320	1.550.160	1.790.365	1.710.675	2.026.465	1.949.855	1.763.665	1.759.435	1.681.960	1.785.395	1.635.135	1.633.065	21.895.095	21.895,10	468,57	0,469
Circuitos ANTEQUERA	1.459	1.620	2.569	2.632	2.480	1.510	2.291	2.564	3.099	2.752	2.543	2.281	2.569	28.902	28,90	19,81	0,020
	4.107	4.560	7.231	7.408	6.990	6.449	7.216	8.707	7.748	7.157	6.419	7.231	81.359	81,36	19,81	0,020	
Totales :	2.802	5.089	4.239	4.715	5.256	5.400	5.243	5.600	5.269	4.739	4.235	4.923	59.937	59,93	21,39	0,021	
Circuitos ANTEQUERA	4.515	3.900	3.500	3.500	3.500	5.663	10.183	11.981	10.598	10.998	11.944	9.326	87.534	87,53	19,39	0,019	
	58.117	1.903.699	1.587.699	1.808.620	1.728.891	2.041.125	1.969.501	1.808.841	1.788.816	1.708.327	1.810.773	1.659.414	1.657.113	21.452.820	21.452,82	369,13	0,369
Totales :																	
Total Anual - Circuito ANTEQUERA :													50.964.667	50.964,67			

RECOGIDA R.S.U. - CONSORCIO PROVINCIAL DE RESIDUOS SÓLIDOS DE MÁLAGA ---- AÑO 2011

MUNICIPIO	Hab. 2010	MESES												Total Kg Municipio / Anual	Total Tn Municipio / Anual	MEDIA ANUAL	
		Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre				
Axarquía																	
Circuito nº 1																	
ALFARRATE	7.037	41.962	35.560	43.246	45.539	53.720	50.285	53.736	43.952	43.536	50.704	65.260	579.640	579.64	433,36	0,433	
ALFARRATEJO	947	17.030	14.431	17.951	18.506	21.005	21.195	21.030	17.796	17.630	20.577	26.430	235.78	235,78	433,36	0,433	
PERPIÑA	3.602	10.020	35.725	116.423	122.756	144.645	105.316	144.823	118.052	117.365	106.495	175.738	1.580.574	1.580,57	433,36	0,433	
Teléfono :	5.467	172.040	145.710	177.220	186.880	220.900	294.020	205.590	179.700	178.660	207.700	287.530	2.208.200	2.208,20	433,36	0,433	
Circuito nº 2																	
BENABADILLA	3.029	53.439	77.214	120.733	103.765	103.765	81.770	103.056	73.328	104.851	103.236	71.659	1.029.250	1.029,25	385,24	0,385	
CÓRTEVA	3.682	118.300	98.251	93.631	132.045	140.362	86.237	49.234	140.045	94.072	133.421	91.160	1.377.947	1.377,94	385,24	0,385	
SAYALONGA	7.533	47.874	33.901	61.850	53.817	56.753	34.722	11.646	56.306	37.877	53.728	36.710	552.407	552,40	385,24	0,385	
CUARTO DE BARRERAS	3.027	28.417	23.462	36.710	31.956	33.687	20.670	11.781	33.470	22.483	31.680	21.750	317.680	317,68	385,24	0,385	
Teléfono :	9.329	268.620	238.516	372.340	320.540	342.860	209.340	119.660	339.360	303.680	325.070	221.340	3.330.330	3.330,33	385,24	0,385	
Circuito nº 3																	
ARCHEZ	469	10.726	9.796	12.643	12.257	12.332	20.169	26.334	17.668	17.046	11.203	12.325	853.487	853,48	379,36	0,380	
ARENAS	33.334	30.306	40.000	23.067	39.070	63.669	85.720	95.897	53.334	35.773	35.443	36.392	506.620	506,62	379,36	0,380	
CANILLAS DE AGUIFAR	2.285	54.954	50.187	64.777	37.354	63.179	103.430	137.917	90.521	57.342	57.397	63.144	689.202	689,20	379,36	0,380	
CANILLAS DE ALBARRAS	314	22.222	20.293	26.194	15.105	25.548	41.625	59.000	38.104	35.379	23.428	25.534	397.080	397,08	379,36	0,380	
SALINES	205	4.330	4.502	5.811	3.351	5.668	9.279	12.306	8.127	7.036	5.781	5.665	77.639	77,63	379,36	0,380	
BEDELLA	697	16.522	15.098	19.476	10.227	18.395	31.097	41.496	27.236	26.260	17.417	12.257	18.995	281.031	281,03	379,36	0,380
