



UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA

INGENIERÍA EN INFORMÁTICA

PROYECTO FIN DE CARRERA

LORO: Herramienta para la localización preliminar de una red de alineamientos horizontales en sistemas de transporte

Eusebio Angulo Sánchez-Herrera

Julio, 2007



UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA

INGENIERÍA EN INFORMÁTICA

PROYECTO FIN DE CARRERA

LORO: Herramienta para la localización preliminar de una red de alineamientos horizontales en sistemas de transporte

Autor: Eusebio Angulo Sánchez-Herrera

Director: Ricardo García Ródenas

Coordinador: José Jesús Castro Sánchez

Julio, 2007

TRIBUNAL:

Presidente:

Vocal:

Secretario:

FECHA DE DEFENSA:

CALIFICACIÓN:

PRESIDENTE

VOCAL

SECRETARIO

Fdo.:

Fdo.:

Fdo.:

*A los que no están.
A mi familia que siempre está
en los momentos difíciles.
A todos mis amigos.*

Agradecimientos

Desearía expresar mi agradecimiento a todas las personas e instituciones que han hecho posible la realización de este proyecto final de carrera.

En primer lugar agradezco al profesor D. Ricardo García, del Departamento de Matemáticas de la UCLM, por haberme introducido y guiado durante todo el año en la realización del presente Proyecto Fin de Carrera. Sin su apoyo e implicación, este PFC no habría podido llevarse a cabo.

Al proyecto de investigación: “Optimización matemática para la planificación robusta y la extensión de sistemas metropolitanos de transporte público con Ref. 2006-BIN-1133-002” y financiado por el Ministerio de Fomento, que ha hecho posible el desarrollo de este proyecto bajo el amparo de una beca de investigación.

Al profesor D. José Jesús Castro por su interés y ayuda en este proyecto.

Al Grupo de Investigación ORETO, por cederme un espacio en su laboratorio para desarrollar mi proyecto y brindarme un ambiente estimulante de trabajo.

A mis padres Eusebio y Encarnación por darme todo su apoyo y todos esos buenos consejos que me han llevado hasta este punto del camino. Sin ellos, no habría estudiado y ahora estaría más que arrepentido.

A mis dos hermanas Rosario y Cristina. Rosario por ser un ejemplo de esfuerzo y dedicación, y la pequeña Cristina por alegrarme cada día.

A todos mis amigos y amigas de la Escuela Superior de Informática, especialmente a Ramón, Ricardo y Vicente con los que he compartido tanto tiempo realizando prácticas.

A todos mis amigos de Pozuelo con los que no he podido pasar el tiempo que hubiera querido, debido a la carrera y el trabajo.

A los compañeros del grupo ORETO y el grupo MAT con los que he compartido

mucho tiempo durante este último año.

Por último no me podía olvidar de Aurora a la que entre carrera, trabajo y proyecto no le he podido dedicar el tiempo que se merece. Gracias por tener tanta paciencia y aguantarme en momentos difíciles.

Resumen

Determinar la localización de una red de infraestructuras lineales de transporte en función de las necesidades de transporte de las distintas poblaciones de una región, es un problema estratégico en el desarrollo regional y vertebración del territorio. En este trabajo se desarrolla una herramienta informática para localizar dichas infraestructuras.

Para realizar la herramienta, se desarrolla un modelo matemático y una metodología de resolución para localizar simultáneamente varias infraestructuras lineales de transporte. Dos aplicaciones relevantes son: i) la localización de una red de autovías, que será en la cual nos centraremos en este PFC y ii) la localización un sistema de metro en una ciudad.

El marco matemático empleado para su modelización es la llamada *programación matemática binivel*. Este tipo de problemas de optimización consta de dos problemas de optimización anidados. En el problema interior (o problema de nivel inferior) se modela la red de transporte y como los usuarios emplean el sistema diseñado y en el problema exterior (o problema de nivel superior) se establece un criterio de diseño para evaluar el beneficio social/privado del sistema diseñado.

Se implementa en el entorno de programación MATLAB un algoritmo de tipo búsqueda dispersa (scatter search) para la resolución del problema propuesto y se realizan varias pruebas numéricas con la herramienta elaborada, denominada LORO, sobre la región de Castilla-La Mancha. LORO está diseñada para consultorías de transporte o agencias públicas de planificación del transporte, con el objeto de auxiliar en el proceso de toma de decisiones sobre la localización preliminar de la red más apropiada en función de la distribución geográfica de la demanda de transporte y atendiendo a una restricción presupuestaria.

Índice general

Índice de figuras	XI
Índice de cuadros	XV
1. Introducción	1
1.1. Introducción	2
1.2. Aportaciones del Proyecto Fin de Carrera	3
2. Objetivos	7
2.1. Objetivos del proyecto	8
2.2. Restricciones	9
2.3. Recursos	11
3. Antecedentes y estado de la cuestión	13
3.1. Optimización de alineamientos	14
3.2. Métodos Evolutivos	16
3.2.1. Introducción al algoritmo de búsqueda dispersa (scatter search) .	17
4. Modelo de Localización de la red	23
4.1. Modelo de oferta: Sistema de transporte	25
4.2. Modelización de la demanda	27

4.3. Interacción oferta-demanda	29
4.3.1. Problema ii): Grafo de estrategia	50
4.4. Criterios de diseño	54
5. Algoritmo metaheurístico de resolución	57
5.1. Un algoritmo metaheurístico tipo Búsqueda Dispersa	58
6. Herramienta LORO	67
6.1. Análisis	68
6.1.1. Especificación de requisitos	68
6.1.2. Diagramas de flujo de datos	69
6.1.3. Diccionario de datos	75
6.2. Diseño	77
6.2.1. Diagramas de estructura	77
6.3. Codificación	84
6.3.1. Interfaz codificación	84
6.3.2. Modelo matemático	86
6.3.3. Generador de autovías	86
6.3.4. Algoritmo de Búsqueda Dispersa	87
7. Pruebas numéricas	89
7.1. Datos para las pruebas	90
7.2. Ajuste de parámetros	94
7.2.1. Parámetros del modelo	95
7.2.2. Interpolación cuadrática	95
7.2.3. Parámetros para el algoritmo del gradiente proyectado	95
7.3. Pruebas y análisis	106

<i>ÍNDICE GENERAL</i>	IX
8. Conclusiones y propuestas de mejora	109
8.1. Conclusiones	110
8.2. Propuestas de mejora	111
A. Manual de usuario	113
A.1. Requisitos de instalación	114
A.2. Manual de funcionamiento	114
B. Manual de código	119
C. CD	123
D. Publicaciones Relacionadas	125
Bibliografía	127

Índice de figuras

4.1. Ilustración del modelo de oferta	27
4.2. Problema i.1)	30
4.3. Incorporaciones óptimas a una infraestructura lineal	32
4.4. Soluciones al lote de pruebas para para el problema i.1)	36
4.5. Prueba 1 para las capitales de provincia de Castilla-La Mancha	40
4.6. Prueba 2 para las capitales de provincia de Castilla-La Mancha	40
4.7. Problema i.2)	41
4.8. Función objetivo a minimizar junto a una parábola interpolante	44
4.9. Caso 1.a en el método de optimización unidimensional	46
4.10. Caso 1.b en el método de optimización unidimensional	47
4.11. Soluciones al lote de pruebas para el problema i.2)	49
4.12. Grafo de estrategia para dos autovías	51
5.1. Espacio de factibilidad	58
5.2. Parámetros que determinan una infraestructura lineal	59
5.3. Ejemplo ilustrativo de la definición de proximidad entre dos segmentos	60
5.4. Recta que contiene al segmento	60
5.5. Distancia de los extremos a la recta	60

5.6. Soluciones generadas inicialmente para $N = 100$	63
5.7. Conjunto de referencia para $b = 10$	63
5.8. Autovías similares	64
5.9. Distancias euclídeas que se van a calcular	64
5.10. Nueva infraestructura lineal	64
6.1. Diagrama de contexto	69
6.2. Diagrama 0. Localización de la red de alineamientos	70
6.3. Diagrama 1. Procesar entrada	71
6.4. Diagrama 2. Generar autovías	72
6.5. Diagrama 3. Algoritmo de Búsqueda Dispersa	73
6.6. Diagrama 2.2. Evaluar Autovías	74
6.7. Diagrama de estructura	78
6.8. Procesar Entrada	79
6.9. Generar Autovías	80
6.10. Evaluar Autovías	81
6.11. Algoritmo de Búsqueda Dispersa	82
6.12. Crear Nuevas Soluciones	83
7.1. Gráfico de Coordenadas UTM	92
7.2. Grafo ejemplo a partir de la tabla	93
7.3. Grafo demanda simplificado	94
7.4. Autovía inicial para la prueba 1	97
7.5. Autovía optimizada para la prueba 1	98
7.6. Autovías iniciales para la prueba 2	99
7.7. Autovías optimizadas para la prueba 2	99

7.8. Autovías iniciales para la prueba 3	100
7.9. Autovías optimizadas para la prueba 3	101
7.10. Búsqueda Lineal con variaciones de Alfa	102
7.11. Comportamiento de Z	103
7.12. Mejor solución con $C = 300$	104
7.13. Mejor solución con $C = 600$	105
7.14. Gráfica con los resultados de Z en función de la restricción C	107
A.1. Ventana principal de la interfaz de usuario	115
A.2. Ventana de error al no cargar alguno de los ficheros	115
A.3. Ventana que muestra la solución	116
A.4. Menú desplegable de la solución	117
A.5. Ventana de información acerca de la herramienta	117
A.6. Ventana de confirmación para salir	118

Índice de cuadros

2.1. Puntos fuertes y débiles de los programas analizados	10
3.1. Algoritmo prototipo de Búsqueda Dispersa	20
4.1. Datos de las pruebas para el problema i.1)	35
4.2. Resultados de las pruebas para el problema i.1)	37
4.3. Caso 1 con capitales de provincia de Castilla-La Mancha	38
4.4. Caso 2 con capitales de provincia de Castilla-La Mancha	39
4.5. Datos de las pruebas para el problema i.2)	48
4.6. Algoritmo modificado de Dijkstra	53
5.1. El algoritmo BD	65
6.1. Diccionario de Datos	75
6.2. Continuación del Diccionario de Datos	76
6.3. Flujos de Datos	77
6.4. Estructura del fichero de entrada Municipios	84
6.5. Estructura del fichero de entrada Trayectos	85
7.1. Definición de los datos de los problemas.	90
7.2. Municipios para el Problema 1	91

7.3. Traslados para el Problema 1	91
7.4. Matriz Demanda Origen-Destino	93
7.5. Codificación del nombre de cada centroide	93
7.6. Entrada de datos para las pruebas al método de mejora	96
7.7. Tabla con los datos de la autovía para la primera prueba	96
7.8. Tabla con los datos de las autovías para la segunda prueba	98
7.9. Tabla con los datos de las autovías para la tercera prueba	100
7.10. Tabla con los datos para las pruebas del <i>Problema 1</i>	104
7.11. Tabla con los datos para las pruebas del <i>Problema 2</i>	106
7.12. Tabla para estudiar la mejora de Z en función del incremento de C . . .	107

Capítulo 1

Introducción

1.1. Introducción

1.2. Aportaciones del Proyecto Fin de Carrera

1.1. Introducción

El transporte es una de las actividades económicas y sociales más importantes en las sociedades actuales. El crecimiento económico ha originado un incremento sustancial de la demanda de transporte, tanto en los países desarrollados como en los que están en vías de desarrollo. Este aumento de la demanda ha conducido a que, en algunas regiones y para ciertos modos de transporte, la demanda supere a la oferta de servicios de transporte, originando la necesidad de expandir los sistemas actuales de transporte o crear nuevos sistemas como metro ligero, tranvía, etc.

Las acciones correspondientes al transporte son tan complejas que se ha vuelto indispensable utilizar modelos matemáticos y nuevas tecnologías para poder organizarlas y realizarlas, con el fin de encontrar buenas soluciones a las necesidades de movilidad de la sociedad.

En este trabajo se aborda la determinación de la localización de una red de alineamientos lineales como un problema de optimización en un espacio factible continuo (el plano) de soluciones. En la literatura científica el problema de localizar alineamientos de metro se ha planteado como problemas de localización en grafos. Este enfoque de localización continua presenta la ventaja frente a la formulación del problema como uno de localización en grafos, que permite abordar de una forma unificada, los dos problemas siguientes:

- Problema 1 . Localización de una red ferroviaria urbana, tipo metro o cercanías.
- Problema 2 . Localización preliminar de una red de autovías.

El problema de diseño de redes abordado en este trabajo se sitúa en un contexto de planificación estratégica. Éste consiste en localizar la infraestructura de una red de transporte (metro o autovía) de modo que se optimice un cierto criterio de diseño para una determinada matriz de demanda origen-destino fija y conocida, y satisfaciendo una restricción presupuestaria.

En este PFC se propone un modelo matemático para el anterior problema y un algoritmo metaheurístico tipo búsqueda dispersa para su resolución. Estos desarrollos teóricos conducen a una colección de algoritmos que se han implementado en el entorno de programación MATLAB, que permite ante una entrada de datos como son la demanda entre distintos pares origen-destino y el presupuesto disponible, determine una buena localización preliminar de autovías posible para un presupuesto dado y un cierto criterio de diseño.

Esta herramienta, denominada LORO, se ha testado sobre el problema de localizar una red de autovías para la comunidad de Castilla-La Mancha.

1.2. Aportaciones del Proyecto Fin de Carrera

Las infraestructuras de transporte son indispensables para el desarrollo de las actividades socioeconómicas y para el asentamiento de la población en el territorio. Las redes de transporte expresan la capacidad de un país o región para estructurar y modelar un espacio, configurando las posibilidades de acceso a los recursos del territorio. Desempeñan sin duda un papel estratégico en el crecimiento económico y en el desarrollo regional, influyendo sobre la capacidad de las distintas áreas del territorio. Por tanto, el impacto de las infraestructuras de comunicación sobre el desarrollo debe ser siempre objeto de especial atención en el marco del desarrollo de cualquier región.

Además en la mayoría de las ciudades del mundo occidental existen serios problemas debido al aumento del tráfico vehicular, con importantes consecuencias en términos de demoras, consumos, accidentes y contaminación tanto acústica como ambiental. Todos estos problemas, podrían paliarse con una red de metro eficiente para la demanda de la población urbana.

Si se consigue una red de autovías eficiente, se reduciría el número de accidentes

de tráfico y también se conseguiría un ahorro energético para los usuarios, al reducir el gasto de combustible y se reduciría el tiempo de viaje en los trayectos. Se obtiene un beneficio para el medio ambiente y otro económico para los usuarios.

Cuando afirmamos que se reducen los accidentes de tráfico, nos basamos en datos como los proporcionados en [12] por AEA (Automovilistas Europeos Asociados) del año 2006. Donde se afirma que la mayor parte de los puntos negros se han localizado en las carreteras convencionales (61,9%) y en las autovías (32,5%), siendo muy pocos significativos aquellos localizados en autopistas de peaje (0,4%) y en las autopistas libres (5,2%). Después de esto, parece más que importante establecer en una determinada región una red de autovías o autopistas eficiente, en función de la demanda de viajes entre las ciudades de dicha región.

Todos estos motivos avalan la necesidad de realizar esfuerzo de investigación en estas materias.

Las aportaciones de este PFC se pueden agrupar:

1. Modelización matemática de localización de alineamientos. En este PFC se propone un nuevo modelo que presenta como novedad frente a los modelos del Estado-del-Arte que aborda la red completa (varios alineamientos) del sistema de transporte, frente a la consideración de un único alineamiento tanto en el Problema 1 como en el Problema 2.
2. Técnicas computacionales. Los algoritmos desarrollados en la literatura científica no son totalmente satisfactorios. En este PFC se ha propuesto una técnica heurística de búsqueda dispersa orientada al problema. Se ha requerido de una estructura de datos eficiente y subrutinas altamente depuradas que permitan aprovechar las características del problema. Las pruebas numéricas parecen indicar que el método es aplicable a problemas de tamaño medio.
3. Desde el punto de vista de la práctica se ha desarrollado un producto software para el problema de localización de alineamientos. Según nuestro conocimiento no

existen programas comerciales para este propósito.

Capítulo 2

Objetivos

2.1. Objetivos del proyecto

2.2. Restricciones

2.3. Recursos

2.1. Objetivos del proyecto

El PFC que vamos a desarrollar, contará con los siguientes dos objetivos principales:

Objetivo 1: Herramienta para la localización preliminar robusta de una red de alineamientos horizontales en sistemas de transporte. Dicha herramienta estará orientada a su utilización en consultorías de transporte que facilitará a los planificadores una red preliminar de alineamientos horizontales para una red de autovías, sin ser necesario que los mismos tengan conocimiento alguno del entorno de programación utilizado. Esta herramienta, será una implementación en MATLAB en la que se integran y documentan todos los algoritmos.

Además, la herramienta contará con las siguientes restricciones funcionales:

- a) Implemente un modelo matemático que recoja los elementos notables en la localización preliminar de la red de alineamientos horizontales en un contexto de planificación estratégica.
- b) Implemente una metodología de resolución aplicable a problemas reales. Varios problemas más sencillos que el abordado en este PFC se ha demostrado que son *NP-duros*. Este hecho nos fuerza a recurrir a una metodología heurística de resolución.

Objetivo 2: Aplicación de dicha herramienta a casos reales. Se validará la herramienta con datos reales de Castilla La Mancha. Se realizarán dos pruebas numéricas con estos datos:

- a) Con las ciudades de Castilla La Mancha de más de 50.000 habitantes.
- b) Con las ciudades de Castilla La Mancha de más de 5.000 habitantes.

El objetivo de estas pruebas es analizar el comportamiento cualitativo y el coste computacional en problemas reales.

2.2. Restricciones

Factores dato

La herramienta a desarrollar se utilizaría en una consultoría de transporte. La Herramienta tiene que ser multiplataforma (Linux/Unix, Windows, Mac), así podrá utilizarse en cualquier consultoría independientemente del Sistema Operativo que tengan.

Factores estratégicos

Para la elección del lenguaje de programación y el entorno de programación utilizado, hemos tenido en cuenta que al tener que implementar un modelo matemático con numerosos cálculos numéricos, había que elegir una herramienta orientada al cálculo numérico, cuya principal virtud sea su velocidad. Por tanto, tras buscar información al respecto en la web, encontramos las siguientes alternativas principales: Octave, MATLAB, Scilab y Scipy.

A continuación mostramos una tabla que resume los puntos fuertes y débiles que cada uno de los programas, según el estudio comparativo de [26].

Programas	Puntos fuertes	Puntos débiles
MatLab	1.Gran número de funciones y ToolBox. 2.Amplia difusión, diseño apoyado en interfaces gráficas. 3.Gran cantidad de información sobre el programa.	1.Licencia
Octave	1.Gran cantidad de funciones y paquetes. 2.Alta compatibilidad con MatLab. 3.Licencia.	2.Rapidez
Scilab	1.Integración de todos los paquetes en un solo programa. 2.Modelado gráfico de sistemas dinámicos. 3.Ayuda del programa. 4.Licencia.	1.Falta de información en la red en castellano. 2.Mala impresión por los 'cuelgues' de las demos. 3.Rapidez.
Scipy	1.Lenguaje de programación. 2.Facilidad para integrar otros programas, lenguajes. 3.Licencia.	1.Información sobre el programa. 2.Proyecto en desarrollo.

Tabla 2.1: Puntos fuertes y débiles de los programas analizados

Al final nos decantamos por MATLAB fundamentalmente porque es el más completo y con mayor información, seguramente por su tradición y sus años en el mercado. MATLAB nos ofrece muchas más ventajas con las que poder codificar nuestro algoritmo de forma satisfactoria. Nos proporciona todas las características de un lenguaje de programación, manejo de estructuras de datos complejas, manejo de ficheros, manejo de volúmenes elevados de información, numerosos paquetes (toolbox) como el de optimización, etc.

Otra ventaja importante es que una aplicación de MATLAB se puede integrar con Java, .NET, C, C++ y Excel, como se puede ver en [27]. Pero para esto, se necesita el Compiler y el builder que corresponda según donde se quiera integrar (MATLAB Builder for Java, MATLAB® Builder for .NET, y MATLAB® Builder for Excel). Con el Compiler y el Builder, se generan ejecutables independientes y componentes C, C++, Java, o Excel. El gran inconveniente es que hay que pagar dos licencias para utilizar el Compiler y el Builder.

La gran desventaja de utilizar MATLAB es que necesita Licencia pero gracias a que se puede integrar con Java, C, C++, .NET o Excel, esto no es un problema para el cliente. El desarrollador es el que tiene que comprar la Licencia de MATLAB, la

del Compiler y el Builder, con esto ya podría ser utilizada por cualquier cliente que cuente con compilador Java por ejemplo. Por tanto, el coste de Licencias sería para el desarrollador, no supondría un coste adicional para los clientes.

Otra cuestión estratégica en la elección del programa MATLAB, es que permite la paralelización de los algoritmos, cuestión altamente importante para el problema que se trata.

2.3. Recursos

Este proyecto fin de carrera va a ser desarrollado contando con los siguientes recursos humanos, hardware y software.

Recursos humanos

1. Autor: Eusebio Angulo Sánchez-Herrera.
2. Director: Ricardo García Ródenas.
3. Coordinador: José Jesús Castro Sánchez.

Recursos hardware

1. Ordenador de sobremesa PC Dell Optiplex GX520 (Pentium 4 a 3.00 GHz) y 1GB de RAM.
2. Ordenador portátil Toshiba Centrino con procesador Intel Pentium M715 a 1.5 GHz 512 MB DDR RAM.

Recursos software

1. Sistema Operativo Windows XP.
2. MATLAB 7.1 para todo el desarrollo de la herramienta, incluida la interfaz de usuario. Para el desarrollo de la herramienta se ha utilizado [28], [29], [30], [31], [32], [33], [34] y [35].
3. Corel 12 para la edición de las gráficas incluidas en la documentación.
4. LaTeX para la generación de la documentación del proyecto, haciendo uso de las herramientas WinEdt, MiKTeX, GSView y Acrobat Reader 7.0. Para generar esta documentación se ha utilizado [36], [37], [38], [39], [40] y [41].
5. Otras aplicaciones como OpenOffice para elaborar la presentación, etc.
6. Derive 6 Trial Edition para facilitar el cálculo simbólico.
7. Además de las referencias señaladas, se ha utilizado más documentación disponible en Internet correspondiente a las distintas herramientas y lenguajes empleados.

Capítulo 3

Antecedentes y estado de la cuestión

3.1. Optimización de alineamientos

3.2. Métodos Evolutivos

3.2.1. Introducción al algoritmo de búsqueda dispersa (scatter search)

En este capítulo analizaremos los antecedentes y el Estado-del-Arte para el diseño de sistemas de transporte como redes de metro, autovías, autopistas, etc. Abordaremos el problema de diseño de redes de transporte en un contexto de planificación estratégica (véase la sección 1.1). Primero veremos los antecedentes en optimización de alineamientos y después describiremos los elementos claves del algoritmo de búsqueda dispersa.

3.1. Optimización de alineamientos

El diseño de sistemas de ferrocarriles urbanos como metro, tranvías, cercanías, etc. o el diseño de una red de autovías en una determinada región, constan de etapas que se desarrollan a lo largo de un período dilatado de tiempo. Se puede abordar bajo tres horizontes temporales, en el problema de localización de sistemas ferroviarios se distinguen:

Planificación estratégica. El objetivo de esta fase es localizar la infraestructura de vías y estaciones.

Planificación táctica. En esta fase se definen las líneas que soportará la anterior infraestructura, así como sus frecuencias.

Planificación operacional. En esta planificación se deciden los horarios y asignación de recursos. Este tipo de problemas están ligados a la explotación ferroviaria.

En el diseño de redes de autovías se identifican las siguientes fases:

Planificación estratégica. Se desea localizar en el plano (alineamientos horizontales) el sistema de carretera que satisfaga una demanda de viajes origen-destino dada.

Planificación táctica. Dados un principio y un final en una autovía se desea determinar el trazado (alineamiento vertical) determinando movimientos de tierras, localización de puentes, intersecciones, niveles, etc.

Planificación operacional. En esta fase se determina el trazado, señalización, etc de la vía, de modo que se cumpla la normativa vigente.

En este PFC se abordan los problemas 1 y 2 en un contexto de planificación estratégica.

Los principales aspectos que se han estudiado en la literatura son: a) alineamientos de sistema ferroviario en el plano y b) la determinación del número de estaciones y su localización. Un modelo para el problema a) ha sido desarrollado en [11] y [6]. Estos autores formulan el problema como una localización de caminos en un grafo. [8] estudiaron el problema b), teniendo en cuenta los viajes de tipo *park 'n ride*. [3] abordan el problema de diseño de la red maximizando la matriz cubierta por la red (líneas y estaciones) ponderando con factores económicos, y restringiendo la distancia entre paradas y la máxima longitud admisible del alineamiento.

La mayor crítica a estos modelos proviene del hecho de que no incorporan el comportamiento de los usuarios en la red diseñada. Implícitamente se asume que una buena red promocionará su uso. Un primer modelo que subsana estas carencias es el dado en [1], donde la elección del nodo es tomada en cuenta mediante el mecanismo de asignación denominado *todo o nada* bajo el cual los usuarios emplean la red que les proporcione los menores costes de transporte. Recientemente [9], [7] introducen un modelo de demanda logit para modelar la elección del sistema de transporte.

La mayoría de las referencias abordan el problema de una única línea, ya que incluso bajo esta simplificación el modelo resultante es demasiado complejo. Un ejemplo de esta dificultad es obtenida por [5], quienes muestran que el problema de localizar las estaciones sobre una línea dada es un problema *NP-duro*. Se han planteado algoritmos heurísticos para resolver estos modelos de localización de alineamientos. En otros casos, como en [4], se aplica un algoritmo de búsqueda tabú. Los algoritmos de [2] y [9] realizan búsquedas locales para determinar estaciones compatibles con el conjunto de restricciones. [10] abordan el problema de localizar la infraestructura de la red ferroviaria como

un problema de localización en grafos. Este modelo recoge las elecciones de los usuarios de modo de transporte y de ruta para satisfacer su viaje.

En el caso de optimizar alineamientos de autovías se ha estudiado el problema de un único alineamiento y consiste en encontrar el mejor trazado entre un par de puntos. Dados los dos extremos de la autovía, las características del área estudiada, hay que encontrar el mejor alineamiento entre las alternativas respecto de una función objetivo dada, considerando las estructuras que sean necesarias y satisfaciendo los requisitos de diseño y operacionales [16],[17],[18],[19].

Para este problema es necesario utilizar algoritmos de búsqueda rápidos y eficientes porque el número de alternativas a evaluar es potencialmente infinito.

Para este problema de optimización se han utilizado algoritmos basados en programación dinámica [20], búsqueda numérica [22], optimización en dos etapas [21], Algoritmos Genéticos [23], etc.

3.2. Métodos Evolutivos

La existencia de una gran cantidad y variedad de problemas difíciles, que aparecen y necesitan ser resueltos de forma eficiente, ha impulsado el desarrollo de procedimientos capaces de encontrar buenas soluciones. En estos métodos la rapidez es tan importante como la calidad de la solución obtenida y se denominan heurísticos o aproximados.

Los procedimientos metaheurísticos constituyen la nueva generación de métodos aproximados, y dan unas reglas o estrategias que guían la construcción o el diseño del algoritmo heurístico concreto que resolverá el problema dado.

Una de las familias de métodos que podemos encontrar dentro de los procedimientos metaheurísticos es la de los llamados algoritmos evolutivos. Tres de estos métodos que

ya han demostrado su eficacia son los algoritmos genéticos, la búsqueda dispersa y el re-encadenamiento de trayectorias. En este apartado nos centraremos en explicar brevemente la búsqueda dispersa, ya que es la que utilizamos en el proyecto.

Para la revisión de este método, se han seguido los trabajos de Laguna y Martí [[13],[14]].

3.2.1. Introducción al algoritmo de búsqueda dispersa (*scatter search*)

La Búsqueda Dispersa (BD) es un método evolutivo cuyos conceptos y principios fundamentales fueron propuestos a comienzo de la década de los setenta, basados en las estrategias para combinar reglas de decisión, especialmente en problemas de secuenciación, así como en la combinación de restricciones. La BD se basa en el principio de que la información sobre la calidad o el atractivo de un conjunto de reglas, restricciones o soluciones puede ser utilizado mediante la combinación de éstas. En concreto, dadas dos soluciones, se puede obtener una nueva mediante su combinación de modo que mejore a las que la originaron.

La primera descripción del método fue publicada en 1977 por Fred Glover donde establece los principios de la BD. En este primer artículo se determina que la BD realiza una exploración sistemática sobre una serie de “buenas” soluciones llamadas conjunto de referencia. Los siguientes comentarios resumen los principales aspectos de este trabajo:

- El método se centra en combinar dos o más soluciones del conjunto de referencia.
- Al combinar dos soluciones se deben de seleccionar pesos apropiados y no tomar valores al azar.
- Se deben de realizar combinaciones “convexas” y “no convexas” de las soluciones.
- La distribución de los puntos en el espacio de soluciones se considera importante

y deben de tomarse dispersos.

