



UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA
Departamento de Matemáticas

PROYECTO FIN DE CARRERA

“LAHIT: Una herramienta para la localización del alineamiento horizontal de una nueva infraestructura de transporte”

Autor: M^a Carmen Rafael Bastante

Director: Ricardo García Ródenas

Coordinador: Luis Jiménez Linares

Julio, 2007

TRIBUNAL:

Presidente:

Vocal:

Secretario:

FECHA DE DEFENSA:

CALIFICACIÓN:

PRESIDENTE

VOCAL

SECRETARIO

Fdo.

Fdo.

Fdo.

Resumen

El diseño de nuevos sistemas de transporte, como pueden ser redes de autovías, sistemas de metro, etc., requiere localizar en el espacio las nuevas infraestructuras. Este tipo de problemas se puede resolver mediante el uso de modelos de optimización, los cuales emplean una compleja formulación de costes y requieren el desarrollo de algoritmos eficientes. La complejidad de este tipo de problemas hace inviable su resolución mediante métodos exactos de optimización.

Recientemente se ha tratado de dar solución a este problema mediante algoritmos genéticos, ya que estos pueden realizar búsquedas de forma muy eficaz en espacios de soluciones complejos y con gran cantidad de óptimos locales.

En este proyecto se aborda la tarea de desarrollar una metodología, y una herramienta software basada en la misma, para abordar la localización de alineamientos para sistemas de transporte atendiendo al patrón de movilidad de los usuarios, evitando zonas prohibidas y restringiéndose a un presupuesto dado. Se propone un modelo matemático de localización y se desarrolla un algoritmo genético para su resolución.

A mi familia, por su apoyo incondicional.
A mis amigos, por todo lo que significan para mí.

Agradecimientos

A Ricardo García, por su apoyo e implicación en este proyecto, ya que sin él no hubiera sido posible llevarlo a cabo.

A Luis Jiménez, por su interés en este proyecto y su ayuda.

A mis padres, por ayudarme en la revisión de este documento.

A todos mis compañeros y amigos de la escuela, por lo maravillosos que han sido estos cinco años en la Universidad.

Al Grupo de Investigación ORETO, por cederme un espacio en su laboratorio para desarrollar este proyecto.

Al Ministerio de Fomento por la financiación del proyecto de investigación: “*Optimización matemática para la planificación robusta y la extensión de sistemas metropolitanos de transporte público*”, que ha hecho posible el desarrollo de este proyecto.

Índice general

Índice de figuras	XV
Índice de tablas	XVII
1. Introducción	1
1.1. Estructura de este documento	3
2. Objetivos	5
2.1. Objetivos del proyecto	6
2.2. Hipótesis de trabajo	7
2.3. Recursos	8
3. Estado del Arte	9
3.1. Algoritmos Genéticos	10
3.2. Optimización de alineamientos	15
3.2.1. Optimización del alineamiento de autovías	15
3.2.2. Optimización del alineamiento de un sistema de metro	18
3.3. Modelización del problema de localización de curvas en el plano	20
3.4. Aportaciones de este Proyecto Fin de Carrera	25
4. Método de trabajo	29
4.1. Organización de los objetivos	30
4.2. Herramientas utilizadas	30
4.3. Ingeniería del Software	32
4.3.1. Ciclo de vida del Software	32
4.3.2. Análisis y Diseño	32
5. Localización de un alineamiento horizontal	33
5.1. Estructura del modelo	34
5.2. Formulación matemática del modelo	35

5.2.1.	Restricciones del sistema de transporte	36
5.2.2.	Restricciones del comportamiento del usuario	40
5.2.3.	Criterios de diseño	41
5.3.	Consideraciones geométricas sobre la formulación del modelo	43
5.3.1.	Necesidad de la rejilla	43
5.3.2.	Existencia de obstáculos	49
5.3.3.	Problema de caminos mínimos	50
5.3.4.	Accesos al alineamiento	54
6.	Resolución del problema mediante un Algoritmo Genético	57
6.1.	Descripción del algoritmo de resolución	58
6.2.	Descripción del Algoritmo Genético	59
6.2.1.	Representación cromosómica	60
6.2.2.	Población inicial	61
6.2.3.	Medida de evaluación	65
6.2.4.	Criterio de selección de individuos	66
6.2.5.	Operaciones de recombinación y mutación	66
6.2.6.	Estructura del algoritmo	66
6.3.	Operaciones de recombinación	68
6.3.1.	Selección de los padres	68
6.3.2.	Cruce simple	69
6.3.3.	Cruce de dos puntos	71
6.4.	Operaciones de mutación	72
6.4.1.	Mutación uniforme	73
6.4.2.	Mutación recta	74
6.4.3.	Mutación no uniforme	75
6.4.4.	Mutación de incorporaciones	77
7.	Herramienta LAHIT	79
7.1.	Análisis	80
7.1.1.	Especificación de requisitos	80
7.1.2.	Diagramas de flujo de datos	81
7.1.3.	Diccionario de Datos	86
7.2.	Diseño	87
7.2.1.	Diagramas de estructura	87
7.3.	Codificación	93
7.3.1.	Interfaz de entrada y salida	93
7.3.2.	Modelo matemático	94

7.3.3. Resolución geométrica	96
7.3.4. Algoritmo genético	96
7.3.5. Sección Áurea	97
8. Pruebas Numéricas	99
8.1. Ajuste de parámetros	101
8.2. Estudio del coste computacional	106
8.3. Análisis de las soluciones obtenidas	108
9. Conclusiones y propuestas futuras	111
9.1. Conclusiones	112
9.2. Propuestas de mejora	112
A. Manual de Usuario	115
A.1. Requisitos de instalación	116
A.2. Ejecución	116
B. Estructura del Código	121
C. CD adjunto	125
Referencias	129

Índice de figuras

3.1. Tiempo de computación vs. densidad del mapa	16
5.1. Centroides	37
5.2. Matriz origen–destino entre los centroides	37
5.3. Alineamiento	38
5.4. Grafo de movimiento del problema	38
5.5. Poligonal no válida para un alineamiento	44
5.6. Rectas caso $m > 0$	46
5.7. Rectas caso $m < 0$	46
5.8. Caso $m < 0$	48
5.9. Caso $m > 0$	48
5.10. Intervalos prohibidos por un obstáculo	49
5.11. Intervalos permitidos de un segmento	50
5.12. Caminos mínimos entre dos centroides en una vía recta	51
5.13. Caminos mínimos entre un centroide y dos incorporaciones	53
5.14. Caminos mínimos entre dos centroides en una poligonal	53
5.15. Acceso desde un centroide a una incorporación de autovía bordeando un obstáculo	55
5.16. Acceso desde un centroide a una estación de metro empleando la distancia Manhattan	55
6.1. Ejemplo de vías generadas factibles y no factibles	64
6.2. Cruce Simple: Padres	70
6.3. Cruce Simple: Descendientes	70
6.4. Cruce de dos Puntos: Padres	71
6.5. Cruce de dos Puntos: Descendientes	72
6.6. Ejemplo de mutación uniforme	74
6.7. Ejemplo de mutación recta	75
6.8. Ejemplo de mutación no uniforme durante las primeras generaciones	76

6.9. Ejemplo de mutación no uniforme durante las últimas generaciones	77
6.10. Ejemplo de mutación de incorporaciones	78
7.1. Diagrama de contexto	81
7.2. Diagrama 0. Localización de alineamientos	82
7.3. Diagrama 1. Procesar ficheros de entrada	83
7.4. Diagrama 2. Generar rejilla	84
7.5. Diagrama 3. Algoritmo genético	85
7.6. Diagrama de estructura	88
7.7. Procesar ficheros de entrada	89
7.8. Generar rejilla	90
7.9. Algoritmo genético	91
7.10. Crear generaciones	92
8.1. Valores de Z_3 en función del tiempo de ejecución	107
8.2. Valores de Z_3 en función del valor de Θ	108
8.3. Solución al <i>Problema 2</i> con $O1$	108
8.4. Solución al <i>Problema 2</i> con $O2$	109
8.5. Solución al <i>Problema 2</i> con $O3$	109
A.1. Ventana principal de LAHIT	116
A.2. Ayuda contextual	117
A.3. Ventana de error	117
A.4. Ventana de solución	118
A.5. Menú de la ventana de solución	118
A.6. Ventana Acerca de LAHIT	119
A.7. Ventana de confirmación	119

Índice de tablas

3.1. Algoritmo Genético Simple	12
3.2. Estudios acerca de la optimización de autovías	17
3.3. Defectos de los métodos actuales de optimización del alineamiento de autovías	18
5.1. Estructura del modelo matemático	34
5.2. Algoritmo utilizado para calcular las incorporaciones/salidas de un centroide $o \in \mathcal{N}_c$ para el sentido \oplus	52
5.3. Cálculo de los caminos mínimos utilizando las incorporaciones calculadas	53
6.1. Algoritmo de la sección áurea para un problema de maximización	60
6.2. Algoritmo de generación de los cromosomas Λ de la población inicial . . .	63
6.3. Algoritmo de generación de los cromosomas Φ de la población inicial . .	64
6.4. Algoritmo Genético	67
7.1. Almacenes de datos	86
7.2. Flujos de datos	87
7.3. Estructura de los ficheros de entrada	93
8.1. Definición de los problemas	100
8.2. Datos de entrada de los municipios para el <i>Problema 1</i>	101
8.3. Datos de los traslados entre municipios para el <i>Problema 1</i>	102
8.4. Valores de los parámetros relacionados con la geometría del problema . .	103
8.5. Valores de σ , \mathcal{Z}_3 y tiempo de ejecución en función de los parámetros . . .	105
8.6. Resultados de la ejecución del <i>Problema 2</i>	107

Capítulo 1

Introducción

1.1. Estructura de este documento

El diseño de nuevos sistemas de transporte como redes de autovías, sistemas de metro, tranvía etc., requiere la localización en el espacio de las nuevas infraestructuras lineales de modo que se optimice algún criterio de interés social o privado y se satisfaga la demanda de transporte considerada. Dependiendo de la características propias de cada sistema, en unos casos se localizará una red de carretera (autovías), en otros una red de túneles y estaciones o una red de vías ferroviarias.

Todos estos problemas admiten una formulación matemática unificada mediante el uso de modelos de optimización. Estos modelos emplean una compleja formulación de costes y requieren el desarrollo de algoritmos eficientes. La complejidad (no dada explícitamente en algunas ocasiones) de las funciones de coste empleadas limita los algoritmos de búsqueda que se pueden emplear, haciendo inviables métodos exactos de optimización.

Recientemente se han desarrollado algoritmos genéticos para dar solución al problema de la alineación de autovías [GC99, SY99]. Estos algoritmos pueden realizar búsquedas muy eficazmente en espacios de soluciones complejos con gran cantidad de valores óptimos locales. Sin embargo, para resolver problemas reales los algoritmos de optimización deben trabajar directamente con un Sistema de Información Geográfica, que almacena información geográfica relevante. Esto conlleva un elevado tiempo computacional, que aumenta significativamente con el número de propiedades afectadas por cada alineación [JS04].

El desarrollo de una metodología y su concreción en una herramienta software para abordar de forma unificada la localización de alineamientos para sistemas de transporte urbanos o interurbanos atendiendo al patrón de movilidad de los usuarios y considerando “obstáculos” es el objeto sobre el que versa este proyecto fin de carrera.

1.1. Estructura de este documento

Este documento se ha estructurado en 10 capítulos y 3 anexos.

El contenido de los capítulos es el siguiente:

- En el capítulo 1 se incluye una introducción al proyecto fin de carrera y la estructura del resto del documento.
- En el capítulo 2 se enumeran los objetivos que se pretenden conseguir con este proyecto fin de carrera.
- En el capítulo 3 se realiza una revisión del estado del arte en las disciplinas asociadas a los objetivos del proyecto.
- En el capítulo 4 se describe la metodología de trabajo que se sigue en este proyecto.
- En el capítulo 5 se detallan la estructura y la formulación matemática del modelo para la localización preliminar de un alineamiento horizontal. También se explica cómo se ha resuelto geoméricamente el problema, para evitar las vueltas atrás y tener en cuenta los posibles obstáculos, así como para definir qué nodos son incorporación y cuáles no.
- En el capítulo 6 se propone un algoritmo genético para la resolución del anterior modelo matemático para la localización de un alineamiento horizontal.
- En el capítulo 7 se explica la estructura de la herramienta LAHIT, que es la herramienta desarrollada en este proyecto fin de carrera, implementada en base a los desarrollos teóricos de los capítulos 5 y 6.
- En el capítulo 8 se muestran los resultados de la ejecución de la herramienta con problemas reales.
- En el capítulo 9 se exponen las conclusiones obtenidas de la realización de este Proyecto Fin de Carrera y se muestran propuestas para la modificación futura de la herramienta.

- En el capítulo Referencias se detalla la bibliografía utilizada para la realización del proyecto.

El contenido de los anexos es el siguiente:

- El anexo A es un manual de usuario de la herramienta LAHIT.
- El anexo B describe la estructura del código fuente de LAHIT.
- El anexo C es un CD que contiene el código fuente de LAHIT, los datos empleados en las pruebas y este documento en formato PDF.

Capítulo 2

Objetivos

-
- 2.1. Objetivos del proyecto
 - 2.2. Hipótesis de trabajo
 - 2.3. Recursos
-

2.1. Objetivos del proyecto

El objetivo de este proyecto fin de carrera es el desarrollo de una herramienta para una consultoría de ingeniería de transportes que permita el asesoramiento en la toma de decisiones sobre la localización preliminar de una nueva infraestructura lineal de transporte.

Esta herramienta debe permitir una entrada y una salida de datos amigable. Además debe incluir las siguientes características:

1. Debe implementar un modelo matemático para el auxilio en la toma de decisiones sobre la localización preliminar de un alineamiento horizontal. Para ello se deben tener en cuenta los siguientes submodelos y criterios de diseño:
 - 1.1 **Submodelo de demanda.** Los usuarios de sistemas de transporte realizan diversas elecciones a la hora de satisfacer su viaje como son: ruta, modo de transporte, destino, etc. Características de los modelos matemáticos:
 - 1.1.1 Modelización de la elección de ruta
 - 1.1.2 Modelización de la elección del sistema (modo) de transporte. Elasticidad de la demanda.
 - 1.2 **Submodelo de oferta (sistema de transporte)**
 - 1.2.1 Definición del alineamiento horizontal.
 - 1.2.2 Restricciones presupuestarias.
 - 1.2.3 Se asume la creación de un sistema de transporte nuevo (autovía o línea de metro) y la existencia de un sistema de transporte alternativo (red de carreteras secundarias o red de tráfico urbana) que proporciona unos servicios a un determinado coste generalizado de transporte conocido.
 - 1.2.4 Existencia de zonas prohibidas en el plano para la localización de alineamientos, como parques naturales, ciudades, monumentos o zonas geotécnicas inestables.
 - 1.3 **Criterio de diseño.** Se plantean tres objetivos como criterios de diseño de infraestructuras de transporte.

- 1.3.1 Captación del mayor número de usuarios en la nueva infraestructura lineal.
 - 1.3.2 Reducción del tiempo total en la red existente.
 - 1.3.3 Minimización del tiempo total de transporte en el conjunto de redes de transporte.
2. Debe desarrollar una metodología de resolución del modelo propuesto. Para ello se deben tener en cuenta los siguientes aspectos:
 - 2.1 Evaluación computacional eficiente del uso de la red de transporte diseñada.
 - 2.2 Evaluación de la utilidad del sistema de transporte.
 - 2.3 Estudio matemático de la geometría del problema.
 - 2.4 Elaboración de estrategias heurísticas de resolución.
 - 2.5 Pruebas computacionales.
 3. Debe ser aplicable a problemas reales. Para ello se realizará una validación computacional sobre los siguientes problemas:
 - 3.1 Autovía en Castilla–La Mancha.
 - 3.2 Análisis cualitativo de las soluciones obtenidas.

2.2. Hipótesis de trabajo

El trabajo desarrollado en este proyecto fin de carrera tiene su génesis en el Trabajo Académicamente Dirigido “Modelización y resolución numérica del problema de localización de curvas en el plano. Aplicación al diseño de redes suburbanas de transporte privado” [RG06].

El modelo matemático desarrollado en [RG06] es un modelo de optimización no convexo, que posee múltiples óptimos locales. Este modelo no es resoluble en tiempo “finito” mediante métodos exactos y es por esto por lo que en este proyecto se desarrolla un algoritmo genético para resolverlo y se generaliza el anterior modelo.

2.3. Recursos

Este proyecto fin de carrera se ha desarrollado contando con los siguientes recursos humanos, hardware y software:

Recursos humanos

- Autor: M^a Carmen Rafael Bastante
- Director: Ricardo García Ródenas
- Coordinador: Luis Jiménez Linares

Recursos hardware

- Ordenador portátil con procesador Intel Pentium M740 1.73Ghz, 1GB de Memoria RAM y 60GB de disco duro.

Recursos software

- Sistema Operativo Windows XP.
- MATLAB 7.1
- LaTeX para la generación de toda la documentación del proyecto y la presentación, así como las herramientas WinEdt, Gimp y Adobe Acrobat 7.0.
- EasyCase para la generación de los diagramas de flujo de datos y los diagramas de estructura.
- TortoiseSVN para el control de versiones del proyecto.
- Documentación disponible en Internet correspondiente a las distintas herramientas y lenguajes empleados.

Capítulo 3

Estado del Arte

-
- 3.1. Algoritmos Genéticos**
 - 3.2. Optimización de alineamientos**
 - 3.2.1. Optimización del alineamiento de autovías
 - 3.2.2. Optimización del alineamiento de un sistema de metro
 - 3.3. Modelización del problema de localización de curvas en el plano**
 - 3.4. Aportaciones de este Proyecto Fin de Carrera**
-

3.1. Algoritmos Genéticos

En este apartado se describen los algoritmos genéticos basándose en el trabajo [ALM03].

Los algoritmos genéticos fueron introducidos por John Holland en [Hol75], inspirándose en el proceso observado en la evolución natural de los seres vivos. Básicamente, los algoritmos genéticos imitan el proceso de evolución natural.

Algunos de los principios generales de la evolución biológica ampliamente aceptados por la comunidad científica son los siguientes:

- La evolución opera en los *cromosomas*, no en los individuos a los que estos representan.
- La selección natural es el proceso por el que los cromosomas con “buenas estructuras” se reproducen más a menudo que los demás.
- En el proceso de reproducción tiene lugar la evolución mediante la combinación de los cromosomas de los progenitores. Se llama *recombinación* a este proceso en el que se forma el cromosoma del descendiente. También hay que tener en cuenta las mutaciones que pueden alterar dichos códigos.
- La evolución biológica tiene una memoria implícita en el sentido de que en la formación de los cromosomas únicamente se considera la información del periodo anterior, pero la representación n -ploide permite almacenar información que resultó de interés en el pasado de un entorno ambiental distinto.

Este modelo *neo-Darwiniano* de la evolución orgánica se refleja en la estructura de un algoritmo genético. Así, los algoritmos genéticos establecen una analogía entre el conjunto de soluciones de un problema y el conjunto de individuos de una población natural, codificando la información de cada solución en una cadena a modo de cromosoma.

soma. Este cromosoma forma junto con su *adecuación* un “*individuo*” sobre el que el algoritmo aplica sus operaciones.

La estructura que codifica cómo se construye el organismo se denomina *cromosoma*. Un individuo de la población puede tener asociados uno o más cromosomas, que forman su *genotipo*. En la mayor parte de los problemas de optimización combinatoria es suficiente un cromosoma para representar una solución. En este caso, el genotipo y el cromosoma son lo mismo. Los símbolos que forman un cromosoma se conocen como *genes*, y los diferentes valores que puede tomar un gen se denominan *alelos*.

Los algoritmos genéticos están basados en integrar e implementar eficientemente dos ideas fundamentales:

- Las representaciones simples de las soluciones del problema.
- La realización de transformaciones simples para modificar y mejorar estas representaciones.

Para implementar el esquema anterior en un algoritmo, hay que especificar los siguientes elementos:

- Una representación cromosómica de los individuos
- Una población inicial
- Una medida de adecuación
- Un criterio de selección/reemplazo de individuos
- Una o varias operaciones de recombinación
- Una o varias operaciones de mutación

En la tabla 3.1 se muestra un esquema básico de un algoritmo genético. Se puede observar que se trata de un algoritmo estocástico en el que es posible distinguir

las tres etapas clásicas [WC97]: *generación de la muestra inicial, paso de optimización y comprobación de la condición de parada.*

Tabla 3.1: Algoritmo Genético Simple

```

t := 0;
inicializar [P(t)];
evaluar [P(t)];
mientras no terminar hacer
    [P'(t)] := seleccion_pareja [P(t)];
    [P'(t)] := recombinacion [P'(t)];
    evaluar [P'(t)];
    [P(t + 1)] := selecc_entorno [P'(t) ∪ P(t)];
    t := t + 1;
fin mientras

```

En el algoritmo de la figura 3.1, $P(t)$ denota una población de μ individuos en la generación t . El algoritmo genera una nueva población $P'(t)$ de λ individuos aplicando a $P(t)$ un conjunto de operadores de variación.

Los algoritmos genéticos inician la búsqueda a partir de una población de soluciones posibles. La población inicial suele ser generada de forma aleatoria, restringiéndose a soluciones que sean factibles, aunque recientemente se están considerando diseños en los que se utiliza información sobre el problema para generar soluciones de cierta calidad.

La adecuación sirve para asociar un valor de calidad, calculado por la función objetivo $f(x_k)$ del problema de optimización a resolver, para cada solución x_k representada por el individuo k -ésimo de $P'(t)$, con $k \in \{1, 2, \dots, \lambda\}$. Aunque se suele utilizar como medida de la bondad el valor de la función objetivo, se puede añadir un factor de penalización para controlar la infactibilidad en algunos problemas. Este factor pue-

de ser estático o ajustarse dinámicamente, lo que produciría un efecto similar al de la *Oscilación Estratégica* en la Búsqueda Tabú [GL97].