HUELA	7.997	47.371	43.770	56.439	32.587	55.105	30.210	120.357	78.161	50.528	50.082	55.179	787.253	787,25	379,36	0,380	
Teléfono :	7.597	191.220	174.620	225.480	129.880	179.840	103.900	480.180	314.360	201.520	193.720	279.720	3.027.040	3.027,04	379,36	0,380	
Circuito nº 4																	
MAHARRADAYA	529	9.935	9.468	5.395	1.425	12.519	6.177	14.167	3.296	6.430	4.294	5.766	67.287	67,28	381,07	0,381	
TOTALÁN	727	14.085	13.445	7.847	2.024	17.778	8.914	9.660	20.116	4.579	12.053	8.168	214.652	214,65	381,07	0,381	
BORQUE (EL)	1.015	19.371	18.576	10.532	2.768	24.484	12.277	13.311	27.707	6.444	16.607	8.338	171.677	171,67	381,07	0,381	
BENAHUE	89	3.371	3.375	1.320	509	4.463	2.298	2.430	5.060	1.175	3.027	1.530	31.287	31,28	381,07	0,381	
CITARR	677	12.806	12.241	6.362	1.043	16.166	8.116	8.103	10.310	4.260	10.379	5.512	107.407	107,40	381,07	0,381	
ALMACHAR	1.889	36.242	34.642	19.704	5.715	45.807	22.965	24.942	51.838	12.056	31.071	15.809	321.885	321,88	381,07	0,381	
MOULINEJO	7.774	24.314	23.241	13.219	3.499	30.771	15.409	16.733	34.777	8.086	20.845	14.154	235.477	235,47	381,07	0,381	
EL VALDES	527	9.361	9.541	5.427	1.426	12.646	6.206	6.868	14.277	3.202	8.551	4.297	69.457	69,45	381,07	0,381	
BENAMARGOSA	1.637	30.726	29.371	16.705	4.422	39.826	19.473	21.146	43.949	10.222	26.343	13.221	222.308	222,31	381,07	0,381	
Teléfono :	6.427	180.340	163.840	87.930	23.160	203.420	102.000	110.760	220.200	53.540	107.980	63.280	1.428.710	1.428,71	381,07	0,381	
Circuito nº 5																	
BENAMARGOSA	1.637	47.227	39.300	46.386	43.852	47.394	46.370	49.948	52.848	50.403	48.121	27.178	588.678	588,67	371,97	0,372	
ROSOFORD	3.053	19.750	14.670	18.133	14.718	19.897	17.936	14.932	19.370	100.412	16.705	11.630	1.076.684	1.076,68	371,97	0,372	
HUELA	7.897	58.474	48.645	57.421	41.711	58.583	57.373	61.830	76.471	85.420	62.380	53.583	707.487	707,48	371,97	0,372	
Teléfono :	6.667	195.460	162.620	191.940	106.280	195.780	191.640	208.680	255.620	218.680	208.560	193.120	2.344.840	2.344,84	371,97	0,372	
Circuito nº 6																	
ALCALÓN	2.689	56.020	56.020	39.680	102.400	111.800	103.260	114.840	141.620	107.200	105.000	113.300	1.263.300	1.263,30	471,97	0,471	
ALGARROBO	6.269	279.470	265.000	291.900	302.660	287.600	290.100	440.240	320.300	280.000	277.320	260.060	3.533.000	3.533,00	505,97	0,506	
COMARES	1.675	48.000	48.000	51.220	44.361	58.590	55.100	66.300	55.000	54.000	44.160	40.540	621.791	621,79	381,07	0,381	
PERPIÑA	3.177	103.600	30.100	105.340	120.060	121.820	126.400	161.300	173.400	128.260	120.060	117.160	1.448.280	1.448,28	468,77	0,467	
HERNIA	27.857	322.300	884.740	976.220	1.100.000	1.162.660	1.207.340	1.475.040	1.637.340	1.250.780	1.140.620	965.940	13.543.580	13.543,58	617,87	0,617	
RECINTO DE LA VENTANA	29.822	1.508.160	1.201.320	1.443.740	1.591.330	1.584.440	1.720.300	2.111.640	2.248.200	1.723.300	1.484.520	1.372.360	14.804.520	14.804,52	494,08	0,494	
TOPOR	17.207	754	593.280	686.280	752.320	765.160	837.160	1.142.580	1.353.460	827.140	711.200	620.580	8.091.644	8.091,64	510,89	0,510	
VELEZMALAGA	75.628	2.979.780	2.628.560	3.071.780	3.077.320	3.263.520	3.397.160	4.208.440	4.208.440	3.263.660	3.028.450	2.738.060	39.427.859	39.427,85	508,08	0,508	