En 1998, Glover publica una versión más específica, la cual tuvo un gran impacto en lo que a la difusión del método se refiere. Numerosos investigadores comenzaron a aplicar la BD a la resolución de problemas de optimización obteniendo resultados de gran calidad.

El método de BD se basa en combinar exhaustivamente las soluciones que aparecen en el llamado conjunto de referencia. En este conjunto se tienen las soluciones buenas que se han ido encontrando. Es importante destacar que el significado de buena no se restringe a la calidad de la solución (valor de la función objetivo), sino que también se considera la diversidad que esta aporta al conjunto de soluciones de referencia. La BD consta básicamente de los siguientes elementos:

1. **Un generador de soluciones diversas.** El método se basa en generar un conjunto P de soluciones diversas, del que extraeremos un subconjunto pequeño con el que realizar las combinaciones y que denominamos conjunto de referencia.
2. **Un conjunto de referencia.** Se extrae del conjunto de soluciones diversas un subconjunto según el criterio de contener soluciones de calidad y diferentes entre sí (Calidad y Diversidad).
 - a) **Creación.** Iniciamos el conjunto de referencia con las $b/2$ mejores soluciones de P . Las $b/2$ restantes se extraen de P por el criterio de máxima distancia con las ya incluidas en el conjunto de referencia. Para ello debemos de definir previamente una función de distancia que generalmente depende del contexto del problema que se está resolviendo.
 - b) **Actualización.** Las soluciones fruto de las combinaciones pueden entrar en el conjunto de referencia y reemplazar a alguna de las ya incluidas si las mejoran. Así, el conjunto de referencia mantiene un tamaño b constante pero va mejorando a lo largo de la búsqueda. Si el método no logra mejorar a la solución, se considera que la salida es la propia solución considerada.

3. **Un método de combinación.** Se consideran subconjuntos de dos o más elementos del conjunto de referencia y se combinan mediante una rutina diseñada a tal efecto. La solución o soluciones que se obtienen de esta combinación pueden ser inmediatamente introducidas en el conjunto de referencia (actualización dinámica) o almacenadas temporalmente en una lista hasta terminar de realizar todas las combinaciones y después ver qué soluciones entran en éste (actualización estática).
4. **Un método de mejora.** Normalmente se trata de un método de búsqueda local para mejorar las soluciones, tanto del conjunto de referencia como las combinadas antes de estudiar su inclusión en el conjunto de referencia.

A continuación, la siguiente tabla 3.2.1, muestra cómo actúan los elementos descritos en un esquema básico del algoritmo. El algoritmo hace referencia a los subconjuntos de R ya que podemos combinar parejas, tríos o cualquier número de soluciones.

El algoritmo se detiene cuando se examinan todos los subconjuntos del conjunto de referencia (la variable *NuevaSolucion* está a *FALSE*).

Laguna y Martí en [13] realizan una revisión exhaustiva del método y sus propuestas más destacables son:

1. **Aplicar la rutina de mejora de forma selectiva.** Las pruebas indican que aplicar el método de mejora a todas las soluciones generadas y combinadas no garantiza obtener mejores resultados finales. Luego para ahorrar coste de computación, sería conveniente establecer umbrales de calidad para no aplicar la mejora a soluciones que difícilmente van a proporcionar la mejor solución. Por otro lado, aplicar el método de mejora a todas las soluciones acelera la convergencia, lo cual puede ser deseable si se dispone de poco tiempo de computación, pero debemos evitarlo si queremos obtener soluciones de gran calidad.

Esta consideración es tenida en cuenta en este PFC, donde el proceso de mejora solo se aplica si la solución combinada es incluida en R .

-
1. Comenzar con $P = \{\emptyset\}$. Utilizar el **método de generación** para construir una solución y el **método de mejora** para tratar de mejorarla; sea x la solución obtenida. Si $x \notin P$ entonces añadir x a P , en otro caso rechazar x . Repetir esta etapa hasta que P tenga un tamaño prefijado.
 2. Construir el **conjunto de referencia** $R = \{x^1, \dots, x^b\}$ con las $b/2$ mejores soluciones de P y las $b/2$ soluciones de P más diversas a las ya incluidas.
 3. Evaluar las soluciones en R y ordenarlas de mejor a peor respecto a la función objetivo.
 4. Hacer *NuevaSolución=TRUE*

Mientras (*NuevaSolucion*)

5. *NuevaSolución=FALSE*
6. Generar los subconjuntos de R en los que haya al menos una nueva solución.

Mientras (*Queden subconjuntos sin examinar*)

7. Seleccionar un subconjunto y etiquetarlo como examinado.
8. Aplicar el **método de combinación** a las soluciones del subconjunto.
9. Aplicar el **método de mejora** a cada solución obtenida por combinación. Sea x la solución mejorada:

Si ($f(x) < f(x^b)$) **y** x **no está en** R

Hacer $x^b = x$ y reordenar R

Hacer *NuevaSolucion=TRUE*

EndSi

EndMientras

EndMientras

Tabla 3.1: Algoritmo prototipo de Búsqueda Dispersa

2. **Equilibrio entre la intensificación y la diversificación.** Hay que estudiar el porcentaje de tiempo que el método está generando soluciones y el tiempo que esta combinando.
3. **Comparar actualizaciones del conjunto de referencia estática y dinámica.** Podemos aplicar dos estrategias al combinar las soluciones, introducirlas en el conjunto nada más generarlas o anotarlas en una “pila” y cuando terminemos de realizar todas las combinaciones, realizar la actualización. La primera estrategia es dinámica y las soluciones buenas están rápidamente en el conjunto de referencia, pero dado que éste es de tamaño constante, esto implica que hay soluciones que pueden salir sin llegar a haber sido utilizadas para realizar combinaciones.

Un estudio en profundidad del método, se realiza en el libro [13] o en el artículo [14].

Capítulo 4

Modelo de Localización de la red

- 4.1. Modelo de oferta: Sistema de transporte
 - 4.2. Modelización de la demanda
 - 4.3. Interacción oferta-demanda
 - 4.3.1. Problema ii): Grafo de estrategia
 - 4.4. Criterios de diseño
-

En este PFC nos centramos en los siguientes problemas:

Problema 1 Localización de una red ferroviaria urbana, tipo metro o cercanías.

Problema 2 Localización preliminar de una red de autovías.

Los modelos matemáticos de optimización son ampliamente utilizados en diversas ramas de la ingeniería para resolver problemas que por su naturaleza son indeterminados, es decir presentan más de una solución posible. La determinación de cual de las múltiples opciones utilizar se hace con el auxilio de una función objetivo.

Dentro de los modelos matemáticos aplicables a los dos anteriores problemas, nos centraremos en la programación matemática binivel que consta de dos modelos de optimización anidados: los denominados *problema del nivel inferior* y *problema del nivel superior*.

Las variables del nivel superior parametrizan la función objetivo y/o las restricciones del nivel inferior. Este es el marco adecuado en este tipo de problemas de diseño de redes de transporte. En el nivel superior se decide el tipo de infraestructura a localizar, atendiendo a restricciones presupuestarias y tecnológicas, mientras que en el nivel inferior se modela el comportamiento de los usuarios en el sistema de transporte diseñado (elección de ruta y elección del sistema de transporte).

Esquemáticamente el modelo propuesto obedece a la siguiente estructura:

1. **Nivel superior:** Criterio de diseño de redes.

Variables: localización de las infraestructuras lineales.

Restricciones: presupuestarias.

2. **Nivel inferior:** modelo de equilibrio entre oferta/demanda.

Variables: flujos, desagregación modal por sistemas de transporte, tiempos de transportes.

Restricciones: enrutamiento, localización, modelo de demanda logit.

El modelo de optimización resultante es un problema no lineal entero y por tanto difícil de resolver de modo exacto.

4.1. Modelo de oferta: Sistema de transporte

El modelo considera un sistema compuesto por tres redes de transporte:

- (a) **Red de transporte actual:** En el Problema 1 la red (a) corresponde a la red de tráfico, en que los usuarios realizan su viaje a unos costes (generalizados) de transporte dados. En el Problema 2 la red (a) corresponde al sistema actual de carreteras.
- (b) **Nueva red de transporte que se desea localizar:** En el Problema 1 ésta red representa el conjunto de alineamientos de la red de metro, en el Problema 2 corresponde al conjunto de nuevas autovías.
- (c) **Red de acceso a la nueva infraestructura.** En el Problema 1 correspondería a la red pedestre, en la que los usuarios acceden a las paradas de metro mientras que en el Problema 2 consistiría en una red de carreteras secundarias. Además se supone que mediante esta red se puede acceder a cualquier punto de la red (b).

La modelización matemática del conjunto de infraestructuras lineales que definen la nueva red de transporte (b) se realiza mediante un conjunto de segmentos. Cada uno de estos segmentos queda determinado por sus extremos. El modelo busca localizar estos extremos y por tanto se tiene que las variables del nivel superior son las coordenadas

$$\begin{aligned} (x_i^+, y_i^+), & \quad i = 1, \dots, n \\ (x_i^-, y_i^-), & \quad i = 1, \dots, n \end{aligned}$$

de los extremos de los n segmentos. El superíndice $+$ y $-$ se emplearán para indicar el sentido de desplazamiento empleado en la infraestructura lineal.

El modelo considera que se desean localizar a lo sumo n infraestructuras lineales (segmentos). Cumpliendo una restricción presupuestaria que restringe la longitud total de la infraestructura localizada. Formalmente,

$$\sum_{i=1}^n \sqrt{(x_i^+ - x_i^-)^2 + (y_i^+ - y_i^-)^2} \leq C \quad (4.1)$$

donde C es el número de kilómetros que se podrían construir con el presupuesto disponible.

El modelo considera que los usuarios viajan a una velocidad constante v en la red de transporte (b) y a una velocidad $w < v$ en la red de acceso (c). El tiempo de transporte en la red (b) es la suma del tiempo de acceso más el tiempo de viaje en los segmentos (infraestructuras lineales). Se emplea la distancia euclídea en el cálculo de los tiempos de transporte. El tiempo total de transporte en la red se denota por u^b .

La red (a) proporciona unos costes de transportes u^a para cada tipo de viaje existente en el sistema, calculados a un nivel de servicio de la red. Por ejemplo, si se considera el Problema 1, la red (a) es la red de tráfico la cual soporta un cierto nivel de congestión, que será tenido en cuenta en el cálculo de los costes de transporte u^a .

Los costes generalizados u^a, u^b dependen del tiempo andando, valor del combustible, tickets, tiempos en vehículos, etc. todos estos conceptos deben ser homogeneizados para poder ser comparados. Se supone que el valor u^a está medido en una escala comparable con u^b .

La siguiente figura 4.1 ilustra el modelo matemático de oferta. La red (b) está formada por dos autovías modeladas mediante dos segmentos. Los usuarios pueden acceder/abandonar estas infraestructuras en cualquier punto. Si un usuario desea ir desde el punto (a', b') al (c, d) empleando la red (b) recorrerá una distancia d_1 sobre la primera autovía a una velocidad v , la abandonará en el nodo (x', y') y viajará a una velocidad w durante d_2 kilómetros hasta el nodo de acceso (x, y) de la segunda autovía,

completará su viaje recorriendo una distancia d_3 de la segunda autovía a una velocidad v . El tiempo total de viaje en la red (b) se calcula por $u^b = \frac{d_1+d_3}{v} + \frac{d_2}{w}$.

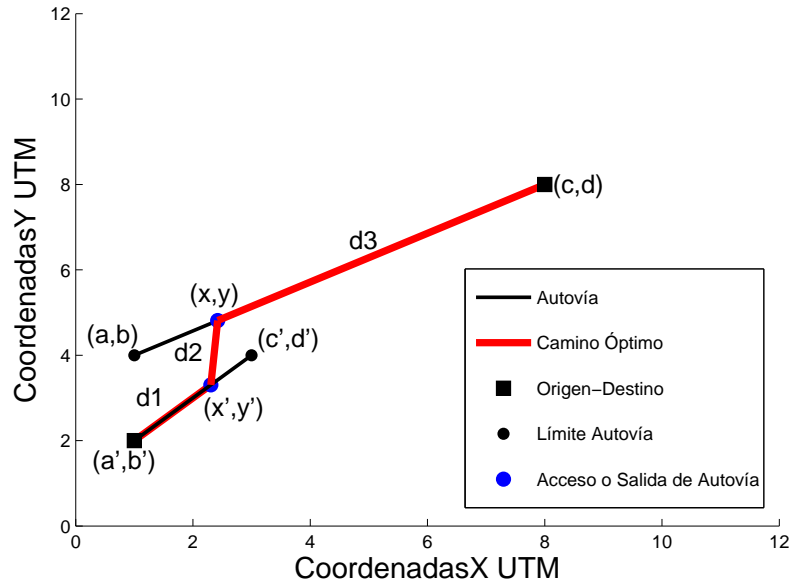


Figura 4.1: Ilustración del modelo de oferta

4.2. Modelización de la demanda

El modelo de demanda representa las elecciones que realizan los usuarios en el sistema de transporte. El modelo considera que existen unos lugares en el plano, los denominados *centroides* desde donde los usuarios comienzan (genera la demanda) o finalizan los viajes (atrae la demanda). Se considera que las coordenadas de estos puntos son conocidas:

$$(x_c, y_c), \quad c \in \mathcal{O}$$

La distancia entre un centroide y un punto arbitrario (x, y) de una infraestructura lineal se calcula empleando la distancia euclídea

$$d = \sqrt{(x - x_c)^2 + (y - y_c)^2}$$

Se consideran un conjunto de pares de demanda $\omega = (o, d)$ entre centroides $o, d \in \mathcal{O}$. El conjunto de todos estos pares se denota por W y se denota por g_ω el número de usuarios que desean realizar el viaje tipo $\omega \in W$. Los datos $\{g_\omega\}_{\omega \in W}$ recibe el nombre de matriz de demanda origen-destino y ésta sintetiza el patrón de movilidad de la demanda potencial del sistema de transporte.

Elección de ruta. El modelo asume el denominado *primer principio de Wardrop* para modelar como los usuarios eligen su ruta para satisfacer su viaje. Este principio afirma que en la situación de equilibrio ningún usuario puede reducir su tiempo de viaje cambiando unilateralmente de ruta. Se ha considerado que los costes de transporte en las redes (b) y (c) son independientes del flujo que soportan (esto es, que se viaja siempre a la misma velocidad w y v respectivamente) y por tanto los usuarios emplearán la ruta de mínimo coste para satisfacer su viaje. Para profundizar en esta cuestión se considera el ejemplo con dos autopistas mostrado en la figura 4.1. Se desea viajar desde el punto (a', b') de una autopista al punto (c, d) de la otra autopista. Los usuarios emplearán la ruta de mínimo coste y por tanto elegirán los nodos de transferencia (salida e incorporación) (x', y') y (x, y) de modo que se minimice el tiempo total de realizar el viaje, esto es, de modo que se minimice $\frac{d_1+d_3}{v} + \frac{d_2}{w}$.

Elección del sistema de transporte. Conocidos los costes de transportes en la nueva red (b) u_ω^b y en la red existente (a) u_ω^a los usuarios del par $w = (o, d)$, $\omega \in W$ deben decidir que sistema de transporte van a emplear. El modelo asume que la proporción de viajes en cada red de transporte $m \in \{a, b\}$ depende exclusivamente de los costes generalizados/tiempos de viajes. Esta relación es descrita por el siguiente modelo logit:

$$p_w^m(u_\omega^a, u_\omega^b) = \frac{\exp\{-(\alpha^m + \beta u_\omega^m)\}}{\sum_{m' \in \{a, b\}} \exp\{-(\alpha^{m'} + \beta u_\omega^{m'})\}}, \quad m \in \{a, b\} \quad (4.2)$$

La distribución de la demanda por redes de transportes viene dada por la expresión:

$$g_{\omega}^m = p_w^m(u_{\omega}^a, u_{\omega}^b)g_{\omega}, \quad m \in \{\mathbf{a}, \mathbf{b}\} \quad (4.3)$$

donde g_{ω}^m es el número de usuarios de la demanda ω que satisfacen su viaje empleando la red $m \in \{\mathbf{a}, \mathbf{b}\}$. El modelo logit permite tener en cuenta preferencias de los usuarios por un sistema de transporte, si un usuario tiene acceso al vehículo privado, etc. Estos factores son muy importantes para el Problema 1 y son tenidos en cuenta con una adecuada parametrización del modelo logit.

Un caso límite del anterior modelo es obtenido con los valores $\alpha^m = 0$ y $\beta \rightarrow +\infty$, obteniéndose:

$$p_w^b(u_{\omega}^a, u_{\omega}^b) = \begin{cases} 1 & \text{si } u_{\omega}^a - u_{\omega}^b > 0 \\ 0 & \text{si } u_{\omega}^a - u_{\omega}^b < 0 \end{cases} \quad (4.4)$$

Esta parametrización del modelo logit haría que todos los usuarios emplearía la red de mínimo coste generalizado. Este caso es adecuado para el Problema 2 .

4.3. Interacción oferta-demanda

En esta sección se desarrollan los mecanismos para calcular la ruta empleada por un usuario en el nuevo sistema de infraestructuras lineales (red **b**). Un usuario para realizar su viaje a través de la red (**b**) debe de decidir:

- i) Las incorporaciones/salidas de cada infraestructura lineal.
- ii) El conjunto de infraestructuras lineales a emplear para satisfacer su viaje.

El problema i) consta de dos subproblemas:

- i.1) Punto de acceso/salida desde un centroide a una infraestructura lineal.

i.2) Puntos de salida/entrada entre dos infraestructuras lineales.

A continuación describiremos los mecanismos para resolver cada uno de estos problemas.

Problema i.1) Punto de acceso/salida desde un centroide a una infraestructura lineal

En esta subsección se aborda el problema de calcular el punto de incorporación a la autovía desde un centroide, de forma que se minimice el tiempo de viaje desde él hasta cada uno de los extremos de la autovía. Para ilustrar este problema considérese la figura 4.2.

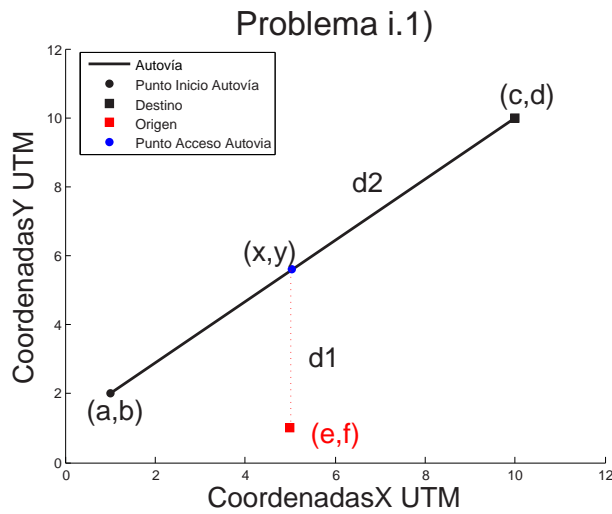


Figura 4.2: Problema i.1)

El segmento de extremos (a, b) y (c, d) modela la autovía. El punto (e, f) representa la localización del centroide (población generadora/atractora de demanda de viajes). Se asume que los usuarios pueden acceder a cualquier punto de la autovía a una velocidad w y que la velocidad de viaje en la autovía es v siendo $v > w$. Hay que calcular las coordenadas de incorporación (x, y) para los casos de ir hasta (a, b) o ir hasta (c, d) .

Una vez expuesto el problema, los parámetros de entrada son:

w : velocidad en el tramo de acceso a la autovía.

v : velocidad en el tramo de autovía.

(a, b) : punto inicial del segmento de autovía.

(c, d) : punto final del segmento de autovía.

(e, f) : localización del centroide desde el cual se quiere acceder a la autovía.

d_1 : distancia euclídea desde el centroide hasta el punto de acceso a la autovía.

d_2 : distancia euclídea desde el punto de acceso a la autovía hasta el punto final del segmento que modela la autovía, que en este caso sería el destino.

Un usuario que decida acceder a una infraestructura lineal tiene dos posibles incorporaciones óptimas en función del sentido que desea tomar. Se denota con los superíndices $+$ y $-$ cada uno de los sentidos posibles. Para ilustrar esta cuestión considérese la figura 4.3.

Si la velocidad de viajar en la autovía fuese $v = +\infty$, los usuarios se incorporarían en el punto de la autovía más cercano al centroide (en el dibujo esta trayectoria está representada por un segmento de puntos). Para valores finitos de v , los usuarios se incorporan en dos puntos equidistantes del punto anterior, dependiendo del sentido de la autovía que va a utilizar. Si los extremos de la autovía estuvieran situados en los extremos definidos por los puntos (x^+, y^+) , (x^-, y^-) , los usuarios accederían directamente sin hacer uso de la misma.

Para determinar el punto de incorporación a la autovía se debe resolver el siguiente problema de optimización. El tiempo a minimizar será $t_1 + t_2$ (tiempo en recorrer el tramo de acceso, más el tiempo en recorrer el tramo de autovía). Calculando las distancias d_1 y d_2 con la norma euclídea, la función a minimizar es:

$$f(x, y) = t_1 + t_2 = d_1/w + d_2/v = \sqrt{(x - e)^2 + (y - f)^2}/w + \sqrt{(c - x)^2 + (d - y)^2}/v \quad (4.5)$$

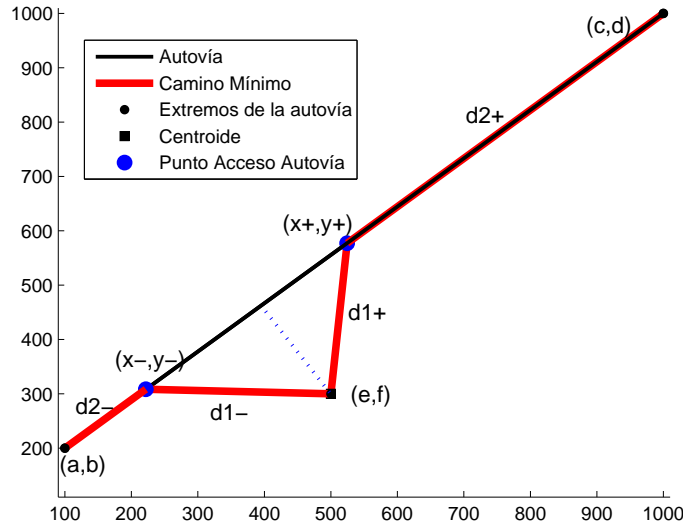


Figura 4.3: Incorporaciones óptimas a una infraestructura lineal

Introducimos una parametrización del segmento.

$$\begin{aligned} \begin{pmatrix} x(\lambda) \\ y(\lambda) \end{pmatrix} &= (1 - \lambda) \begin{pmatrix} a \\ b \end{pmatrix} + \lambda \begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} (1 - \lambda)a + \lambda c \\ (1 - \lambda)b + \lambda d \end{pmatrix} \\ &= \begin{pmatrix} \lambda(c - a) + a \\ \lambda(d - b) + b \end{pmatrix} \text{ con } \lambda \in [0, 1] \end{aligned} \quad (4.6)$$

Esta parametrización nos permite expresar la función objetivo en término de λ y el problema de optimización es:

$$\text{Minimizar } f(\lambda) = \frac{\sqrt{[\lambda(c-a)+a-e]^2 + [\lambda(d-b)+b-f]^2}}{w} + \frac{\sqrt{[c-(\lambda(c-a)+a)]^2 + [d-(\lambda(d-b)+b)]^2}}{v} \quad (4.7)$$

$$\text{sujeto a: } \lambda \in [0, 1]$$

Antes de resolverlo, vamos hacer una traslación geométrica que simplificará los cálculos. Se trata de trasladar las coordenadas de manera que $a = 0$ y $b = 0$, de modo que se modifican los valores de todas las coordenadas y tras obtener la solución del

problema trasladado, deshacemos los cambios y obtendremos la solución en función de la parametrización original.

$$\bar{a} = a - a = 0;$$

$$\bar{b} = b - b = 0;$$

$$\bar{c} = c - a;$$

$$\bar{d} = d - b;$$

$$\bar{e} = e - a;$$

$$\bar{f} = f - b;$$

Una vez realizada la traslación, a y b valen 0, con lo cual la nueva función nos quedaría de la siguiente forma simplificada:

$$f_t(\lambda) = \frac{\sqrt{(\lambda\bar{c} - \bar{e})^2 + (\lambda\bar{d} - \bar{f})^2}}{w} + \frac{\sqrt{(\bar{c} - \lambda\bar{c})^2 + (\bar{d} - \lambda\bar{d})^2}}{v} \quad (4.8)$$

Por tanto, vamos a minimizar $f_t(\lambda)$ con $\lambda \in [0, 1]$. La solución de un problema de optimización de una función continua en un intervalo, se alcanzará en uno de los siguientes tipos de candidatos:

- i) Puntos que anulan la primera derivada: $f'_t(\lambda) = 0$ con $\lambda \in [0, 1]$
- ii) Extremos del intervalo: $\lambda = 0$; , $\lambda = 1$;
- iii) Puntos donde la función no sea derivable. En este problema no existen este tipo de candidatos si el centroide no está en la autovía.

Posteriormente, con este parámetro λ calculamos las coordenadas (x, y) de acceso a la infraestructura lineal desde el centroide hasta el destino (que puede ser cualquiera de los extremos de la infraestructura) para que el tiempo sea lo menor posible.

Calculamos dicha derivada con el programa Derive 6, para poder estudiar el caso i , y el resultado obtenido es el siguiente:

$$f'_t(\lambda) = \frac{v(\bar{c}^2\lambda - \bar{c}\bar{e} + \bar{d}(\bar{d}\lambda - \bar{f}))}{vw\sqrt{\bar{c}^2\lambda^2 - 2\bar{c}\bar{e}\lambda + \bar{d}^2\lambda^2 - 2\bar{d}\bar{f}\lambda + \bar{e}^2 + \bar{f}^2}} - \frac{w\sqrt{\bar{c}^2 + \bar{d}^2}\sqrt{(\bar{c}^2\lambda^2 - 2\bar{c}\bar{e}\lambda + \bar{d}^2\lambda^2 - 2\bar{d}\bar{f}\lambda + \bar{e}^2 + \bar{f}^2)}}{vw\sqrt{\bar{c}^2\lambda^2 - 2\bar{c}\bar{e}\lambda + \bar{d}^2\lambda^2 - 2\bar{d}\bar{f}\lambda + \bar{e}^2 + \bar{f}^2}} \quad (4.9)$$

Despejamos λ , utilizando nuevamente el programa Derive 6 y obtenemos la siguiente solución de λ :

$$\lambda = \frac{\sqrt{(v^2 - w^2)(\bar{c}\bar{e} + \bar{d}\bar{f}) + w|\bar{c}\bar{f} - \bar{d}\bar{e}|}}{(\bar{c}^2 + \bar{d}^2)\sqrt{(v^2 - w^2)}} \quad (4.10)$$

Por último, faltaría deshacer la traslación realizada, quedando:

$$\lambda = \frac{\sqrt{(v^2 - w^2)((c - a)(e - a) + (d - b)(f - b)) + w|(c - a)(f - b) - (d - b)(e - a)|}}{[(c - a)^2 + (d - b)^2]\sqrt{(v^2 - w^2)}} \quad (4.11)$$

Necesitamos también la incorporación óptima para ir al otro extremo, para obtenerlo, por la simetría del problema, cambiamos en la ecuación 4.11, (c, d) por (a, b) y (a, b) por (c, d) , obteniendo:

$$\lambda = \frac{\sqrt{(v^2 - w^2)((a - c)(e - c) + (b - d)(f - d)) + w|(a - c)(f - d) - (b - d)(e - c)|}}{[(a - c)^2 + (b - d)^2]\sqrt{(v^2 - w^2)}} \quad (4.12)$$

El valor de λ que minimiza el tiempo en llegar desde el centroide hasta el extremo (a, b) , lo vamos a denotar λ^- . Y el que minimiza el tiempo en el otro sentido lo vamos a denotar λ^+ .

El valor óptimo del problema de optimización (4.9) se denota λ^{+*} y respectivamente λ^{-*} , denota el valor óptimo del problema en sentido opuesto. La consideración del caso

i) y caso ii) conduce a:

$$\lambda^{-*} = \min\{1, \max\{0, \lambda^{-}\}\} \quad (4.13)$$

$$\lambda^{+*} = \max\{0, \min\{1, \lambda^{+}\}\} \quad (4.14)$$

En algunos casos la mejor solución será ir directamente hasta el extremo sin utilizar la autovía, lo que se denomina enrutamiento directo y correspondería al valor $\lambda^{+*} = 1$. Otro caso sería que para llegar al extremo que es el destino por ejemplo (c, d) , se incorpore en el otro extremo (a, b) , con lo cual el usuario utilizaría la autovía completa y correspondería al caso $\lambda^{+*} = 0$. Finalmente el caso $0 < \lambda^{+*} < 1$ corresponde al de la figura 4.3.