El proceso de selección en un algoritmo genético consta de dos etapas: la primera decide quiénes compiten por la reproducción (emparejamientos) y la segunda decide cuáles de entre todos los individuos, los nuevos y los viejos, van a sobrevivir, simulando así el proceso de selección del entorno.

Ambas etapas de la selección utilizan los valores de adecuación asociados a cada individuo para guiar el proceso hacia soluciones mejores.

Los diferentes tipos de selección utilizan el valor de adecuación directamente (*fitness proportionate*) o bien la relación entre dichos valores (torneo estocástico, *ranking* lineal, etc.). El reemplazo es, en la mayor parte de aplicaciones, elitista, de forma que se asegura la conservación de las μ mejores soluciones presentes en la población actual cuando se va a generar la siguiente población. En el reemplazo se distingue entre técnicas que usan la población actual de μ individuos más la nueva de λ individuos para generar la próxima población, los algoritmos $(\mu + \lambda)$, y los que únicamente usan los λ nuevos individuos para reemplazar a los μ individuos antiguos, los algoritmos (μ, λ) .

En general, un algoritmo genético es un metaheurístico poblacional en el que se pueden mezclar las ideas de combinación de bloques de construcción básicos con la modificación de bloques parcialmente útiles, para avanzar hacia la región más prometedora de la búsqueda. Los operadores de recombinación se aplican con alta probabilidad y usualmente han utilizado uno o varios puntos de cruce en los individuos para intercambiar las porciones resultantes entre los dos padres. También existen variantes que consideran más de dos padres en el proceso de recombinación.

Algunos de los operadores de recombinación mas utilizados son:

- **De un punto:** Se elige aleatoriamente un punto de ruptura en los padres y se intercambian el valor de sus genes.

- De 2 puntos: Se eligen dos puntos de ruptura al azar para intercambiar.
- Uniforme: En cada gen se elige al azar un padre para que contribuya con su gen al del hijo, mientras que el segundo hijo recibe el gen del otro padre.
- PMX, SEX: Son operadores más sofisticados fruto de mezclar y aleatorizar los anteriores.

La mutación produce cambios aleatorios en los descendientes cambiando el valor de los alelos de algunos genes. La mutación tiene el efecto de perturbar un determinado cromosoma para producir nuevas características no presentes en ningún elemento de la población. Las mutaciones más comunes son las que modifican un bit aleatoriamente a baja probabilidad, o las que utilizan un ruido con distribución normal, en el caso de genotipos definidos en espacios continuos.

Dado que el algoritmo genético opera con una población en cada iteración, se espera que el método converja de modo que al final del proceso la población sea muy similar, y en el infinito se reduzca a un sólo individuo. Se ha desarrollado una extensa teoría para estudiar la convergencia de estos algoritmos en el caso de vectores binarios. Esta teoría se basa principalmente en considerar que un individuo es realmente un representante de una clase de equivalencia o **esquema**, reinterpretando la búsqueda en lugar de entre vectores, entre esquemas.

Como ocurre con los otros metaheurísticos, aún cuando existan explicaciones teóricas de funcionamiento, el problema es que en la práctica no se suelen respetar las condiciones necesarias para garantizar la convergencia del método, ya que son difíciles de seguir y probar. Por ello, en ocasiones los algoritmos genéticos resuelven satisfactoriamente un problema de optimización dado y otras se quedan bastante alejados del óptimo.

Por último, mencionar que entre las técnicas más prometedoras para el avance en estos metaheurísticos y su aplicación a problemas del mundo real, se pueden destacar

la inclusión de técnicas paralelas y descentralizadas, en las que la población se divide con algún criterio de entorno y se hace evolucionar de forma separada a zonas distintas dentro de la tradicional población única [AT02].

3.2. Optimización de alineamientos

3.2.1. Optimización del alineamiento de autovías

La optimización del alineamiento de autovías implica encontrar el mejor trazado entre un par de puntos [JJLS00, Jha03, KJLS04, JS04]. El problema puede enunciarse como sigue:

Dados dos extremos en el área estudiada y permitiendo que las condiciones existentes en el estudio del área sean variables, encontrar el mejor alineamiento entre las alternativas para optimizar una función objetivo dada, considerando las estructuras necesarias y satisfaciendo requisitos de diseño y operacionales.

Para aplicaciones más fiables y realistas, los procesos de optimización de alineamientos deben considerar muchos factores, lo que aumenta la complejidad del problema [KJS05]. Los factores pueden incluir estructuras, topografía, factores socio-económicos, ecología, geología, tipos de terreno, patrones de uso del terreno, medio ambiente e incluso preocupaciones de la sociedad. Estos factores son considerados con diferentes énfasis y niveles de detalle en distintas fases de los procesos de selección de alineamientos. Tradicionalmente, estos procesos han consumido mucho tiempo y esfuerzo de agencias, planificadores, ingenieros y residentes. Se han desarrollado varios modelos para dar solución a esta necesidad. Dichos modelos pueden ahorrar mucho tiempo y esfuerzo, comparados con los métodos manuales tradicionales.

Recientemente se ha desarrollado una solución [JS03] basada en algoritmos genéticos para alineamientos de autovías en tres dimensiones (alineamientos verticales). Las ventajas de los algoritmos genéticos sobre los métodos tradicionales para optimizar alineamientos de autovías están cubiertas extensamente en [Jon98] y [JS03].

Aunque el primer objetivo de los modelos desarrollados es obtener el mejor alineamiento (óptimo global o al menos cerca del óptimo global), la eficiencia computacional del modelo es una gran preocupación, ya que afecta en gran medida el grado de aplicabilidad del modelo. La carga computacional aumenta especialmente [Jha02] cuando el número de propiedades analizadas aumentan. La figura 3.1 muestra los resultados de un experimento llevado a cabo en [Jha02] con un Pentium II a 450 MHz y 96MB de memoria RAM. Dicho experimento mostró que mientras que la carga computacional aumentaba linealmente hasta una densidad del mapa de 10,000, la variación no era lineal cuando la densidad del mapa excedía de 10,000. Aunque se espera que la carga computacional se alivie un poco cuando se usan ordenadores más rápidos, ésta sigue siendo una preocupación cuando es necesario analizar miles elementos geográficos.

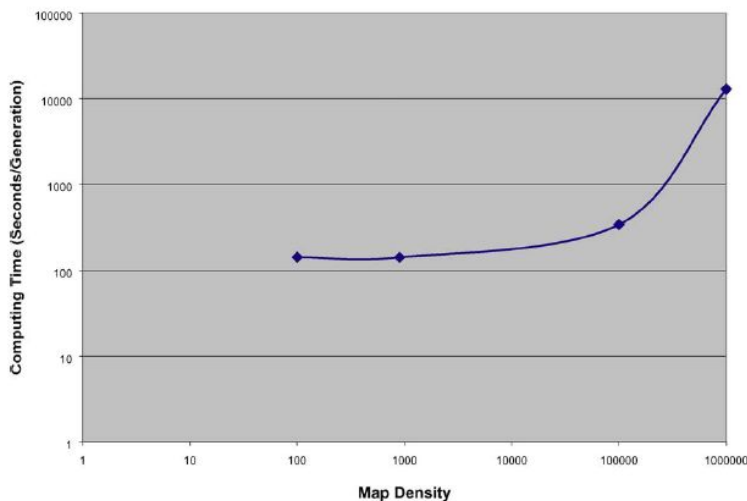


Figura 3.1: Tiempo de computación vs. densidad del mapa

Teóricamente, el problema de la optimización del alineamiento de autovías implica evaluar un número infinito de alternativas. En aplicaciones previas [JJLS00, JS03,

JS04], el problema de optimización era formulado como un problema de minimización del coste en el que las funciones de coste eran no-diferenciables, con ruido e implícitas. De esta manera, es inevitable usar algoritmos de búsqueda rápidos y eficientes para resolver dicho problema.

De acuerdo con la tabla 3.2, se han usado siete métodos de búsqueda para modelos de optimización del alineamiento. Dichos métodos de búsqueda tienen algunos defectos críticos a la hora de ser aplicados a problemas de optimización del alineamiento de autovías, excepto los algoritmos genéticos. Se puede ver un resumen de dichos defectos en la tabla 3.3.

Tabla 3.2: Estudios acerca de la optimización de autovías

Objetivo de la optimización	Enfoques	Referencias
Alineamiento horizontal	Cálculo de variaciones	[Wan95, HBS68, TS88, SH81, SH82]
	Optimización de redes	[OEC73, TM71, AC73, Par77, Tri87a, Tri87b]
	Programación dinámica Algoritmos Genéticos	[Hog73, NEW76] [Jon98]
Alineamiento vertical	Enumeración	[Eas98]
	Programación dinámica	[Puy73, Mur73, GCF88, Fwa89]
	Programación lineal	[RWW97, CC88]
	Búsqueda numérica	[Hay70, GCF88, Rob73, Fwa89, OEC73]
	Algoritmos genéticos	[Jon98]
Alineamiento horizontal y vertical simultáneo	Programación dinámica	[Hog73, NEW76]
	Búsqueda numérica	[CGF89]
	Optimización en dos etapas	[Par77, Tri87a]
	Algoritmos genéticos	[Jon98, Jha00, Kim01]

Tabla 3.3: Defectos de los métodos actuales de optimización del alineamiento de autovías

Métodos	Defectos
Cálculo de variaciones	<ul style="list-style-type: none"> • Requiere funciones objetivo derivables • No es apto para factores discontinuos • Tendencia a quedarse atrapado en óptimos locales
Optimización de redes	<ul style="list-style-type: none"> • Las salidas no son suaves • No es válido para espacios de búsqueda continuos
Programación dinámica	<ul style="list-style-type: none"> • Las salidas no son suaves • No es apto para espacios de búsqueda continuos • No es aplicable a funciones implícitas • Requiere independencia entre subproblemas
Enumeración	<ul style="list-style-type: none"> • No es apto para espacios de búsqueda continuos • Es ineficiente
Programación lineal	<ul style="list-style-type: none"> • No es apta para funciones de coste no lineales • Sólo cubre un número limitado de puntos para restricciones de gradientes y curvaturas
Búsqueda numérica	<ul style="list-style-type: none"> • Tendencia a quedarse atrapado en óptimos locales • Modelado complejo • Dificultad para manejar elementos de coste discontinuos

3.2.2. Optimización del alineamiento de un sistema de metro

El diseño de sistemas de ferrocarriles urbanos como metro, tranvías, cercanías, etc. consta de etapas sucesivas que se desarrollan a lo largo de un período dilatado de tiempo. Se puede abordar bajo tres horizontes temporales:

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

- **Planificación táctica.** En esta fase se define 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.

Los principales aspectos de la planificación estratégica que se han estudiado en la literatura son:

- a) La localización de los alineamientos.
- b) La determinación del número de estaciones y su localización.

Un modelo para el problema a) ha sido desarrollado en [MB96] y [LLRM98], donde se formula el problema como un problema de localización de caminos en un grafo. En [LMO98] se estudia el problema b), teniendo en cuenta los viajes de tipo *park'n ride*. En [CRC85] se aborda el problema de diseño de la red maximizando la demanda 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 esta carencia es el dado en [BGI98], donde la elección del modo de transporte se tiene en cuenta mediante el mecanismo de asignación *todo-o-nada*, bajo el cual los usuarios emplean la red que les proporcione los menores costes de transporte. Recientemente, en [LMOS05] y [LMMO04] se introduce un modelo de demanda logit para modelar la elección del sistema de transporte.

Las mayorías 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 se obtiene en [HLS⁺01], que muestra que el problema de localizar la estaciones sobre una línea dada es un problema *NP-duro*. Se han planteado algoritmos

heurísticos para resolver estos modelos de localización de alineamientos. En [DGL96] se aplica un algoritmo de búsqueda tabú. Los algoritmos de [BGL02] y [LMOS05] realizan búsquedas locales tipo *Greedy* para determinar estaciones compatibles con el conjunto de restricciones. En [MG07] se aborda el problema del 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.

Como conclusión todos los modelos anteriores se basan en técnicas de optimización de grafos y no existe una metodología de resolución aplicable a problemas reales.

3.3. Modelización del problema de localización de curvas en el plano

En el Trabajo Académicamente Dirigido “*Modelización y resolución numérica del problema de localización de curvas en el plano. Aplicación al diseño de redes suburbanas de transporte privado*” [RG06] se desarrolla un modelo matemático para localizar una autovía teniendo en cuenta el comportamiento de los usuarios de la red y una matriz de viaje origen–destino que define el patrón de movilidad.

Para resolver el problema hay que tener en cuenta los siguientes aspectos:

- **Aspectos de la demanda.** La elasticidad de la demanda en función de los costes/-tiempo de viaje, la elección modal de los usuarios, la ruta y el destino
- **Aspectos de la oferta de transporte.** La localización de los nodos y arcos en el plano, la implantación temporal de la vía durante el periodo de planificación, las restricciones de cobertura de nodos y los tipos de arcos.

Estos aspectos son excesivamente complejos por lo que se consideran las si-

güentes simplificaciones del problema:

- **Aspectos de la demanda.** Se considera una demanda fija y conocida entre un conjunto de pares origen-destino. Se considera que no hay competencia entre modos de transporte y se modela la elección de la ruta.
- **Aspectos de la oferta de transporte.** El objetivo es localizar una única vía de un único tipo, asumiendo una simplificación en la red de carreteras previas. Las únicas restricciones a considerar son presupuestarias. Se considera un único periodo de implantación.

Se considera que existen unos puntos en el plano, denominados *centroides*, donde existen usuarios con ciertas necesidades de transportarse (*nodos orígenes*) y otros puntos del plano donde estos usuarios quieren viajar (*nodos destinos*). Estos centroides se corresponden con centros urbanos. Los centroides (ciudades) se denotan por c_1, \dots, c_s .

Se supone que existen unas demandas de viaje entre estos centroides. Se denotará por w un par de demanda origen(o)–destino(d) y por \mathcal{W} el conjunto de todos estos pares $w = (o, d)$.

Se denota por g_w el número de usuarios del par $w \in \mathcal{W}$. Esta demanda constituye la llamada matriz de demanda origen-destino.

La vía de transporte constará de n puntos p_1, \dots, p_n y se representa mediante segmentos rectos entre dos puntos consecutivos p_i p_{i+1} . Desde cada centroide c_i se podrá acceder a todos los puntos $p_1 \dots p_n$. Los arcos de autovía serán de la forma (p_j, p_{j+1}) y los arcos de incorporación a la autovía serán de la forma (c_i, p_j) .

El grafo resultante, \mathcal{G} consta de un conjunto de arcos \mathcal{A} formado por arcos de autovía \mathcal{A}_a y arcos de carretera de acceso \mathcal{A}_c y un conjunto de nodos \mathcal{N} formado por nodos centroides \mathcal{N}_c y por las localizaciones de la autovía \mathcal{N}_p . Los centroides pueden ser orígenes, destinos o las dos cosas (una ciudad genera viajes hacia otras ciudades

y también recibe viajes de otras ciudades). Denotamos por $\mathcal{N}_o \subset \mathcal{N}_c$ el conjunto de orígenes y por $\mathcal{N}_d \subset \mathcal{N}_c$ el conjunto de destinos.

Se crean arcos ficticios directos entre origen–destino que permiten enrutar la demanda si este hecho es más ventajoso que emplear la autovía.

En este grafo se consideran unas variables de flujo, denotada x_{ij}^o , que representa el flujo en el arco $(i, j) \in \mathcal{A}$ originado en el centroide–origen o .

Se consideran dos velocidades de transporte. Una velocidad v_1 para los arcos de autovía (p_j, p_{j+1}) y una velocidad v_2 para los arcos de incorporación (c_i, p_j) . Asumiremos $v_1 > v_2$.

El modelo de optimización está compuesto por unas variables de decisión, unas restricciones y un criterio de optimización.

Las variables de decisión del modelo son la localización en el plano de las coordenadas de los n puntos por los que pasará la carretera: $(x_1, y_1) \dots (x_n, y_n)$.

El modelo debe cumplir las siguientes restricciones:

- **Presupuestarias.** La longitud de la carretera debe ser menor o igual a una constante C dada. Esta restricción se modela con la ecuación (3.1).

$$\sum_{i=1}^{n-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} < C \quad (3.1)$$

donde: x_i es la coordenada en el eje x del nodo p_i

y_i es la coordenada en el eje y del nodo p_i

x_{i+1} es la coordenada en el eje x del nodo p_{i+1}

y_{i+1} es la coordenada en el eje y del nodo p_{i+1}

Se ha asumido la distancia euclídea.

- **Satisfacción de la demanda.** La suma de los flujos que entran a cada nodo no centroide debe ser igual a la suma de los que salen. Esta restricción se modela con

la ecuación (3.2).

$$\sum_{k \in E(i)} x_{ik}^o - \sum_{k \in S(i)} x_{ki}^o = 0, \quad \forall i \in \mathcal{N}_p, \forall o \in \mathcal{N}_c \quad (3.2)$$

donde: $E(i) = \{k / (k, i) \in \mathcal{A}\}$ son los nodos con arcos entrantes al nodo i

$S(i) = \{k / (i, k) \in \mathcal{A}\}$ son los nodos con arcos salientes al nodo i

La demanda total saliente de un centroide origen (denotada g_o) se expresa como muestra la ecuación (3.3).

$$g_o = \sum_{k \in S(o)} x_{ok}^o, \quad \forall o \in \mathcal{N}_c \quad (3.3)$$

La demanda del par $w = (o, d)$ se expresa como muestra la ecuación (3.4).

$$g_w = \sum_{k \in E(d)} x_{kd}^o, \quad \forall o \in \mathcal{N}_c \quad (3.4)$$

- **No negatividad.** El flujo que lleva el arco (i, j) para el origen o debe ser positivo, para todo arco $(i, j) \in \mathcal{A}$, para todo origen o . Se modela en la ecuación (3.5).

$$x_{ij}^o \geq 0, \quad \forall o \in \mathcal{N}_c, \forall (i, j) \in \mathcal{A} \quad (3.5)$$

donde x_{ij}^o es el flujo que lleva el arco (i, j) generado en el origen o .

El criterio de optimización empleado consiste en minimizar el tiempo total de viaje en la red, representado mediante la variable z . La ecuación (3.6) define el criterio de optimización:

$$z = \sum_{o \in \mathcal{N}_c} \sum_{(i,j) \in \mathcal{A}} x_{ij}^o \cdot \frac{d_{ij}}{v_{ij}} \quad (3.6)$$

donde: x_{ij}^o es el flujo que lleva el arco (i, j) para el origen o .

$v_{i,j}$ es la velocidad del arco (i, j) .

$d_{i,j}$ es la distancia euclídea entre los puntos p_i y p_j y se define en la ecuación (3.7).

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (3.7)$$

donde: x_i es la coordenada x del nodo p_i
 x_j es la coordenada x del nodo p_j
 y_i es la coordenada y del nodo p_i
 y_j es la coordenada y del nodo p_j

Finalmente el modelo propuesto se puede escribir como el siguiente problema de optimización:

Minimizar

$$z = \sum_{o \in \mathcal{N}_c} \sum_{(i,j) \in \mathcal{A}} x_{ij}^o \cdot \frac{d_{ij}}{v_{ij}} \quad (3.8)$$

sujeto a:

$$\sum_{k \in E(i)} x_{ik}^o - \sum_{k \in S(i)} x_{ki}^o = r_{oi}, \quad \forall i \in \mathcal{N}_p, \forall o \in \mathcal{N}_c \quad (3.9)$$

$$\sum_{i=1}^{n-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} < C \quad (3.10)$$

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \quad \forall (i, j) \in \mathcal{A} \quad (3.11)$$

donde

$$x_{ij}^o \geq 0, \quad \forall o \in \mathcal{N}_c, \forall (i, j) \in \mathcal{A} \quad (3.12)$$

$$r_{oi} = \begin{cases} 0 & \text{si } i \in \mathcal{N}_p \\ g_w & \text{si } i \in \mathcal{N}_c \text{ y } w = (o, i) \in \mathcal{W} \\ \sum_{(o,k) \in \mathcal{W}} g_{o,k} & \text{si } o = i \end{cases} \quad (3.13)$$

Se asume implícitamente que los n primeros nodos corresponden a la localización de la autovía y el resto a los centroides.

El número de variables del problema es:

$$\underbrace{n}_{\text{localidades}} + \underbrace{|\mathcal{N}_o| \times |\mathcal{A}|}_{\text{flujo}} \quad (3.14)$$

El número de restricciones es:

$$1 + |\mathcal{N}_o| \times |\mathcal{A}| + |\mathcal{A}| + |\mathcal{N}_o| \times |\mathcal{A}| \quad (3.15)$$

Por otro lado el número de arcos es:

$$|\mathcal{A}| = n_c \cdot n + (n - 1) \quad (3.16)$$

$$|\mathcal{N}| = n + n_c \quad (3.17)$$

Finalmente queda:

$$\text{variables} = n + n_o(n_c \cdot n + (n - 1)) + |\mathcal{W}| \quad (3.18)$$

$$\text{restricciones} = 1 + n_o(n + n_c) + n_c \cdot n + n_o(n_c \cdot n + (n - 1)) \quad (3.19)$$

Si se observa el hecho que el trazado de una autovía entre dos incorporaciones–salidas debe ser una línea recta para que este trazado sea de mínimo tiempo y por tanto, óptimo, se tiene que el modelo localizará estas entradas–salidas de la autovía y por tanto cualquier valor de $n \geq n_o + n_d$ conducirá siempre a la misma solución del modelo.

Como conclusión el tamaño del problema crece como un polinomio cúbico en el número de centroides.