Teléfono :	89.642	5.096.584	5.897.120	6.665.680	7.091.910	7.362.420	7.743.660	10.283.960	10.283.960	7.812.840	6.910.960	6.311.560	87.186.450	87.186,45	516,67	0,516	
Total Anual - Circuito AXARQUÍA : 89.642,85																	

RECOGIDA R.S.U. - CONSORCIO PROVINCIAL DE RESIDUOS SÓLIDOS DE MÁLAGA ---- AÑO 2.011

MUNICIPIO	Hab. 2010	MESES												Total Kg. Municipio / Anual	Total Tn. Municipio / Anual	MEDIA ANUAL	
																Kg/habitado	Tn/habitado
		Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre				
Guadalhorce																	
Circuito nº 1																	
CASABUELA	2.723	45.969	90.815	5.424.840	76.825	112.612	84.623	128.794	113.697	118.899	100.363	100.363	81.683	6.471.498	6.471.500	2.376,61	2,377
GUARO	2.895	38.574	76.297	4.552.244	64.468	94.498	71.011	101.364	95.499	99.766	84.279	84.279	68.544	5.420.545	5.420.545	2.376,61	2,377
MOGA	2.470	41.697	82.377	4.928.886	69.687	102.149	76.780	109.571	103.134	107.844	91.638	91.638	74.093	5.670.215	5.670.215	2.376,61	2,377
Totales :	7.679	126.240	249.490	14.897.990	219.960	369.260	232.364	331.730	312.240	326.500	275.620	275.620	224.326	17.772.254	17.772.254	2.376,61	2,377
Circuito nº 2																	
ALCORNIA	2.244	78.344	65.000	3.684.583	93.789	79.262	87.030	88.621	98.896	83.955	75.982	68.625	65.321	4.481.607	4.481.611	1.987,11	1,987
BURGO EL	1.987	68.162	57.382	3.182.121	82.797	69.972	76.830	71.172	87.383	74.116	67.877	68.582	57.665	3.956.268	3.956.271	1.987,11	1,987
YUJUELA	3.265	111.895	92.837	5.148.257	133.954	113.286	134.361	115.147	141.391	119.949	108.521	98.013	93.294	6.400.728	6.400.730	1.987,11	1,987
Totales :	7.479	259.400	215.220	11.934.960	310.540	262.440	268.940	266.940	327.760	277.989	251.500	227.220	216.286	14.838.505	14.838.507	1.987,11	1,987
Circuito nº 3																	
CARTANA	22.173	775.793	499.182	1.076.894	599.296	798.827	799.292	883.469	887.747	771.537	734.721	734.619	736.457	9.067.032	9.067.033	408,92	0,409
ALH DE LA TORRE	35.829	1.189.056	806.697	1.728.988	919.983	1.298.919	1.291.672	1.298.422	1.305.336	1.246.819	1.187.323	1.187.199	1.196.129	14.652.508	14.652.509	408,92	0,409
ALH EL SPANKE	23.675	785.635	532.996	1.148.988	607.860	662.939	853.656	667.896	862.464	823.801	784.491	784.382	786.345	9.681.233	9.681.235	408,92	0,409
ALORA	13.474	447.124	303.341	653.916	345.947	485.426	485.711	488.248	498.848	488.844	446.472	446.410	447.527	5.509.818	5.509.821	408,92	0,409
COÍN	22.000	771.847	495.982	1.069.154	595.624	793.675	794.137	798.288	802.537	766.661	729.982	729.881	731.707	9.008.558	9.008.561	408,92	0,409
POZUÑA	8.990	298.326	202.262	436.300	239.820	323.883	324.071	325.765	327.499	312.818	297.891	297.850	298.995	3.678.211	3.678.213	408,92	0,409
Totales :	126.174	4.186.980	2.840.960	6.123.440	3.235.540	4.545.670	4.548.320	4.572.090	4.596.430	4.360.380	4.188.880	4.188.380	4.198.760	51.585.250	51.585.253	408,92	0,409
TOLIX	2.346	82.125	78.548	56.620	58.745	65.840	91.320	104.120	154.880	92.880	92.546	77.060	77.040	1.000.844	1.000.846	430,49	0,430
Totales :	2.346	82.125	78.548	56.620	58.745	65.840	91.320	104.120	154.880	92.880	92.546	77.060	77.040	1.000.844	1.000.846	430,49	0,430
Total Anual - Circuito GUADALHORCE :														85.236.948	85.236.95		