Ejemplos ilustrativos del problema i.1)

Mostramos en primer lugar, una tabla con los datos de entrada para cada una de las pruebas.

	(a,b)	(c,d)	(e,f)	v	w
Prueba 1	(100,200)	(1000,1000)	(500,0)	110 km/h	80 km/h
Prueba 2	(100,200)	(1000,1000)	(0,500)	110 km/h	80 km/h
Prueba 3	(100,200)	(1000,1000)	(0,0)	110 km/h	80 km/h
Prueba 4	(100,1000)	(500,500)	(500,100)	110 km/h	80 km/h
Prueba 5	(0,100)	(1000,100)	(10,0)	110 km/h	80 km/h
Prueba 6	(0,0)	(1000,0)	(10,100)	110 km/h	80 km/h

Tabla 4.1: Datos de las pruebas para el problema i.1)

Con esta tabla resumimos todos los datos de entrada. En los cuales, hemos recogido los tipos de casos posibles, desde casos en los que la autovía no es utilizada por los usuarios hasta casos donde los usuarios utilizan toda la autovía para llegar al destino.

Los resultados son mostrados en las gráficas de la figura 4.4, donde se indica de color rojo el camino óptimo en cada uno de las casos para llegar al destino desde el centroide.

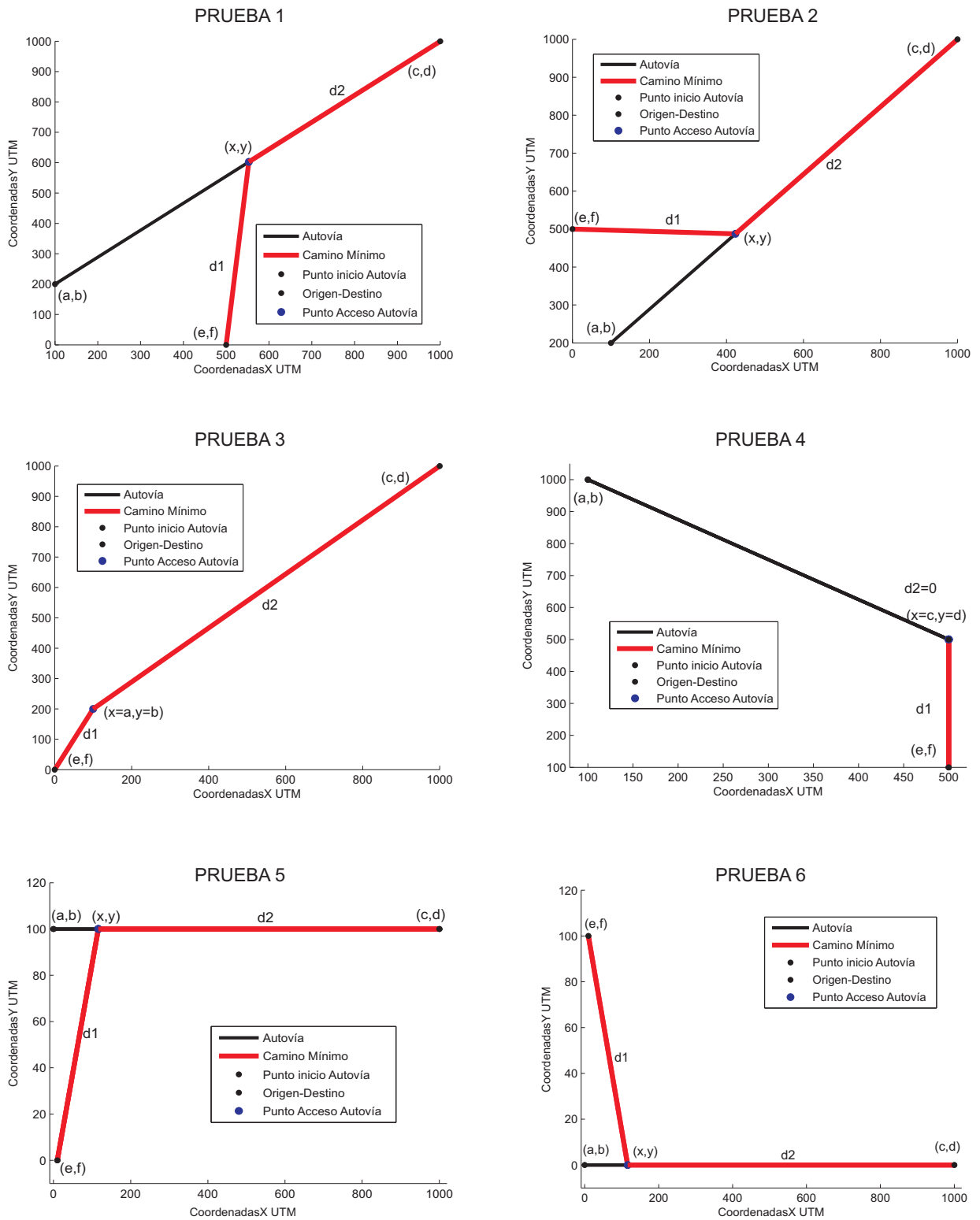


Figura 4.4: Soluciones al lote de pruebas para para el problema i.1)

A continuación, resumimos en una tabla los resultados numéricos obtenidos en cada una de las pruebas:

	λ	(x, y)	tiempo empleado (min)
Prueba 1	0,503319	(552,987344, 602,655417)	0,779962
Prueba 2	0,359220	(423,298037, 487,376033)	0,738488
Prueba 3	0	(100, 200)	0,824519
Prueba 4	1	(500, 500)	0,3
Prueba 5	0,115963	(115,962589, 100)	0,591476
Prueba 6	0,115963	(115,962589, 0)	0,591476

Tabla 4.2: Resultados de las pruebas para el problema i.1)

Los tiempos obtenidos son pequeños, ya que las distancias son muy pequeñas en estas pruebas simples. En cada una de las pruebas mostramos el valor de λ que es el parámetro con el que obtenemos las coordenadas de acceso óptimas. También mostramos dichas coordenadas y el tiempo mínimo en minutos obtenido.

En la Prueba 3, los usuarios recorren la autovía completa para llegar al destino, sus coordenadas de acceso son igual a las de inicio de la autovía.

En la Prueba 4, pasa todo lo contrario los usuarios van directamente al destino sin utilizar la autovía, sus coordenadas de acceso son igual a las de destino.

En las pruebas 5 y 6 como se puede observar que los resultados de λ y el tiempo son iguales, esto se debe a que son dos casos simétricos y las distancias son las mismas por tanto λ y el tiempo empleado para llegar al destino deben ser igual en ambos casos.

Pruebas con capitales de provincia de Castilla-La Mancha

Vamos a realizar dos pruebas del problema i.1) tomando datos reales sobre las cinco capitales de provincia de Castilla-La Mancha. Vamos a ver cual sería el recorrido mínimo para los siguientes pares de demanda origen-destino, para dos autovías dadas.

- 1) Ciudad Real-Albacete
- 2) Toledo-Albacete
- 3) Guadalajara-Albacete
- 4) Cuenca-Albacete

Suponiendo que la velocidad media por una carretera convencional es de 80 km/h y por la autovía de 110 km/h. Además en el primer caso, existe una autovía desde las coordenadas UTM (459000,4350000) hasta Albacete cuyas coordenadas UTM son (599200,4317149). Se calculará como los usuarios de Ciudad Real, Toledo, Guadalajara y Cuenca emplean la citada autovía de forma que tarden el mínimo tiempo posible en llegar hasta Albacete.

A continuación, resumimos en la tabla 4.3 los resultados numéricos obtenidos en el primer caso:

par de demanda	(x, y)	tiempo empleado (min)
Ciudad Real-Albacete	(472645, 4346803)	117,039926
Toledo-Albacete	(459000, 4350000)	137,039817
Guadalajara-Albacete	(599200, 4317149)	160,105572
Cuenca-Albacete	(599200, 4317149)	91,697704

Tabla 4.3: Caso 1 con capitales de provincia de Castilla-La Mancha

La solución obtenida es representada gráficamente en la figura 4.5.

Los resultados del primer caso, indican que la autovía sería importante para los usuarios de Ciudad Real y Toledo y no se emplearía para los de Guadalajara y Cuenca.

Ahora analizaremos un segundo caso. Si consideramos que la autovía comienza en (560000,4420000) y llega hasta Albacete, los resultados obtenidos en este caso se muestran en la tabla 4.4 y la solución obtenida es representada gráficamente en la figura 4.6.

par de demanda	(x, y)	tiempo empleado (min)
Ciudad Real-Albacete	(599200, 4317149)	117,710086
Toledo-Albacete	(599200, 4317149)	157,272820
Guadalajara-Albacete	(560000, 4420000)	140,680381
Cuenca-Albacete	(560000, 4420000)	76,527750

Tabla 4.4: Caso 2 con capitales de provincia de Castilla-La Mancha

Los resultados del segundo caso, indican que la autovía sería empleada por usuarios de Guadalajara y Cuenca y no por los de Ciudad Real y Toledo.

Una observación importante es que en el caso 1 los usuarios de Ciudad Real-Albacete emplean la autovía durante gran parte de su recorrido frente al caso 2 que elige las carreteras actuales, sin embargo los tiempos son similares. Esto motiva la necesidad de elaborar criterios cuidadosos para evaluar el beneficio de la infraestructura. No solo es importante que la empleen los usuarios sino que produzca reducciones de tiempo significativas.

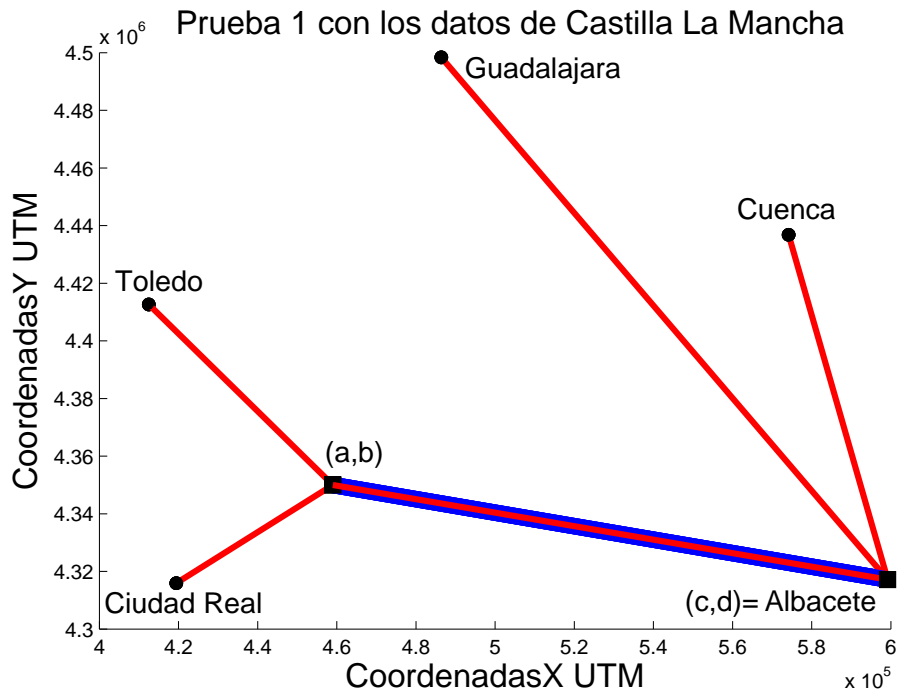


Figura 4.5: Prueba 1 para las capitales de provincia de Castilla-La Mancha

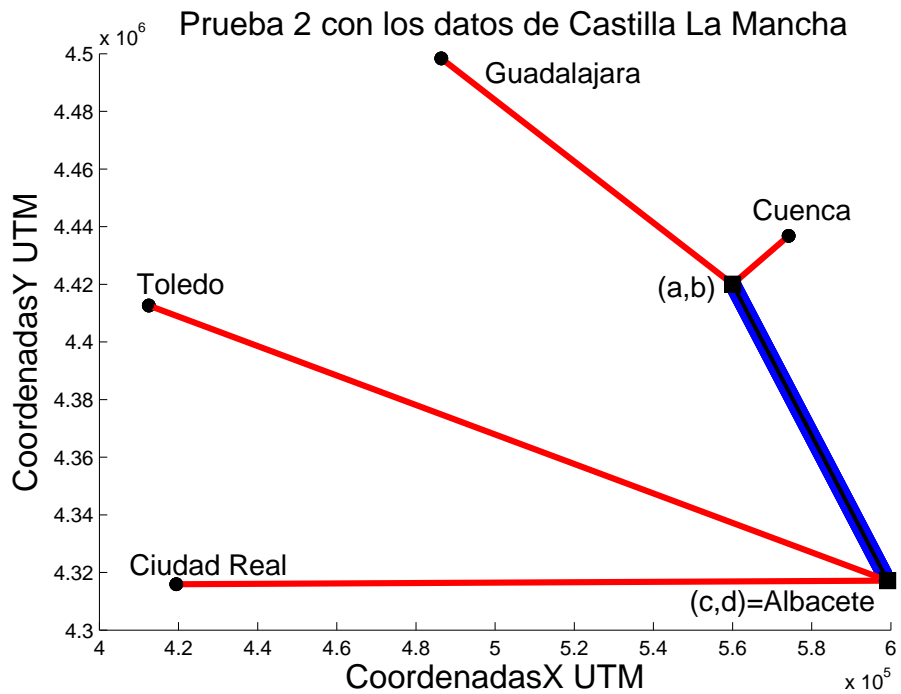


Figura 4.6: Prueba 2 para las capitales de provincia de Castilla-La Mancha

Problema i.2): Puntos de salida/entrada entre dos infraestructuras lineales

Se trata de calcular el camino óptimo entre el punto de inicio del primer segmento (que representa una de las autovías) y el punto del final del segundo segmento (que representa la otra autovía). Para ilustrar esta cuestión, considérese el ejemplo con dos autovías mostrado en la figura 4.7.

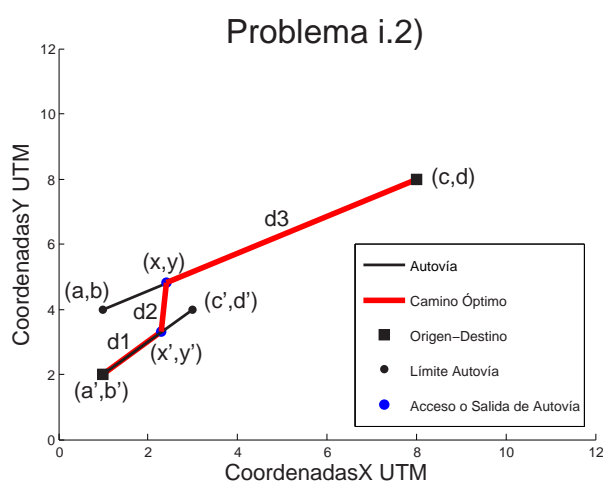


Figura 4.7: Problema i.2)

Tenemos dos segmentos que modelan las dos autovías, la primera va del punto (a',b') al (c',d') y la segunda va del punto (a,b) al (c,d) . Calculamos el camino mínimo para ir desde (a',b') hasta (c,d) , este camino mínimo en la gráfica anterior está representado en color rojo. Se asume que los usuarios pueden acceder a cualquier punto de la autovía a una velocidad w y que la velocidad de viaje en la autovía es v siendo $v > w$. Las coordenadas (x',y') son las de salida de la primera autovía y las (x,y) son las de incorporación a la segunda autovía para minimizar el tiempo en ir desde (a',b') hasta (c,d) .

Una vez expuesto el problema, los parámetros de entrada son:

w : velocidad en el tramo de acceso a la autovía.

v : velocidad en el tramo de autovía.

(a, b) : punto inicial del segundo segmento.

(c, d) : punto final del segundo segmento.

(a', b') : punto inicial del primer segmento.

(c', d') : punto final del primer segmento.

d_1 : distancia euclídea desde (a', b') hasta (x', y') .

d_2 : distancia euclídea desde (x', y') hasta (x, y) .

d_3 : distancia euclídea desde (x, y) hasta (c, d) .

Los usuarios emplearán la ruta de mínimo coste y por tanto elegirán los nodos de transferencia (salida e incorporación) (x', y') y (x, y) de modo que se minimice el tiempo total de realizar el viaje, esto es, de modo que se minimice $\frac{d_1+d_3}{v} + \frac{d_2}{w}$. El camino óptimo viene dado por la solución del siguiente problema de optimización.

$$\begin{aligned}
 \text{Minimizar} \quad & \frac{\sqrt{(x'-a')^2+(y'-b')^2}}{v} + \frac{\sqrt{(x-x')^2+(y-y')^2}}{w} + \frac{\sqrt{(c-x)^2+(d-y)^2}}{v} \\
 \text{sujeto a:} \quad & x = \lambda(c - a) + a \\
 & y = \lambda(d - b) + b \\
 & \lambda \in [0, 1] \\
 & x' = \lambda'(c' - a') + a' \\
 & y' = \lambda'(d' - b') + b' \\
 & \lambda' \in [0, 1]
 \end{aligned} \tag{4.15}$$

Reemplazando las restricciones asociadas a x, x', y, y' en la función objetivo el problema (4.15) admite una expresión genérica del tipo:

$$\begin{aligned}
 \text{Minimizar} \quad & f(\lambda, \lambda') \\
 \text{sujeto a:} \quad & \lambda \in [0, 1] \\
 & \lambda' \in [0, 1]
 \end{aligned} \tag{4.16}$$

El objetivo de la siguiente discusión, es expresar el problema (4.16) en dos variables como un problema de optimización unidimensional. El principio de optimalidad de Bellman aplicado a este problema afirma que el camino óptimo de ir de (a', b') a (c, d) está compuesto de subcaminos óptimos, por tanto, si el punto de salida de la primera autovía (x', y') fuese conocido, el punto de incorporación a la segunda autovía (x, y) debe ser óptimo. Entonces si λ' fuese conocida tendríamos que:

$$\lambda(\lambda') = \max \left\{ 0, \min \left\{ 1, \frac{((c-a)(x'(\lambda') - a) + (d-b)(y'(\lambda') - b))}{[(c-a)^2 + (d-b)^2]} + \frac{w|(c-a)(y'(\lambda') - b) - (d-b)(x'(\lambda') - a)|}{[(c-a)^2 + (d-b)^2]\sqrt{(v^2 - w^2)}} \right\} \right\} \quad (4.17)$$

donde

$$\begin{aligned} x'(\lambda') &= \lambda'(c' - a') + a' \\ y'(\lambda') &= \lambda'(d' - b') + b' \end{aligned}$$

Finalmente el problema unidimensional a resolver es;

$$\begin{aligned} \text{Minimizar } & g(\lambda') = f(\lambda(\lambda'), \lambda') \\ \text{sujeto a: } & \lambda' \in [0, 1] \end{aligned} \quad (4.18)$$

Para minimizar la función $g(\lambda')$ calculamos su derivada con el programa Derive 6 y el resultado de esta derivada es excesivamente grande y complicada, lo cual dificulta el trabajo analítico con ella. Este resultado, ha motivado trabajar con un método aproximado de optimización de funciones unidimensionales como el de interpolación cuadrática.

Método de interpolación cuadrática para la optimización de una función de una variable

Este método interpola la función objetivo mediante una parábola, lo que requiere conocer tres puntos de la gráfica de la función $g(\lambda')$, y considera que este polinomio cuadrático es una aproximación de la función objetivo. En nuestro caso para iniciar este algoritmo calculamos $g(0)$, $g(0,5)$ y $g(1)$. Así obtenemos tres puntos de $g(\lambda')$. En la figura 4.8, podemos observar un ejemplo con los tres puntos y el polinomio cuadrático que se formaría.

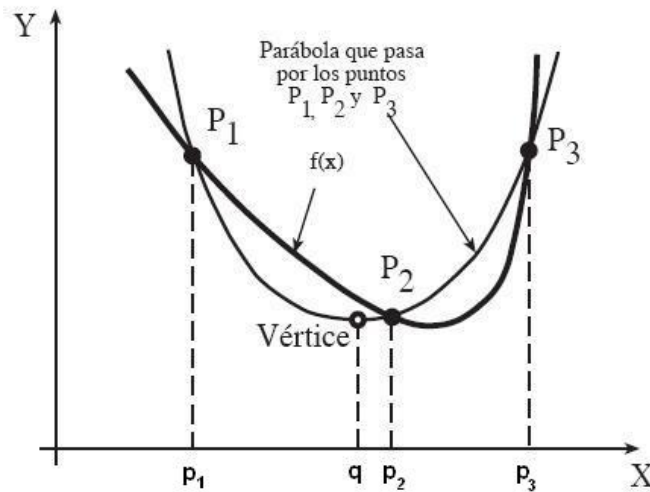


Figura 4.8: Función objetivo a minimizar junto a una parábola interpolante

Por tanto, el método toma el vértice de la parábola (denotado por q) como una aproximación del mínimo de la función objetivo. El proceso se repite con el vértice hallado y con dos de los tres puntos iniciales de interpolación. Más formalmente, se supone que se minimiza la función (convexa) $g(x)$, y se consideran tres puntos $p_1 < p_2 < p_3$ tales que se cumple (véase la figura 4.8) $g(p_1) \geq g(p_2) \leq g(p_3)$.

Se puede suponer que al menos una de las dos desigualdades anteriores se cumple estrictamente. En caso contrario, por la convexidad de g sería constante y todos los puntos del intervalo $[p_1, p_3]$ serían mínimos. Seguidamente, se ajusta una parábola que

pase por los puntos $(p_1, g(p_1))$, $(p_2, g(p_2))$ y $(p_3, g(p_3))$. Su vértice viene dado por la expresión;

$$q = p_2 - \frac{1}{2} \frac{(p_2 - p_1)^2 [g(p_2) - g(p_3)]^2 - (p_2 - p_3)^2 [g(p_2) - g(p_1)]^2}{(p_2 - p_1)[g(p_2) - g(p_3)] - (p_2 - p_3)[g(p_2) - g(p_1)]} \quad (4.19)$$

Si los tres puntos fuesen colineales, el denominador sería cero y la expresión anterior no estaría definida, pero esta situación es imposible bajo la suposición de $g(p_1) > g(p_2)$ ó $g(p_2) < g(p_3)$. Así garantizamos que el vértice de la parábola es un mínimo (y no un máximo), y que está situado en el intervalo (p_1, p_3) .

Una vez calculado el vértice q de dicha parábola, nos queda un conjunto con cuatro puntos $\{p_1, p_2, p_3, q\}$, del cual nos quedaremos con tres para que se pueda repetir el proceso y los tres deben cumplir $g(p_1) \geq g(p_2) \leq g(p_3)$. Pueden aparecer tres casos:

Caso 1 ($q < p_2$). Dentro del Caso 1 tenemos dos posibilidades:

- a. Si $g(p_2) \geq g(q)$ el mínimo de la función se encuentra en el intervalo (p_1, p_2) ; entonces, el nuevo conjunto de tres puntos es $\{p'_1, p'_2, p'_3\} = \{p_1, q, p_2\}$.
- b. Si $g(p_2) < g(q)$ el mínimo de la función se encuentra en el intervalo (q, p_3) ; entonces, el nuevo conjunto de tres puntos es $\{p'_1, p'_2, p'_3\} = \{q, p_2, p_3\}$.

Caso 2 ($q > p_2$). Dentro del Caso 2 tenemos dos posibilidades:

- a. Si $g(p_2) \geq g(q)$ el mínimo de la función se encuentra en el intervalo (p_1, q) ; entonces, el nuevo conjunto de tres puntos es $\{p'_1, p'_2, p'_3\} = \{p_1, p_2, q\}$.
- b. Si $g(p_2) < g(q)$ el mínimo de la función se encuentra en el intervalo (p_2, p_3) ; entonces, el nuevo conjunto de tres puntos es $\{p'_1, p'_2, p'_3\} = \{p_2, q, p_3\}$.

Caso 3 ($q = p_2$). En este caso se obtienen los tres puntos iniciales y es imposible decidir en qué intervalo reducido se encuentra la solución. Para evitar este problema, se

reemplaza p_2 por $(p_1 + p_2)/2$ y se repite el proceso, con la garantía de que en el siguiente intento se estará en el Caso 1 o en el Caso 2.

La técnica de interpolación cuadrática, descrita anteriormente, tiene asegurada la convergencia bajo la hipótesis de que g es una función convexa. Una modificación que simplifica la codificación del procedimiento consiste en sistemáticamente reemplazar el vértice q por el último punto generado. Este nuevo procedimiento presenta una convergencia local, esto es, la convergencia depende de lo cerca que estén los puntos iniciales de la solución óptima.

Con esta método basado en la interpolación cuadrática nos vamos aproximando a la solución.

A continuación mostramos las dos figuras 4.9 y 4.10, para ilustrar los casos 1a y 1b.

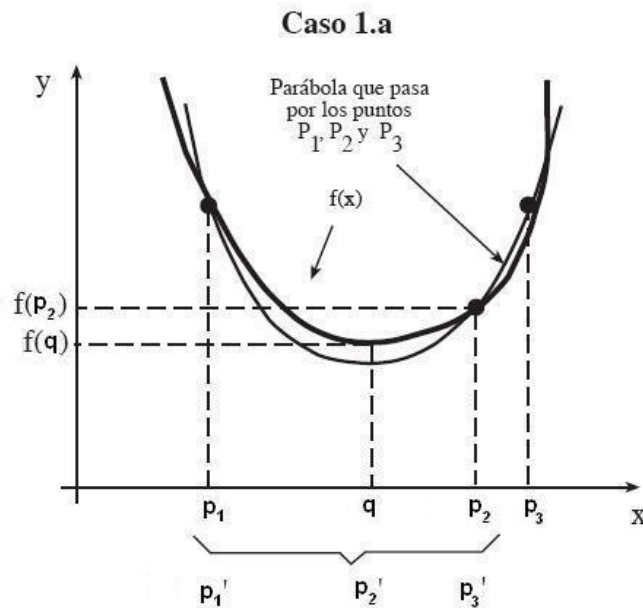


Figura 4.9: Caso 1.a en el método de optimización unidimensional

Este método está desarrollado para problemas unidimensionales sin restricciones, y el problema 4.18 considera la restricción $\lambda' \in [0, 1]$ por lo que lo hemos modificado del siguiente modo.

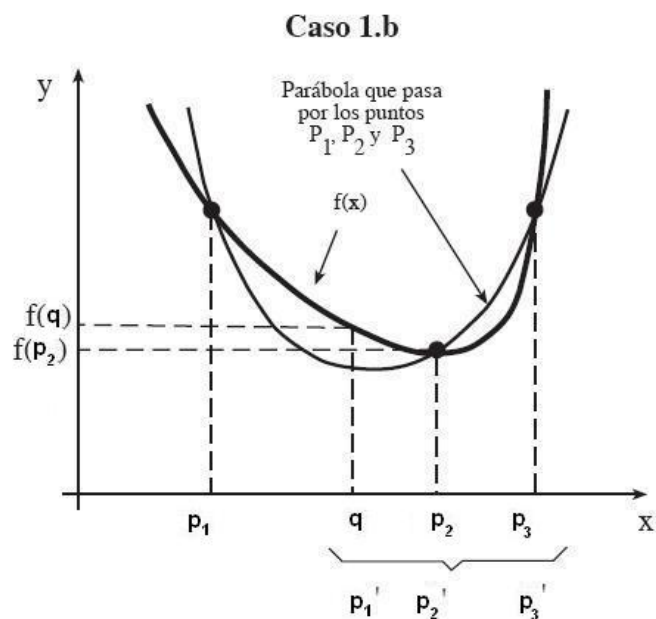


Figura 4.10: Caso 1.b en el método de optimización unidimensional

$\lambda' \in [0, 1]$. Por tanto si $q > 1$ o $q < 0$ tendremos: ($q > 1$) En este caso si se cumple, $(q-1) > (1-p_1)$ el λ' óptimo aproximado es uno. Si no se cumple damos otra iteración al algoritmo con el nuevo conjunto de tres puntos:

$$\{p'_1, p'_2, p'_3\} = \left\{ \frac{p_1 + p_2}{2}, \frac{p_2 + p_3}{2}, 1 \right\} \quad (4.20)$$

($q < 0$) En este caso si se cumple, $(-q) > p_3$ el λ' óptimo aproximado es cero. Si no se cumple damos otra iteración al algoritmo con el nuevo conjunto de tres puntos:

$$\{p'_1, p'_2, p'_3\} = \left\{ 0, \frac{p_1 + p_2}{2}, \frac{p_2 + p_3}{2} \right\} \quad (4.21)$$

Ejemplos numéricos ilustrativos del problema i.2)

La tabla 4.3 muestra los datos de entrada para cada una de las pruebas realizadas.