Las pruebas numéricas realizadas en GAMS demostraron que se trata de un problema inabordable mediante técnicas de cálculo numérico. A la hora de realizar pruebas numéricas con las 919 poblaciones de Castilla–La Mancha, tanto en un PC como en el servicio de Super Computación de la UCLM, GAMS (que implementa métodos exactos) no fue capaz de trabajar con la cantidad de datos que utiliza el modelo.

3.4. Aportaciones de este Proyecto Fin de Carrera

Las aportaciones realizadas en este PFC se pueden agrupar en tres categorías:

- **Modelización matemática del problema de localización de alineamientos.** En la literatura científica se han abordado separadamente el problema de la localización de autovías y el de los alineamientos de trenes ligeros. El primer problema se ha abordado en un entorno continuo (localización permitida en todo el plano) mientras que para el segundo problema se ha formulado como un problema de localización en grafos (problema discreto).

El modelo propuesto presenta las siguientes novedades:

1. Un marco unificado de modelización para ambos problemas.
 2. Considera un espacio continuo (novedoso frente al problema de la localización de alineamientos en sistemas de metro)
 3. Considera una matriz de demanda origen–destino dada a satisfacer y por tanto no se asume conocido el inicio y final del alineamiento (novedoso en la literatura de localización de autovías).
 4. Se considera la existencia de obstáculos (novedoso en la literatura científica de ambos problemas)
 5. Consideraciones de elasticidad de la demanda para el caso de autovías.
- **Técnicas computacionales de resolución.** Los modelos propuestos son difíciles de resolver. En el problema de alineamientos de metros son escasos los algoritmos propuestos y no son enteramente satisfactorios los resultados obtenidos siendo aplicables a problemas de muy reducidas dimensiones. Para este tipo de modelos se ha explorado una nueva vía y se ha demostrado computacionalmente su viabilidad. Las técnicas desarrolladas son extensibles a otros problemas de geometría computacional.

Desde el punto de vista de la Ingeniería Informática se ha requerido una estructura

de datos eficientes y subrutinas altamente depuradas que permitan aprovechar al máximo las características del problema.

- Desde el punto de vista de la práctica se ha desarrollado una herramienta informática que implementa lo anterior. No existe software comercial para este propósito. Los programas desarrollados están en el campo de la investigación y todavía no son comercializables.

Como conclusión decir que la dificultad de la problemática abordada ha requerido exprimir al máximo la formación matemática, en ciencias de la computación, algorítmica, estructuras de datos, etc., recibida a lo largo de la carrera.

Capítulo 4

Método de trabajo

- 4.1. Organización de los objetivos
 - 4.2. Herramientas utilizadas
 - 4.3. Ingeniería del Software
 - 4.3.1. Ciclo de vida del Software
 - 4.3.2. Análisis y Diseño
-

4.1. Organización de los objetivos

Este proyecto tiene tres objetivos principales, cuyo desarrollo se puede ver en detalle en el apartado 2.

1. Debe implementar un modelo matemático para el auxilio en la toma de decisiones sobre la localización preliminar de un alineamiento horizontal.
2. Debe desarrollar una metodología de resolución e implementarla.
3. Debe ser aplicable a problemas reales.

En primer lugar se desarrolló e implementó el modelo matemático que resuelve el problema (ver apartado 5).

Una vez hecho esto, se desarrolló una metodología de resolución, en este caso un Algoritmo Genético (ver apartado 6).

Finalmente se realizaron pruebas numéricas con datos de problemas reales. Los resultados y conclusiones de dichas pruebas se pueden ver en el apartado 8.

4.2. Herramientas utilizadas

El entorno de programación utilizado para este proyecto es MATLAB. MATLAB ofrece una gran potencia de cómputo a la hora de trabajar con vectores. Esto resulta muy útil, ya que organizando grandes cantidades de datos en vectores, MATLAB trabaja rápidamente con los datos. Además, MATLAB proporciona todas las características de un lenguaje de programación de alto nivel, como son el manejo de estructuras de datos complejas, el manejo de ficheros, etc.

Para la implementación de la interfaz gráfica de usuario se ha utilizado el

entorno de desarrollo GUIDE (presente en el entorno MATLAB), que proporciona un conjunto de herramientas para crear interfaces gráficas de usuario (GUIs). Estas herramientas simplifican el proceso de diseño y construcción de GUIs y su conexión con código escrito en el entorno MATLAB.

Para la generación de la documentación se ha usado \LaTeX . Los documentos se han compilado con `pdflatex`. \LaTeX ofrece muchas ventajas, como son la facilidad para componer fórmulas matemáticas, la facilidad para componer documentos estructurados, la generación automática de pies de página, referencias, tablas de contenido y bibliografías. Además, \LaTeX es portable.

Para la generación de las diapositivas se ha usado la clase Beamer de \LaTeX . Las principales ventajas de Beamer frente a otros programas de creación de diapositivas son básicamente las mismas que posee \LaTeX .

Para editar el documento y la presentación se ha utilizado la herramienta WinEdt.

Para la generación de los gráficos incluidos en la documentación se ha utilizado MATLAB, que permite exportar los gráficos a formato *Encapsulated PostScript Language (EPS)*. Estos gráficos se han exportado posteriormente a formato *Portable Document Format (PDF)* utilizando Adobe Acrobat Distiller. Para la modificación de algunos de los gráficos se ha usado Gimp.

Para la generación de los diagramas de flujos de datos y los diagramas de estructura se ha utilizado EasyCase.

Para visualizar la documentación y la presentación se ha usado Adobe Acrobat Reader.

4.3. Ingeniería del Software

En el proceso de creación de la herramienta LAHIT ha influido el ciclo de vida elegido y la decisión de seguir el paradigma de la *Programación Estructurada*, condicionado a la elección del entorno MATLAB.

4.3.1. Ciclo de vida del Software

Según la definición dada en ISO 12207-1 el ciclo de vida del software es

“Un marco de referencia que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando la vida del sistema desde la definición de los requisitos hasta la finalización de su uso”.

El modelo de ciclo de vida empleado en este proyecto, es el *Modelo en Espiral* [DP04], propuesto por Boehm en 1988. En cada una de las iteraciones de la espiral, se plantean nuevos objetivos evaluando ciertas alternativas y posibles riesgos. Al finalizar cada una de ellas se obtiene un nuevo producto que debe cumplir los requisitos establecidos en un principio y funcionar correctamente.

4.3.2. Análisis y Diseño

Debido a la herramienta usada para implementar el sistema, MATLAB, éste sigue el paradigma de la *Programación Estructurada* [DP04]. El análisis y el diseño de la herramienta se detallarán mediante Diagramas de Flujo de Datos y Diagramas de Estructura (Véase apartado 7).

Capítulo 5

Modelo matemático para la localización preliminar de un alineamiento horizontal

5.1. Estructura del modelo

5.2. Formulación matemática del modelo

5.2.1. Restricciones del sistema de transporte

5.2.2. Restricciones del comportamiento del usuario

5.2.3. Criterios de diseño

5.3. Consideraciones geométricas sobre la formulación del modelo

5.3.1. Necesidad de la rejilla

5.3.2. Existencia de obstáculos

5.3.3. Problema de caminos mínimos

5.3.4. Accesos al alineamiento

5.1. Estructura del modelo

En la revisión de la literatura realizada en el capítulo 3 se muestra que la herramienta matemática empleada en la localización de alineamientos son los modelos de optimización.

En este proyecto se abordan simultáneamente dos tipos de problemas:

- **Problema 1.** Localización preliminar del alineamiento horizontal de una autovía.
- **Problema 2.** Localización preliminar del alineamiento horizontal de una vía ferroviaria, tipo metro o cercanías.

El modelo matemático propuesto para la localización preliminar de un alineamiento horizontal que se expone en este capítulo es aplicable a ambos problemas. La estructura de dicho modelo se recoge en la tabla 5.1.

Tabla 5.1: Estructura del modelo matemático

Optimizar	(A) Criterio de diseño de una red <ul style="list-style-type: none"> (A.1) Maximización del cubrimiento de la nueva infraestructura de transporte (A.2) Minimización de la congestión de la infraestructura de transporte existente (A.3) Minimización del tiempo total de transporte en la infraestructura nueva y la existente
Sujeto a:	(B) Modelado del sistema de transporte <ul style="list-style-type: none"> (B.1) Modelado de la red de transporte (B.2) Restricciones presupuestarias (B.3) Evitación de obstáculos (C) Modelado del comportamiento del usuario <ul style="list-style-type: none"> (C.1) Elección modal (Modelo Logit) (C.2) Elección de ruta (Caminos mínimos)

5.2. Formulación matemática del modelo

A continuación se explican en detalle todos los elementos que forman parte del modelo.

Conjuntos

\mathcal{G} : Es el grafo que representa el sistema de transporte. Está formado por un conjunto de arcos, \mathcal{A} y un conjunto de nodos, \mathcal{N} : $\mathcal{G} = (\mathcal{A}, \mathcal{N})$.

\mathcal{N} : Es el conjunto de nodos: $\mathcal{N} = \mathcal{N}_c \cup \mathcal{N}_p$.

\mathcal{N}_c : Es el conjunto de nodos centroides c_i .

\mathcal{N}_p : Es el conjunto de puntos de la poligonal que definen la vía principal p_i .

\mathcal{N}_i : Es el subconjunto de \mathcal{N}_p que son incorporaciones a la vía.

\mathcal{A}_p : Es el conjunto de arcos de vía principal. Los arcos $a \in \mathcal{A}$ son de la forma $a = (p_i, p_{i+1})$.

\mathcal{A}_s : Es el conjunto de arcos de red secundaria. Los arcos $a \in \mathcal{A}$ son de la forma $a = (c_i, p_j)$ o $a = (c_i, c_j)$ con $p_j \in \mathcal{N}_i$.

w : Es el par de demanda entre dos centroides de la red. El primer elemento se denomina origen de la demanda y el segundo, destino.

\mathcal{W} : Es el conjunto de todos los pares w de demanda origen–destino.

Datos

x_i : Es la coordenada x UTM del centroide $i \in \mathcal{N}_c$.

y_i : Es la coordenada y UTM del centroide $i \in \mathcal{N}_c$.

v_{ij} : Es la velocidad del arco (i, j) . Se consideran tres velocidades de transporte. Una velocidad v_1 para los arcos de autovía (p_i, p_{i+1}) , una velocidad v_2 para los arcos de incorporación (c_i, p_j) y una velocidad v_3 para viajar directamente de un origen a un destino. Se supone que $v_1 > v_2$.

g_w : Es el número de usuarios del par $w \in \mathcal{W}$.

C_1 : Es el presupuesto disponible para la construcción del alineamiento.

C_2 : Es el presupuesto disponible para la localización de estaciones/incorporaciones.

Variables

d_{ij} Es la distancia euclídea entre los nodos p_i y p_j . Se define como muestra la expresión

$$(5.1): \quad d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (5.1)$$

x_i : Es la posición en el eje x del nodo $i \in \mathcal{N}_p$.

y_i : Es la posición en el eje y del nodo $i \in \mathcal{N}_p$.

5.2.1. Restricciones del sistema de transporte

(B.1) Modelado de la red de transporte

Se supone que existen unos puntos en el plano, denominados centroides, donde existen usuarios con ciertas necesidades de transportarse (nodos orígenes) y otros puntos del plano donde estos usuarios quieren viajar (nodos destinos). Estos centroides corresponden con centros urbanos (Véase Figura 5.1). Los centroides se denotan por c_1, \dots, c_s .

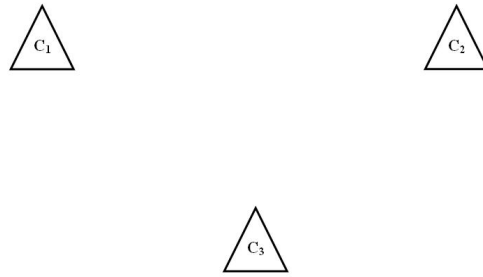


Figura 5.1: Centroides

Se supondrá que existen unas demandas de viaje entre estos centroides. Se denotará por w un par de demanda origen(o)–destino(d) y por \mathcal{W} el conjunto de todos estos pares $o-d$ (Véase Figura 5.2).

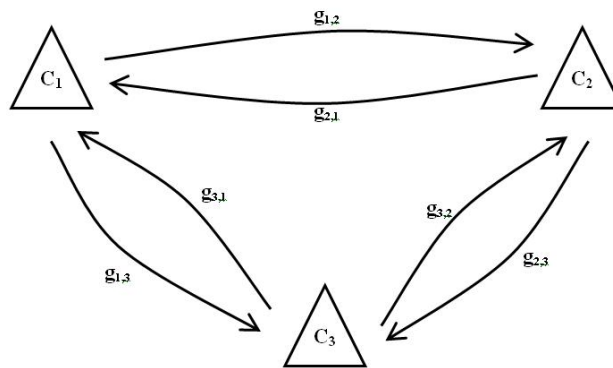


Figura 5.2: Matriz origen–destino entre los centroides

Se denotará por g_w el número de usuarios del par $w \in \mathcal{W}$. Esta demanda constituye la llamada matriz de demanda origen-destino.

La vía de transporte se modelará por una poligonal de n puntos p_1, \dots, p_n y se representará mediante segmentos rectos entre dos puntos consecutivos p_i p_{i+1} . La figura 5.3 representa la red de transporte. Esta conduce a un grafo que determina los posibles movimientos entre los nodos de la red. Desde cada centroide c_i se podrá acceder

a todos los puntos $p_j \in \mathcal{N}_i$ (incorporaciones). Los arcos de la red principal serán de la forma (p_i, p_{i+1}) y los arcos de la red secundaria serán de la forma (c_i, p_j) o (c_i, c_j) . El grafo resultante es el mostrado en la figura 5.4.

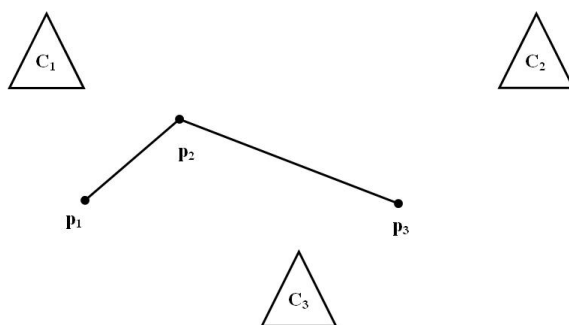


Figura 5.3: Alineamiento

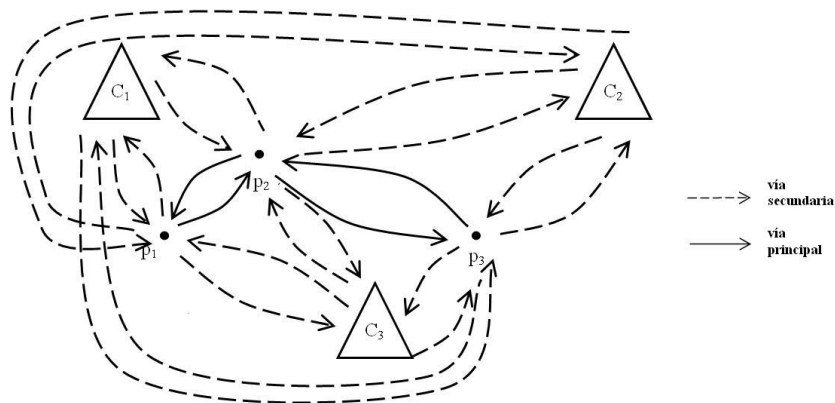


Figura 5.4: Grafo de movimiento del problema

Este grafo \mathcal{G} consta de un conjunto de arcos \mathcal{A} formado por arcos de la red principal \mathcal{A}_p y arcos de la red secundaria \mathcal{A}_s y un conjunto de nodos \mathcal{N} formado por nodos centroides \mathcal{N}_c y por las localizaciones de la red principal \mathcal{N}_p . Los centroides pueden ser orígenes, destinos o las dos cosas. Se denota por $\mathcal{N}_o \subset \mathcal{N}_c$ el conjunto de orígenes y por $\mathcal{N}_d \subset \mathcal{N}_c$ el conjunto de destinos.

Se han creado arcos ficticios directos entre origen–destino que modelan una red

de transporte existente y permiten enrutar la demanda si este hecho es más ventajoso que emplear la red principal.

Se denota con $\mathcal{P} \subset \mathbb{R}^2$ un alineamiento genérico definido por la poligonal de nodos p_1, p_2, \dots, p_n .

(B.2) Restricciones presupuestarias

Hay un presupuesto máximo C para la construcción de la vía. Dicho presupuesto se divide por el usuario en dos partes, C_1 y C_2 , tales que $C_1 + C_2 = C$.

C_1 : Es el presupuesto máximo destinado a incorporaciones. El coste de las incorporaciones debe ser menor o igual a C_1 .

$$\hat{c}_1 \cdot \sum_{i=1}^n \delta_i \leq C_1 \quad (5.2)$$

donde: \hat{c}_1 es el coste de construir una estación/incorporación

$$\delta_i = \begin{cases} 1 & \text{si } p_i \in \mathcal{N}_i \\ 0 & \text{si } p_i \notin \mathcal{N}_i \end{cases}$$

C_2 : Es el presupuesto máximo destinado al trazado de la vía. El coste del trazado de la vía debe ser menor o igual a C_2 .

$$\hat{c}_2 \cdot \sum_{i=1}^{n-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \leq C_2 \quad (5.3)$$

donde: \hat{c}_2 es el coste por kilómetro de vía

x_i es la coordenada en el eje x del nodo $p_i \in \mathcal{P}$

y_i es la coordenada en el eje y del nodo $p_i \in \mathcal{P}$

x_{i+1} es la coordenada en el eje x del nodo $p_{i+1} \in \mathcal{P}$

y_{i+1} es la coordenada en el eje y del nodo $p_{i+1} \in \mathcal{P}$

(B.3) Evitación de obstáculos

Se considera que hay un conjunto $\mathcal{C} \subset \mathbb{R}^2$ de puntos en el plano por los que no puede pasar el alineamiento \mathcal{P} . Esto se modeliza con la ecuación (5.4).

$$\mathcal{C} \cap \mathcal{P} = \{\emptyset\} \quad (5.4)$$

Este conjunto de restricciones se introduce para modelar que la autovía no puede pasar por parques naturales, monumentos, ciudades o en general zonas a preservar. Para el problema de alineamientos de tipo metro permite recoger zonas geotécnicas prohibidas. El hecho de que el algoritmo de resolución sea de naturaleza heurística hace que esta restricción no se haya definido explícitamente a través de un sistema de ecuaciones.

5.2.2. Restricciones del comportamiento del usuario**(C.1) Elección modal (Modelo Logit)**

Se considera que los usuarios tienen dos alternativas para llevar a cabo sus viajes:

- (E) Alineamiento existente. El número de usuarios que realizan sus viajes utilizando la red de transporte existente se denota por g_w^E y el coste de esta alternativa para el par $w \in \mathcal{W}$ se denota por u_w^E .
- (N) Alineamiento nuevo. El número de pasajeros que utilizan nuevo alineamiento para el par de demanda $w \in \mathcal{W}$ se denota por g_w^N y su coste se denota por u_w^N .

Se supone que la proporción de usuarios en cada alternativa de transporte depende exclusivamente de los costes de transporte de cada red, y esta relación viene dada por un modelo de demanda. Un modelo ampliamente usado en la planificación

de transportes es el modelo Logit, que da la proporción de usuarios en el alineamiento nuevo para el par w , p_w^N , mediante la expresión (5.5).

$$p_w^N(u_\omega^N, u_\omega^E) = \frac{\exp\{-(\alpha^N + \beta u_\omega^N)\}}{\sum_{m \in \{N, E\}} \exp\{-(\alpha^m + \beta u_\omega^m)\}} \quad (5.5)$$

donde: α^N y α^E son los parámetros de distribución que simulan la cuota de mercado de cada modo de transporte.

β pondera la importancia de los costes de transportes frente a la cuota modal en la elección de la alternativa.

Para los valores $\alpha^N = \alpha^E = 0$ y $\beta \rightarrow +\infty$, el modelo Logit se comporta como un modelo de asignación *todo-nada* y conduce a la expresión (5.6).

$$p_w^N(u_\omega^N, u_\omega^E) = \begin{cases} 1 & \text{if } u_\omega^E - u_\omega^N > 0 \\ 0 & \text{if } u_\omega^E - u_\omega^N < 0 \end{cases} \quad (5.6)$$

(C.2) Elección de ruta (Caminos mínimos)

El modelo considera que los usuarios eligen su ruta dentro del nuevo alineamiento de modo que se minimice su tiempo de viaje.

Estas elecciones consisten en determinar los nodos de incorporación y salida del nuevo alineamiento de modo que minimice el tiempo de viaje. Estas elecciones conducen a un problema de caminos mínimos dentro del grafo de movimientos \mathcal{G} .

5.2.3. Criterios de diseño

Los criterios de diseño definen la función objetivo del problema de optimización. Los problemas de diseño de infraestructuras de transporte se pueden abordar bajo dos ópticas:

1. **Competencia entre redes.** Este enfoque considera que la nueva infraestructura compite con las existentes. 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 existente. El objetivo que se persigue es optimizar la red resultante.

Si se desea implementar una infraestructura ferroviaria con el objeto de reducir la congestión en una red de tráfico, el enfoque de competencia entre redes de transporte es el más adecuado. Los criterios utilizados son (A.1) y (A.2), que se describirán a continuación.

Para el problema de localizar una nueva autovía sobre una red de carreteras existente, 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. Un criterio utilizado es (A.3).

(A.1) Maximización del cubrimiento de la nueva infraestructura de transporte

Este criterio de optimización consiste en captar el mayor número de usuarios posible en la nueva infraestructura de transporte. Para ello hay que maximizar el valor de Z_1 .

$$Z_1 = \sum_{w \in W} g_w^N \quad (5.7)$$

(A.2) Minimización de la congestión de la infraestructura de transporte existente

Este criterio de optimización consiste en conseguir reducir al máximo el tiempo empleado en la red de transporte existente. Para ello hay que minimizar el valor de \mathcal{Z}_2 .

$$\mathcal{Z}_2 = \sum_{w \in \mathcal{W}} u_w^E g_w^E \quad (5.8)$$

(A.3) Minimización del tiempo total de transporte en la infraestructura nueva y la existente

Este criterio de optimización consiste en minimizar el tiempo total de viaje en la red. Para ello hay que minimizar el valor de \mathcal{Z}_3 .

$$\mathcal{Z}_3 = \sum_{w \in \mathcal{W}} \left(u_w^E g_w^E + u_w^N g_w^N \right) \quad (5.9)$$

5.3. Consideraciones geométricas sobre la formulación del modelo

5.3.1. El problema de las vueltas hacia atrás: la necesidad de la rejilla

A priori se puede considerar que no toda poligonal \mathcal{P} define una carretera o el trazado de una línea de metro. La figura 5.5 muestra una situación inusual. Para evitar estos alineamientos se introduce una rejilla.

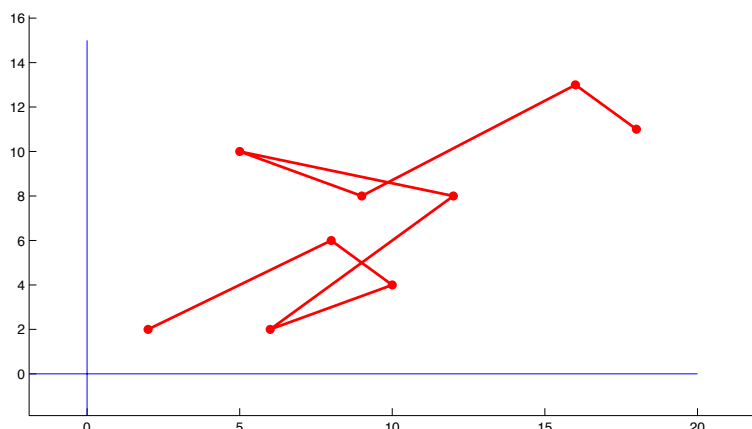


Figura 5.5: Poligonal no válida para un alineamiento

Se va a utilizar una rejilla para evitar la vuelta atrás en la vía. Una rejilla es un conjunto de líneas paralelas, es decir, un conjunto de líneas con la misma pendiente.

En primer lugar se realizará una traslación de los centroides:

$$\begin{aligned}x_{nuevo} &= x - x_{min} \\ y_{nuevo} &= y - y_{min}\end{aligned}\tag{5.10}$$

donde: x_{min} e y_{min} son las coordenadas x e y UTM mínimas entre todos los centroides del problema considerado.

Esta traslación hace que todos los centroides estén en el rectángulo definido por los vértices que se muestran en (5.11).

$$\begin{aligned}v_1 &= (0, 0) \\ v_2 &= (0, y_{max} - y_{min}) \\ v_3 &= (x_{max} - x_{min}, y_{max} - y_{min}) \\ v_4 &= (x_{max} - x_{min}, 0)\end{aligned}\tag{5.11}$$

Después se considera una pendiente de las rectas que forman la rejilla. Para ello se considera una variable aleatoria Θ , que denota un ángulo, distribuida uniformemente

en $(-\pi, \pi)$. Esta variable representa el ángulo formado por la recta que pasa por el origen y el eje de las X. A continuación se calcula la pendiente usando (5.12).

$$m = \tan(\Theta) \quad (5.12)$$

El valor de la pendiente condiciona fuertemente los alineamientos, por lo que se tendrá que ir variando m (o alternativamente Θ) en el proceso de optimización.

La siguiente decisión es determinar cuantas rectas forman la rejilla. Si se pretende calcular trabajar con una precisión aproximada de ε (1 kilómetro por ejemplo), se toma un número de rectas n que cumpla (5.13).

$$n \geq \frac{\text{longitud diagonal}}{\varepsilon} + 1 \quad (5.13)$$

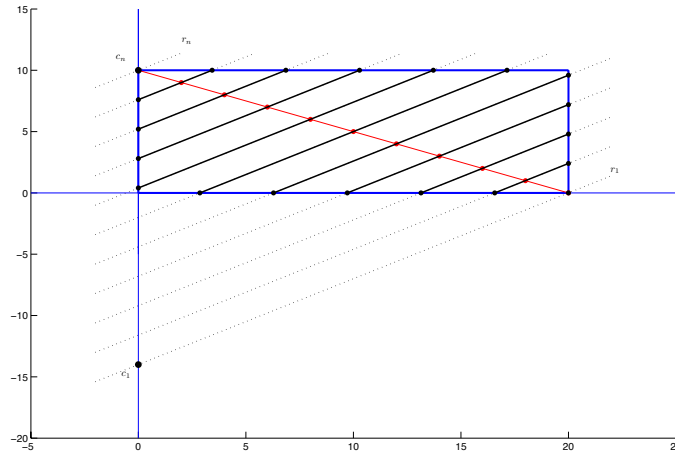
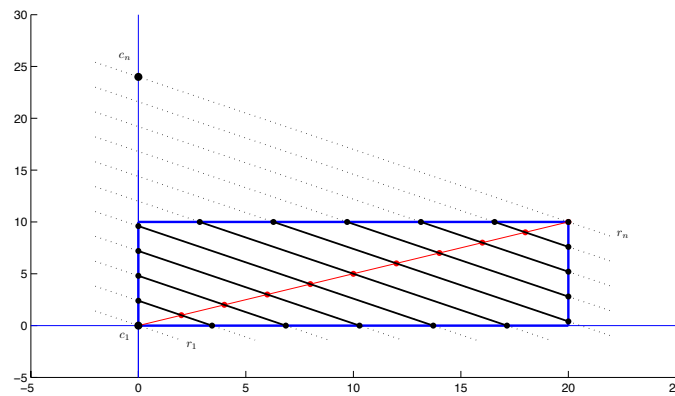
La rejilla tiene que cubrir el rectángulo donde se sitúan los centroides. Lo primero es calcular todas las ecuaciones de las rectas. Como todas las rectas tienen pendiente m , sólo hay que calcular el término de interceptación, es decir, la constante. Introducimos la notación que se muestra en la expresión (5.14).

$$y_i = mx_i + c_i, \quad i = 1, \dots, n \quad (5.14)$$

Se plantean dos situaciones:

- $m > 0$ La primera recta tiene $c_1 = -m \cdot (x_{max} - x_{min})$ y la última $c_n = (y_{max} - y_{min})$. (Véase Figura 5.6)
- $m < 0$ La primera recta tiene $c_1 = 0$ y la última $c_n = (y_{max} - y_{min}) - m \cdot (x_{max} - x_{min})$. (Véase Figura 5.7)

Se considera que la primera recta es la que tiene un término de interceptación menor y la última la que tiene un valor mayor.

Figura 5.6: Rectas caso $m > 0$ Figura 5.7: Rectas caso $m < 0$

A partir de estos datos, las demás c_i se calculan mediante la expresión (5.15).

$$c_i = \frac{c_n - c_1}{(n-1)}(i-1) + c_1, \quad i = 1, \dots, n \quad (5.15)$$

Aunque la disposición de las rectas esté en el plano, con dos variables x e y , se considerarán estas como segmentos con un solo grado de libertad. Se considerará que la x es la variable independiente, que se puede mover, y la y es la variable dependiente,

que queda completamente determinada por las anteriores ecuaciones de las rectas. El siguiente problema es determinar que rangos de las x son admisibles para que los segmentos no se salgan del rectángulo de vértices v_1, v_2, v_3, v_4 calculados en la expresión (5.11).

Para cada caso de los expuestos anteriormente, $m < 0$ y $m > 0$, se consideran tres tipos diferentes de rectas en función de qué lado del rectángulo corten, como se puede ver a continuación.

- Rejilla con pendiente $m < 0$ (Véase Figura 5.8):

Caso 1: Rectas con $c_i \leq y_{max} - y_{min}$

$$x \in \left[0, \frac{-c_i}{m} \right] \quad (5.16)$$

Caso 2: Rectas con $y_{max} - y_{min} < c_i \leq -m \cdot (x_{max} - x_{min})$

$$x \in \left[\frac{y_{max} - y_{min} - c_i}{m}, \frac{-c_i}{m} \right] \quad (5.17)$$

Caso 3: Rectas con $c_i > -m \cdot (x_{max} - x_{min})$

$$x \in \left[\frac{y_{max} - y_{min} - c_i}{m}, x_{max} - x_{min} \right] \quad (5.18)$$

- Rejilla con pendiente $m > 0$ (Véase Figura 5.9):

Caso 1: Rectas con $c_i \geq 0$

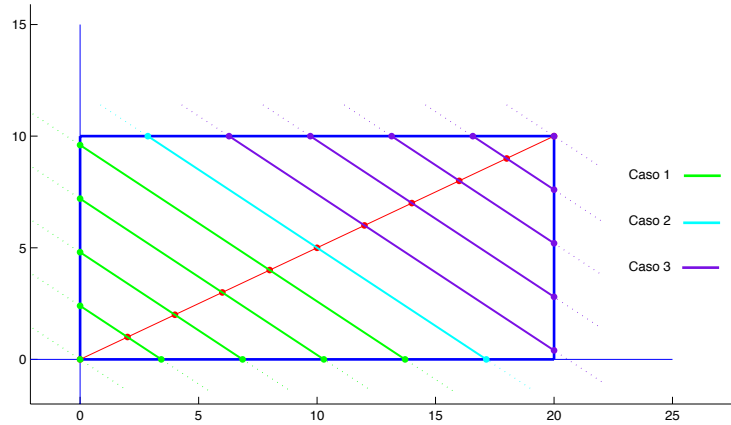
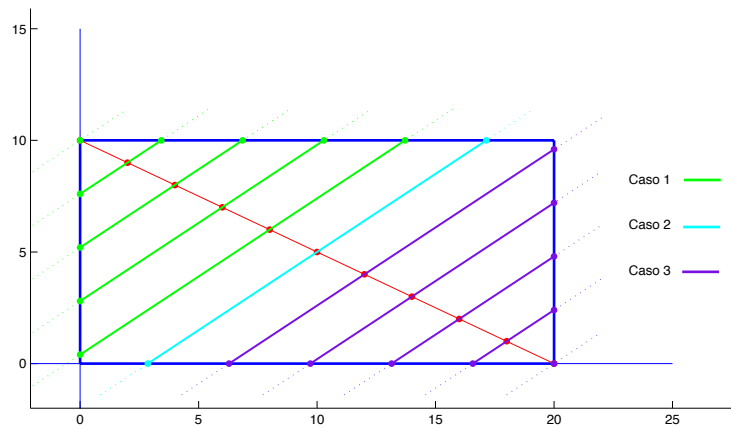
$$x \in \left[0, \frac{y_{max} - y_{min} - c_i}{m} \right] \quad (5.19)$$

Caso 2: Rectas con $0 > c_i \geq y_{max} - y_{min} - m \cdot (x_{max} - x_{min})$

$$x \in \left[\frac{-c_i}{m}, \frac{y_{max} - y_{min} - c_i}{m} \right] \quad (5.20)$$

Caso 3: Rectas con $c_i < y_{max} - y_{min} - m \cdot (x_{max} - x_{min})$

$$x \in \left[\frac{-c_i}{m}, x_{max} - x_{min} \right] \quad (5.21)$$

Figura 5.8: Caso $m < 0$ Figura 5.9: Caso $m > 0$

Utilidad de la rejilla. Se exigirá que los nodos de los alineamientos (poligonal \mathcal{P}) pertenezcan a la rejilla, forzando un avance en el alineamiento. Por simplicidad se forzará a que nodos consecutivos del alineamiento estén en segmentos consecutivos de la rejilla.

5.3.2. Restricciones a la localización en el plano: existencia de obstáculos

Para este proyecto se van a considerar obstáculos rectangulares, disjuntos dos a dos, orientados de la misma manera que el rectángulo que contiene a los centroides. Un obstáculo queda definido por sus cuatro vértices.

Se considera que los obstáculos prohíben ciertos intervalos de x para cada segmento de la rejilla. Se puede ver un ejemplo en la figura 5.10.

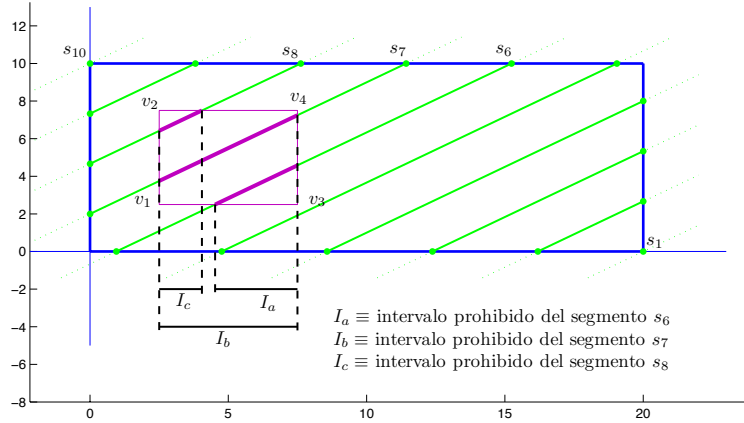


Figura 5.10: Intervalos prohibidos por un obstáculo

Cuando un segmento s se ve afectado por más de un obstáculo $o = \{o_1, \dots, o_q\}$, sus intervalos prohibidos se calculan como la unión de los intervalos prohibidos por cada obstáculo, como se puede ver en la ecuación (5.22). Sus intervalos permitidos se calcularán como la intersección entre el intervalo de x del segmento y los intervalos prohibidos, como se puede ver en la ecuación (5.23).

$$\text{intervalosProhibidos}(s) = \bigcup_{i=1}^q \text{intervalo}_{o_i} \quad (5.22)$$

$$\text{intervalosPermitidos}(s) = [x_{min}^s, x_{max}^s] \cap \text{intervalosProhibidos}(s) \quad (5.23)$$

Se puede ver un ejemplo gráfico de los intervalos posibles en un segmento en la figura 5.11.

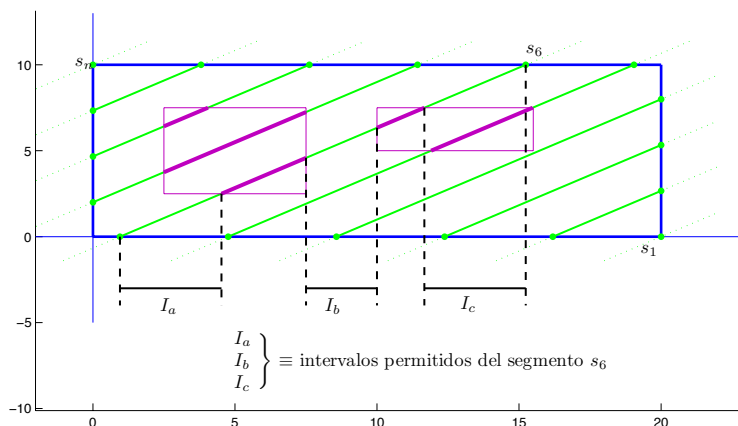


Figura 5.11: Intervalos permitidos de un segmento

5.3.3. Modelización de las elecciones del usuario: un problema de caminos mínimos

Las incorporaciones desde los centroides a la vía, y desde ésta a los centroides, se eligen para que el coste (tiempo) del camino recorrido sea mínimo.

Cuando la vía es una línea recta, con incorporaciones discretas a lo largo de su trazado, esta decisión se simplifica (Véase Figura 5.12). Cada nodo o (origen) tiene dos incorporaciones/salidas a la red. Una para viajar en un sentido, denominado \oplus , y otro para ir en sentido opuesto, denominado \ominus . Estas incorporaciones/salidas son óptimas independientemente del destino del usuario. Esto se puede demostrar por reducción al absurdo utilizando el principio de optimalidad de Bellman.

Considérese que los usuarios del origen o tuvieran dos incorporaciones óptimas I_1 y I_2 para ir a dos destinos empleando la autovía usando la dirección \oplus (Véase Figura 5.13). El principio de optimalidad de Bellman afirma que todo camino mínimo

está formado por subcaminos mínimos. Por tanto los dos caminos para ir de o a I_2 , directo o usando la incorporación I_1 deben tener exactamente el mismo coste. Por tanto I_2 también es una incorporación óptima para el otro destino y lo mismo le sucede a I_1 .

El mismo proceso ocurre con los destinos. Estas consideraciones plantean un algoritmo para calcular incorporaciones y salidas óptimas para el par w . El algoritmo empleado para calcular las incorporaciones/salidas en sentido \oplus (Véase Tabla 5.2) es análogo al utilizado para calcular las incorporaciones/salidas en sentido \ominus .

Una vez calculadas las incorporaciones/salidas de los orígenes y destinos, el camino mínimo se calcula como el mínimo entre las cuatro posibilidades existentes, que se muestran en la tabla 5.3.

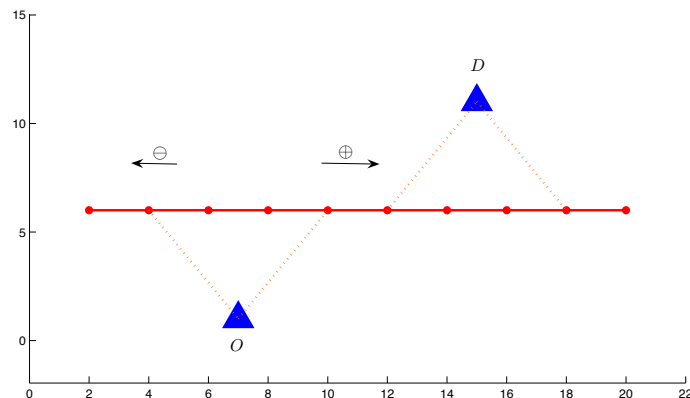


Figura 5.12: Caminos mínimos entre dos centroides en una vía recta

Si la vía no es una línea recta, como es el caso de las vías que se tratan en este proyecto, para calcular las incorporaciones es necesario aplicar un algoritmo de *Caminos mínimos* al grafo \mathcal{G} de movimientos del problema (Véase Figura 5.4 en la página 38).

Estos algoritmos consumen mucho tiempo, y esto, unido a la naturaleza heurística del algoritmo de optimización, hace que para el caso de una poligonal (Véase Figura 5.14) se emplee el algoritmo de la tabla 5.2, aun no siendo la vía una línea recta, basándose en que cierto avance en la poligonal asegura la validez del algoritmo.

Tabla 5.2: Algoritmo utilizado para calcular las incorporaciones/salidas de un centroide $o \in \mathcal{N}_c$ para el sentido \oplus

Entradas:

I	(* Vector de incorporaciones)
o	(* Centroide origen)
$vViaP$	(* Velocidad de la vía principal)
$vViaS$	(* Velocidad de la vía secundaria)

Salidas:

$incDcha$	(* Incorporación \oplus)
$incIzda$	(* Incorporación \ominus)

Inicio (caminoMinimo)

$i := 2$ (* Sentido \oplus)

$tiempo1 := \frac{distancia(o, I_1)}{vViaS}$

$optimo := \text{falso}$

Mientras ($i \leq tamañoI$) **Y** $optimo = \text{falso}$) **hacer**

$tiempo1 := tiempo1 + \frac{distancia(I_{i-1}, I_i)}{vViaP}$

$tiempo1 := \frac{distancia(o, I_i)}{vViaS}$ (* Tiempo directo)

Si $tiempo1 > tiempo2$ **entonces**

$tiempo1 := tiempo2$ (* Seguir adelante)

Si no entonces

$incDcha := i - 1$

$optimo := \text{verdadero}$

Fin Si

$i := i + 1$

Fin Mientras

Si $optimo := \text{falso}$ **entonces**

$nDcha := tamaño(I)$ (* Óptima la última incorporación)

Fin Si

Fin

Tabla 5.3: Cálculo de los caminos mínimos utilizando las incorporaciones calculadas

		destino d	
		\oplus	\ominus
origen o	\oplus	distancia $_{\oplus\oplus}$	distancia $_{\oplus\ominus}$
	\ominus	distancia $_{\ominus\oplus}$	distancia $_{\ominus\ominus}$

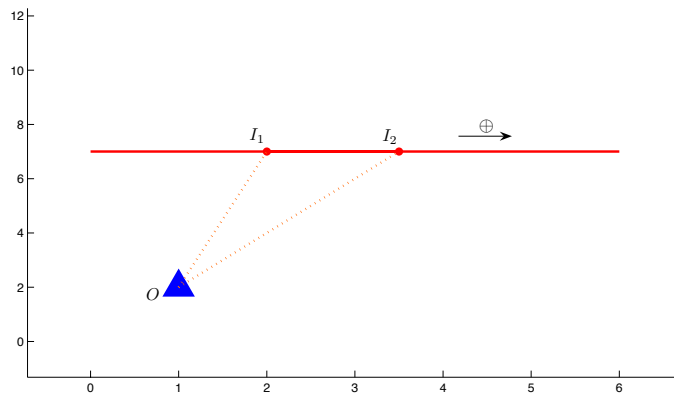


Figura 5.13: Caminos mínimos entre un centroide y dos incorporaciones

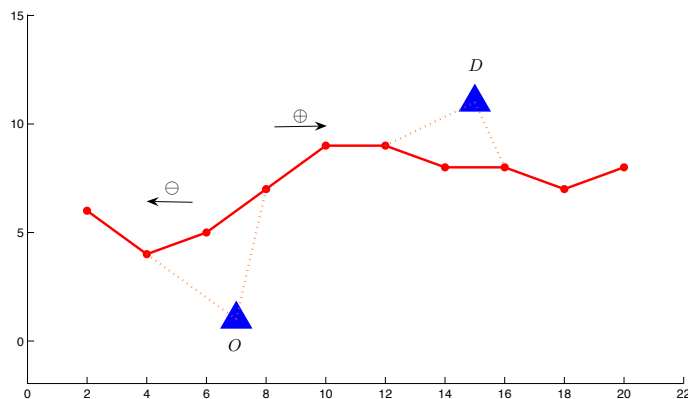


Figura 5.14: Caminos mínimos entre dos centroides en una poligonal

5.3.4. Accesos al alineamiento

En un sistema de metro los usuarios entran a través de las estaciones y en una autovía, por las incorporaciones. Los nodos de la poligonal representan, según el problema, estaciones o incorporaciones.