	(a,b)	(c,d)	(a',b')	(c',d')	v	w
Prueba 1	(0,0)	(1000,1000)	(300,100)	(1000,200)	110 km/h	80 km/h
Prueba 2	(400,400)	(1000,700)	(100,0)	(500,200)	110 km/h	80 km/h
Prueba 3	(400,600)	(0,400)	(200,300)	(600,300)	110 km/h	80 km/h
Prueba 4	(200,500)	(300,700)	(100,100)	(1000,300)	110 km/h	80 km/h
Prueba 5	(0,200)	(500,500)	(0,0)	(200,400)	110 km/h	80 km/h
Prueba 6	(400,600)	(1000,400)	(200,300)	(600,450)	110 km/h	80 km/h

Tabla 4.5: Datos de las pruebas para el problema i.2)

Con la subrutina programada con MATLAB se calcula el recorrido óptimo en cada una de las pruebas. Una vez hechos los cálculos, se muestran las gráficas con los resultados de cada una de las pruebas. Donde se indica de color rojo el camino óptimo en cada uno de los casos para llegar al destino:

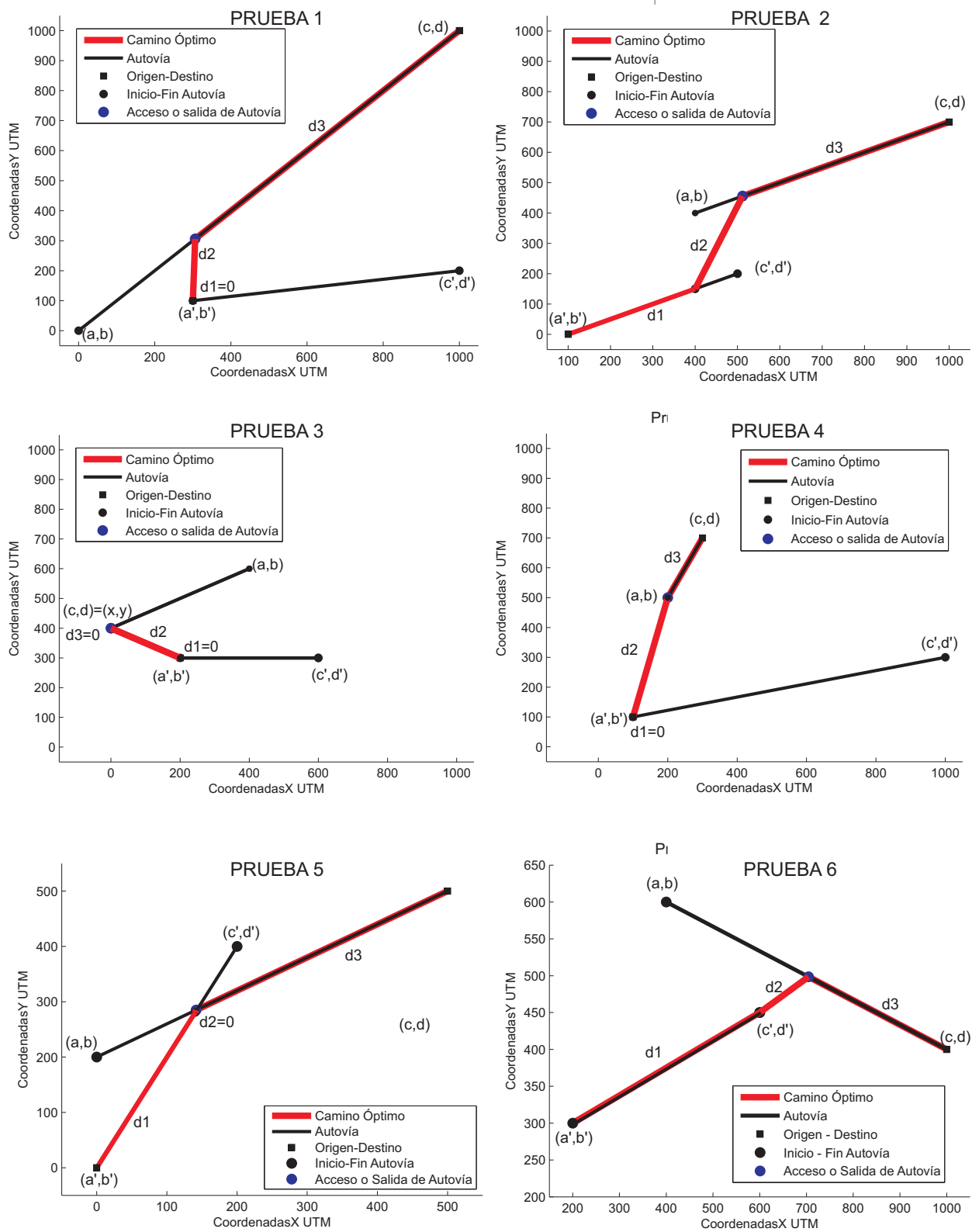


Figura 4.11: Soluciones al lote de pruebas para el problema i.2)

A continuación, resumimos en una tabla los resultados numéricos obtenidos en cada una de las pruebas:

	λ'	(x', y')	λ	(x, y)	t (min)
Prueba 1	0	(300, 100)	0,306	(305,9626, 305,9282)	0,6899
Prueba 2	0,75	(400, 150)	0,1865	(511,92, 455,96)	0,7249
Prueba 3	0	(200, 300)	1	(0, 400)	0,1677
Prueba 4	0	(100, 100)	0	(200, 500)	0,4312
Prueba 5	0,7019	(140,37, 280,75)	0,2831	(141,55, 284,93)	0,4025
Prueba 6	1	(600, 450)	0,5074	(704,47, 498,51)	0,4893

Los tiempos obtenidos son pequeños, ya que las distancias son muy pequeñas en estas pruebas simples. En cada una de las pruebas mostramos el valor de λ' que es el parámetro con el que obtenemos las coordenadas de salida de la primera autovía óptima. También mostramos λ para calcular las coordenadas de acceso óptima a la segunda autovía. Además mostramos las coordenadas de salida de la primera autovía y las de acceso a la segunda autovía. Por último mostramos el tiempo mínimo en minutos obtenido.

4.3.1. Problema ii): Grafo de estrategia

Se ha desarrollado el denominado *grafo de estrategia* con el objeto de calcular la ruta óptima de los usuarios empleando las nuevas infraestructuras lineales. Este grafo contiene como nodos un centroide origen O , todos los centroides destinos para el origen O y cada sentido de una infraestructura lineal (recogido por un nodo). Los arcos modelan las distintas elecciones que un usuario puede realizar. Se han omitido los arcos que unen los dos sentidos de una misma infraestructura, ya que esos arcos nunca formarán parte de la estrategia óptima de los usuarios. Un camino en esta red define el conjunto y orden en utilizar las distintas infraestructuras que forman la red.

Para ilustrar el concepto de *grafo de estrategia*, considérese el grafo de la figura 4.12 para el caso de dos autovías. Un usuario para realizar el viaje tipo $\omega = (O, D)$ puede

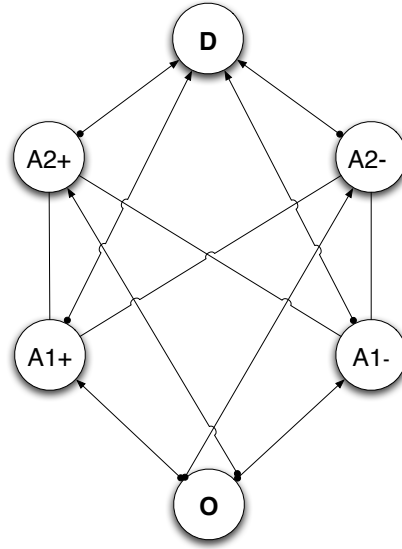


Figura 4.12: Grafo de estrategia para dos autopistas

elegir emplear la infraestructura lineal 1 ó 2 en un sentido (+) o en el otro sentido (-). Si el usuario decide incorporarse a la autopista 1 en un determinado sentido, y tras recorrer un tramo, el usuario puede elegir abandonarla para acceder a la infraestructura 2 en un sentido determinado o para ir directamente al centroide destino D .

Los arcos del grafo de estrategia se dividen en dos tipos, en función del modo en que se calculan sus costes:

- a) **Adyacentes con centroides.** El coste de este tipo de arco es el tiempo de viaje desde el centroide a la incorporación/salida. El punto óptimo de esta incorporación es calculado con uno de los valores $\lambda^{+*}, \lambda^{-*}$. Notar que estamos calculando la estrategia óptima desde un origen fijo O . Para este origen dado los puntos de incorporación a todas las autopistas y en los dos sentidos se ha calculado resolviendo problema i.1).

El número de incorporaciones desde los centroides es $2m$, siendo m el número de infraestructuras lineales de la red \mathbf{b} . Denotamos por $\mathcal{N}_I = \{1, \dots, 2m\}$ el conjunto de estas incorporaciones y por $t_{(O,i)}$, para todo $i \in \mathcal{N}_I$ el coste (tiempo) en incorporarse a la infraestructura.

De forma similar las fórmulas desarrolladas en (4.13) y (4.14) permiten calcular el punto de salida que se define el coste (tiempo) de salida desde una infraestructura

lineal hasta un destino D . La diferencia respecto al caso anterior es que este coste recoge el tiempo empleado dentro de la infraestructura lineal más el tiempo desde ella al destino D . Estos costes se almacenarán en $(i, D) \forall i \in \mathcal{N}_I$

- b) Arcos entre infraestructuras lineales. Estos arcos contienen el coste desde la incorporación a la primera infraestructura hasta el acceso a la segunda. Estos costes son calculados aproximadamente resolviendo un problema de optimización unidimensional. En este trabajo se ha considerado el método de interpolación parabólica descrito anteriormente.

Consideramos que se puede calcular

$$t_{ij}, \forall i, j \in \mathcal{N}_I$$

Dado $i \in \mathcal{N}_I$ denotamos por \bar{i} la misma infraestructura lineal pero en sentido opuesto.

Algoritmo de Dijkstra modificado para calcular el recorrido mínimo del Grafo de estrategia

El cálculo de los costes de cada arco en el grafo de estrategia requiere resolver un problema de optimización. Esta operación es costosa computacionalmente y debe de evitarse lo más posible. En esta sección adaptamos el algoritmo de Dijkstra al grafo de estrategia, cuya topología es conocida de antemano, con el objeto de calcular el menor número de costes de arco.

Este algoritmo se muestra en la tabla 4.6, los datos de salida son la estrategia óptima desde un origen O a todos sus destinos y el correspondiente tiempo en llegar a cada uno de los destinos.

Tabla 4.6: Algoritmo modificado de Dijkstra

1. (Inicialización):

$$U = \{o\},$$

$$V = \{d \in \mathcal{O} / (o, d) \in W\}$$

$$g(o) = 0,$$

$$x(o) = x_o,$$

$$y(o) = y_o,$$

Calcular t_{oj} para toda $j \in \mathcal{N}_I = \{1, \dots, 2m\}$

$$h(j) = \begin{cases} t_{oj} & \forall (o, j) \in j \in \mathcal{N}_I \\ +\infty & \text{en otro caso} \end{cases}$$

$$p(j) = \begin{cases} o & \forall j \in \mathcal{N}_I \\ +\infty & \text{en otro caso} \end{cases}$$

2. Sea

$$h(i) = \text{minimizar}_{j \notin U} h(j)$$

$$i = \text{Arg minimizar}_{j \notin U} h(j)$$

Tomar $U = U \cup \{i\}$ y $g(i) = h(i)$. Si $U = V$ parar.

3. Si $i \in \mathcal{N}_I$, hacer para todo $j \notin U \cup \{\bar{i}\}$ calcula t_{ij} y denota por (\tilde{x}, \tilde{y}) su incorporación óptima. Actualiza

$$h(j) \leftarrow \text{mín}\{h(j), g(i) + t_{ij}\}$$

Si $h(j) = g(i) + t_{ij}$ tomar $p(j) = i$ y actualizar la incorporación a j tomando $x(j) = \tilde{x}$ y $y(j) = \tilde{y}$. Ir al paso 2.

Salida: El coste del camino mínimo a través del sistema de infraestructuras lineales desde el origen o a todos los centroides destinos, y esta información está contenida en la variable $g(d)$. La estrategia óptima para el par $(o, d) \in W$ es

$$d \rightarrow p(d) \rightarrow p(p(d)) \rightarrow p(p(p(d))) \rightarrow \dots \rightarrow o$$

4.4. Criterios de diseño

Los criterios de diseño definen la función objetivo del problema de optimización.

Los problemas de diseño de redes de transporte se pueden abordar bajo dos ópticas:

1. **Competencia entre redes.** Este enfoque considera que las nuevas infraestructuras compiten con la red de transporte existente. El objetivo perseguido es captar la mayor demanda, reduciendo así el nivel de servicio de la red existente.
2. **Complementariedad de redes.** Este enfoque considera la nueva infraestructura como una ampliación de la red existente. El objetivo que se persigue es optimizar la red resultante, por ejemplo, minimizando el tiempo total de transporte en el sistema de transporte. Un ejemplo de esta metodología es el seguido en la promoción de los denominados viajes en modos combinados como el llamado *park'n ride*.

Si se desea implementar una red de metro con el objeto de reducir la congestión en una red de tráfico, el enfoque de competencia entre redes de transporte es el más adecuado. Dos criterios utilizados son:

Maximización del cubrimiento de la red (b) .

$$\text{Maximizar } Z_b = \sum_{\omega} g_{\omega}^b \quad (4.22)$$

Minimización de la congestión de la red (a) .

$$\text{Minimizar } Z_a = \sum_{\omega} u_{\omega}^a g_{\omega}^a \quad (4.23)$$

Para el problema de localizar una red de autovías sobre una red de carreteras existentes el enfoque más adecuado sería un enfoque de complementariedad de redes, siendo el objetivo el de minimizar el tiempo total de transporte en el sistema

Minimización del tiempo total de transporte en la red (a) \cup (b) .

$$\text{Minimizar } Z = \sum_{\omega} (u_{\omega}^a g_{\omega}^a + u_{\omega}^b g_{\omega}^b) \quad (4.24)$$

Capítulo 5

Algoritmo metaheurístico de resolución

5.1. Un algoritmo metaheurístico tipo Búsqueda Dispersa

5.1. Un algoritmo metaheurístico tipo Búsqueda Dispersa

En este PFC se propone un método metaheurístico de tipo BD Búsqueda Dispersa 3.2.1, en el que se ha explotado las cuatro características fundamentales del modelo:

1. Evaluar la función objetivo tiene un alto coste computacional.
2. El número de variables de decisión es pequeño. Cada autovía queda determinado por las coordenadas de los extremos y por tanto el número de variables de decisión es $4m$, donde m es el número de infraestructuras lineales a localizar.
3. Es un problema con óptimos locales.
4. Es un problema con una única restricción (4.1). Las infraestructuras lineales se pueden localizar en cualquier lugar del plano. A priori se puede reducir el espacio de factibilidad a un rectángulo, que lo denotaremos por \mathcal{R} , definido por:



Figura 5.1: Espacio de factibilidad

donde

$$x_{MIN} = \min_{c \in \mathcal{O}} x_c,$$

$$x_{MAX} = \max_{c \in \mathcal{O}} x_c,$$

$$y_{MIN} = \min_{c \in \mathcal{O}} y_c,$$

$$y_{MAX} = \max_{c \in \mathcal{O}} y_c$$

Definiciones y notación empleada en la descripción del el algoritmo.

- Un segmento viene determinado por las coordenadas cartesianas de sus extremos (4 coordenadas). En este trabajo emplearemos otro sistema de coordenadas, definidas por las coordenadas cartesianas (a, b) de un extremo, el ángulo θ formado por el segmento y el vector $(0, 1)$ y su longitud ℓ . Denotaremos una infraestructura lineal por la 4-tupla $L = (a, b, \theta, \ell)$. Esta parametrización de las soluciones se ha elegido para facilitar la operación de contracción de la longitud total de la red.

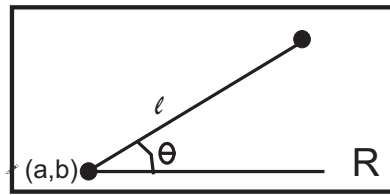


Figura 5.2: Parámetros que determinan una infraestructura lineal

- Una solución factible es un conjunto de segmentos $S = \{L_i\} = \{(a_i, b_i, \theta_i, \ell_i)\}$ con $i = 1, \dots, m$, $m \leq n$ y cumpliéndose que $\sum_{i=1}^m \ell_i \leq C$ (ecuación 4.1).
- Se trabaja con un concepto de proximidad entre infraestructuras lineales que engloba consideraciones de geometría del problema con calidad de las soluciones. Denotamos por $\omega \notin L_k$ a los usuarios del par origen-destino w que no emplean la infraestructura L_k . Las funciones objetivos consideradas son lineales y por tanto se pueden expresar como la suma de las contribuciones en todos los pares origen-destino w , esto es, $Z = \sum_{\omega \in W} z_\omega$. Denotamos por $z_\omega(L_k)$ el valor de la función objetivo para la infraestructura lineal L_k . Por ejemplo si $\omega \notin L_k$ entonces $z_\omega(L_k) = 0$. Se define la distancia de L_i a L_j por

$$d(L_i, L_j) = \sum_{\omega \in L_i} |z_\omega(L_i) - z_\omega(L_j)|$$

Se define la *proximidad* de una infraestructura L_i a una solución S por

$$d(L_i, S) = \min_{L_j \in S} d(L_i, L_j)$$

Este concepto de proximidad es el más adecuado de los que se nos han ocurrido, ya que capta la competencia existente entre infraestructuras lineales, pero tiene un alto coste computacional en problemas reales. Suponiendo un problema donde el número de pares $w \in W$ es del orden de 50,000, el número de autopistas a localizar es 2 – 5, y el número de soluciones del conjunto de referencia de 10, conduce a almacenar y operar con varios millones de datos en el cálculo de la proximidad.

Este problema nos ha motivado a introducir un segundo concepto, aunque teóricamente más débil, pero con un bajo coste computacional.

Para calcular la distancia entre los dos segmentos,

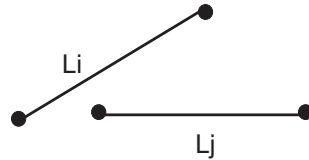


Figura 5.3: Ejemplo ilustrativo de la definición de proximidad entre dos segmentos

Hacemos:

- 1) Calculamos la recta que contiene a L_i .

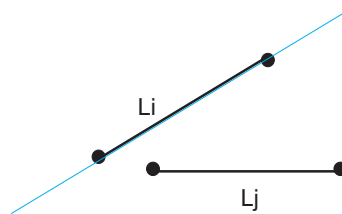


Figura 5.4: Recta que contiene al segmento

- 2) Calculamos distancia de cada extremo de L_j a dicha recta.

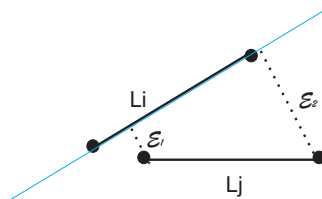


Figura 5.5: Distancia de los extremos a la recta

3) Se define \tilde{d}

$$\tilde{d}(L_i, L_j) = \max\{\varepsilon_1, \varepsilon_2\}$$

Finalmente

$$d(L_i, S) = \min_{L_j \in S} \tilde{d}(L_i, L_j)$$

Los elementos claves en un algoritmo BD son:

- I. **Un método de mejora.** El modelo posee pocas variables, un total de $4m$, siendo m el número infraestructuras lineales de la actual solución S . Este hecho motiva emplear un método aproximado de tipo gradiente proyectado. El primer paso es aproximar el gradiente (en los punto donde exista) de la siguiente forma.

$$\nabla_k Z(S) \approx \frac{Z(S + \varepsilon e_k) - Z(S)}{\varepsilon},$$

$$k = 1, \dots, 4m.$$

donde e_k es el vector que contiene un 1 en la posición k -ésima y ceros en el resto de posiciones. Esta aproximación se guarda en el vector d . La infraestructura mejorada es

$$\hat{S} = S - \alpha d$$

donde α es un parámetro de proyección. El nuevo punto puede ser no factible ya que la nueva longitud total construida puede exceder al presupuesto disponible, esto es: $\hat{C} = \sum_{i=1}^m (\ell_i - \alpha d_i) > C$. En dicho caso se contraen proporcionalmente (proyección proporcional) las componentes para garantizar la factibilidad, mediante la expresión:

$$\hat{\ell}_i \leftarrow \frac{C}{\hat{C}} \hat{\ell}_i, i = 1, \dots, m$$

Esta contracción se efectuaría en un solo extremo, el opuesto a (a, b) . En la herramienta la reducción se reparte igualmente entre los dos extremos.

El procedimiento se repite varias veces hasta que no se obtiene una mejora significativa.

II. Un generador de soluciones diversas.

Se parte de un conjunto $\mathcal{L} = \{L_i\}$ de alrededor de 10,000 soluciones con una sola infraestructura lineal, que constituyen una exploración uniforme de $\mathcal{R} \times [-\pi/2, \pi/2) \times \{C\}$, donde C es la longitud máxima de una infraestructura lineal permitida por el presupuesto disponible ver figura 5.6. Se evalúan cada una de las soluciones que forman el mallado. Además se elimina la parte de las autovías no contenida en el rectángulo de factibilidad \mathcal{R} .

El total de las N soluciones, se dividen en b grupos, agrupando la N/b consecutivas infraestructuras lineales generadas. Dentro de cada uno de estos N/b grupos se elige la infraestructura lineal que produce mejor resultado, para inicializar con las b mejores autovías el **conjunto de referencia**. Esta elección permite generar soluciones diferentes y además elegir las mejores. Con un valor de $b = 10$ y un valor de $N = 10,000$, cada grupo tiene unas 1,000 autovías.

El conjunto de referencia se somete a mejora con el algoritmo del gradiente proyectado.

A continuación mostramos en la figura 5.6 un ejemplo de las autovías generadas con $N = 100$. Los centroides aparecen con un grosor en función de la demanda de personas que tiene dicho centroide como destino o como origen.

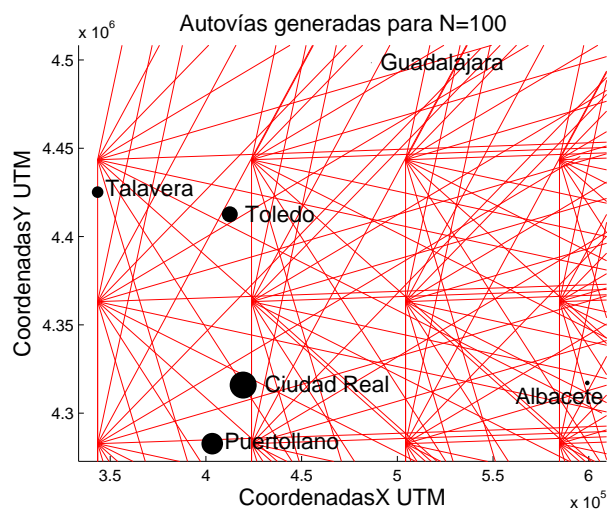


Figura 5.6: Soluciones generadas inicialmente para $N = 100$

De todas las soluciones generadas en 5.6, las soluciones mejores que formarían el conjunto de referencia con $b = 10$ se muestran en la figura 5.7, teniendo en cuenta que $C = 300$ km y que $N = 100$.

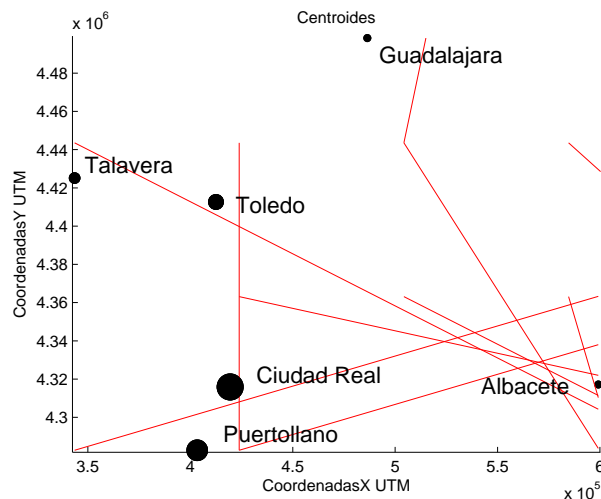


Figura 5.7: Conjunto de referencia para $b = 10$

III. **Un método de combinación.** Dadas dos soluciones S_1, S_2 la combinación de ambas se realiza mediante: i) la **unión** de todas sus infraestructuras, ii) Eliminación de infraestructuras similares y obtención de una resultante. Supongamos que tenemos dos autovías similares como muestra la figura 5.8

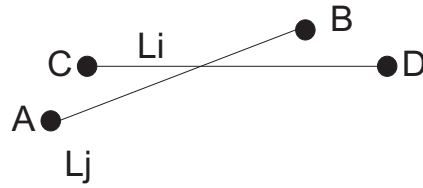


Figura 5.8: Autovías similares

Se calculan las distancias euclídeas representadas con cuadro blanco (ver figura 5.9). Por último se toma como nueva infraestructura, el segmento definido por los

	A	B	C	D
A	X			
B	X	X		
C	X	X	X	
D	X	X	X	X

Figura 5.9: Distancias euclídeas que se van a calcular

puntos más alejados de las dos autovías. Esto es, se calcula el máximo de todas las celdillas y el par de puntos que la definan, definirán los extremos de la misma, ver figura 5.10. De esta forma se eliminarán todas las infraestructuras lineales parecidas y se reemplaza por la anterior.

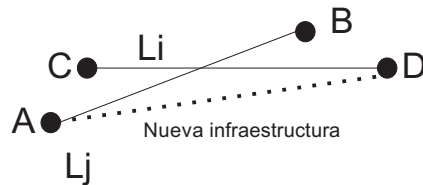


Figura 5.10: Nueva infraestructura lineal

y iii) mantenimiento de la factibilidad mediante el anterior mecanismo de *proyección proporcional* descrito en el método del gradiente proyectado.

Todos estos elementos se combinan en el algoritmo BD descrito en 5.1, donde b es el número de elementos del conjunto de referencia, que en nuestro caso serán 10.

Tabla 5.1: El algoritmo BD

-
0. (*Inicialización*). Sea $\mathcal{R} = \{L_1, \dots, L_b\}$ el conjunto de referencia generado como anteriormente se ha descrito en II) y ordenadas de mejor a peor respecto del valor de la función objetivo. El conjunto de referencia se somete a mejora del gradiente proyectado descrito en I).
1. Hacer NuevaSolucion=TRUE
- Mientras** (NuevaSolucion)
2. Actualización del conjunto de referencia.
- 2.1 Generación de todos los subconjuntos de dos elementos de \mathcal{R} . Se combinan dos a dos las soluciones.
- 2.1.1. Eliminación de autovías redundantes.
- 2.1.2. Contracción de autovías.
- 2.1.3. Evaluación con la función objetivo de cada una de las nuevas combinaciones y se ordenan de menor a mayor respecto del valor de la función objetivo.
- 2.2 Selección del mejor conjunto de referencia: de las b del conjunto de referencia, más las $\frac{(b-1)b}{2}$ combinaciones. Las nuevas soluciones que han entrado se someten a mejora con el algoritmo del gradiente proyectado.
- 2.3 Si el conjunto de referencia no ha cambiado se para NuevaSolucion=FALSE. Si cambia se vuelve al paso 2.1
-

Capítulo 6

Herramienta LORO

6.1. Análisis

- 6.1.1. Especificación de requisitos
- 6.1.2. Diagramas de flujo de datos
- 6.1.3. Diccionario de datos

6.2. Diseño

- 6.2.1. Diagramas de estructura

6.3. Codificación

- 6.3.1. Interfaz codificación
 - 6.3.2. Modelo matemático
 - 6.3.3. Generador de autovías
 - 6.3.4. Algoritmo de Búsqueda Dispersa
-

En este capítulo se describen el análisis y diseño de la herramienta LORO, siguiendo el paradigma de la *Programación Estructurada*. La elección de este paradigma está condicionada por el lenguaje de programación elegido (MATLAB) y motivada por la necesidad de desarrollar una herramienta modular, que le permita ir desarrollándose integrando nuevas funcionalidades.

6.1. Análisis

6.1.1. Especificación de requisitos

Teniendo en cuenta los objetivos expuestos en 2.1, los requisitos del sistema serán los siguientes:

- Debe implementar el modelo matemático descrito en el capítulo 4 para el caso de una localización preliminar de una red de alineamientos, obteniendo como resultado una localización óptima de la red.
- De los criterios de diseño expuestos en la sección 4.5 del capítulo 4, debe implementar el enfoque más adecuado para el problema de localizar una red de autovías sobre una red de carreteras existentes. El criterio de diseño a implementar es el de *Minimización del tiempo total de transporte en la red, teniendo en cuenta la infraestructura nueva, la existente y sus restricciones presupuestarias*.
- Debe desarrollar una metodología de resolución del modelo matemático propuesto en el capítulo 4.
- Debe ser aplicable a casos reales.
- Debe permitir una entrada y salida de datos amigable para los usuarios, del tipo consultoría de transportes o agencias públicas de planificación del transporte.