Se supone que no todos los nodos de la poligonal son estaciones/incorporaciones, y que dichos nodos están localizados en segmentos de la rejilla. Para cada nodo se deberá decidir si es o no una incorporación. El presupuesto restringirá el número de incorporaciones que tiene la vía.

El coste de las incorporaciones dependerá del problema que se esté tratando. Para líneas de metro este coste llega a suponer 2/3 del presupuesto total.

En el problema de una autovía, el acceso desde un centroide a una incorporación se realiza en línea recta. En el caso de que dicha línea corte a algún obstáculo, se bordeará dicho obstáculo de forma que se minimize el tiempo en la incorporación. Para calcular la distancia de la incorporación, que es proporcional al tiempo, se utiliza la expresión (5.24). Se puede ver un ejemplo en la figura 5.15.

$$d(c_i, p_j) = d(c_i, l_1) + \text{mín}(a, b) + d(l_2, p_j), \quad \forall c_i \in \mathcal{C}, \forall n_j \in \mathcal{N} \quad (5.24)$$

donde: $a = d(l_1, v_1) + d(v_1, v_2) + d(v_2, l_2)$

$b = d(l_1, v_4) + d(v_4, v_3) + d(v_3, l_2)$

$d(q_1, q_2)$ representa la distancia euclídea entre dos puntos $q_1, q_2 \in \mathbb{R}^2$

En el problema de un sistema de metro, el acceso desde un centroide a una estación de metro no se ve afectado por los obstáculos, ya que estos representan dificultades geotécnicas para los túneles y no dificultan el acceso en la superficie. Para calcular la distancia de la incorporación se utiliza la expresión (5.25), que mide la distancia Manhattan del centroide a la estación. Se puede ver un ejemplo en la figura 5.16.

$$d(c_i, p_j) = |x_{c_i} - x_{p_j}| + |y_{c_i} - y_{p_j}| \quad \forall c_i \in \mathcal{N}_c, \forall p_j \in \mathcal{N}_i \quad (5.25)$$

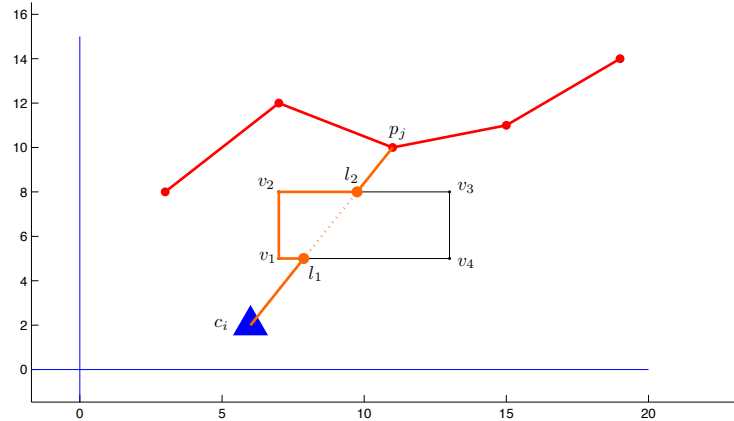


Figura 5.15: Acceso desde un centroide a una incorporación de autopista bordeando un obstáculo

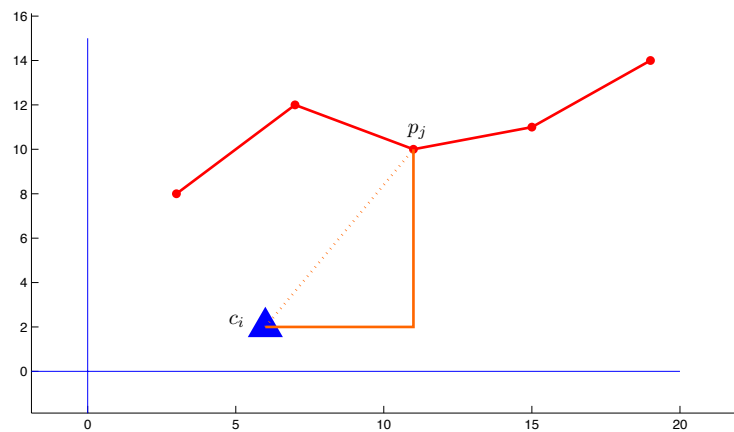


Figura 5.16: Acceso desde un centroide a una estación de metro empleando la distancia Manhattan

Capítulo 6

Resolución del problema mediante un Algoritmo Genético

6.1. Descripción del algoritmo de resolución

6.2. Descripción del Algoritmo Genético

- 6.2.1. Representación cromosómica
- 6.2.2. Población inicial
- 6.2.3. Medida de evaluación
- 6.2.4. Criterio de selección de individuos
- 6.2.5. Operaciones de recombinación y mutación
- 6.2.6. Estructura del algoritmo

6.3. Operaciones de recombinación

- 6.3.1. Selección de los padres
- 6.3.2. Cruce simple
- 6.3.3. Cruce de dos puntos

6.4. Operaciones de mutación

- 6.4.1. Mutación uniforme
 - 6.4.2. Mutación recta
 - 6.4.3. Mutación no uniforme
 - 6.4.4. Mutación de incorporaciones
-

En este capítulo se explica en detalle el algoritmo heurístico que se ha usado para resolver el problema del alineamiento de una autovía o una línea de metro con obstáculos. Este método de resolución optimiza una función unidimensional cuya evaluación requiere resolver un problema de optimización.

En el primer apartado se describe el algoritmo empleado en la optimización del problema unidimensional. En el segundo apartado se detalla el algoritmo genético (AG) usado para aproximar la función objetivo y los elementos de los que consta, sin entrar a detallar las operaciones de recombinación y mutación. En el tercer apartado se exponen las diferentes operaciones de recombinación del algoritmo genético. En el último apartado se detallan las operaciones de mutación de las que consta el algoritmo genético.

6.1. Descripción del algoritmo de resolución

Las vías generadas están condicionadas por la orientación de la rejilla, Θ (el ángulo de los segmentos que componen la rejilla), por lo que ésta debe modificarse a lo largo del algoritmo. Se define $\mathcal{Z}(\Theta)$ como el alineamiento óptimo dentro de la rejilla de orientación Θ . El objetivo que se persigue es optimizar el valor de $\mathcal{Z}(\Theta)$ sujeto a todas las orientaciones posibles $\Theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$.

Este problema de optimización consta de una única variable, Θ , esto es, es un problema de optimización unidimensional con una restricción de intervalo. En la literatura se han propuesto varios algoritmos para resolver este tipo de problemas. En este proyecto hemos adaptado el algoritmo de la *sección áurea* [Ven02], que ha sido propuesto para optimizar funciones con un único óptimo local (funciones unimodales).

La característica esencial de este método es que solamente emplea evaluaciones funcionales (método libre de cálculo de derivadas). Para poder evaluar la función

objetivo $\mathcal{Z}(\Theta)$ se debe resolver el problema de optimización de un alineamiento sobre una rejilla dada. Este valor se aproxima aplicando el algoritmo genético descrito en la sección 6.2.

Para evitar pendientes infinitas se ha restringido el intervalo de rejillas factibles quedando el problema en:

$$\begin{aligned} & \text{Optimizar} && \mathcal{Z}(\Theta) \\ & \text{sujeto a:} && \Theta \in \left[-\frac{\pi}{2} - \varepsilon, \frac{\pi}{2} - \varepsilon\right] \end{aligned} \tag{6.1}$$

Este algoritmo es uno de los mejores de la familia de métodos para reducir intervalos y es sencillo de implementar. El algoritmo de la sección áurea reduce el intervalo en la misma fracción en cada iteración. Los intervalos se derivan de la *razón áurea*.

La adaptación del método de la sección áurea a este problema para una formulación de maximización viene recogida en la tabla 6.1.

6.2. Descripción del Algoritmo Genético

El algoritmo genético usado en este proyecto consta de los siguientes elementos, que se desarrollarán a continuación.

- Una representación cromosómica, (Λ, Φ)
- Una población inicial, P_0 , de tamaño M
- Una medida de evaluación (o adecuación), σ
- Un criterio de selección / reemplazo de individuos
- Varias operaciones de recombinación o cruce
- Varias operaciones de mutación

Tabla 6.1: Algoritmo de la sección áurea para un problema de maximización

Paso 1.	Elegir $\Theta^{min} = -\frac{\pi}{2} + \varepsilon$, $\Theta^{max} = \frac{\pi}{2} - \varepsilon$ $\tau := 0,38197$ $\varepsilon := \text{tolerancia} := (\Delta\Theta)_{final}/(\Theta^{max} - \Theta^{min})$ $N := \text{número de iteraciones} := -2,078 \ln \varepsilon$ $i := 1$
Paso 2.	$\Theta_1 := (1 - \tau)\Theta^{min} + \tau\Theta^{max}$; $\mathcal{Z}_1 := \mathcal{Z}(\Theta_1)$ $\Theta_2 := \tau\Theta^{min} + (1 - \tau)\Theta^{max}$; $\mathcal{Z}_2 := \mathcal{Z}(\Theta_2)$ Los puntos son equidistantes de los extremos
Paso 3.	Si $i < N$ Si $\mathcal{Z}_1 < \mathcal{Z}_2$ $\Theta^{min} := \Theta_1$; $\Theta_1 := \Theta_2$; $\mathcal{Z}_1 := \mathcal{Z}_2$ $\Theta_2 := \tau\Theta^{min} + (1 - \tau)\Theta^{max}$; $\mathcal{Z}_2 := \mathcal{Z}(\Theta_2)$ Si $\mathcal{Z}_1 > \mathcal{Z}_2$ $\Theta^{max} := \Theta_2$; $\Theta_2 := \Theta_1$; $\mathcal{Z}_2 := \mathcal{Z}_1$ $\Theta_1 := (1 - \tau)\Theta^{min} + \tau\Theta^{max}$; $\mathcal{Z}_1 := \mathcal{Z}(\Theta_1)$ $i := i + 1$ Ir al Paso 3

6.2.1. Representación cromosómica

Una vía, o individuo de la población, se va a representar mediante dos cromosomas. Para un alineamiento representado por n puntos en una rejilla, el genotipo correspondiente está formado por 2 cromosomas de n genes cada uno. El genotipo se puede definir como se indica en la ecuación (6.2).

$$\begin{aligned}
\Lambda &= [\lambda_1, \lambda_2, \dots, \lambda_n] \\
\Phi &= [\phi_1, \phi_2, \dots, \phi_n]
\end{aligned} \tag{6.2}$$

donde: Λ es el cromosoma que representa los nodos del alineamiento en una rejilla determinada

$$\lambda_i \in [x_{min}^i, x_{max}^i]$$

Φ es el cromosoma que representa si el nodo es o no una incorporación

$$\phi_i \in \{0, 1\}$$

En el cromosoma Λ cada gen representa el valor en el eje X de un nodo de la vía. Inicialmente el problema consistía en seleccionar puntos en el plano, con coordenadas x e y . Gracias a la rejilla, los nodos quedan completamente determinados por su valor en el eje X . Su valor en el eje Y se calcula según el segmento de rejilla correspondiente, como se explica en el apartado 5.3.1.

En el cromosoma Φ cada gen indica si el nodo correspondiente es incorporación o no. Un valor de $\phi_i = 1$ indica que el nodo i es una incorporación, mientras que un valor de $\phi_i = 0$ indica que dicho nodo no lo es. Los nodos $\phi_i = 0$ sirven para evitar obstáculos, mientras que los nodos $\phi_i = 1$ sirven para incorporar usuarios al alineamiento, además de para evitar obstáculos. Se considera que los genes ϕ_1 y ϕ_n son incorporaciones a la vía, por lo que $\phi_1 = \phi_n = 1$.

6.2.2. Población inicial

La población inicial P consiste en un determinado número M de segmentos rectilíneos que recorren uniformemente el espacio de soluciones y cada uno de los cuales representa una vía. Esta población se genera teniendo en cuenta que todos los cromosomas representan vías factibles. El número de individuos de la población inicial, M , se mantendrá constante a lo largo del algoritmo genético.

La generación de la población inicial es una heurística que se realiza en dos partes. En la primera parte se generan las posiciones de los nodos de las vías, es decir, los cromosomas Λ de las vías de la población inicial. A continuación se generan las

incorporaciones, es decir, los cromosomas Φ .

Se generarán más de M vías, por lo que habrá que evaluar todos los individuos generados y descartar aquellos que tengan los peores valores de adecuación (valores de la función objetivo empleada en el diseño).

Para poder generar los cromosomas Λ de la población inicial es necesario establecer los valores de los siguientes parámetros:

n : Número de segmentos de la rejilla.

L : Longitud máxima que se puede construir según el presupuesto. Es un valor dado por el usuario.

d : Longitud de la diagonal del rectángulo que encierra los centroides.

N : Número de alineamientos totales.

Δn : Número aproximado de segmentos que intersecta una vía de la población inicial. Su valor viene dado por:

$$\Delta n = \frac{0,7 \cdot L}{d/(n-1)} \quad (6.3)$$

r : Pares de segmentos que se pueden formar. Su valor viene dado por:

$$r = n - \Delta n \quad (6.4)$$

Δd : Incremento de distancia entre un punto elegido para ser origen o destino de las vías generadas. Su valor viene dado por:

$$\Delta d = \sqrt{\frac{r}{N} \cdot \left(\frac{d}{2}\right)^2} \quad (6.5)$$

El algoritmo empleado para la generación de los cromosomas Λ se muestra en la tabla 6.2.

Los obstáculos existentes provocan que algunas de las soluciones generadas no sean factibles, por lo que dichas soluciones se descartan. Esto causa una disminución en el número de vías generadas.

Tabla 6.2: Algoritmo de generación de los cromosomas Λ de la población inicial

```

Inicio (GeneracionNodosPoblacionInicial)
  incX := incD · cos( $\Theta$ ) (* Avance en el eje X)
  via := 1 (* Número de vías generadas)
  Para i := 1 hasta m hacer
    xOrigen := xMin(rejilla, i)
    Mientras xOrigen  $\leq$  xMax(rejilla, i)
      yOrigen := calcularY(xOrigen, i)
      xDestino := xMin(rejilla, i+incN)
      Mientras xDestino  $\leq$  xMax(rejilla, i+incN)
        yDestino := calcularY(xDestino, i)
        Si la vía es factible
          pVia := añadirVia(xOrigen, xDestino)
          via := via + 1
        Fin Si
      xDestino := xDestino + incX
    Fin Mientras
  xOrigen := xOrigen + incX
Fin Mientras
Fin Para
Fin

```

Se puede ver un ejemplo de las vías factibles y las no factibles generadas mediante el algoritmo de la tabla 6.2 en la figura 6.1.

Una vez generados los cromosomas Λ se pasa a generar los cromosomas Φ , utilizando el algoritmo que se muestra en la tabla 6.3. Para generar las incorporaciones hay que tener en cuenta dos parámetros, p_1 y p_2 .

p_1 : Es la probabilidad de poner una incorporación en un nodo y, por tanto, debe tomar valores pequeños.

p_2 : Es la proporción máxima de incorporaciones permitidas que se pueden poner en una vía de la población inicial.

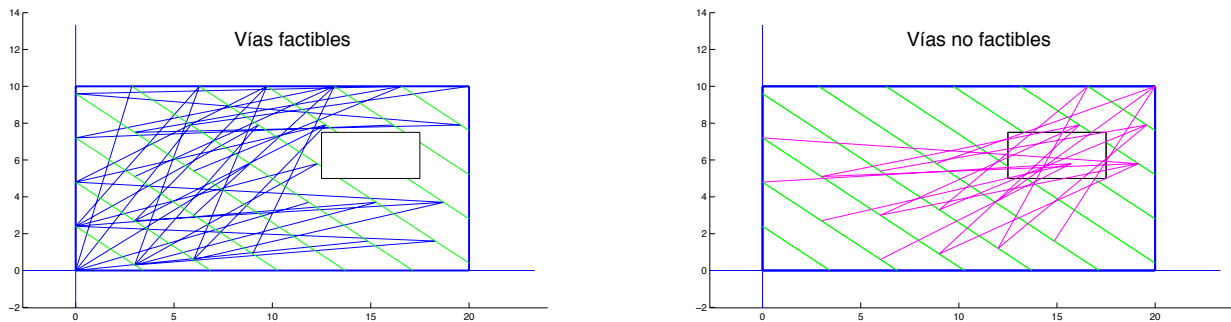


Figura 6.1: Ejemplo de vías generadas factibles y no factibles

Tabla 6.3: Algoritmo de generación de los cromosomas Φ de la población inicial

```

Inicio (GeneracionIncorporacionesPoblacionInicial)
  Para todo pVia:pVias (* Todos los cromosomas  $\Lambda$ )
    incorporaciones := 1
    Para todo nodo:pVia (* Todos los nodos de la vía pVia)
      Si nodo := primero O nodo := último
        iVia(nodo) := 1
        incorporaciones := incorporaciones + 1
      Si no
        Si valorAleatorio <  $p_1$  Y incorporaciones <  $p_2 \cdot \text{maxIncorporaciones}$ 
          iVia(nodo) := 1
          incorporaciones := incorporaciones + 1
        Si no
          iVia(nodo) := 0
        Fin Si
      Fin Si
    Fin Para
  Fin Para
Fin
  
```

Otra opción, además de la anterior, es que el usuario proporcione un conjunto de alineamientos preliminares, tratando de que las vías iniciales ya sean *buenas* vías y el algoritmo intente mejorar soluciones establecidas por los planificadores de transporte.

6.2.3. Medida de evaluación

El valor de adecuación de un individuo es un número positivo que mide la bondad de dicho individuo. El modelo se ha formulado mediante un problema de optimización y emplea como medida de adecuación el valor de la función objetivo. En el problema tratado en este proyecto se han propuesto tres criterios de diseño, dos formulados como minimizar un cierto índice y otro como un problema de maximización. La adecuación, denotada por σ , se define como uno de los tres criterios de diseño introducidos modificándolos en caso necesario.

Criterio 1: Maximización del cubrimiento de la nueva infraestructura de transporte (ver ecuación (5.7) en la página 42).

$$\sigma = \sum_{w \in \mathcal{W}} g_w^N \quad (6.6)$$

Criterio 2: Minimización de la congestión de la infraestructura de transporte existente (ver ecuación (5.8) en la página 43).

$$\sigma = \frac{1}{\sum_{w \in \mathcal{W}} u_w^E g_w^E} \quad (6.7)$$

Criterio 3: Minimización del tiempo total de transporte en la infraestructura nueva y la existente (ver ecuación (5.9) en la página 43).

$$\sigma = \frac{1}{\sum_{w \in \mathcal{W}} (u_w^E g_w^E + u_w^N g_w^N)} \quad (6.8)$$

6.2.4. Criterio de selección de individuos

En una generación se crean n_d descendientes, por lo que el tamaño de la población crece. Para mantener constante el número de individuos de una población hay que seleccionar qué individuos permanecerán en ella. Si el tamaño de la población inicial es M , después de cada generación habrá $M + n_d$ individuos en la población. Se utiliza un *Mecanismo de Selección Avaro*, como propone [SY99], para seleccionar en cada iteración los M mejores individuos del total de la población.

6.2.5. Operaciones de recombinación y mutación

Las operaciones de recombinación y mutación usadas en este proyecto se basan en las descritas en [KJS05].

Las operaciones de recombinación, como se verá más detalladamente en el apartado 6.3, sirven para generar nuevos individuos, descendientes, a partir de dos individuos, padres, de la población. Los nuevos descendientes creados no podrán ser elegidos como padres hasta la siguiente iteración.

Las operaciones de mutación sirven para modificar los individuos existentes en la población. En este problema, se aplicarán operaciones de mutación únicamente sobre los descendientes creados en cada generación. Las operaciones de mutación se ven con más detalle en el apartado 6.4.

6.2.6. Estructura del algoritmo

La estructura del algoritmo desarrollado en este proyecto, basada en la propuesta en [SY99], se puede ver en la tabla 6.4.

Se elige un número de iteraciones n_g a realizar y en cada una de ellas se efectúan dos conjuntos de operaciones:

- Generar nuevos individuos

- Mutar (transformar) algunos de estos descendientes

Tabla 6.4: Algoritmo Genético

Inicio (AlgoritmoGenetico)

$P := \text{PoblacionInicial}(M)$ (* Generar población inicial)

Para $i := 1$ **hasta** M **hacer**

$\sigma_i := \text{evaluar}(P_i)$ (* Evaluar la población inicial)

Fin Para

Para $t := 1$ **mientras** $t < n_g$ **hacer** (* Para cada generación \equiv n° de iteraciones)

Para $i := 1$ **mientras** $i < n_d$ **incremento** $i := i + 2$ **hacer**

$(x, y) := \text{seleccionarPadres}(P, \sigma)$ (* Seleccionar dos padres)

$(D_i, D_{i+1}) := \text{recombinar}(x, y)$ (* Generar descendientes)

Fin Para

Para $i := 1$ **mientras** $i < n_d$ **hacer** (* Para cada descendiente)

mutar := decidirMutar() (* Decidir si mutar)

Si mutar **entonces**

$D_i := \text{mutar}(D_i)$ (* Mutar descendiente)

Fin Si

$\sigma_i := \text{evaluar}(P_i)$ (* Evaluar descendiente)

Fin Para

$P := \text{seleccionarMejores}(P \cup D)$ (* Seleccionar las M mejores soluciones)

Fin Para

Devolver la mejor solución de P

Fin

6.3. Operaciones de recombinación

Una operación de recombinación toma dos genotipos (padres) y genera descendientes a partir de ellos.

En este proyecto se utilizan dos operaciones de recombinación diferentes:

- Cruce simple
- Cruce de dos puntos

En las operaciones de recombinación hay que tener en cuenta la factibilidad de los descendientes, es decir, puede darse el caso de que los padres sean factibles, pero su recombinación deje de serlo. Esto puede suceder por diferentes motivos:

- El nuevo descendiente atraviesa algún obstáculo.
- La vía descendiente tiene un mayor coste que el presupuesto disponible.

Para evitarlo, se comprobará la factibilidad de cada descendiente, y en el caso de que alguno no sea factible, se descartará dicho descendiente.

6.3.1. Selección de los padres

Para poder aplicar una operación de recombinación, en primer lugar hay que elegir los padres que se cruzarán. De los diversos métodos existentes, en este proyecto se ha elegido el *Método de la Ruleta* [SY99], que permite simular una variable aleatoria discreta.

En primer lugar se asigna a cada posible padre una probabilidad proporcional a su adecuación, σ , como se puede ver en la ecuación (6.9).

$$p_i = \frac{\sigma_i}{\sum_{i=1}^M \sigma_i} \quad \forall i \in \{1, \dots, M\} \quad (6.9)$$

Después se asigna a cada posible padre i un intervalo entre 0 y 1, como se puede ver en la ecuación (6.10).

$$\begin{aligned} [\min_i, \max_i] &= [0, p_i] && \text{si } i = 1 \\ (\min_i, \max_i) &= \left(\sum_{j=1}^{i-1} p_j, \sum_{j=1}^i p_j \right) && \forall i \in \{2, \dots, M\} \end{aligned} \quad (6.10)$$

A continuación se generan dos variables aleatorias independientes, p_1 y p_2 , en el intervalo $[0, 1]$ tales que $p_1 = r_c[0, 1]$ ¹ y $p_2 = r_c[0, 1]$. El primer padre, (Λ_i, Φ_i) , es aquel para el que la variable aleatoria p_1 esté en su intervalo de valores, calculado con la ecuación (6.10). Si el segundo padre, (Λ_j, Φ_j) , resulta ser el mismo que el primero ($p_1 = p_2$), se genera otra variable aleatoria p_2 tal que $p_2 = r_c[0, 1]$, hasta obtener dos padres diferentes.

6.3.2. Cruce simple

El cruce simple consiste en dividir en dos cada genotipo padre, seleccionando aleatoriamente un gen, e intercambiar las partes resultantes de cada genotipo padre, cromosoma a cromosoma, para generar dos descendientes, cada uno con la primera parte de un padre y la segunda del otro.

Más formalmente, dados dos padres (Λ_i, Φ_i) y (Λ_j, Φ_j) , con $\Lambda_i = [\lambda_1^i, \lambda_2^i, \dots, \lambda_n^i]$, $\Phi_i = [\phi_1^i, \phi_2^i, \dots, \phi_n^i]$, $\Lambda_j = [\lambda_1^j, \lambda_2^j, \dots, \lambda_n^j]$ y $\Phi_j = [\phi_1^j, \phi_2^j, \dots, \phi_n^j]$, generados aleatoriamente empleando el *Método de la ruleta* (ver apartado 6.3.1) y una variable aleatoria discreta uniforme, k , en el intervalo $[1, n]$ tal que $k = r_d[1, n]$. Se generan dos descendientes según se puede ver en la ecuación (6.11).

¹Se denotará con $r_d[x_1, x_2]$ una variable aleatoria discreta uniforme que toma los valores enteros comprendidos en el intervalo $[x_1, x_2]$, y con $r_c[x_1, x_2]$ una variable aleatoria continua uniforme en el intervalo $[x_1, x_2]$

$$\begin{aligned} \Lambda'_i &= [\lambda_1^i, \lambda_2^i, \dots, \lambda_k^i, \lambda_{k+1}^j, \lambda_{k+2}^j, \dots, \lambda_{n-1}^j, \lambda_n^j] \\ \Phi'_i &= [\phi_1^i, \phi_2^i, \dots, \phi_k^i, \phi_{k+1}^j, \phi_{k+2}^j, \dots, \phi_{n-1}^j, \phi_n^j] \end{aligned} \tag{6.11}$$

$$\begin{aligned} \Lambda'_j &= [\lambda_1^j, \lambda_2^j, \dots, \lambda_k^j, \lambda_{k+1}^i, \lambda_{k+2}^i, \dots, \lambda_{n-1}^i, \lambda_n^i] \\ \Phi'_j &= [\phi_1^j, \phi_2^j, \dots, \phi_k^j, \phi_{k+1}^i, \phi_{k+2}^i, \dots, \phi_{n-1}^i, \phi_n^i] \end{aligned}$$

En las figuras 6.2 y 6.3 se puede ver un ejemplo de Cruce Simple.

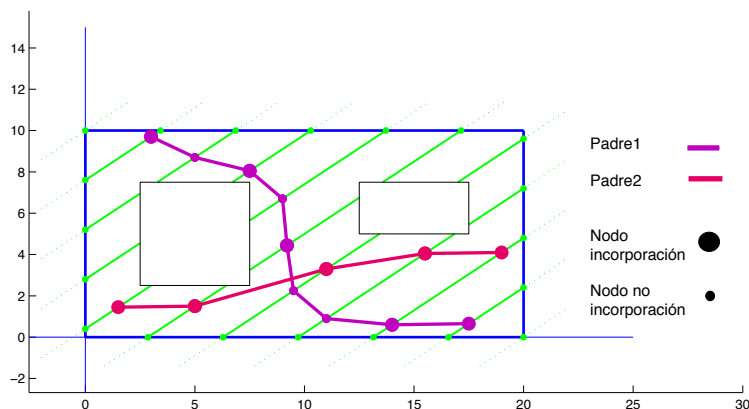


Figura 6.2: Cruce Simple: Padres

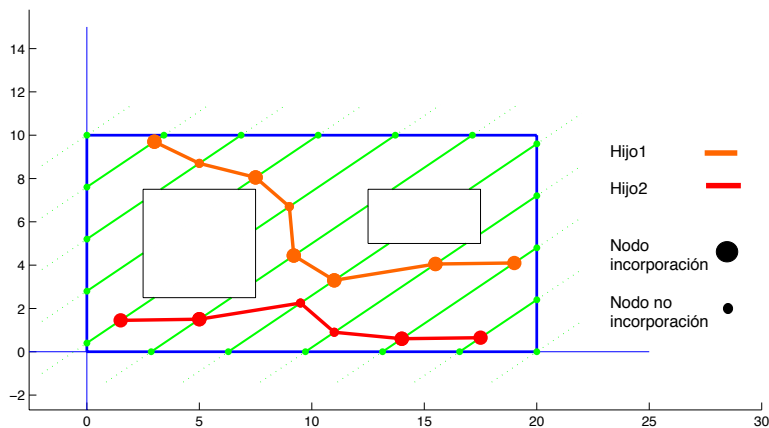


Figura 6.3: Cruce Simple: Descendientes

6.3.3. Cruce de dos puntos

El cruce de dos puntos es similar al cruce simple, pero selecciona dos genes, dividiendo los genotipos padre en tres partes, cromosoma a cromosoma. Los descendientes que se generan tienen la primera y la tercera parte de un padre, y la parte central del otro.

Dados dos padres (Λ_i, Φ_i) y (Λ_j, Φ_j) , con $\Lambda_i = [\lambda_1^i, \lambda_2^i, \dots, \lambda_n^i]$, $\Phi_i = [\phi_1^i, \phi_2^i, \dots, \phi_n^i]$, $\Lambda_j = [\lambda_1^j, \lambda_2^j, \dots, \lambda_n^j]$ y $\Phi_j = [\phi_1^j, \phi_2^j, \dots, \phi_n^j]$, seleccionados aleatoriamente como se puede ver en el apartado 6.3.1, se generan dos variables aleatorias discretas uniformes k y l tales que $k = r_d[1, n]$, $l = r_d[1, n]$, $k \neq l$ y $(k + 1) < l$. Se generan dos descendientes según se puede ver en la ecuación (6.12).

$$\begin{aligned}\Lambda'_i &= [\lambda_1^i, \dots, \lambda_k^i, \lambda_{k+1}^j, \dots, \lambda_{l-1}^j, \lambda_l^i, \dots, \lambda_n^i] \\ \Phi'_i &= [\phi_1^i, \dots, \phi_k^i, \phi_{k+1}^j, \dots, \phi_{l-1}^j, \phi_l^i, \dots, \phi_n^i]\end{aligned}\tag{6.12}$$

$$\begin{aligned}\Lambda'_j &= [\lambda_1^j, \dots, \lambda_k^j, \lambda_{k+1}^i, \dots, \lambda_{l-1}^i, \lambda_l^j, \dots, \lambda_n^j] \\ \Phi'_j &= [\phi_1^j, \dots, \phi_k^j, \phi_{k+1}^i, \dots, \phi_{l-1}^i, \phi_l^j, \dots, \phi_n^j]\end{aligned}$$

En las figuras 6.4 y 6.5 se puede ver un ejemplo de Cruce de dos Puntos.

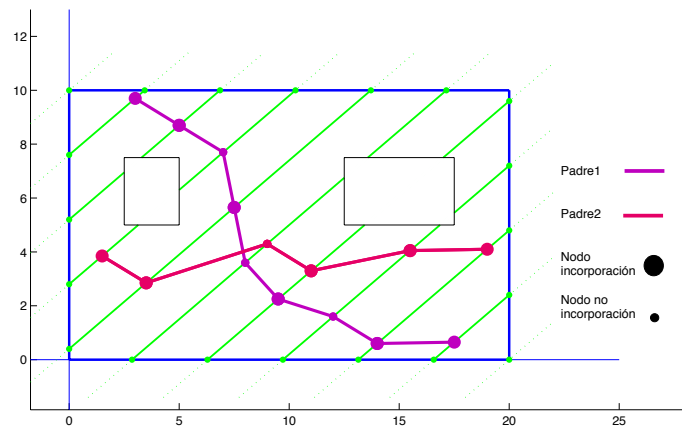


Figura 6.4: Cruce de dos Puntos: Padres

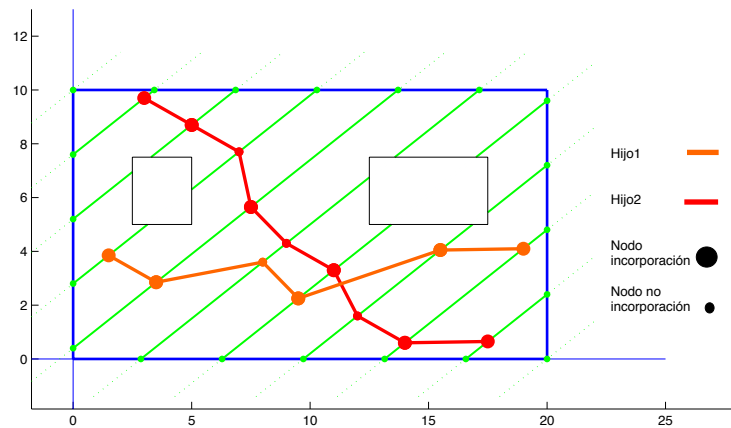


Figura 6.5: Cruce de dos Puntos: Descendientes

6.4. Operaciones de mutación

Las operaciones de mutación son operaciones que modifican un genotipo dado.

En este proyecto se usan las siguientes operaciones de mutación:

- Mutación Uniforme
- Mutación Recta
- Mutación No Uniforme
- Mutación de Incorporaciones

Las tres primeras operaciones de mutación modifican únicamente el cromosoma Λ , manteniendo el mismo cromosoma Φ que tuvieran. La cuarta, modifica el cromosoma Φ , manteniendo el cromosoma Λ .

La probabilidad de que se produzca una mutación de un descendiente es muy pequeña, ya que no se pretende introducir grandes modificaciones en los descendientes, sino que algunos de ellos tengan características que los padres no incorporaban.

En cada operación de mutación no se muta un único gen, sino que se muta un intervalo de genes $\{i, \dots, j\}$. Esto se debe a que computacionalmente resulta muy costoso calcular la adecuación de un genotipo, ya que para ello es necesario obtener un modelo de uso de la vía correspondiente con dicho genotipo.

En las operaciones de mutación hay que tener en cuenta la factibilidad de los individuos mutados, ya que puede que éstos tengan una longitud o un número de incorporaciones mayores que los permitidos. Para solucionar este problema, los individuos mutados que no sean factibles se descartarán.

6.4.1. Mutación uniforme

La mutación uniforme consiste en mover uniformemente, dentro del subintervalo de x al que pertenece el alelo original, los valores de un intervalo de genes del cromosoma Λ de un individuo.

Dado un cromosoma $\Lambda = [\lambda_1, \lambda_2, \dots, \lambda_n]$ se generan dos variables aleatorias discretas independientes i y j tales que $i = r_d[0, n]$ y $j = r_d[0, n]$, $i \neq j$ y $i < j$. Se intercambiarán los genes $\lambda_l, \forall l \in \{i, \dots, j\}$, por λ'_l según se puede ver en la ecuación (6.13).

$$\lambda'_l = r_c[x_{min}^l, x_{max}^l] \quad (6.13)$$

donde: x_{min}^l es el máximo del subintervalo de x al que pertenece λ_l
 x_{max}^l es el mínimo del subintervalo de x al que pertenece λ_l

El cromosoma Φ del individuo mutado coincide con el del original.

En la figura 6.6 se puede ver un ejemplo de mutación uniforme. En este ejemplo, se mutan los nodos 4, 5 y 6 de la vía.

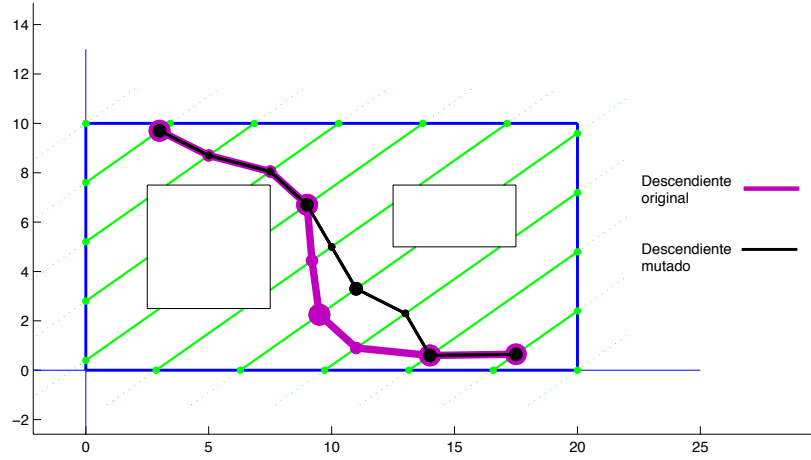


Figura 6.6: Ejemplo de mutación uniforme

6.4.2. Mutación recta

La mutación recta consiste en convertir en un segmento recto la poligonal existente entre dos incorporaciones consecutivas de un individuo, seleccionadas al azar.

Dados un cromosoma $\Lambda = [\lambda_1, \lambda_2, \dots, \lambda_n]$ y un cromosoma $\Phi = [\phi_1, \phi_2, \dots, \phi_n]$, se genera una variable aleatoria discreta i tal que $i = r_d[0, n]$ y $\phi_i = 1$. Una vez generada i se toma una variable discreta j tal que $\phi_j = 1$ y $\phi_l = 0, \forall l \in \{i + 1, \dots, j - 1\}$. Se intercambian los genes $\lambda_l, \forall l \in \{i + 1, \dots, j - 1\}$, por λ'_l según se puede ver en la ecuación (6.14).

$$\lambda'_l = \begin{cases} \lambda_i + (l - i) \left(\frac{\lambda_j - \lambda_i}{j - i} \right), & \text{si } m < 0 \\ \lambda_i - (l - i) \left(\frac{\lambda_j - \lambda_i}{j - i} \right), & \text{si } m > 0 \end{cases} \quad (6.14)$$

El cromosoma Φ del individuo mutado coincide con el del original.

En la figura 6.7 se puede ver un ejemplo de mutación recta.

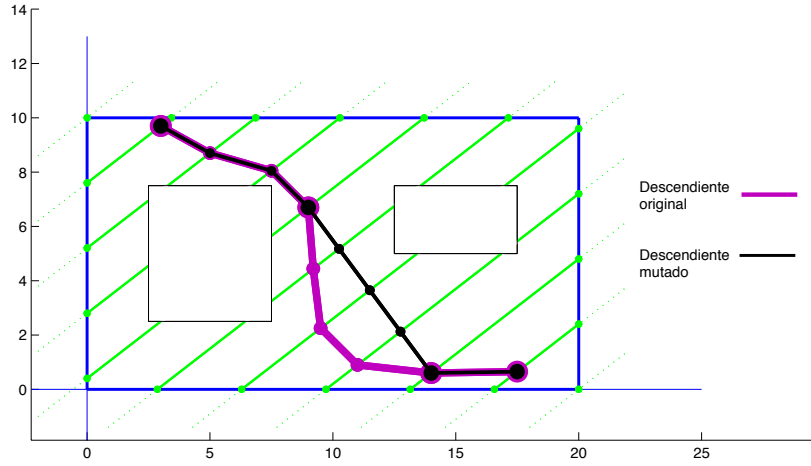


Figura 6.7: Ejemplo de mutación recta

6.4.3. Mutación no uniforme

La mutación no uniforme consiste en mover, dentro del subintervalo de x al que pertenece el alelo original, los valores de un intervalo de genes del cromosoma Λ de un individuo. Estos movimientos podrán ser más amplios en las generaciones iniciales y más limitados en las generaciones posteriores.

Dado un cromosoma $\Lambda = [\lambda_1, \lambda_2, \dots, \lambda_n]$ se generan dos variables aleatorias discretas independientes i y j tales que $i = r_d[0, n]$ y $j = r_d[0, n]$, $i \neq j$ y $i < j$. Además, se genera un dígito binario aleatorio, b . Los genes λ_l , $\forall l \in \{i, \dots, j\}$, se intercambian por λ'_l dependiendo del valor de b , según se puede ver en la ecuación (6.15).

$$\lambda'_l = \begin{cases} \lambda_l - f(t, \lambda_l - x_{min}^l), & \text{si } b = 0 \\ \lambda_l + f(t, x_{max}^l - \lambda_l), & \text{si } b = 1 \end{cases} \quad (6.15)$$

donde: t es el número de la actual generación

x_{min}^l es el mínimo del subintervalo de x al que pertenece λ_l

x_{max}^l es el máximo del subintervalo de x al que pertenece λ_l

El cromosoma Φ del individuo mutado coincide con el del original.

La función $f(t, x)$ devuelve un valor aleatorio en el intervalo $[0, x]$ de tal forma que la probabilidad de que $f(t, x)$ se aproxime a 0 aumenta a medida que aumenta el valor de t . De esta forma se consigue que la operación de Mutación No Uniforme examine el espacio de búsqueda uniformemente en las generaciones iniciales y de forma muy local en generaciones posteriores, donde las soluciones que se examinan están más refinadas.

La función $f(t, x)$ se define en la ecuación (6.16).

$$f(t, x) = x \cdot r_c[0, 1] \cdot \left(1 - \frac{t}{n_g}\right)^\zeta \quad (6.16)$$

donde: t es el número de la generación actual

n_g es el número máximo de generaciones

ζ es un parámetro definido por el usuario que determina el grado de no uniformidad

En las figuras 6.8 y 6.9 se pueden ver dos ejemplos de mutación no uniforme. En la figura 6.8 la mutación se realiza en las primeras generaciones, y se puede ver cómo el rango de movimiento es muy amplio. En la figura 6.9 la mutación se realiza en las últimas generaciones, y el rango de movimiento de los nodos es mucho más limitado.

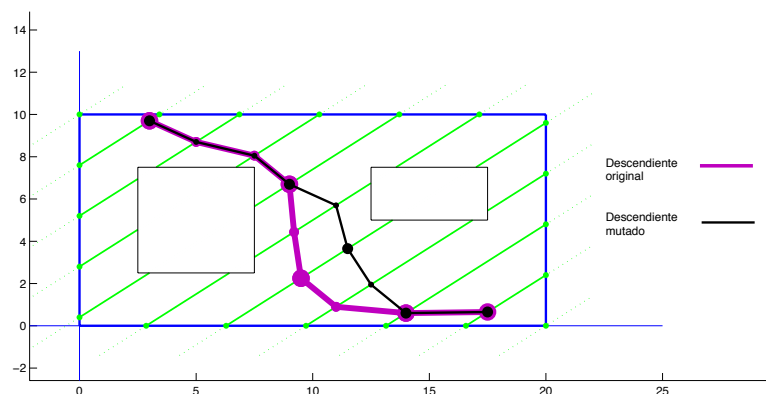


Figura 6.8: Ejemplo de mutación no uniforme durante las primeras generaciones

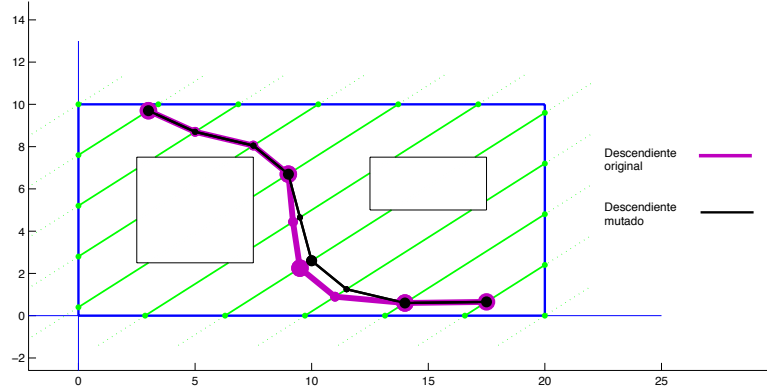


Figura 6.9: Ejemplo de mutación no uniforme durante las últimas generaciones

6.4.4. Mutación de incorporaciones

La mutación de incorporaciones consiste en modificar el cromosoma Φ de un individuo de forma que se puedan poner incorporaciones nuevas y se puedan eliminar incorporaciones existentes. Esta operación de mutación tiene la restricción de que no se pueden mutar el comienzo ni el final de una vía.