6.1.2. Diagramas de flujo de datos

Siguiendo la técnica de análisis estructurado, en esta sección se muestran los *Diagramas de flujo de datos (DFD)* que representan el sistema. Vamos a ver mediante las figuras 6.1, 6.2, 6.3, 6.4, 6.5 y 6.6 la descomposición de los procesos paso a paso.



Figura 6.1: Diagrama de contexto

En este caso, se descompone el proceso Localización de la red de alineamientos, y de los cuatro procesos que lo forman, el **4. Mostrar Salida** es primitivo, simplemente recoge el conjunto de referencia final con las diez mejores redes de alineamientos y muestra la mejor por la interfaz de usuario.

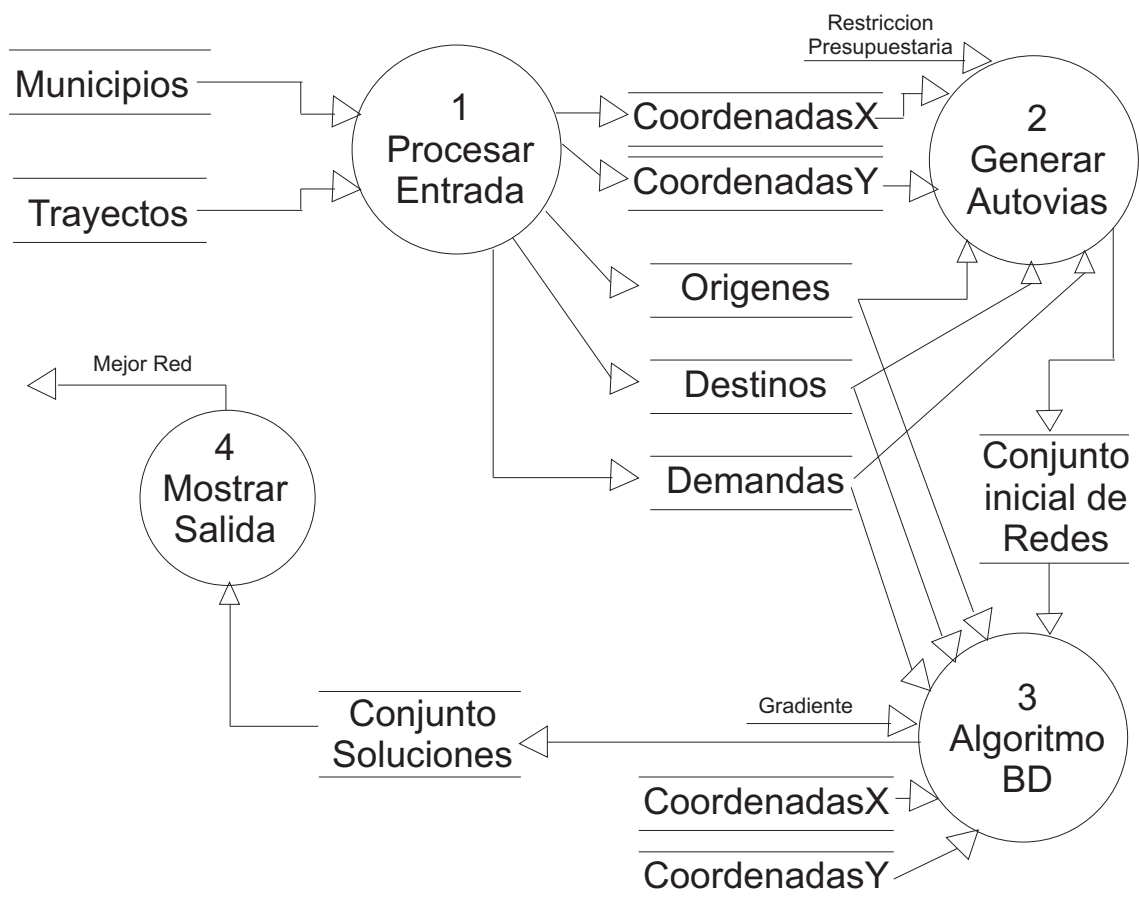


Figura 6.2: Diagrama 0. Localización de la red de alineamientos

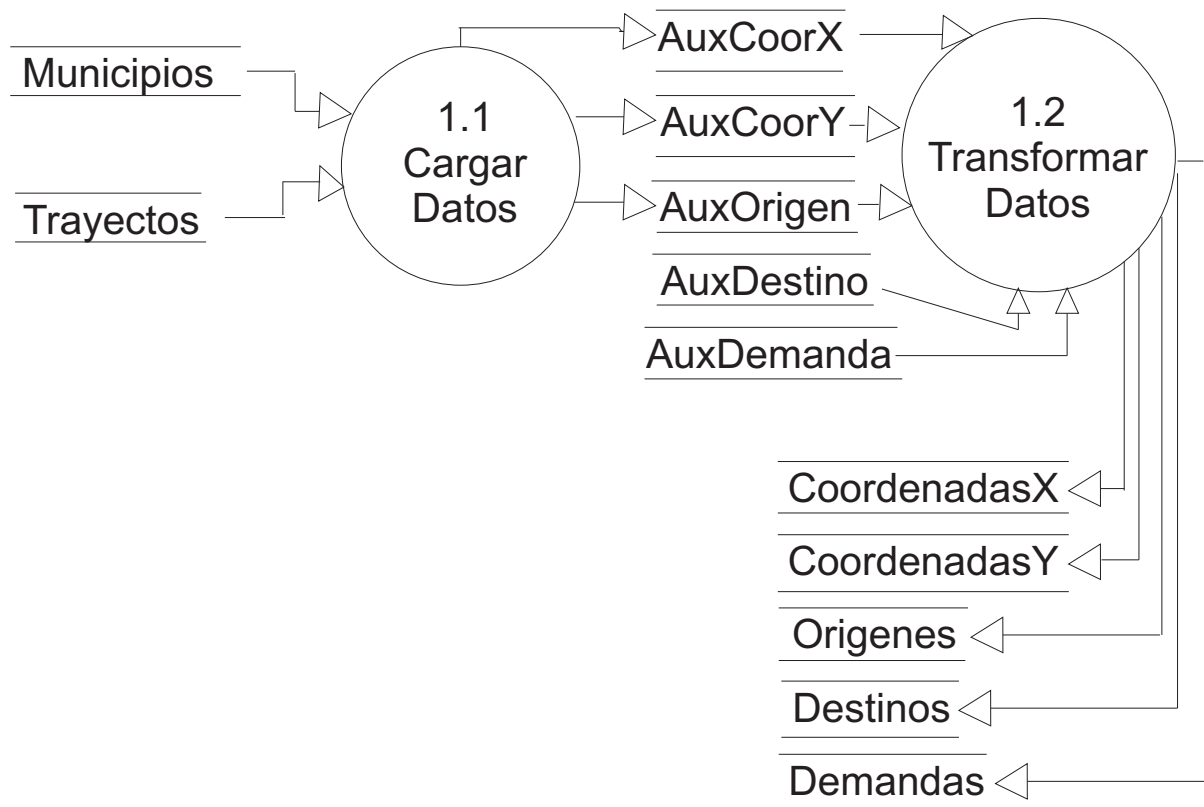


Figura 6.3: Diagrama 1. Procesar entrada

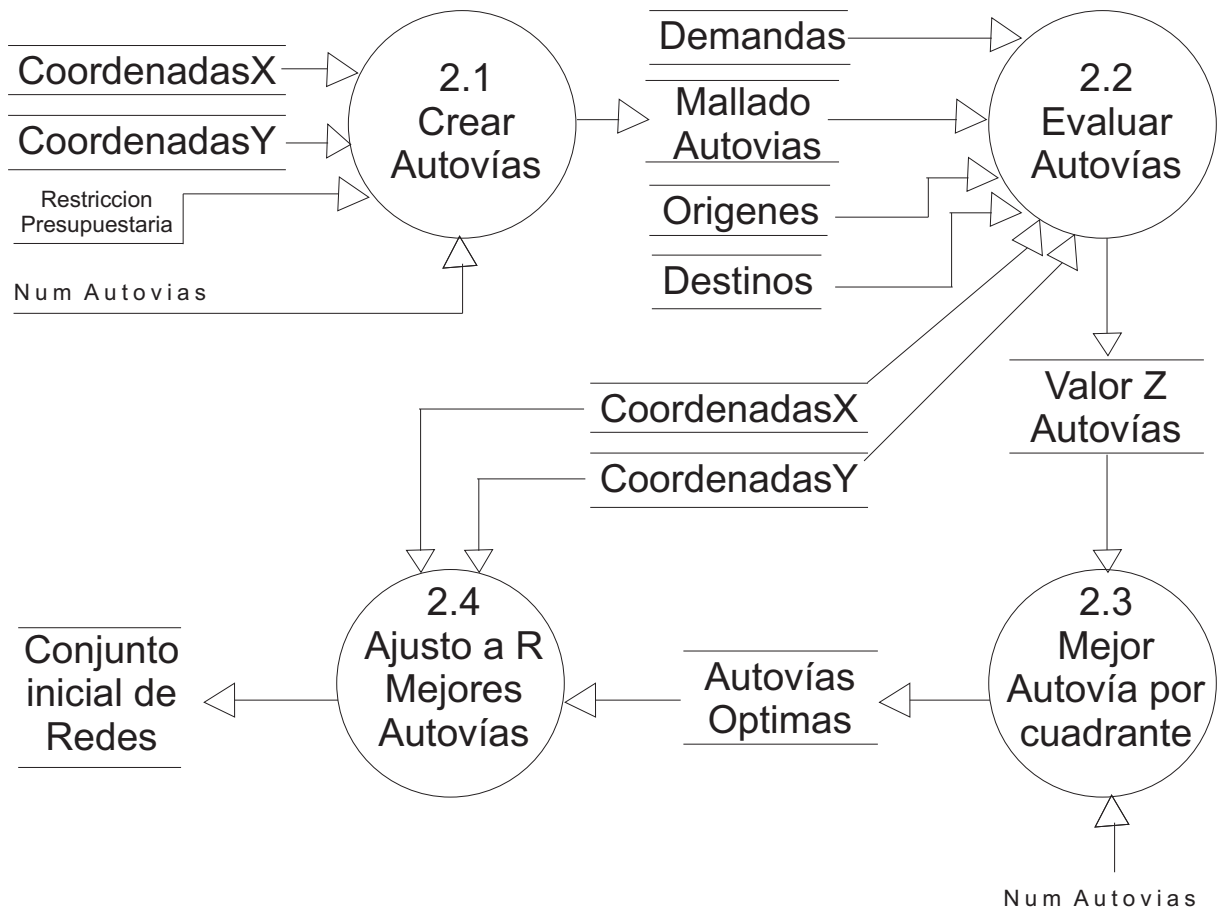


Figura 6.4: Diagrama 2. Generar autovías

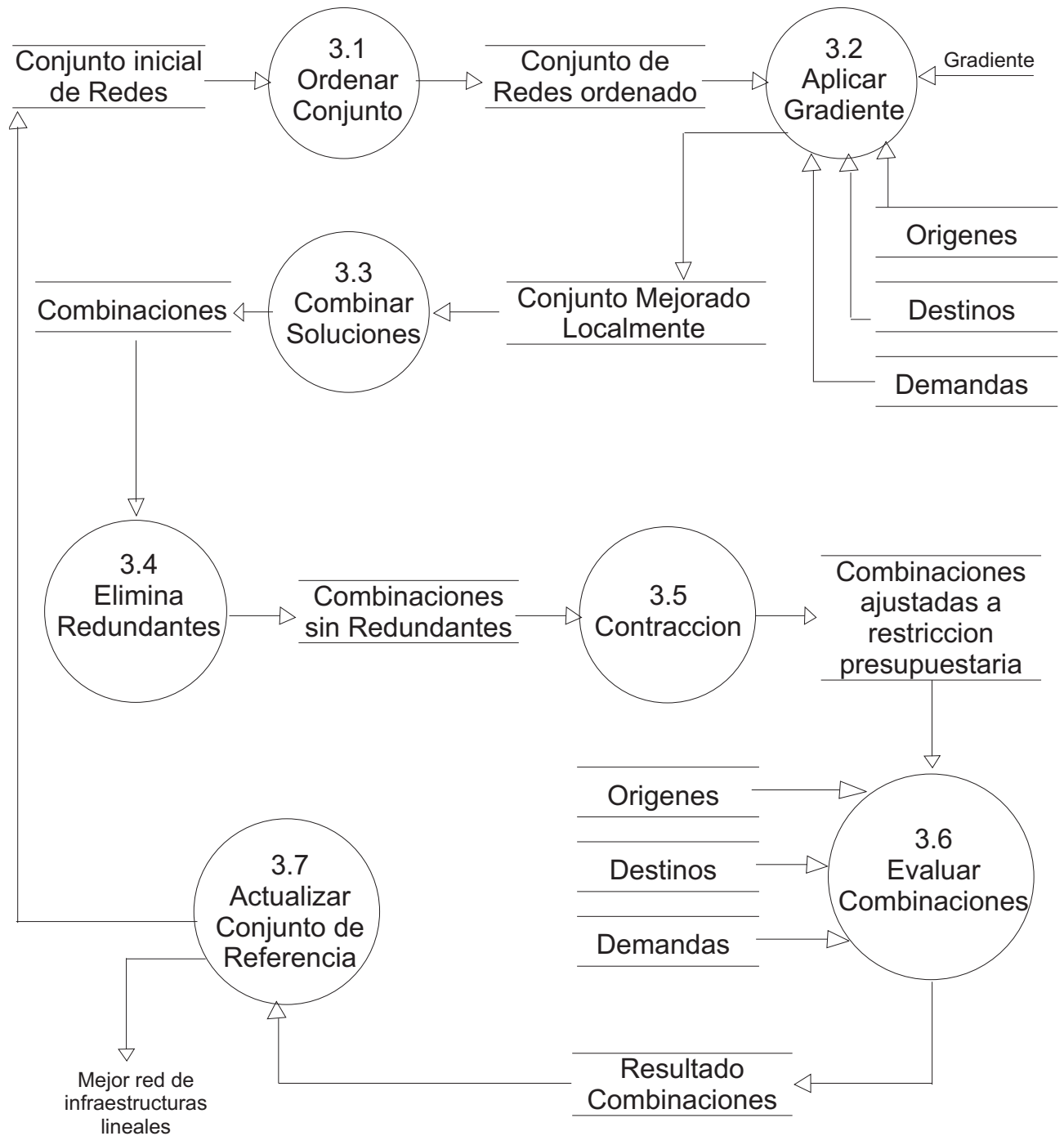


Figura 6.5: Diagrama 3. Algoritmo de Búsqueda Dispersa

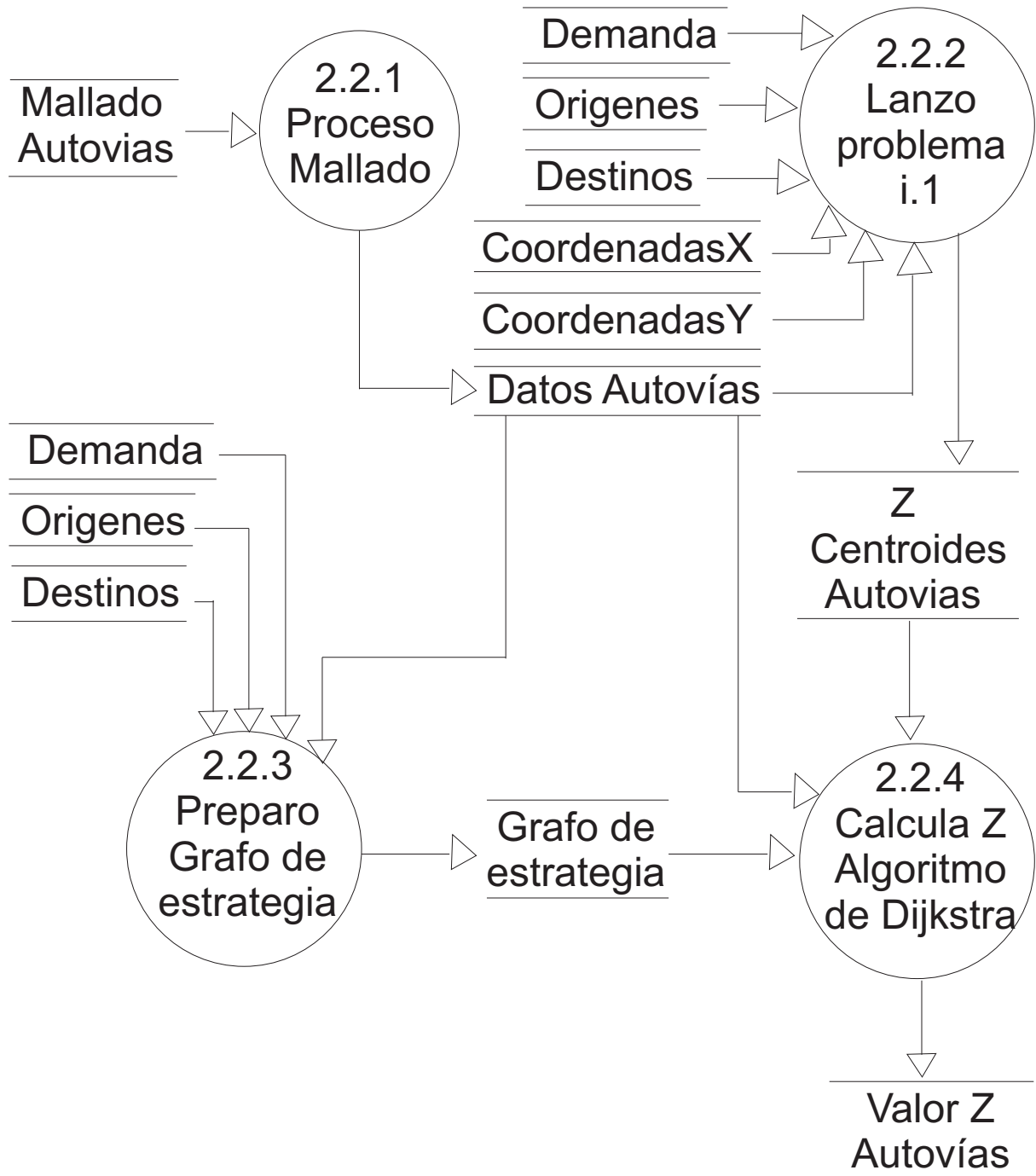


Figura 6.6: Diagrama 2.2. Evaluar Autovías

6.1.3. Diccionario de datos

En este apartado se detallan los almacenes que aparecen en el conjunto de DFDs, explicando los datos que utiliza el sistema.

En las tablas 6.1 y 6.2 se muestra utilizando un enfoque descendente *top-down*, la composición de cada uno de los almacenes de datos.

Tabla 6.1: Diccionario de Datos

Municipios	=	{@id + id_Municipio + coordenadaX + coordenadaY}
Trayectos	=	{@id + id_Origen + id_Destino + Demanda}
AuxCoorX	=	{@id + coordenadaX}
AuxCoorY	=	{@id + coordenadaY}
AuxOrigen	=	{@id + id_Municipio}
AuxDestino	=	{@id + id_Municipio}
AuxDemanda	=	{@id + Demanda}
CoordenadasX	=	{@id + coordenadasX_modificadas}
CoordenadasY	=	{@id + coordenadasY_modificadas}
Origenes	=	{@id + id_Municipio}
Destinos	=	{@id + id_Municipio}
Demandas	=	{@id + Demanda}
Mallado Autovías	=	{@id_Autovia + Coor_X_ini + Coor_Y_ini + Angulo + Longitud}
Datos Autovías	=	{@id_Autovia + Coor_X_ini + Coor_Y_ini + Angulo + Longitud}
Tiempos Centroides Autovías	=	{@id_Municipio + @id_Autovia + Tiempo}
Grafo de estrategia	=	{Origenes + {Destinos} + {Demandas} + {DatosAutovas}}
Valor Z Autovías	=	{@id_Autovia + Z}

Tabla 6.2: Continuación del Diccionario de Datos

Autovías Óptimas = $1\{\text{@id_Autovia} + \text{Coor_X_ini} + \text{Coor_Y_ini} + \text{Angulo} + \text{Longitud} + \text{ValorZ}\}10$

Conjunto inicial de Redes = $\{\text{@id_Autovia} + \text{Coor_X_ini_modificado} + \text{Coor_Y_ini_modificado} + \text{Angulo} + \text{Longitud_modificado} + \text{ValorZ}\}$

Conjunto de Redes ordenado = $\{\text{@id_Autovia} + \text{Coor_X_ini_modificado} + \text{Coor_Y_ini_modificado} + \text{Angulo} + \text{Longitud_modificado} + \text{ValorZ}\}$

Conjunto Mejorado Localmente = $\{\text{@id_Autovia} + \text{Coor_X_ini_modificado} + \text{Coor_Y_ini_modificado} + \text{Angulo} + \text{Longitud_modificado} + \text{ValorZ}\}$

Combinaciones = $\{\text{@id_Combinacion} + \text{ConjuntoMejoradoLocalmente}\}$

Combinaciones sin Redundantes = $\{\text{@id_Combinacion} + \text{ConjuntoMejoradoLocalmenteModificado1}\}$

Conjunto Mejorado Localmente Modificado1 = $\{\text{@id_Autovia} + \text{Coor_X_ini_modificado} + \text{Coor_Y_ini_modificado} + \text{Angulo} + \text{Longitud_modificado} + \text{ValorZ}\}$

Combinaciones ajustadas a restricción = $\{\text{@id_Combinacion} + \text{ConjuntoMejoradoLocalmenteModificado2}\}$

Conjunto Mejorado Localmente Modificado2 = $\{\text{@id_Autovia} + \text{Coor_X_ini_modificado} + \text{Coor_Y_ini_modificado} + \text{Angulo} + \text{Longitud_modificado} + \text{ValorZ}\}$

Resultado Combinaciones = $\{\text{@id_Combinacion} + \text{Z_nuevo}\}$

A continuación en la tabla , mostramos los flujos de datos empleados.

Tabla 6.3: Flujos de Datos

Inf Centroides	→ Datos de todos los centroides (id municipio, coordenada X, coordenada Y)
Inf Trayectos	→ Datos de todos los arcos (origen, destino y demanda).
Restricción Presupuestaria	→ Número de Kilómetros que podemos distribuir por las distintas autovías.
Gradiente	→ Número de iteraciones que vamos a realizar para el Algoritmo del gradiente Proyectado
Num Autovías	→ Número de infraestructuras lineales que tendrán el mallado inicial generado.
Mejor Red	→ Red de infraestructuras lineales óptima para los usuarios en función de la carga inicial de datos.

6.2. Diseño

6.2.1. Diagramas de estructura

En este apartado se muestran los Diagramas de estructura (DEs) del sistema LO-RO. Se ha seguido la estrategia de diseño de Análisis de transformación, a continuación en las Figuras [6.7](#), [6.8](#), [6.9](#), [6.10](#), [6.11](#), y [6.12](#) se observan los diagramas de estructuras.