Dado un cromosoma $\Phi = [\phi_1, \phi_2, \dots, \phi_n]$ se generan dos variables aleatorias discretas independientes i y j tales que $i = r_d[a + 1, b - 1]$ y $j = r_d[a + 1, b - 1]$, $i \neq j$ y $i < j$, donde a es el primer segmento de la vía (Λ, Φ) y b es el segmento final de dicha vía. Además se generan $j - i + 1$ variables aleatorias discretas independientes b_l tales que $b_l = r_c[0, 1]$. Los genes $\phi_l, \forall l \in \{i, \dots, j\}$, se intercambian por ϕ'_l según se puede ver en la ecuación (6.17).

$$\phi'_l = \begin{cases} 1 - \phi_l, & \text{si } b_l \leq P_l \\ \phi_l, & \text{si } b_l > P_l \end{cases} \quad (6.17)$$

donde: P_l es la probabilidad de que un nodo cambie su tipo

El cromosoma Λ del individuo mutado coincide con el del original.

El valor P_l se calcula en función de la probabilidad de algún cambio, P_G , y disminuye a medida que aumenta el número de generaciones realizadas, t . En la ecuación (6.18) se puede ver cómo se calcula P_l .

$$P_l = 1 - \sqrt[t]{1 - P_G} \quad (6.18)$$

donde: t es el número de la generación actual

El valor de P_G es un valor fijo determinado por el usuario que debe ser muy pequeño, ya que la probabilidad de que se realicen cambios en las incorporaciones debe ser muy pequeña.

En la figura 6.10 se puede ver un ejemplo de mutación de incorporaciones. En él se puede observar que el descendiente mutado tiene incorporaciones que el descendiente original no tenía, y que dos de las que tenía inicialmente han desaparecido.

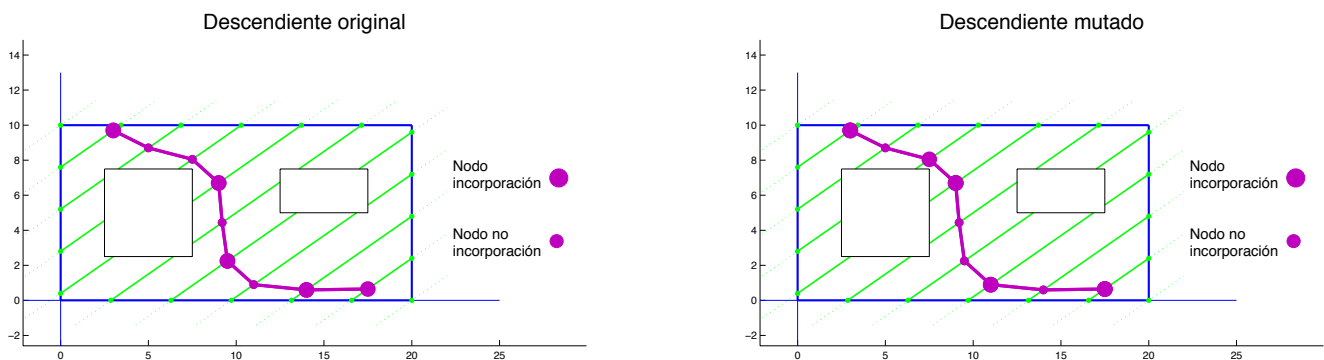


Figura 6.10: Ejemplo de mutación de incorporaciones

Capítulo 7

Herramienta LAHIT

7.1. Análisis

- 7.1.1. Especificación de requisitos
- 7.1.2. Diagramas de flujo de datos
- 7.1.3. Diccionario de Datos

7.2. Diseño

- 7.2.1. Diagramas de estructura

7.3. Codificación

- 7.3.1. Interfaz de entrada y salida
 - 7.3.2. Modelo matemático
 - 7.3.3. Resolución geométrica
 - 7.3.4. Algoritmo genético
 - 7.3.5. Sección Áurea
-

En este capítulo se describen el análisis y el diseño de LAHIT, siguiendo el paradigma de la *Programación Estructurada*.

7.1. Análisis

7.1.1. Especificación de requisitos

Los requisitos del sistema, basados en los objetivos expuestos en el apartado 2.1, son los siguientes:

- Debe encontrar una localización preliminar (casi) óptima para una nueva infraestructura lineal de transporte.
- Debe implementar el modelo matemático descrito en el capítulo 5, así como sus submodelos.
- Debe poder resolver los dos tipos de problemas que se plantean:
 - Una nueva autovía.
 - Una nueva línea de metro.
- Debe implementar los criterios de diseño de redes propuestos en el capítulo 5:
 - Maximización del cubrimiento en la nueva infraestructura de transporte.
 - Minimización de la congestión de la nueva infraestructura de transporte.
 - Minimización del tiempo total de transporte en la infraestructura nueva y la existente.
- Debe desarrollar una metodología de resolución del modelo matemático propuesto en el capítulo 5.
- Debe ser aplicable a problemas reales.
- Debe permitir una entrada y salida de datos amigable.

7.1.2. Diagramas de flujo de datos

En este apartado se muestran los *Diagramas de flujo de datos* (DFDs) que representan el sistema, los datos y los procesos que modifican dichos datos, siguiendo la técnica de análisis estructurado.

En las figuras 7.1, 7.2, 7.3, 7.4 y 7.5 se puede ver la descomposición de los procesos. Como se puede observar, el proceso 4 *Mostrar resultados* es un proceso primitivo.

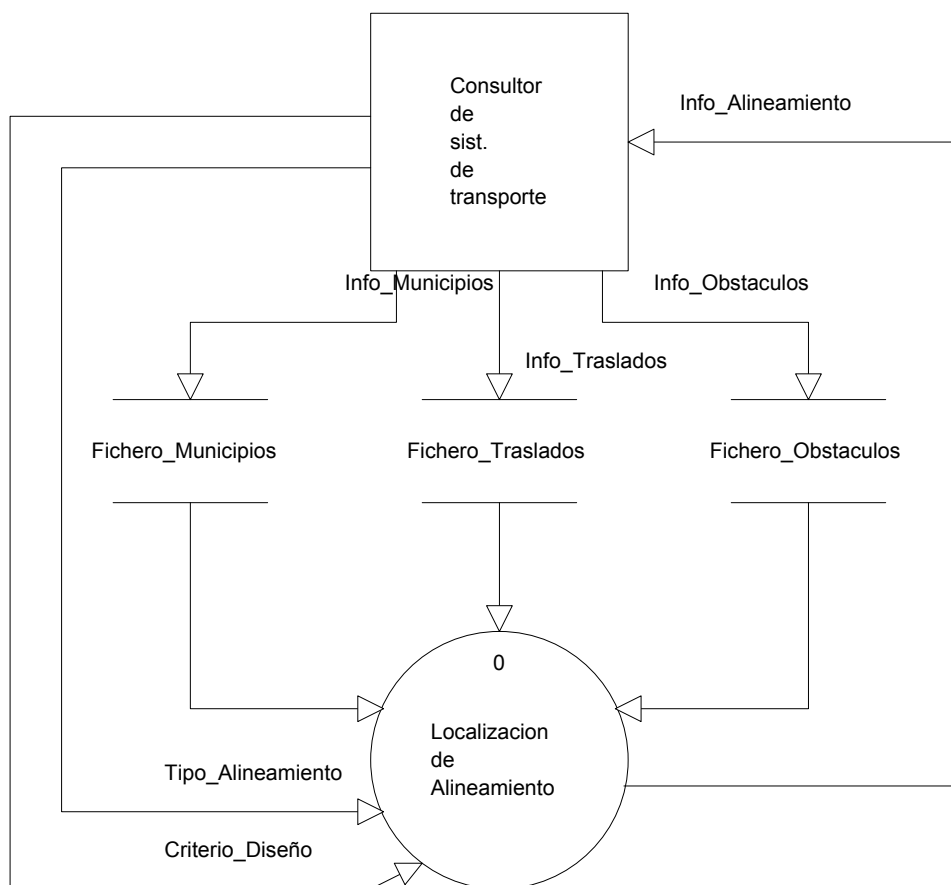


Figura 7.1: Diagrama de contexto

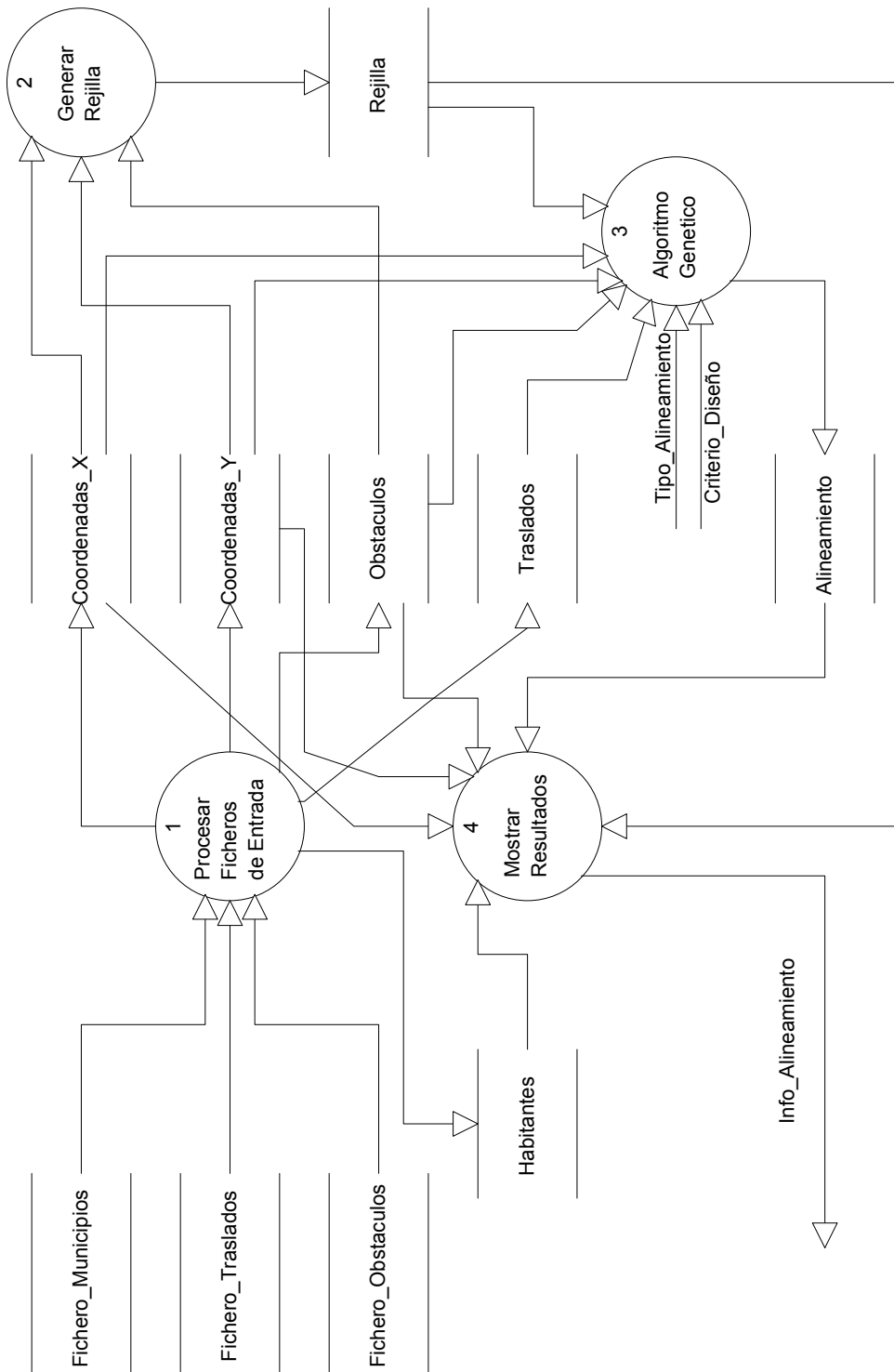


Figura 7.2: Diagrama 0. Localización de alineamientos

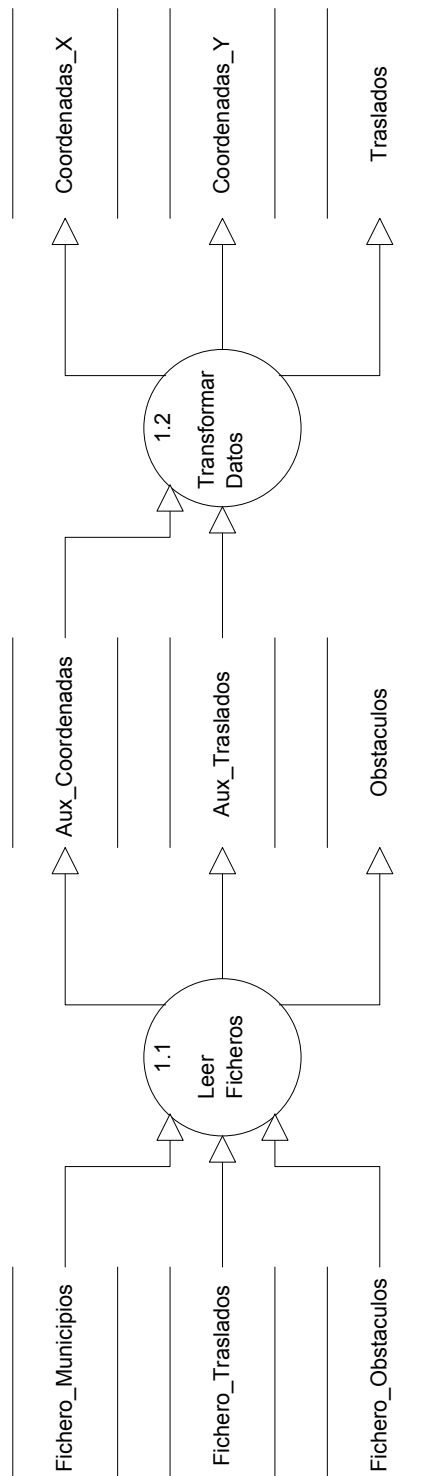


Figura 7.3: Diagrama 1. Procesar ficheros de entrada

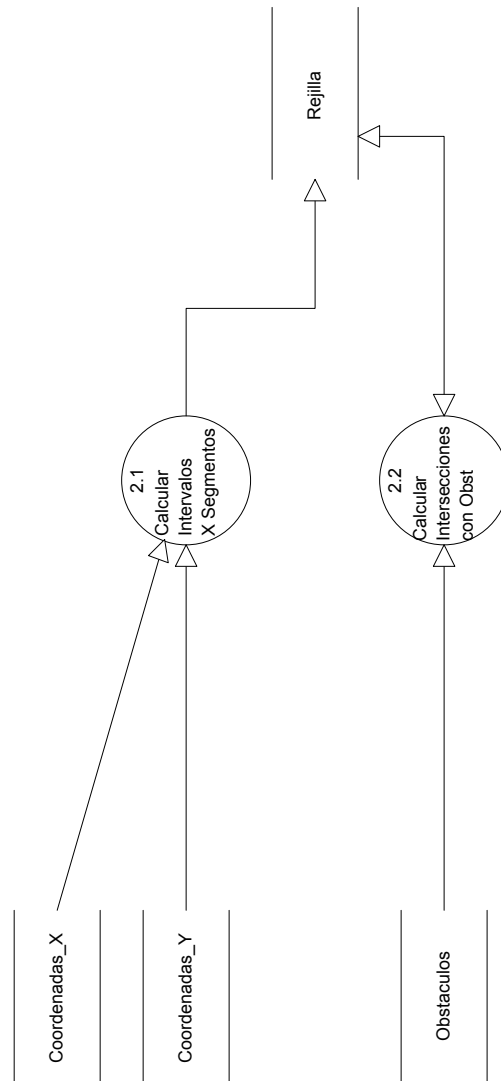


Figura 7.4: Diagrama 2. Generar rejilla

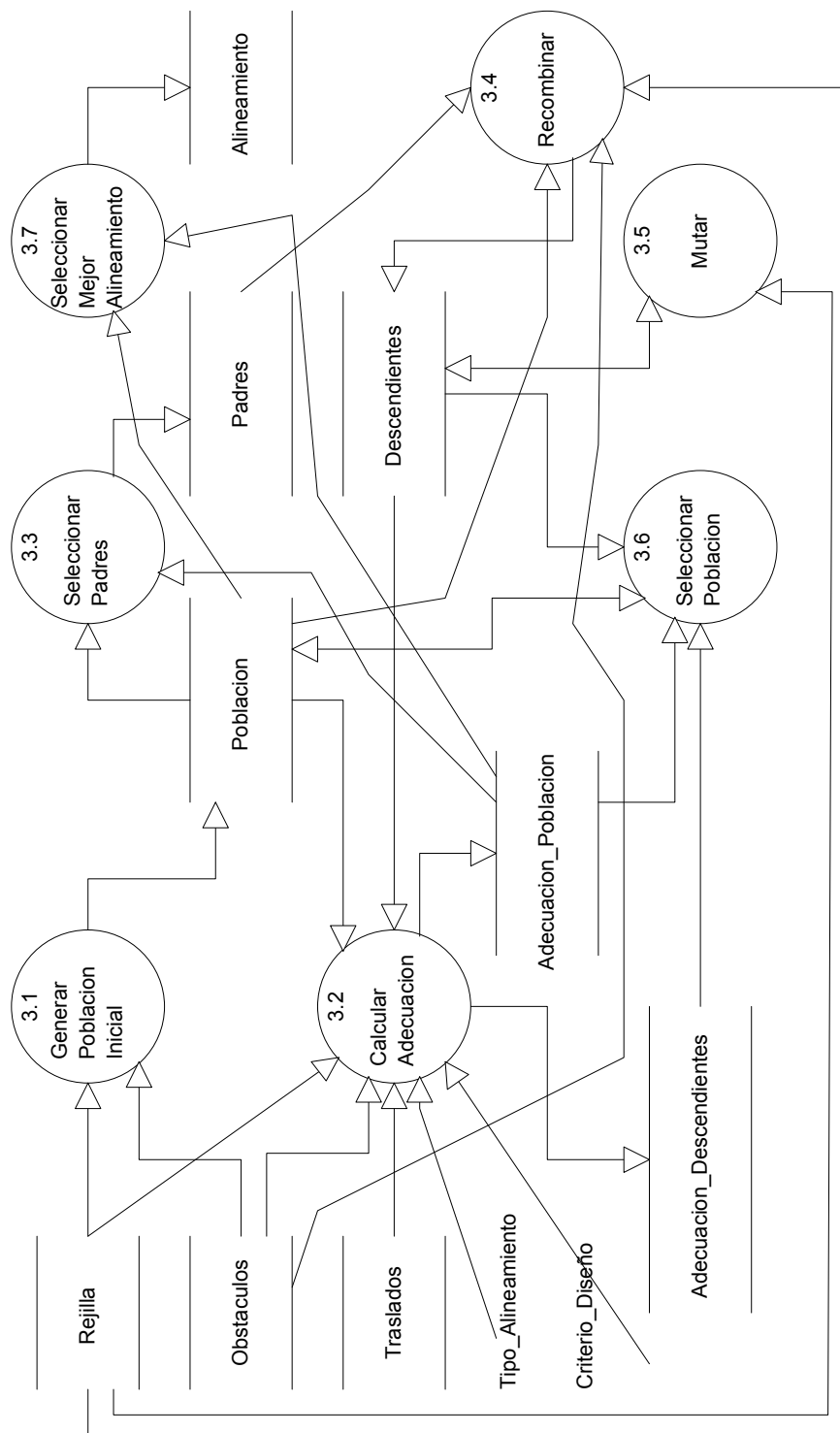


Figura 7.5: Diagrama 3. Algoritmo genético

7.1.3. Diccionario de Datos

En este apartado se detallan los datos utilizados por el sistema, que gráficamente se encuentran representados por los flujos de datos y los almacenes presentes sobre el conjunto de DFDs.

En la tabla 7.1 se muestra la composición de cada almacén existente, utilizando un enfoque descendente (top-down).

Tabla 7.1: Almacenes de datos

Fichero.Obstáculos	= @xMin + @xMax + @yMin + @yMax
Fichero.Municipios	= @id + coordenadasX + coordenadasY + poblacion
Fichero.Traslados	= @origen + @destino + demanda
Coordenadas_X	= @id + coordenadasX_modificadas
Coordenadas_Y	= @id + coordenadasY_modificadas
Traslados	= @origen + @destino + demanda_conjunta
Obstáculos	= @xMin + @xMax + @yMin + @yMax
Habitantes	= @id + poblacion
Rejilla	= @id + c_segmento + intervalo_x + Obstáculo + x_posibles
Obstáculo	= id_obstaculo + xmin_obstaculo + xmax_obstaculo
Aux_Coordenadas	= @id + coordenadasX + coordenadasY
Aux_Traslados	= @origen + @destino + demanda
Población	= @id + cromosoma_lambda + cromosoma_phi
Adecuación_Población	= @id + adecuación
Padres	= @id_padre1 + @id_padre2
Descendientes	= @id + cromosoma_lambda + cromosoma_phi
Adecuación_Descendientes	= @id + adecuación
Alineamiento	= @id + cromosoma_lambda + cromosoma_phi

En la tabla 7.2 se muestran los flujos de datos empleados y una breve explicación de lo que representan, excluyendo de la misma los flujos de lectura y escritura de los almacenes de datos.