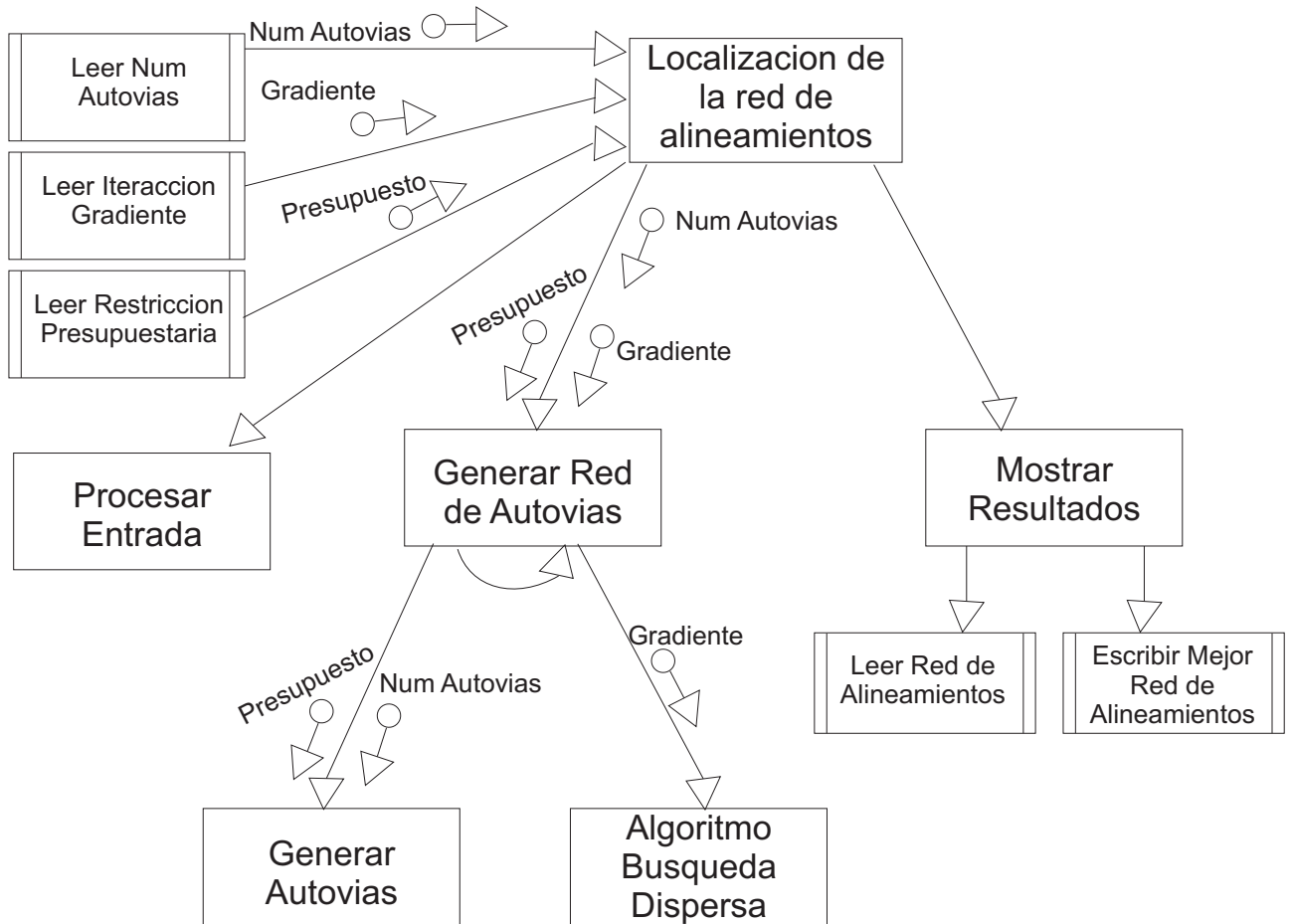


Figura 6.7: Diagrama de estructura

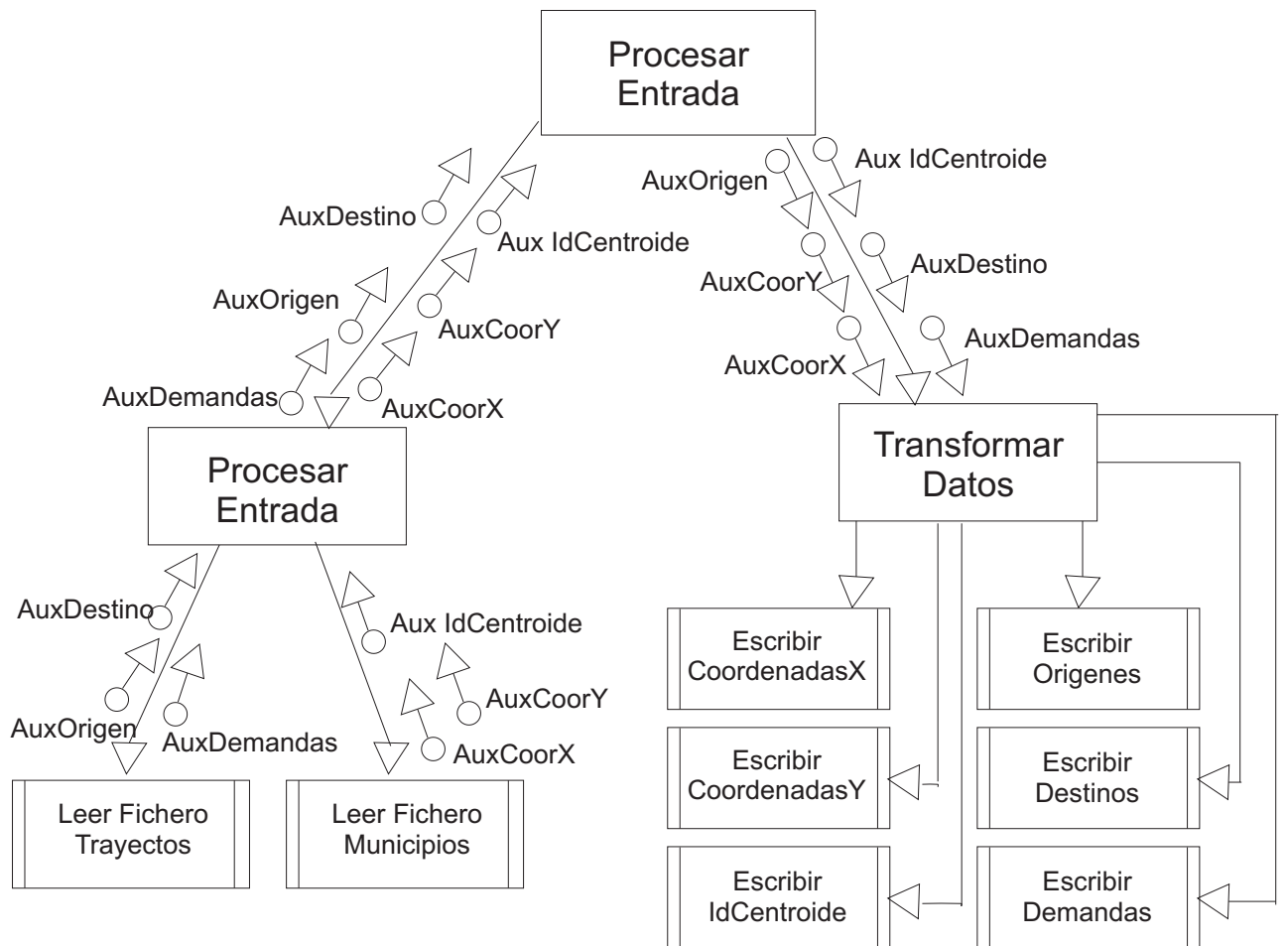


Figura 6.8: Procesar Entrada

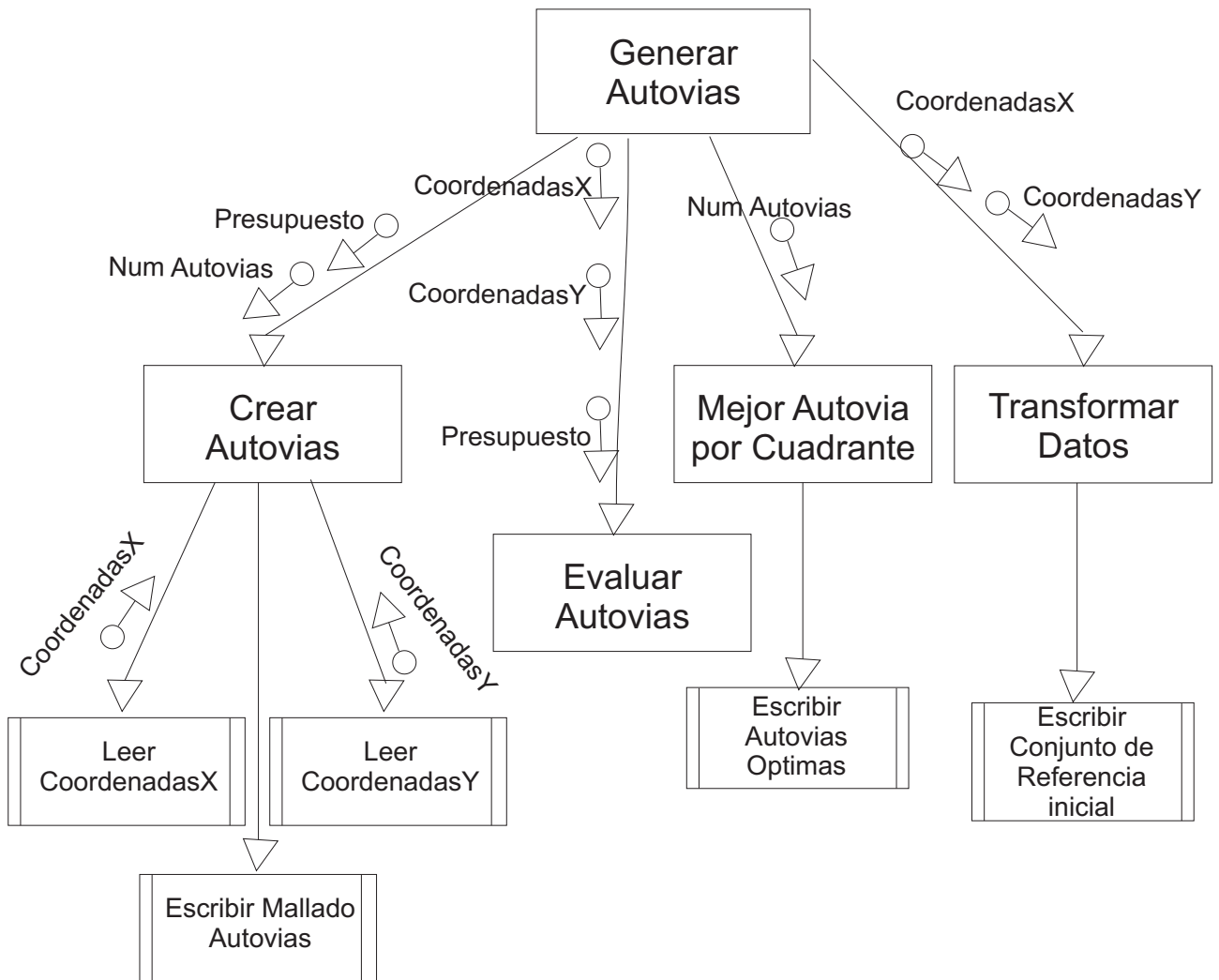


Figura 6.9: Generar Autovías

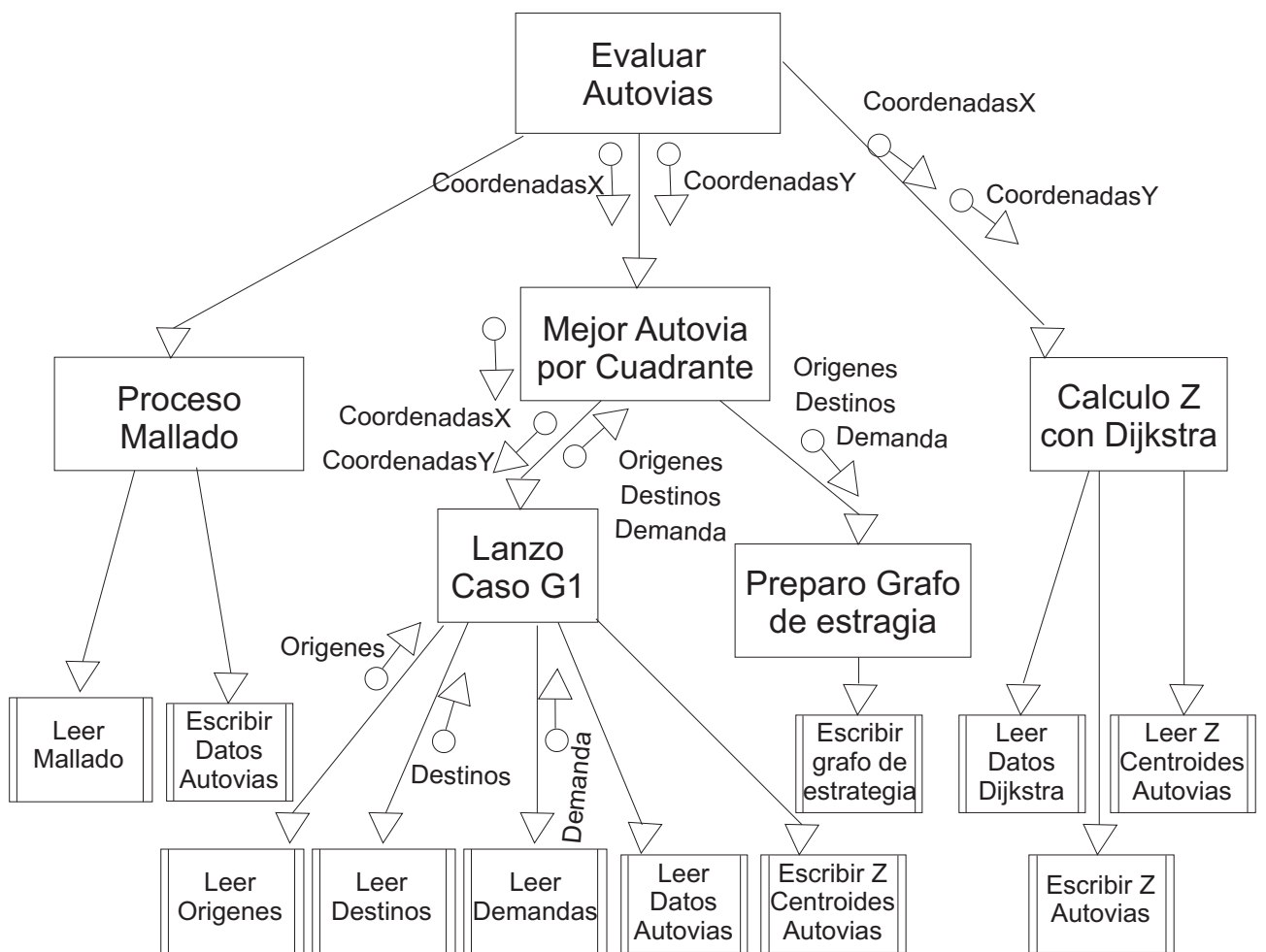


Figura 6.10: Evaluar Autovías

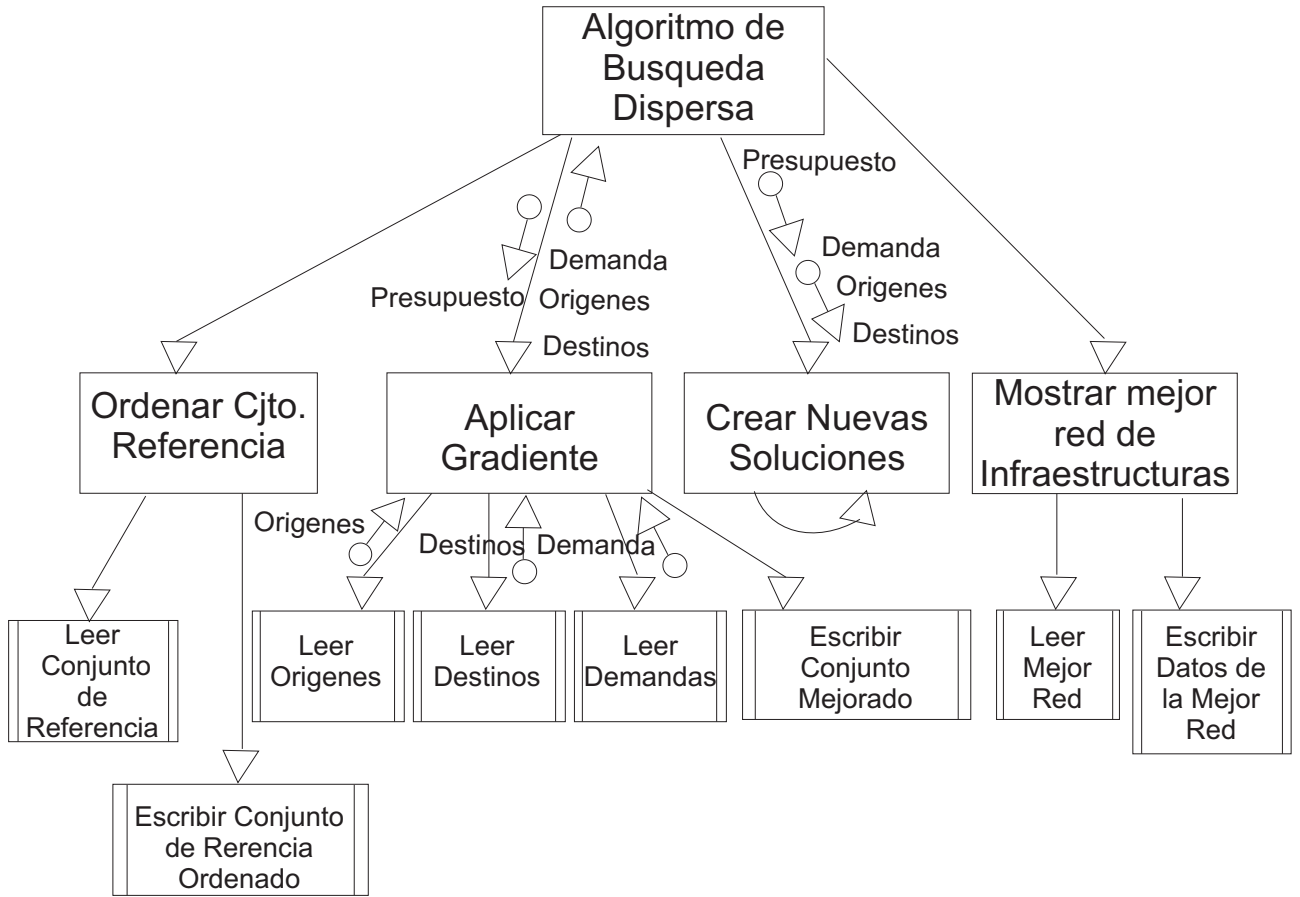


Figura 6.11: Algoritmo de Búsqueda Dispersa

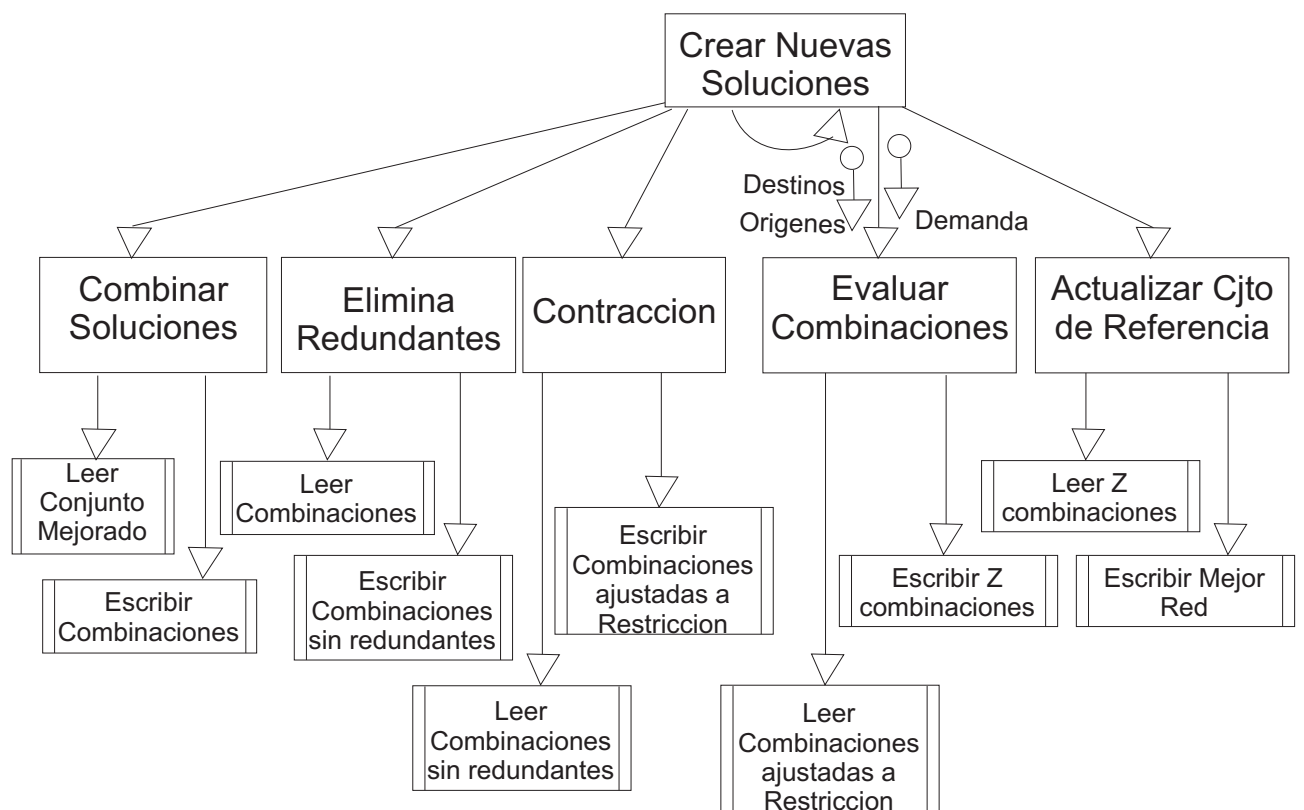


Figura 6.12: Crear Nuevas Soluciones

6.3. Codificación

En este apartado se muestra como se ha construido la herramienta LORO a partir de las fases de análisis y diseño descritas en los apartados anteriores.

6.3.1. Interfaz codificación

La Interfaz de usuario de la herramienta LORO, se ha programado para que el usuario cargue los ficheros de datos que necesita, introduzca los parámetros de entrada para los distintos algoritmos, y muestre la red de infraestructuras lineales generadas para dicha entrada de datos.

Los ficheros que utiliza LORO son los siguientes:

- **Municipios:** Contiene los datos de los centroides (por ejemplo municipios), que formarán el conjunto de centroides.
- **Trayectos:** Contiene traslados entre los centroides del fichero anterior. Formará los arcos del grafo donde se almacenan los datos de ambos archivos.

Estos ficheros están formados por líneas de datos, separados entre sí mediante tabulaciones. La estructura que tendrían cada uno de los ficheros se muestra en las tablas 6.4 y 6.5. Estos ficheros se cargan y procesan mediante la función `ProcesarEntrada`.

Tabla 6.4: Estructura del fichero de entrada Municipios

	código	coordenadasX	coordenadasY
Ejemplo de fila 1	02003	599200	4317149
Ejemplo de fila 2	13034	419450	4315899
Ejemplo de fila 3	13071	403350	4282749
...

Tabla 6.5: Estructura del fichero de entrada Trayectos

	código_origen	código_destino	demanda
Ejemplo de fila 1	45168	02003	52
Ejemplo de fila 2	45165	02003	36
Ejemplo de fila 3	13034	02003	74
...

Los parámetros de entrada serán:

- Restricción presupuestaria: introduzca los Km. de autovías que se pueden construir.
- Gradiente: Para el algoritmo del gradiente proyectado el número iteraciones que queremos hacer para mejorar una autovía localmente. Este parámetro influye sustancialmente en el Tiempo de CPU. El valor 0 indicaría que no se quiere utilizar dicha opción.
- Número de autovías: El número de autovías que vamos a generar en el mallado inicial.
- Epsilon interpolación cuadrática: Parámetro para ajustar la precisión que queremos que tenga la interpolación cuadrática, que calcula el tiempo mínimo entre dos infraestructuras lineales (Problema i.2).

Una vez cargados y procesados todos los datos de entrada, se lanza el algoritmo de optimización y cuando este ha realizado todos los cálculos, la interfaz muestra al usuario una gráfica con los centroides y la red de infraestructuras lineales generada.

La herramienta guarda toda la información de la ejecución en el fichero “Resultados.txt”. Este fichero contiene la información acerca de la ejecución y la red de infraestructuras que se obtiene como solución. El contenido del fichero será la fecha en la que se realiza la ejecución, el tiempo que tarda en realizar la ejecución, el número de centroides, el número de desplazamientos, las autovías generadas en el mallado inicial, el número de soluciones en el conjunto de referencia, el número de iteraciones en el algoritmo del gradiente proyectado, el valor de la función objetivo y los datos de cada una de las infraestructuras lineales (autovías) que forman la red que es la solución.

6.3.2. Modelo matemático

El modelo matemático se ha implementado en varias subrutinas. La primera es **funcion_objetivo_3_Autoviasm** que calcula el valor Z de la función objetivo expuesta en 4.24. Esta subrutina utiliza la función **Calcular_Recorridos_Optimos** que a su vez utiliza las funciones **Caso_G1_centroide_autovia** que calcula el camino mínimo desde un centroide hasta una infraestructura lineal (Problema i.1) y **Algoritmo_Dijkstra** explicado en 4.6. Éste, expande los nodos y utiliza **Caso_G1_autovia_centroide** y **Caso_G2** que calcula el camino mínimo entre dos infraestructuras lineales (Problema i.2).

La estructura de datos empleada son varios vectores de una o varias dimensiones. Elegimos esta opción en lugar de una estructura porque aunque hacen el código menos legible, los cálculos son más rápidos en MATLAB.

6.3.3. Generador de autovías

El parámetro de entrada N indicado en la interfaz de usuario de la herramienta LORO representa el número de autovías que tendrá el mallado inicial de autovías. Además, el generador de Autovías evalúa cada una de las mismas con la subrutina **funcion_objetivo_3_Autovias.m** y selecciona la mejor de cada uno de los b cuadrantes, siendo b el número de soluciones que tendrá el conjunto de referencia. Este parámetro puede variarse en la interfaz de usuario, pero en la literatura científica se recomienda emplear diez soluciones en el conjunto de referencia para el Algoritmo de Búsqueda Dispersa.

El mallado inicial de autovías se genera con la subrutina **generador_autovias.m**. Cada autovía inicialmente tendrá una longitud igual a C , que es el total de kilómetros que podemos construir sin sobrepasar la restricción presupuestaria. Este parámetro también se indica en la entrada de datos de la interfaz de usuario.

6.3.4. Algoritmo de Búsqueda Dispersa

Con el Algoritmo de Búsqueda Dispersa se calcula la red de infraestructuras lineales óptima. Para realizar estos cálculos se utilizan:

- **Burbuja** función que ordena de menor a mayor las soluciones en base al valor Z de cada una de las soluciones.
- **Combinacion_de_Soluciones** función que combina las soluciones del conjunto de referencia dos a dos.
- **Elimina_Redundantes** función que elimina las autovías similares que aparecen al combinar las soluciones, sustituyéndolas por una generada con los extremos más distantes de las dos anteriores.
- **Algoritmo_Gradiente_Proyectado** mediante derivadas aproximadas intenta mejorar localmente cada una de las autovías de las nuevas soluciones que pasan a formar parte del conjunto de referencia.

Para este algoritmo se ha optado de nuevo por utilizar varios vectores de una o más dimensiones en lugar de estructuras, por motivos de rapidez en los cálculos.

Capítulo 7

Pruebas numéricas

7.1. Datos para las pruebas

7.2. Ajuste de parámetros

7.2.1. Parámetros del modelo

7.2.2. Interpolación cuadrática

7.2.3. Parámetros para el algoritmo del gradiente proyectado

7.3. Pruebas y análisis

En este capítulo se exponen los resultados de las pruebas numéricas realizadas con la herramienta LORO.

Las pruebas se han realizado en un ordenador ordenador de sobremesa PC Dell Optiplex GX520 (Pentium 4 a 3.00 GHz) y 1GB de RAM.

7.1. Datos para las pruebas

Las pruebas se han realizado con datos de la región de Castilla-La Mancha. El número de poblaciones de esta región es demasiado elevado para considerarlas todas, con todos sus respectivos desplazamientos, por tanto se han analizado dos situaciones:

- Problema 1. Considera todas las poblaciones de más de 50.000 habitantes.
- Problema 2. Considera todas las poblaciones de más de 5.000 habitantes.

En la tabla 7.1 se muestra para cada problema el número de poblaciones consideradas, el número de pares de demanda origen-destino analizados y el porcentaje de la demanda que representa sobre el total de las novecientas diecinueve poblaciones de Castilla-La Mancha.

Tabla 7.1: Definición de los datos de los problemas.

Problema	Poblaciones	Nº pares $o - d$	%demanda
Problema 1	6	15	27,0
Problema 2	67	2211	65,2

Para el Problema 1, los datos de los municipios se muestran en la tabla 7.2 y los datos acerca de los traslados se pueden ver en la tabla 7.3.

Tabla 7.2: Municipios para el Problema 1

Municipio	CPM	UTM X	UTM Y	Habitantes
Albacete	02003	599200	4317149	159518
Ciudad Real	13034	419450	4315899	69063
Puertollano	13071	403350	4282749	50082
Guadalajara	19130	486401	4498388	73719
Talavera de la Reina	45165	4425152	4317149	82975
Toledo	45168	412492	4412611	75533

Tabla 7.3: Traslados para el Problema 1

	Albacete	Ciudad Real	Puertollano	Guadalajara	Talavera	Toledo
Albacete	—	130	4	19	20	135
Ciudad Real	74	—	451	10	30	183
Puertollano	23	1646	—	11	19	81
Guadalajara	7	7	2	—	0	24
Talavera	36	98	3	13	—	849
Toledo	52	176	5	7	147	—

En esta experiencia computacional, la matriz de demanda origen-destino se ha estimado mediante encuestas, por el Instituto Nacional de Estadística (INE) en el año 2002. La encuesta se realizó a todos los domicilios de España donde se preguntaba en qué población se residía y a qué población se iba a trabajar o estudiar.

Para calcular las distancias entre los distintos centroides, utilizamos la distancia euclídea, será una estimación en metros, ya que los datos de entrada son coordenadas UTM y éstas son estimadas en metros.

Las coordenadas UTM también son conocidas como husos. Lo que hacen es dividir la Tierra en 60 husos y cada zona UTM se divide en 20 bandas. La banda 30 sería (Centro de España y España occidental) y la banda 31 (España oriental e Islas Baleares). A continuación veremos un gráfico establecido por Peter H. Dana que representa la separación para establecer las coordenadas UTM.

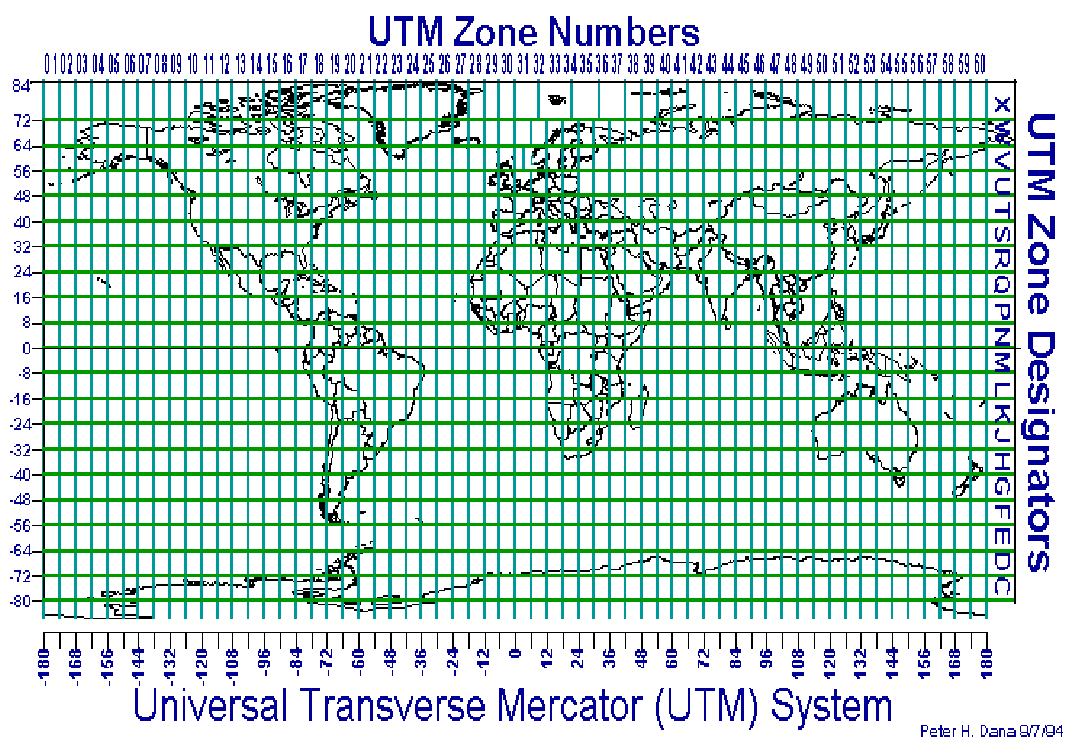


Figura 7.1: Gráfico de Coordenadas UTM

Respecto a las unidades empleadas se ha optado por trabajar con las coordenadas UTM en metros, ya que es su unidad métrica por defecto, las velocidades en Km/h debido a que los conductores están más familiarizados con estas unidades y en minutos para evaluar la función objetivo.

Para obtener más información sobre las coordenadas UTM mirar [15].

La estructura de datos de la matriz origen-destino (ver tabla 7.4, entre capitales de provincia de Castilla La Mancha), se representa mediante un grafo dirigido.

Tabla 7.4: Matriz Demanda Origen-Destino

Origen	Destino	Demanda
Albacete	Cuenca	100
Albacete	Albacete	300
Cuenca	Albacete	50
Ciudad Real	Cuenca	100
Cuenca	Guadalajara	150
Cuenca	Toledo	75

Para ilustrar este procedimiento considérese el ejemplo de la tabla 7.4. Inicialmente se codifica el nombre de los centroides (véase 7.5):

Tabla 7.5: Codificación del nombre de cada centroide

Centroides	Etiqueta
Albacete	1
Ciudad Real	2
Cuenca	3
Guadalajara	4
Toledo	5

La tabla 7.5 quedaría representada en el grafo de la figura 7.2:

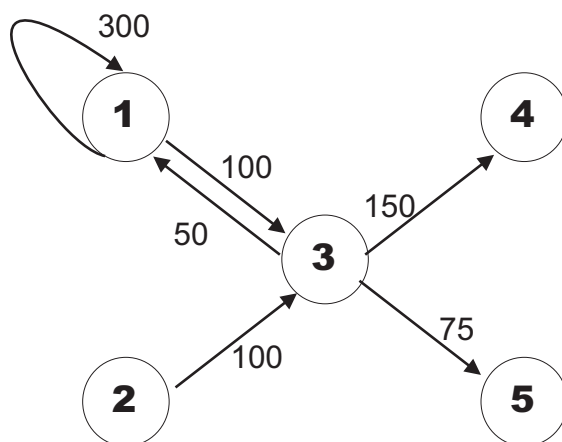


Figura 7.2: Grafo ejemplo a partir de la tabla

Del grafo de la figura 7.2, tenemos que eliminar los pares $w = (o, d)$, $w \in W$ no relevantes para el problema tratado. Estos arcos serán aquellos donde el origen es igual al destino (viajes intrazonales), en este caso de Albacete hasta Albacete.

Por otro lado, el problema es simétrico, es decir el camino mínimo para ir de Ciudad Real hasta Albacete es el mismo que para ir desde Albacete a Ciudad Real y la localización de una infraestructura lineal permite su uso en ambos sentidos. Esto nos permite agrupar los desplazamientos inversos, por ejemplo, de Albacete a Cuenca una demanda 100 y otra de Cuenca hasta Albacete con una demanda 50, sustituiremos ambas demandas por una demanda igual a la suma de las dos anteriores, que en este caso sería de 150. Entre cada par de centroides solo puede haber un arco con la suma de demandas si se desplazan personas en ambos sentidos, ver figura 7.3. Esta operación reduce el tamaño del problema.

El grafo anterior agrupando viajes inversos queda para nuestro problema:

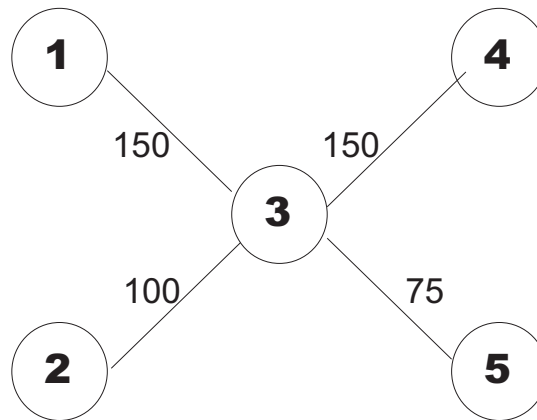


Figura 7.3: Grafo demanda simplificado

7.2. Ajuste de parámetros

En este apartado se ajustan los parámetros para el Problema 1. Una vez ajustados los parámetros, se realizan las pruebas para el Problema 2 que al tener mayor número

de centroides y traslados requiere mucho más tiempo de CPU.

7.2.1. Parámetros del modelo

Conocidos los costes de transportes en la nueva red y en la red existente los usuarios del par $w = (o, d)$, $\omega \in W$ deben decidir que sistema de transporte van a emplear. El modelo asume que los usuarios eligen la red de transporte $m \in \{\mathbf{a}, \mathbf{b}\}$ de mínimo coste. Esto se consigue con $\alpha^m = 0$ y $\beta \rightarrow +\infty$ en la fórmula 4.2, que se considera el caso límite establecido en 4.3.

Luego para calcular los tiempos en los desplazamientos, suponemos que los usuarios pueden acceder a cualquier autovía de las que forman la red, en cualquier punto. Los recorridos por autovía (nueva red) se realizan a una velocidad media de $v = 110Km/h$ y los recorridos por la red existente a una velocidad media de $w = 80Km/h$. Se han tomado dos presupuestos $C = 600Km$ y $C = 300Km$.

7.2.2. Interpolación cuadrática

En el método de interpolación cuadrática descrito en 4.3, utilizamos el parámetro ε para medir la precisión con la que queremos calcular el camino mínimo para el problema i.2). Tomaremos el valor $\varepsilon = 0'1$, porque tras las pruebas para el problema i.2) del capítulo 4 observamos que los resultados en la función objetivo probando con las precisiones $\varepsilon = 0'01$, $\varepsilon = 0'1$ y $\varepsilon = 1$ eran prácticamente los mismos.

7.2.3. Parámetros para el algoritmo del gradiente proyectado

El algoritmo descrito en la sección 5.1, es el método de mejora elegido en el algoritmo de búsqueda dispersa, el procedimiento se repite varias veces hasta que no se

obtiene una mejora significativa.

En este apartado se trata de analizar la convergencia y a partir de ahí, establecer el número de iteraciones que se deben aplicar, para que se produzca una mejora significativa en cada una de las autovías que forman la red.

Se realizarán tres pruebas para la siguiente entrada de datos:

Matriz Demanda Origen-Destino (Castilla La Mancha)

Origen	Destino	Demanda
Cuenca	Albacete	107
Ciudad Real	Albacete	204
Guadalajara	Albacete	7
Toledo	Ciudad Real	176
Guadalajara	Ciudad Real	17
Toledo	Cuenca	175
Guadalajara	Cuenca	32
Toledo	Guadalajara	31

Tabla 7.6: Entrada de datos para las pruebas al método de mejora

Primera prueba con una infraestructura lineal. Tendremos una autovía con los siguientes parámetros:

	Coordenada X inicial	Coordenada Y inicial	Ángulo	Longitud
Autovía1	486401	4484388	4.3524	180000

Tabla 7.7: Tabla con los datos de la autovía para la primera prueba

La autovía de la tabla anterior, se corresponde a la figura 7.4, el valor de la función objetivo expuesta en 4.24 para esta autovía es $Z = 79683,069432$.

Tras la primera iteración el valor de la función objetivo pasa a valor $Z = 79184,464724$. En la siguiente iteración el valor de la función objetivo será $Z = 79185,118613$.

Se observa que el método de mejora converge con rapidez, el criterio de paro que se adoptará es parar cuando la diferencia en valor absoluto entre el valor de Z

óptimo hasta el momento y la Z actual sea menor que un valor constante K , en dicho caso se saldrá del bucle que va mejorando la autovía óptima y devuelve los datos de la autovía hasta dicho instante. Cuando la mejora que obtengamos no compense al coste computacional se para el proceso.

Por tanto en nuestro ejemplo devolverán los valores de la autovía que se muestra en la figura 7.5, que se obtienen tras la primera iteración.

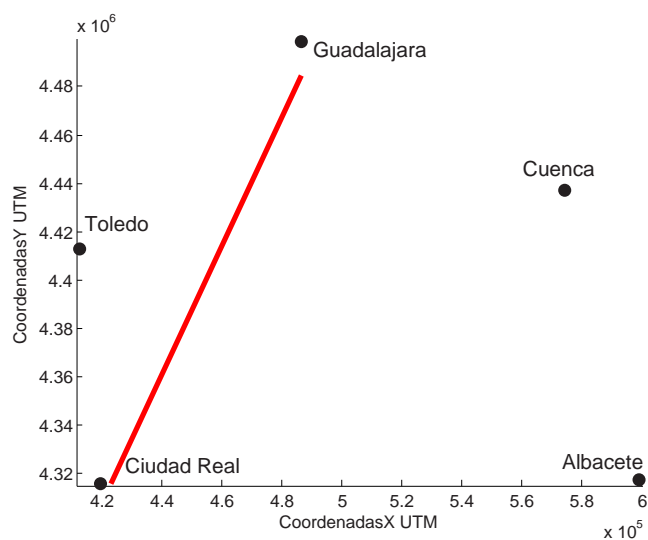


Figura 7.4: Autovía inicial para la prueba 1

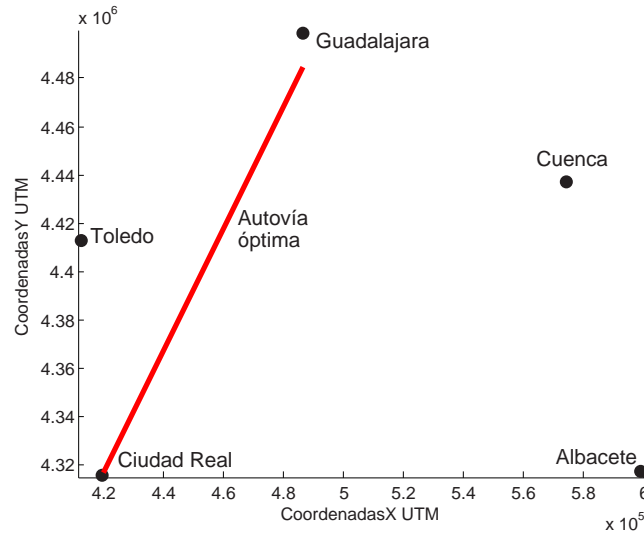


Figura 7.5: Autovía optimizada para la prueba 1

Segunda prueba con tres infraestructuras lineales.

Tendremos tres autopistas con los siguientes parámetros:

	Coordenada X inicial	Coordenada Y inicial	Ángulo	Longitud
Autovía1	486401	4484388	4.3524	180000
Autovía2	415492	4412611	0.1900	140000
Autovía3	412492	4402611	4.7920	80000

Tabla 7.8: Tabla con los datos de las autopistas para la segunda prueba

Las autopistas de la tabla anterior, se corresponde a la figura 7.6, el valor de la función objetivo expuesta en 4.24 para esta autopista es $Z = 71974,320339$.

Tras la segunda iteración el valor de la función objetivo pasa a valer $Z = 71885,530000$. En la siguiente iteración el valor de la función objetivo será $Z = 71840,301355$.

Como la diferencia absoluta entre las dos iteraciones es 45,23 minutos, que es muy pequeña respecto el valor de Z , terminará el proceso por las razones expuestas en la prueba anterior y devolverá los valores de las autopistas que se muestran en la figura 7.7, que se obtienen tras la tercera iteración.

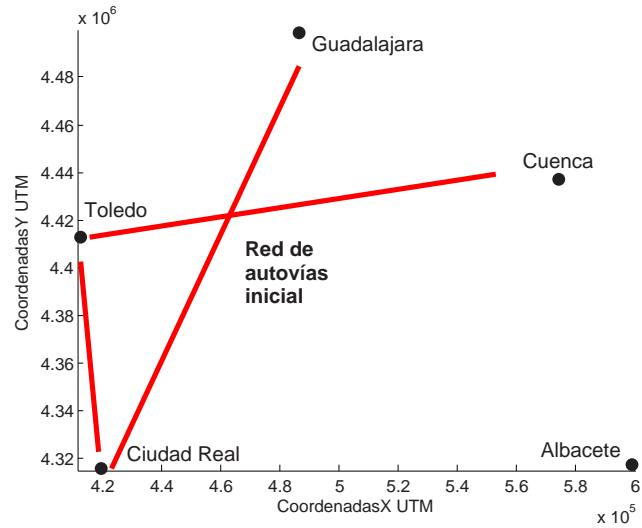


Figura 7.6: Autovías iniciales para la prueba 2

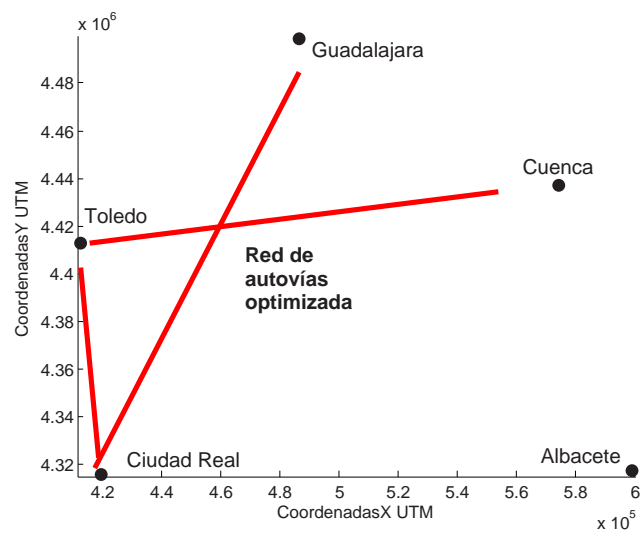


Figura 7.7: Autovías optimizadas para la prueba 2

Tercera prueba con cinco infraestructuras lineales.

Tendremos cinco autovías con los siguientes parámetros:

	Coordenada X inicial	Coordenada Y inicial	Ángulo	Longitud
Autovía1	486401	4484388	4.3524	180000
Autovía2	415492	4412611	0.1900	140000
Autovía3	412492	4402611	4.7920	80000
Autovía4	425450	4315899	0.0400	150000
Autovía5	574160	4416821	4.9020	90000

Tabla 7.9: Tabla con los datos de las autovías para la tercera prueba

La autovías de la tabla anterior, se corresponde a la figura 7.8, el valor de la función objetivo expuesta en 4.24 para esta autovía es $Z = 63889,007009$.

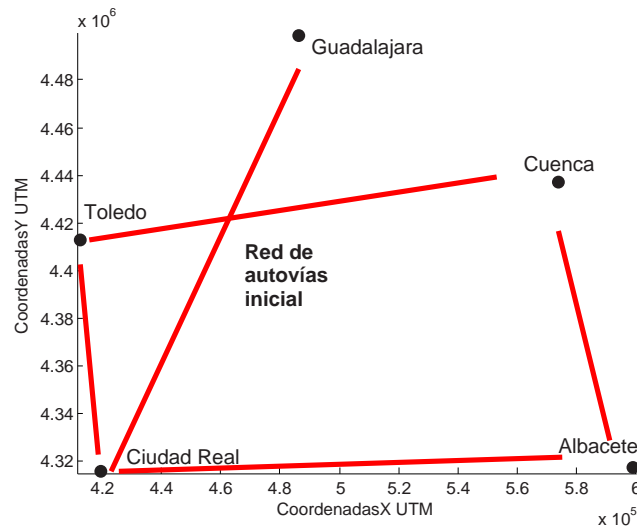


Figura 7.8: Autovías iniciales para la prueba 3

Tras la segunda iteración el valor de la función objetivo pasa a valer $Z = 63684,445093$. En la tercera iteración el valor de la función objetivo es $Z = 63586,689747$. Realiza una nueva iteración con el resultado $Z = 63622,589731$.

Terminará el método de mejora y devolverá los valores de las autovías que se muestran en la figura 7.9, que se obtienen en la tercera iteración. Estos valores son los óptimos de las cuatro iteraciones realizadas.

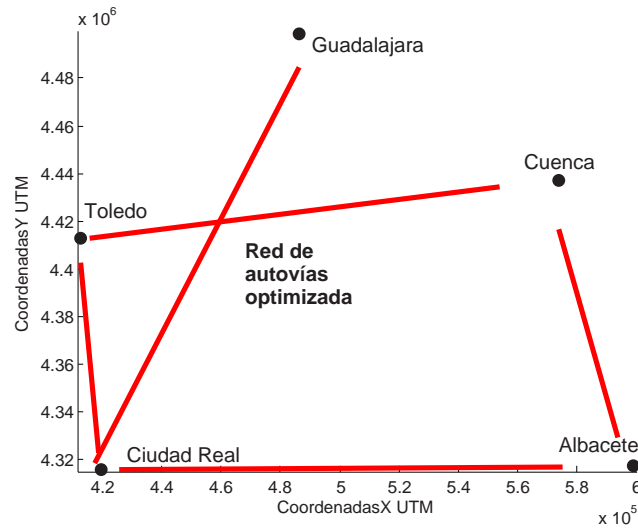


Figura 7.9: Autovías optimizadas para la prueba 3

Este ejemplo de la tabla 7.2.3, muestra que el algoritmo no exhibe una convergencia monótona (mejora la función objetivo en todas las iteraciones) sino oscilante. A priori se podría pensar que el método encuentra una dirección de descenso, ya que se basa en el gradiente, pero como el algoritmo no realiza una búsqueda lineal en dicha dirección, debido al alto coste computacional de esta operación, no se produce el descenso. La figura 7.10, es la representación del resultado de la función objetivo, realizando variaciones de un grado del ángulo en el intervalo $[130^\circ - 157^\circ]$ y manteniendo el resto de variables fijas.

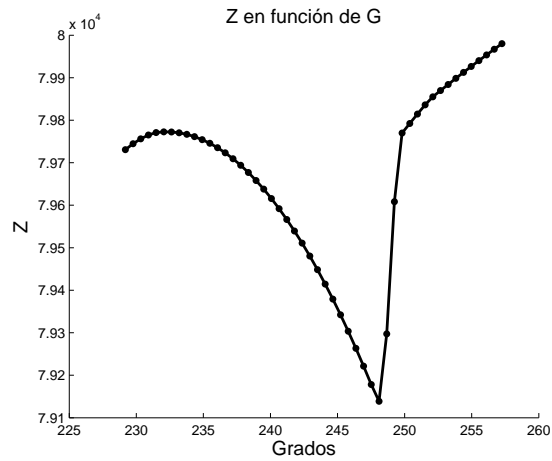


Figura 7.10: Búsqueda Lineal con variaciones de Alfa

La subrutina implementada emplea como criterio de paro un número máximo de iteraciones (G) a realizar. Tras dar las G iteraciones devuelve la mejor solución encontrada.

Tras las pruebas realizadas, se comprueba que el algoritmo del gradiente proyectado tiene un comportamiento oscilante, ver figura 7.11. Tiene una rápida convergencia para conseguir una mejora de la autovía, por tanto tomaremos para las pruebas los valores $G = 5$ y $G = 10$.

Conclusión: Se observa que la función no es convexa y parece no diferenciable. Estas propiedades pueden afectar a la convergencia del método. El método tiene una marcada convergencia local y se mueve siempre en el mismo corredor.

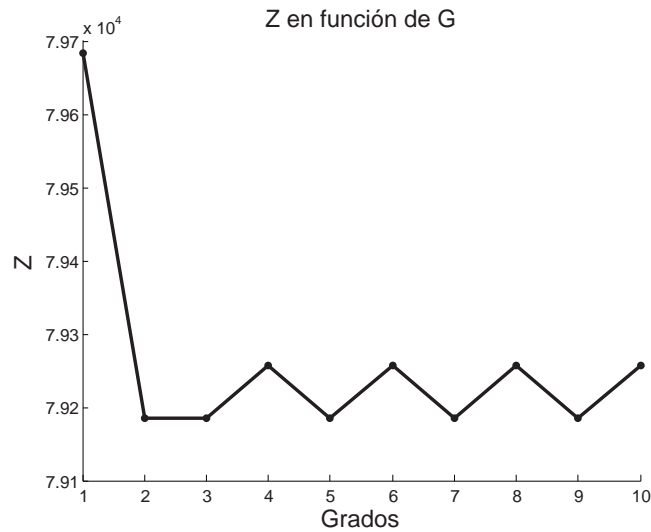


Figura 7.11: Comportamiento de Z

Algoritmo de Búsqueda Dispersa

Para probar este algoritmo es necesario definir los valores de los siguientes parámetros:

1. C : Restricción presupuestaria, kilómetros de autovía que como máximo pueden sumar todos los segmentos de la red. Vamos a realizar pruebas con $C = 600$ y con $C = 300$.
2. b : Número de soluciones que tiene el conjunto de referencia. La bibliografía examinada en el Estado-del-Arte indica que un buen conjunto de referencia es $b = 10$.
3. N : Número de autovías generadas inicialmente para seleccionar las b mejores para el primer conjunto de referencia. Vamos a realizar pruebas con $N = 1,000$, $N = 10,000$ y $N = 100,000$.

Ajustados todos los parámetros de entrada, en la tabla 7.2.3 se muestran los valores de la función objetivo (véase 4.24) Z y el tiempo de ejecución en segundos t_{cpu} , según los valores de los parámetros.

C (km)	b	N	G	ε	Z	t_{cpu} (sg)
600	10	1000	5	0,1	209855,84	46,22
300	10	1000	5	0,1	219047,46	40,5
600	10	10000	5	0,1	205617,15	126,05
300	10	10000	5	0,1	220602,30	102,56
600	10	100000	5	0,1	202211,14	1923,97
300	10	100000	5	0,1	215370,33	1479
600	10	1000	10	0,1	208229,54	73,67
300	10	1000	10	0,1	216660,99	39,97
600	10	10000	10	0,1	205617,67	151,48
300	10	10000	10	0,1	214811,13	273,11
600	10	100000	10	0,1	202211,12	1618,05
300	10	100000	10	0,1	216305,55	1433,36

Tabla 7.10: Tabla con los datos para las pruebas del *Problema 1*

A continuación mostramos la red optimizada que calcula LORO para los casos más interesantes de la tabla anterior:

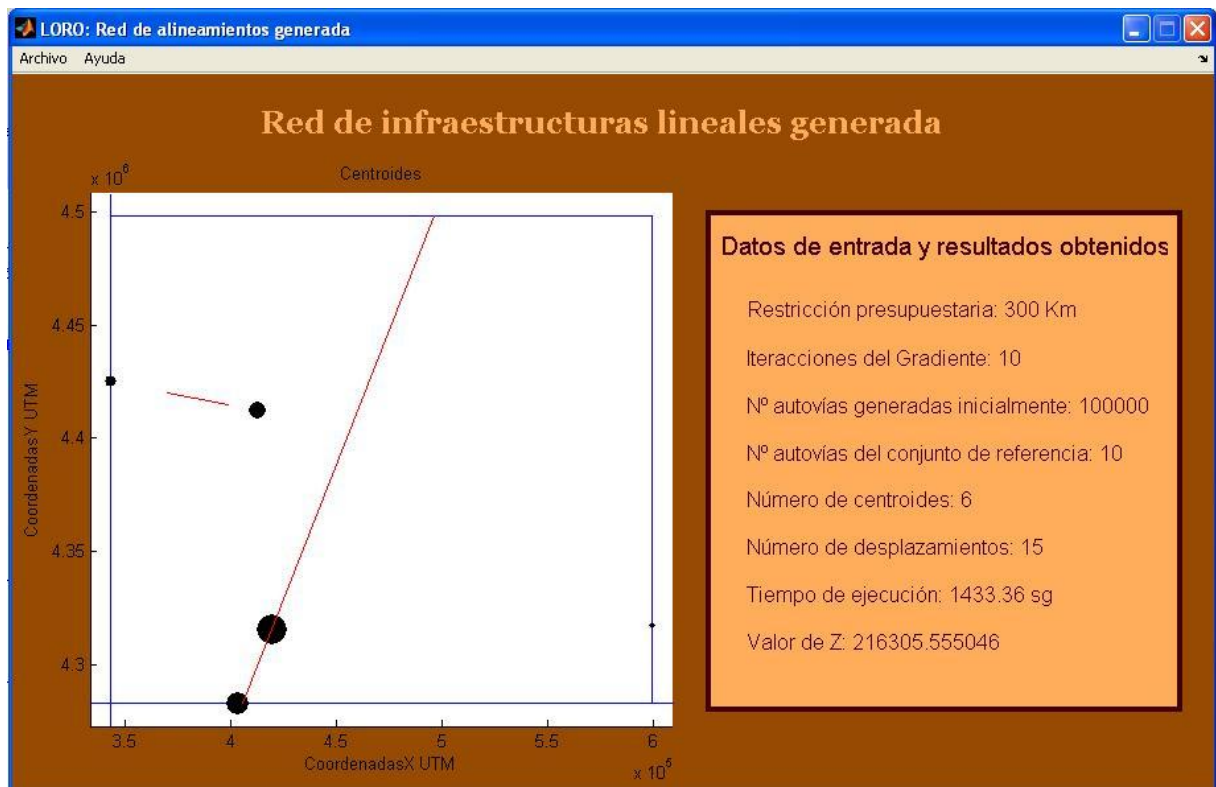


Figura 7.12: Mejor solución con $C = 300$

La figura 7.12, corresponde a la fila 12 de la tabla anterior 7.2.3. Es la prueba que obtiene mejor resultado de la función objetivo con 300 km de restricción.

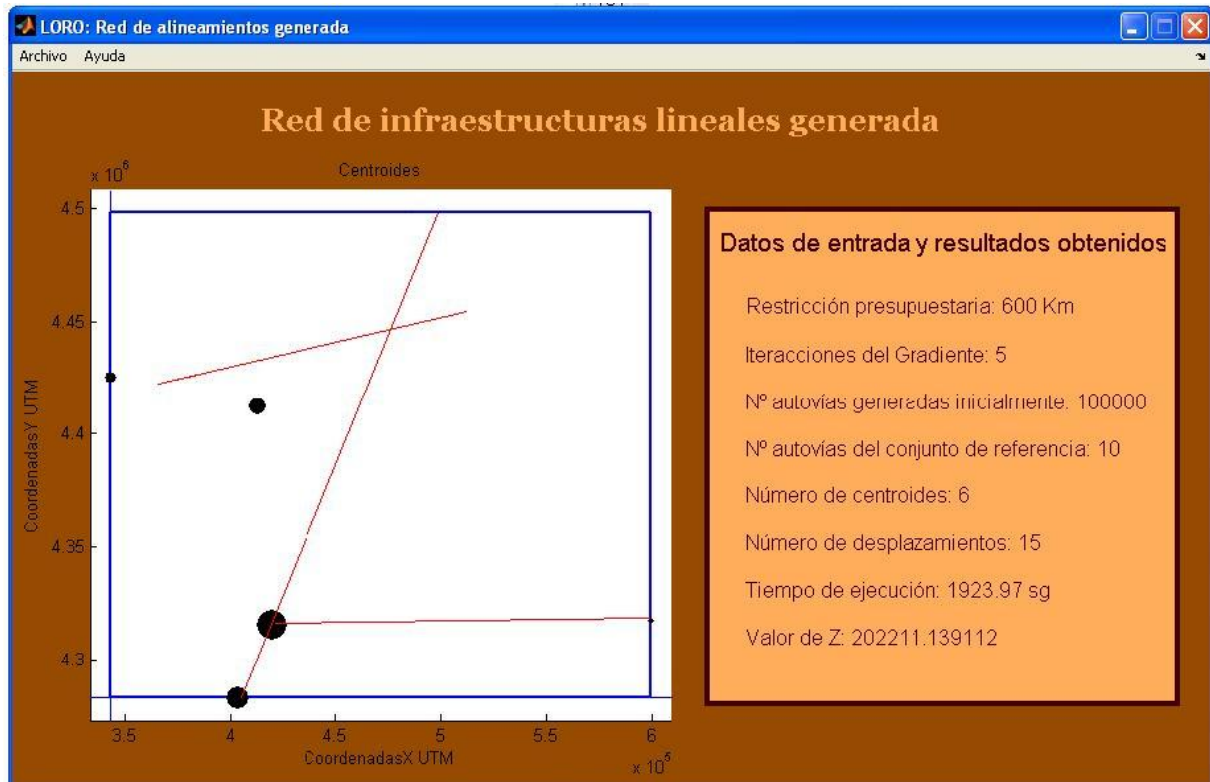


Figura 7.13: Mejor solución con $C = 600$

La figura 7.13, corresponde a la fila 5 de la tabla anterior 7.2.3. Es la prueba que obtiene mejor resultado de la función objetivo con 600 km de restricción. La red de infraestructuras obtenida es la misma que en la fila 11.

A continuación indicamos los valores asignados a los parámetros para el *Problema 2*, basándonos en los resultados obtenidos en la tabla 7.2.3.

Como señalamos anteriormente, el conjunto de referencia será $b = 10$ tal y como se recomienda en el Estado-del-Arte y el margen de error en la interpolación cuadrática ε será menor que 0,1 minutos, ya que no aumenta mucho el tiempo de CPU y obtiene resultados muy próximos al camino mínimo entre dos infraestructuras lineales.

Con respecto al parámetro G apenas influye en los resultados de Z , solo mejora la función objetivo cuando N es igual a 1000 porque con este valor de N no se obtiene un conjunto de referencia próximo a la solución.

Por tanto asignamos $G = 5$ para las pruebas del *Problema 2* y probamos dicho problema con las restricciones $C = 600$ y $C = 300$ para $N = 10000$ y $N = 100000$.

7.3. Pruebas y análisis

En este apartado se resuelve el *Problema 2* para los parámetros asignados en el apartado anterior (ver tabla 7.11).

C (km)	b	N	G	ε	Z	t_{cpu} (sg)
600	10	10000	5	0,1	1510208	5907
300	10	10000	5	0,1	1522246	4497
600	10	100000	5	0,1	1488307	46366
300	10	100000	5	0,1	1538945	41806

Tabla 7.11: Tabla con los datos para las pruebas del *Problema2*

Realizadas las pruebas para el *Problema 1* y el *Problema 2*, se va hacer un estudio de como varían los resultados obtenidos en la función objetivo en función de la variación en la restricción presupuestaria C . Para estas pruebas se cogen los Traslados y Municipios del *Problema 1* con los parámetros de entrada $N = 10000$, $G = 5$, $b = 10$, $\varepsilon = 0,1$ y vamos variando la restricción presupuestaria con valores $C \in [0, 700]$. La tabla 7.12 resume los resultados obtenidos en estas pruebas.

La siguiente figura 7.14, muestra la gráfica que resume la tabla anterior 7.12 con la función objetivo Z en función de C .

Para terminar con el análisis de los resultados obtenidos, vamos a comprobar cuanto vale la función objetivo sin ninguna autovía. En este caso, el tiempo para cada traslado será la distancia euclídea entre las coordenadas origen y destino entre la velocidad

C (km)	Z	t_{cpu} (sg)
0	245866	0,1
100	231444	79,64
200	228714	81,36
300	220602	105,64
400	209863	110,11
500	206647	126,89
600	205796	216,34
700	201966	156,77

Tabla 7.12: Tabla para estudiar la mejora de Z en función del incremento de C

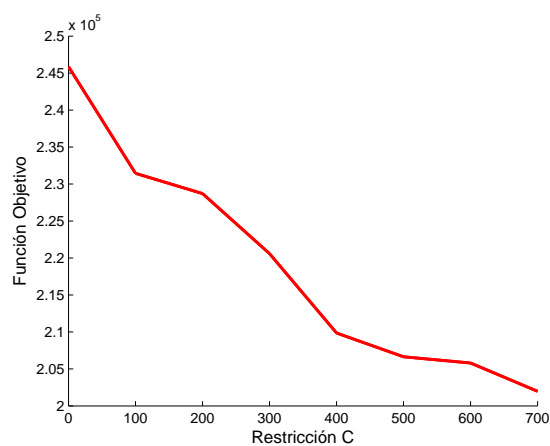


Figura 7.14: Gráfica con los resultados de Z en función de la restricción C

considerada para las carreteras convencionales ($w = 80km/h$).

El resultado para el Problema 1 sin autopistas sería $Z = 245866$ es mucho peor que los mejores resultados de la función objetivo obtenidos con las restricciones de entrada $C = 300$ y $C = 600$, que son 215370 y 202211 respectivamente.

El resultado para el Problema 2 sin autopistas sería $Z = 1642586$ es mucho peor que los mejores resultados de la función objetivo obtenidos con las restricciones de entrada $C = 300$ y $C = 600$, que son 1521945 y 1488307 respectivamente.

Capítulo 8

Conclusiones y propuestas de mejora

8.1. Conclusiones

8.2. Propuestas de mejora

8.1. Conclusiones

Computacionalmente el modelo matemático y el algoritmo de Búsqueda Dispersa son aplicables a problemas de tamaño medio. Para resolver problemas de mayor complejidad (como el caso de Castilla-La Mancha con todas sus poblaciones), se deberá utilizar el servicio de Supercomputación de la UCLM. Se intentó probar el caso de Castilla-La Mancha con todas sus poblaciones y tras más de 70 horas se pararon las pruebas al no obtener resultados.

Para mejorar la eficiencia sería recomendable utilizar una paralelización de procesos, ya que MATLAB lo permite (Stara-p/matlab), y se obtendría la solución de forma más rápida. Por tanto tenemos dos alternativas para mejorar la eficiencia, incorporar el paralelismo de los procesos o utilizar el centro de Supercomputación de la UCLM y con esta mejora, se podrían tratar problemas de mayor tamaño.

Aunque los resultados para problemas complejos se obtienen con grandes tiempos de CPU, estos problemas no necesitan respuesta en un tiempo real, porque los resultados no se requieren de forma inmediata. Una autovía no se hace de forma instantánea son proyectos a largo plazo.

El producto software obtenido en este PFC, está abierto a nuevas mejoras y objetivos que se agregarían al producto, todas las propuestas de mejora expuestas en el siguiente apartado, mejorarían los resultados y las prestaciones del producto.

Una vez analizadas las soluciones, se comprueba que en ocasiones hay tramos de autovía en los extremos que no se utilizan, por tanto se optimizarían los resultados realizando una contracción en todas las autovías del conjunto de referencia desde la incorporación de los usuarios más próxima a las coordenadas iniciales hasta la más próxima a las coordenadas finales en cada una de sus respectivas autovías. Este procedimiento se repetiría en cada iteración del algoritmo de búsqueda dispersa y de esta forma, no se desperdicia ni un kilómetro de autovía.

8.2. Propuestas de mejora

No se contempló en el algoritmo quitar la parte de autovía que no es utilizada en ningún camino óptimo y esta es la razón por la cual se obtienen buenos resultados, pero no los óptimos. Esto es una propuesta de mejora y mejorará los resultados de la función objetivo, consiguiendo que todos los kilómetros de la red final sean recorridos.

Incrementar la eficiencia computacional mediante una paralelización de los procesos. El problema se puede paralelizar mediante un cluster de ordenadores, con b ordenadores y cada ordenador en el caso de generar el mallado de autovías, seleccionaría la mejor de cada cuadrante. Además al combinar las soluciones se repartirían los ordenadores la evaluación de cada combinación. Otra parte que se podría paralelizar es cuando se mejora con el algoritmo del gradiente proyectado cada una de las b soluciones del conjunto de referencia, mejorar una solución en cada ordenador.

Implantar una red de autovías requiere un largo periodo de tiempo, por lo que entre su fase de estudio y la aplicación existe un gran desfase. Asumir una matriz $o - d$ puede conducir a grandes sesgos, porque la demanda de los trayectos puede variar con el paso del tiempo, por tanto, para que la herramienta sea robusta, tiene que hacer frente a distintas matrices verosímiles. Esta es una mejora para ganar robustez, lo mejor es trabajar con varias matrices $o - d$ aleatorias.

Una ampliación de la herramienta sería plantear una planificación táctica y abordar también los alineamientos verticales. Que la herramienta considere puentes, movimientos de tierra, túneles, etc. Como superar todo tipo de obstáculos manteniendo los resultados óptimos de Z en la mayor medida de lo posible.

Por último que la herramienta localice la red de infraestructuras óptima para redes de metro, basándonos en las funciones objetivo 4.22 o 4.23 expuestas en el criterio de diseño en el punto 4.4, ya que es un caso de competencia entre redes donde el objetivo es que la red de metro acapare el mayor número de usuarios. Esta opción se

presenta teóricamente en el modelo y aumentaría las prestaciones de la herramienta. Este supuesto se expone de forma teórica en el modelo matemático del PFC en el capítulo 4.

Apéndice A

Manual de usuario

A.1. Requisitos de instalación

Para que la aplicación funcione correctamente es necesario que el equipo tenga instalado el entorno de programación MATLAB 7.1.

Para instalar la herramienta LORO únicamente tendrá que pegar la carpeta *fuentes* del CD adjunto en el disco duro del ordenador en el que desee ejecutar dicha herramienta.

A.2. Manual de funcionamiento

Para ejecutar LORO hay que ejecutar MATLAB y modificar el directorio actual del programa, indicando como tal el nuevo directorio *fuentes* que ha copiado en su ordenador. Una vez hecho esto, ejecuta el comando LORO y la interfaz de usuario muestra la ventana que se muestra en la figura [A.1](#).

En esta ventana, el usuario deberá cargar los ficheros para los municipios (centroides), pulsando el botón *Municipios* y los trayectos (pares origen-destino) pulsando el botón *Trayectos*.

A continuación el usuario debe introducir correctamente todos los parámetros de entrada:

Restricción presupuestaria en kilómetros C .

Nº iteraciones en algoritmo del gradiente proyectado G .

Número de autovías generadas inicialmente N .

Número de soluciones del conjunto de referencia b .

Una vez que el usuario ha seleccionado los ficheros de entrada y ha introducido

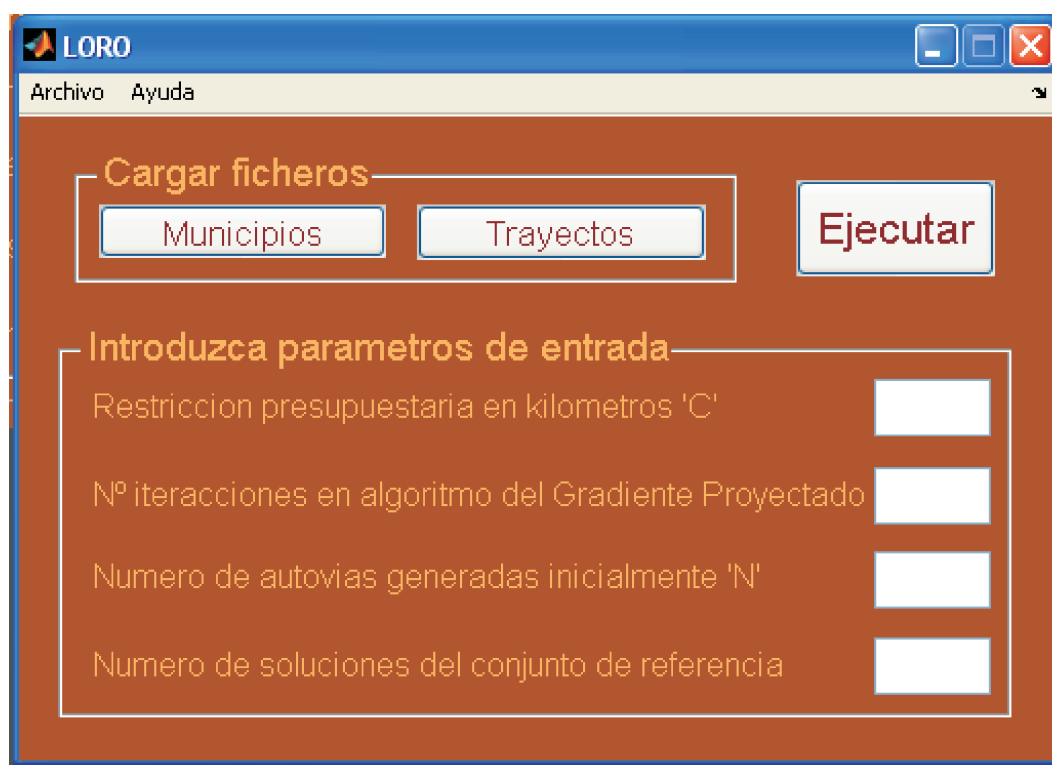


Figura A.1: Ventana principal de la interfaz de usuario

correctamente todos los parámetros de entrada, debe pulsar el botón *Ejecutar* para que comience el algoritmo de optimización.

Si el usuario no selecciona el fichero de los municipios o el fichero de los trayectos, y selecciona el botón *Ejecutar*, aparece la ventana de error (véase figura A.2) indicándolo.

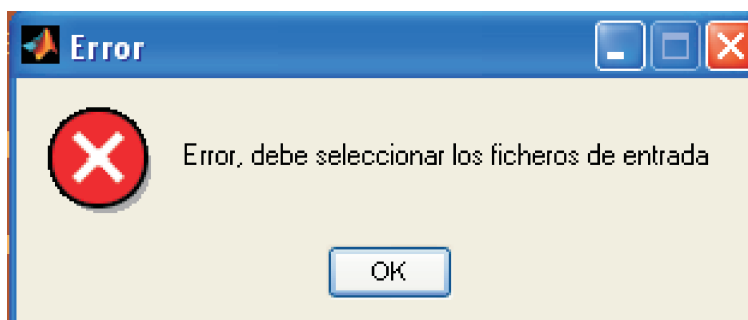


Figura A.2: Ventana de error al no cargar alguno de los ficheros

Si selecciona el botón *Ejecutar* y no ha introducido los parámetros de entrada, aparece la misma ventana de error pero con diferente mensaje.

Cuando el algoritmo de búsqueda dispersa termina su ejecución, aparece la ventana de la figura A.3 con todos los datos sobre la ejecución y una gráfica con la red de autovías generada y todos los municipios con un grosor en función del flujo de entradas y salidas que se produzcan en cada municipio.

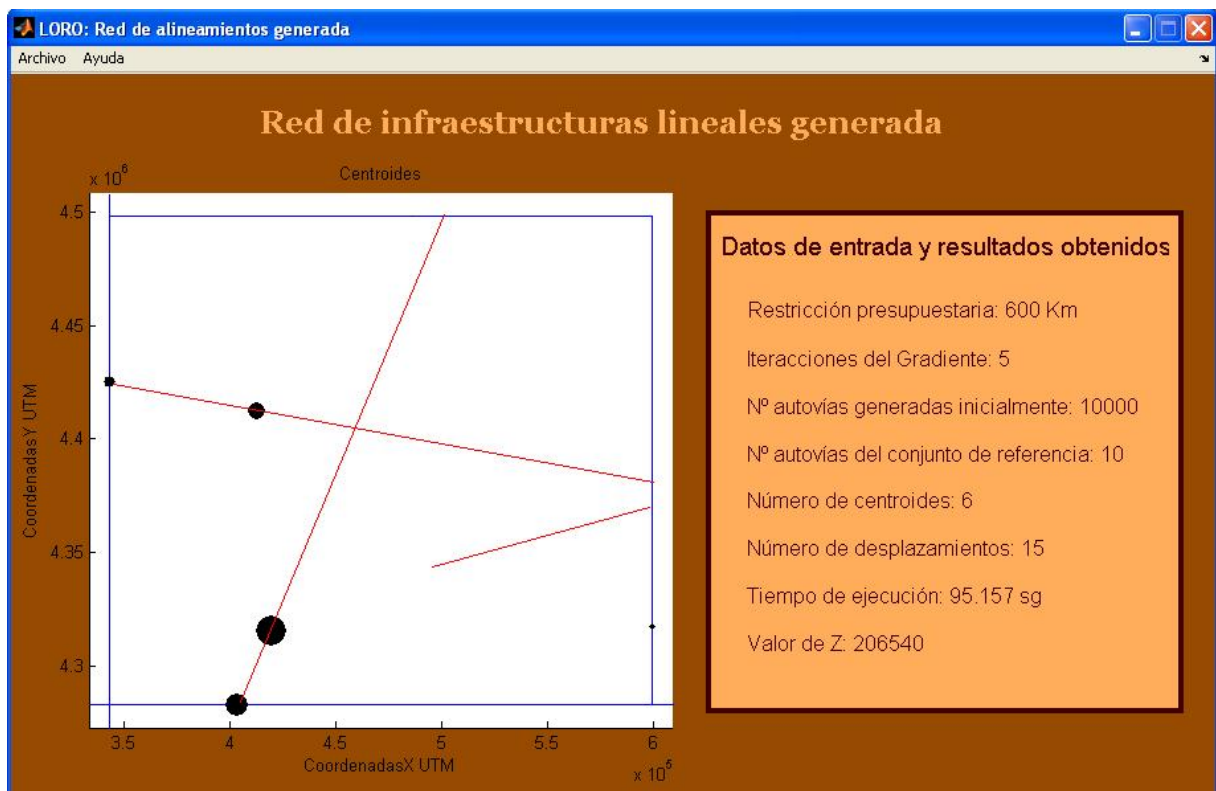


Figura A.3: Ventana que muestra la solución

En el menú de esta ventana tenemos los botones desplegables Archivo y Ayuda. Si pulsa Archivo (ver figura A.4), le aparecen las opciones *Abrir Resultados.txt* y *Cerrar*. Si pulsa *Abrir Resultados.txt*, se abrirá dicho archivo donde se van almacenando todos los datos de cada ejecución. Y si pulsa *Cerrar* se cerrará la ventana.



Figura A.4: Menú desplegable de la solución

Tanto en el menú de la figura A.1 como en el de la figura A.3 aparece la opción ayuda, si el usuario selecciona la misma se muestra la ventana (véase la figura A.5) con información acerca de la herramienta.

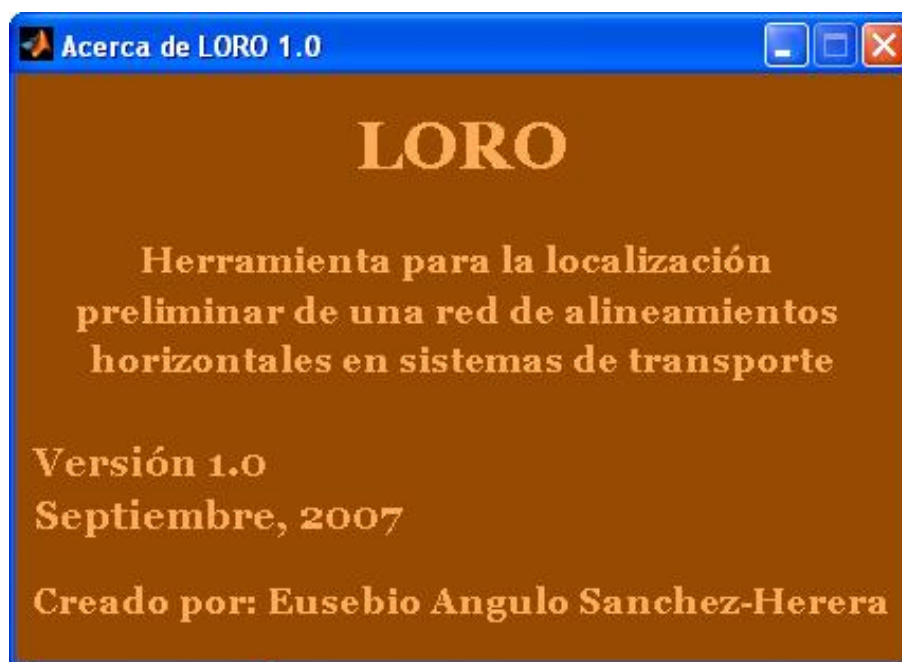


Figura A.5: Ventana de información acerca de la herramienta

Para terminar, cuando el usuario intente cerrar la ventana principal de la herramienta, se muestra una ventana solicitando la confirmación (véase figura A.6).

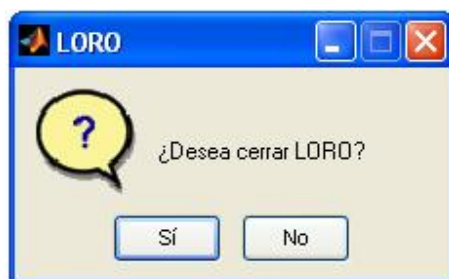


Figura A.6: Ventana de confirmación para salir

Apéndice B

Manual de código

En este capítulo se describe brevemente la estructura del código fuente de la herramienta LORO. Para realizar una consulta del código fuente más detallada consultar el directorio **fuentes** del cd adjunto.

Los ficheros más importantes del código fuente son:

- `leerDatosCentroidesDemandas.m`: Hace una lectura de los ficheros con los datos necesarios y procesa los datos de los mismos.
- `main.m`: función principal que se lanza desde la interfaz de usuario principal de LORO al pulsar el botón *Ejecutar*. Implementa el algoritmo de Búsqueda Dispersa descrito en el capítulo 5, utilizando varias subrutinas como `generador_autovias.m`, `Algoritmo_Gradiente_Proyectado.m`, `Combinacion_de_soluciones.m`, `Elimina_Redundantes.m`, `Contraccion.m`,...
- `Calcular_Recorridos_Optimos.m`: Prepara los datos para Dijkstra y lanza Dijkstra una vez para cada origen.
- `Algoritmo_Dijkstra.m`: Implementación del algoritmo de Dijkstra *modificado*. Para cada origen se va expandiendo el grafo, calculando el peso de sus arcos a la vez que lo recorremos. Decidimos llevar a cabo esta estrategia porque calcular los pesos de los casos problema i.1) y problema i.2) expuestos en el capítulo 4, tiene un alto coste computacional. Los nodos que sean autovías que no se utilizan para los caminos óptimos se evita calcular sus pesos, lo cual es innecesario para nuestro objetivo.

Las distintas posibilidades que podemos encontrar al expandir el grafo y calcular el peso de los arcos son:

- 1 *Los nodos origen y destino son poblaciones*. El tiempo que será el peso del arco que uno los dos nodos, se calcula dividiendo la distancia euclídea de sus coordenadas UTM entre la velocidad que estime el usuario para desplazarse por carretera convencional, es decir sin utilizar la red de autovías. Dicha velocidad sería el parámetro u y sería la estimada por el usuario, siendo

ésta menor que la velocidad de desplazamiento por las autovías. Este caso se denomina enrutamiento directo, ya que los usuarios van directamente al destino sin utilizar la nueva infraestructura.

- 2 *El nodo origen es población y el destino es autovía.* Se calcula el tiempo del arco, que en este caso será lo que tarda desde el origen hasta que se incorpora a la autovía.
 - 3 *Los nodos origen y destino son autovías.* Calcula el camino mínimo y el tiempo es la suma de lo que tarda en el trayecto de la primera autovía que recorre más lo que tarda desde que abandona la primera autovía hasta que se incorpora a la segunda autovía.
 - 4 *El nodo origen es autovía y el destino es población.* Calcula el tiempo del arco como en las anteriores y en este caso será lo que tarda en recorrer el tramo de autovía que recorre más lo que tarda desde que deja la autovía hasta el destino.
- `funcion_objetivo_3_Autovias.m`: Calcula el valor de la función objetivo Z , el valor que se intenta minimizar.
 - `Algoritmo_Gradiente_Proyectado.m`: implementa el método de mejora del algoritmo de Búsqueda Dispersa, expuesto en el capítulo 5.
 - `Combinacion_de_soluciones.m`: Combina dos a dos todas las soluciones y todos sus datos del conjunto de referencia.
 - `Contraccion.m`: Contrae proporcionalmente a su longitud las autovías si no se cumple la restricción.
 - `generador_autovias.m`: Genera un mallado inicial de autovías de las que selecciona las b mejores para inicializar el conjunto de referencia.
 - `Burbuja_R.m` y `Burbuja_Combinaciones.m`: ordenan mediante el método de la Burbuja de menor a mayor todos los datos del conjunto de referencia y de todas las combinaciones respectivamente.

- `Caso_G1_autovia_centroide.m` y `Caso_G1_centroide_autovia.m`: Calculan el camino óptimo problema i.1) uno desde el punto de incorporación a la autovía hasta el centroide destino y el otro desde el centroide origen hasta que se incorpora a la autovía.
- `Caso_G2.m`: Calcula el camino óptimo para el problema i.2).
- `LORO.m`, `LORO.fig`, `mdAcercaDe.m`, `mdAcercaDe.fig`, `mostrarSolucion.m` y `mostrarSolucion.fig`: forman la interfaz de usuario de la herramienta.

Apéndice C

CD

El CD adjunto consta de los siguientes directorios:

1. **doc**: Contiene la documentación del PFC en formato pdf.
2. **fuentes**: Contiene todo el código fuente con todos los ficheros **.m** de la herramienta LORO.
3. **pruebas**: Contiene los datos utilizados para ejecutar las pruebas numéricas del capítulo 7.

Apéndice D

Publicaciones Relacionadas

Artículo: **Localización simultánea de varias infraestructuras lineales de transporte**

Autores: Ricardo García Ródenas, Eusebio Angulo Sánchez-Herrera

Este artículo se presentó en el Simposio: I Jornadas sobre Algoritmos Evolutivos y Metaheurísticas, JAEM'2007. En el Congreso Nacional de Informática (CEDI) 2007.

Bibliografía

Localización preliminar robusta de una red de alineamientos horizontales en sistemas de transporte

- [1] Bruno, G., G. Ghianini, and G. Improta (1998). A multi-modal approach to the location of a rapid transit line. *European Journal of Operational Research* 104, 321–332.
- [2] Bruno, G., M. Gendreau, and G. Laporte (2002). A heuristic for the location of a rapid transit line. *Computers and Operations Research* 29, 1–12.
- [3] Current, J., C. S. ReVelle, and C. Cohon (1985). The maximum covering/shortest path problems: A multiobjective network design and routing formulation. *European Journal of Operational Research* 21, 189–199.
- [4] Dufourd, H., M. Gendreau, and G. Laporte (1996). Locating a transit line using tabu search. *Location Science* 4, 1–19.
- [5] Hamacher, H. W., A. Liebers, A. Schöbel, D. Wagner, and F. Wagner (2001). Locating New Stops in a Railway Network. *Electronic Notes in Theoretical Computer Science*, 50 No I. <http://www.elsevier.nl/locate/entcs/volume50.html>
- [6] Labbe, M., L. G., and Rodríguez-Martín (1998). Path tree and cycle location. In T. Carinic and G. Laporte (Eds.), *Fleet Management and Logistics*, pp. 187–204. Kluwer.

- [7] Laporte, G., A. Marín, J. Mesa, and F. Ortega (2004). An Integrated Methodology for the Rapid Transit Network Design Problem. Accepted to be published in Lectures Notes in Computer Science.
- [8] Laporte, G., J. Mesa, and F. Ortega (1998). Location stations on rapid transit. Technical report, Centre for Research on Transportation, Publication CRT-98-22, Montreal.
- [9] Laporte, G., J. A. Mesa, F. A. Ortega, and I. Sevillano (2005). Maximizing trip coverage in location of a single rapid transit alignment. *Annals of Operations Research* 136, 49–63.
- [10] Marín, A and R. García (2007). *Location of infrastructure in Urban Railway Networks. Computers & Operations Research*.
- [11] Mesa, J. A. and T. B. Boffey (1996). A review of extensive facility location in networks. *European Journal of Operational Research* 95, 592–603.
- [12] <http://www.aeaclub.com/puntosnegros/>
- [13] Glover F., Laguna M., Martí R., Scatter Search and Path Relinking: Advances and Applications, Handbook of MetaHeuristics, 1-36, F.Glover and G. Kochenberger (Eds.),Kluwer, 2003.
- [14] <http://www.uv.es/sestio/TechRep/tr03-04.pdf>
- [15] <http://www.elgps.com/documentos/utm/coordenadasutm.html>
- [16] J.C. Jong, M.K. Jha, D.J. Lovell and P. Schonfeld. Preliminary highway design with genetic algorithms and geographic information Systems. *Computer Aided Civil and Infrastructure Engineering*, 15:231-271,2000.
- [17] M. K. Jha and Paul Schonfeld. A highway alignment optimization model using geographic information systems. *Transportation Research Part A*, 38:339-360,2004.
- [18] M. K. Jha. Criteria-based decision support system for selecting highway alignments. *Journal of Transportation Engineering*, 129:33-41,2003.

- [19] E. Kim, M. K. Jha, D. J. Lovell, and P. Schonfeld. Intersection cost modeling for highway alignment optimization. *Computer-Aided Civil and Infrastructure Engineering*, 125:297-304,2004.
- [20] J.D. Hogan. Experience with optloc - optimum location of highways by computer. PTRC Seminar proceedings on Cost Models and Optimization in Highways, Session L10, 1973.
- [21] D. Trietsch. Comprehensive design of highway networks. *Transportation Science*, 21:26-35,1987.
- [22] E.P. Chew, C.J. Goh, and T.F. Fwa. Simultaneous optimization of horizontal and vertical alignments for highways. *Transportation Research Part B*, 23:314-329,1989.
- [23] E. Kim. Modeling intersections and other structures in highway alignment optimization. PhD thesis, University of Maryland, College Park, 2001.
- [24] Campos V., Laguna M. y Marti R., Scatter Search for the Linear Ordering Problem, *New Ideas in Optimisation*, McGraw-Hill. 331-341,1999.
- [25] **Matlab**
- [26] <http://www.euitt.upm.es/taee06/papers/S1/p46.pdf>
- [27] <http://www.mathworks.es/company/pressroom/articles/article14275.html>
- [28] Matlab Optimization Toolbox User's guide. Version 3. MathWorks Inc, 2004.
- [29] Matlab Genetic Algorithm and Direct Search Toolbox. User's guide Version 1. MathWorks Inc, 2004.
- [30] Using Matlab Graphics. Version 7. MathWorks Inc, 2004.
- [31] Matlab Programming. Version 7. MathWorks Inc, 2004.
- [32] Matlab Mathematics. Version 7. MathWorks Inc, 2004.

- [33] Matlab Desktop Tools and Development Environment. Version 7. MathWorks Inc, 2004.
- [34] Getting Started with Matlab. Version 7. MathWorks Inc, 2004.
- [35] Matlab Optimization Toolbox User´s guide. Version 3. MathWorks Inc, 2004.

LATEX

- [36] <http://www.maths.tcd.ie/~dwilkins/LaTeXPrimer/Matrices.html>
- [37] <http://www.aq.upm.es/Departamentos/Fisica/agmartin/webpublico/latex/FAQ-CervanTeX/FAQ-CervanTeX.txt>
- [38] <http://www.maths.tcd.ie/~dwilkins/LaTeXPrimer/Matrices.html>
- [39] <http://www.fceia.unr.edu.ar/lcc/cdrom/Instalaciones/LaTex/latex.html>
- [40] <http://www.ctan.org/tex-archive/info/epslatex/>
- [41] <http://aristarco.dnsalias.org/recetario>