Tabla 7.2: Flujos de datos

Info_Traslados	* Información de los traslados entre municipios *
Info_Municipios	* Información de los municipios *
Info_Obstáculos	* Información de los obstáculos *
Info_Alineamiento	* Información del alineamiento generado *
Tipo_Alineamiento	* Autovía o línea de metro *
Criterio_Diseño	* Criterio de diseño elegido por el usuario *

7.2. Diseño

7.2.1. Diagramas de estructura

En este apartado se muestran los *Diagramas de estructura* (DEs) del sistema. En ellos se describe el diseño de la estructura de LAHIT, que está basado en módulos. (Véanse Figuras 7.6, 7.7, 7.8, 7.9 y 7.10).

La estrategia de diseño que se ha seguido a la hora de elaborar los diagramas ha sido la estrategia de *Análisis de transformación*, ya que todos los flujos se representan como transformaciones.

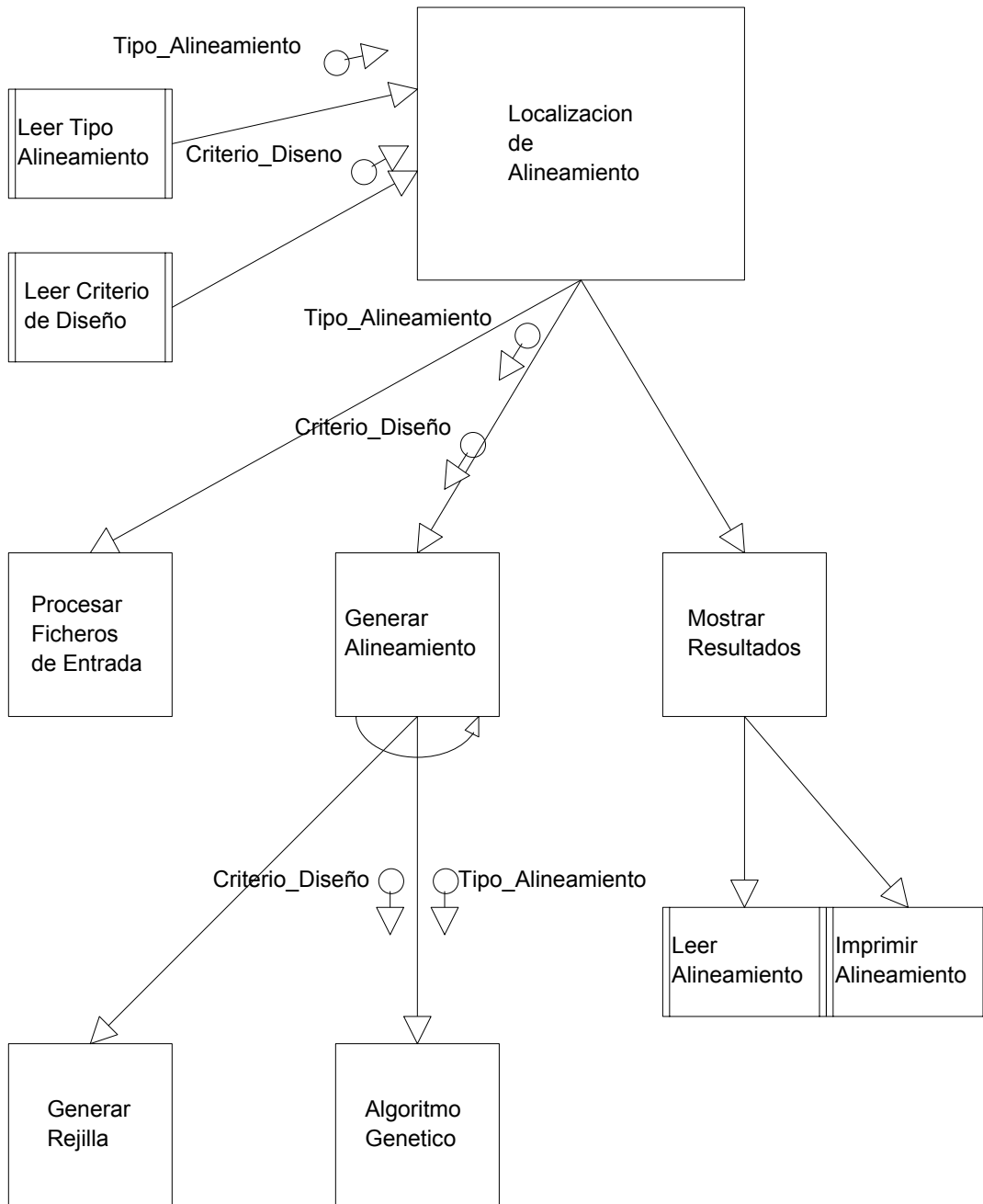


Figura 7.6: Diagrama de estructura

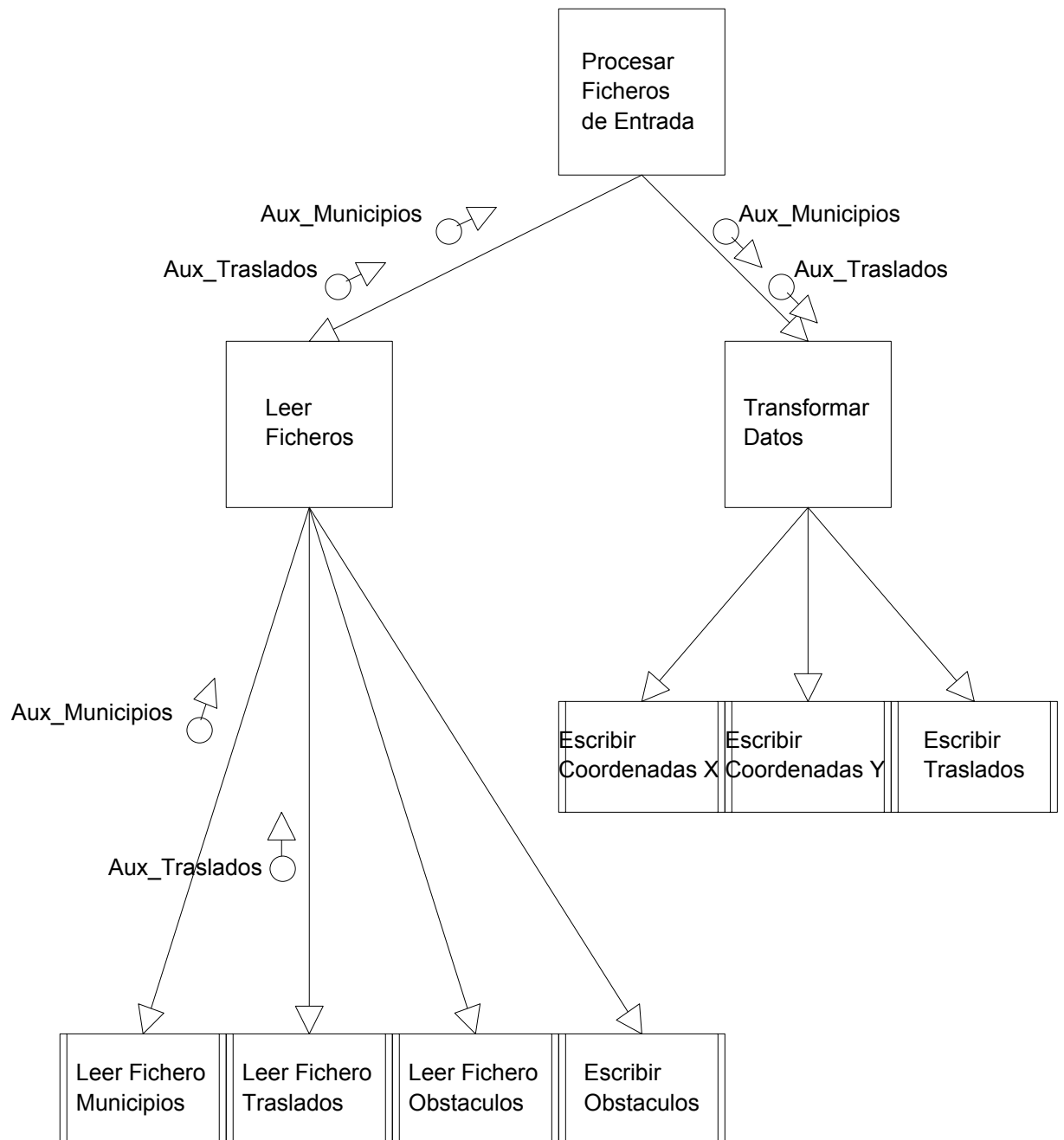


Figura 7.7: Procesar ficheros de entrada

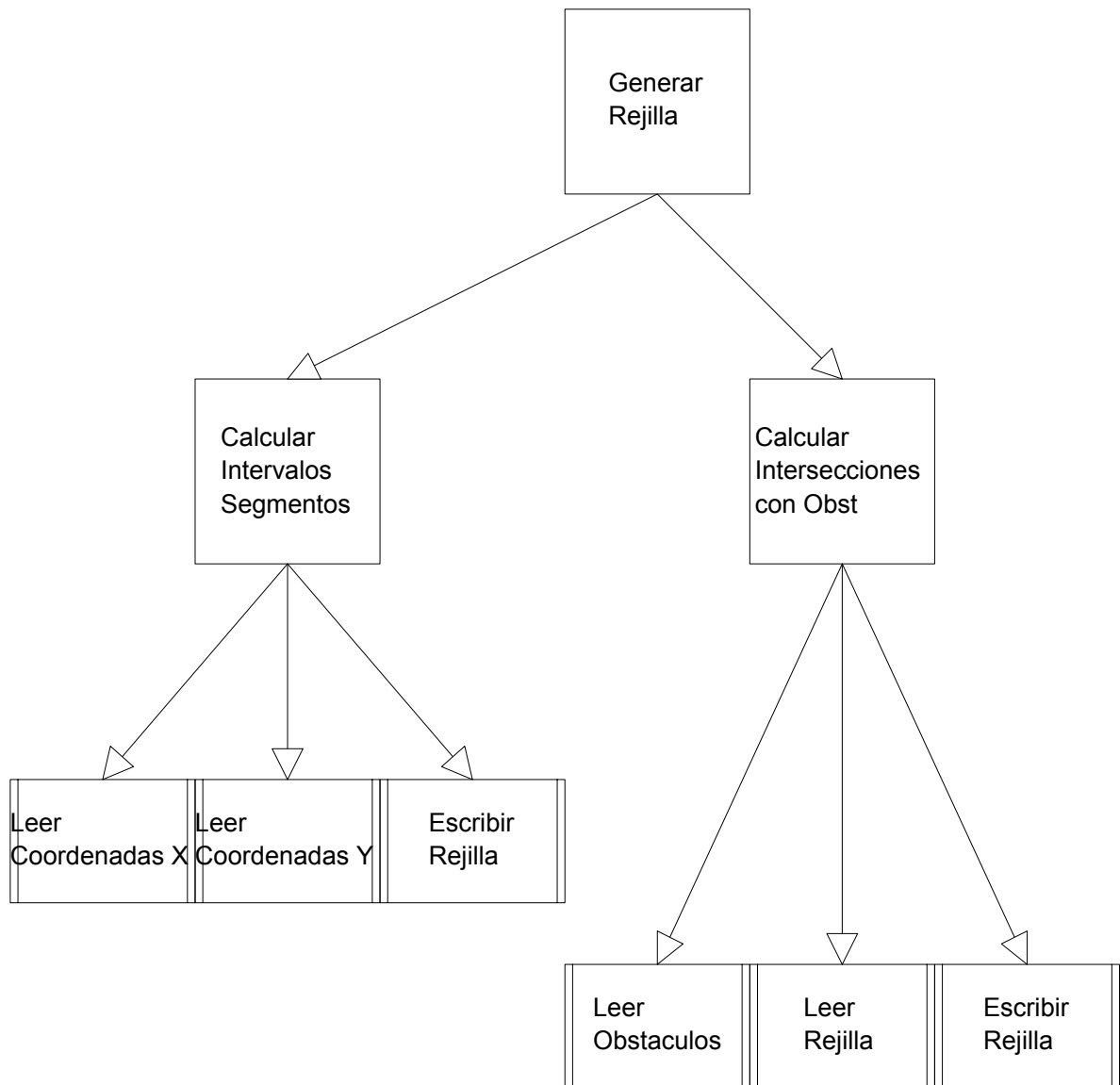


Figura 7.8: Generar rejilla

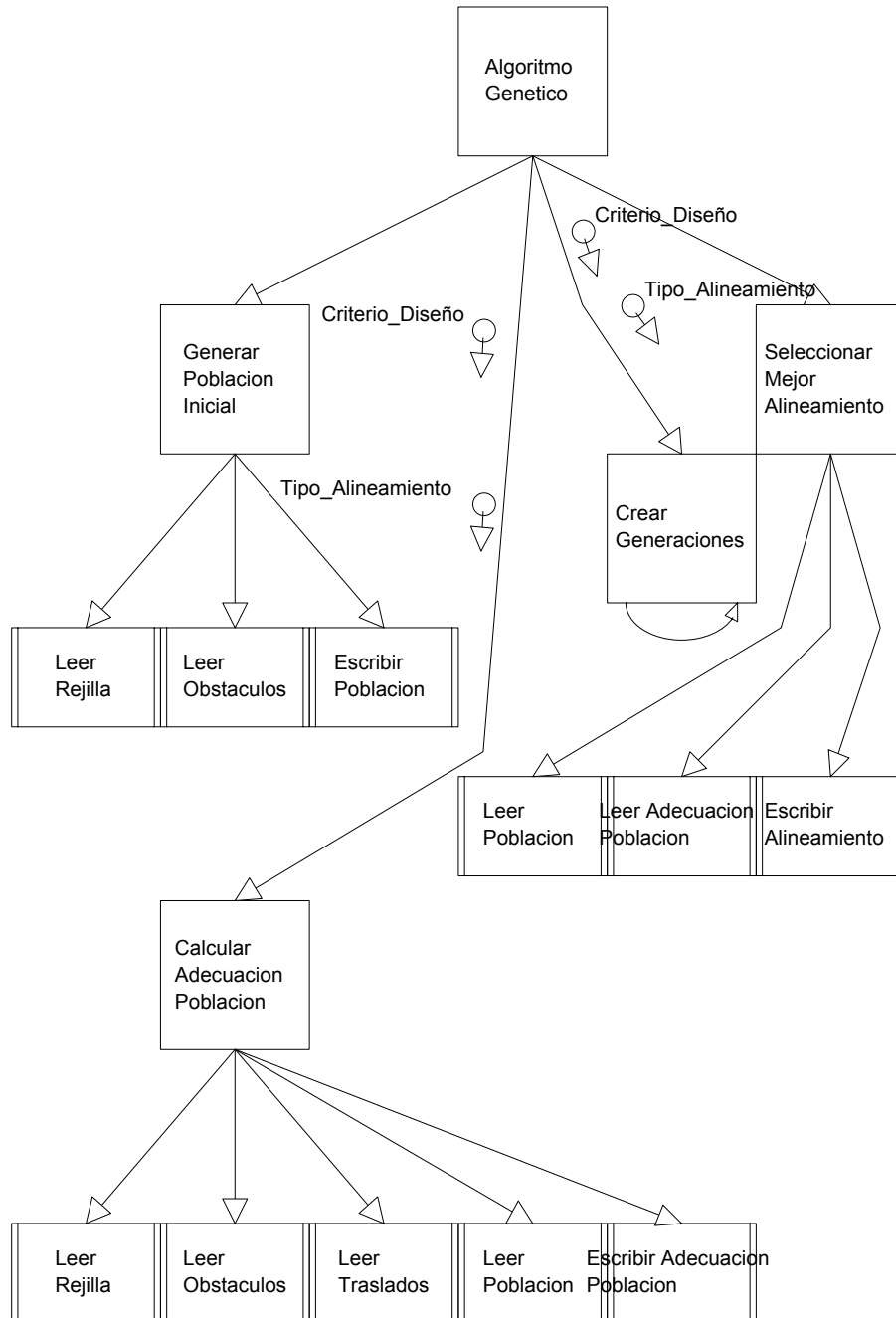


Figura 7.9: Algoritmo genético

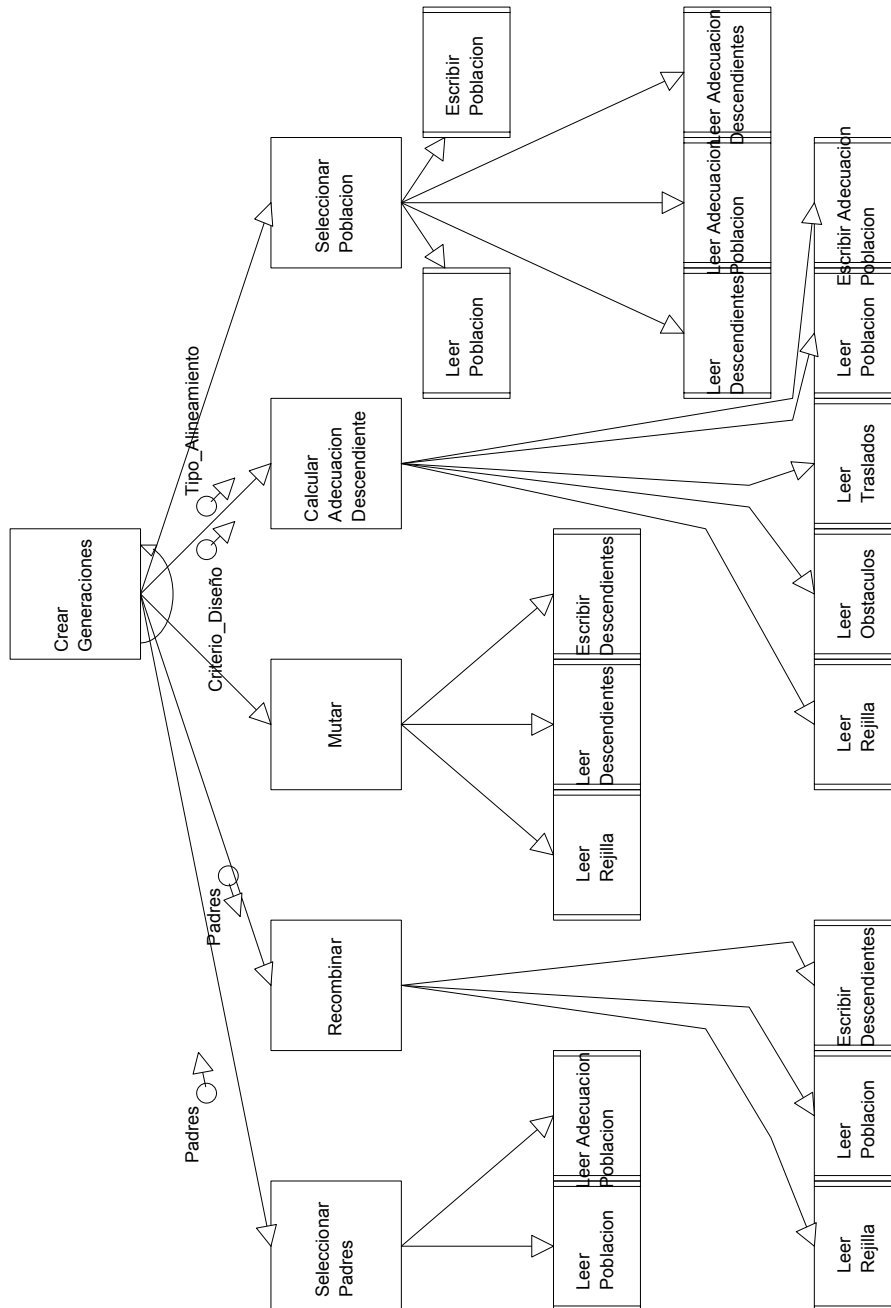


Figura 7.10: Crear generaciones

7.3. Codificación

En este apartado se muestra cómo se ha construido LAHIT a partir de las fases de análisis y diseño anteriormente descritas.

7.3.1. Interfaz de entrada y salida

La interfaz gráfica de LAHIT se ha programado para que pida al usuario los nombres de los ficheros de datos que necesita y las opciones del programa, y muestre una gráfica con la vía generada.

Los ficheros que utiliza LAHIT son los siguientes:

- **Municipios:** Contiene los datos de los municipios.
- **Traslados:** Contiene los datos de los traslados entre los municipios del fichero anterior.
- **Obstáculos:** Contiene los datos de los obstáculos que existen en el trazado.

Estos ficheros están formados por líneas de datos, separados entre sí mediante tabulaciones. Cada línea de los diferentes ficheros tiene la estructura que se muestra en la tabla 7.3. Estos ficheros se leen y procesan mediante la función `procesarFicherosDeEntrada`.

Tabla 7.3: Estructura de los ficheros de entrada

Fichero	Estructura de una línea			
Municipios	codigo	coordenadasX	coordenadasY	poblacion
Traslados	codigo_origen	codigo_destino	demanda	
Obstáculos	xMín	xMáx	yMín	yMáx

Las opciones de entrada son las siguientes:

- **tipoAlineamiento**: Indica si se quiere obtener el alineamiento de una autovía o de una línea de metro.
- **criterioOptimizacion**: Indica cual de los criterios de diseño descritos en el apartado 5.2.3 se desea aplicar.

Una vez que el algoritmo de optimización ha realizado los cálculos, la interfaz muestra al usuario una gráfica con los centroides, la rejilla, los obstáculos y la vía generada.

Además de mostrar la vía, LAHIT guarda información acerca de la ejecución en dos ficheros: “log.txt” y “log2.txt”.

El fichero “log.txt” contiene información general acerca de la ejecución, como es la fecha, el tiempo de ejecución, el tipo de alineamiento, el criterio de diseño, el número de centroides, el número de obstáculos, el número de segmentos de la rejilla y los valores de la pendiente, la vía y las incorporaciones de la solución.

El fichero “log2.txt” contiene información general acerca de la ejecución además de información de las iteraciones del algoritmo de la sección áurea. Para cada iteración almacena el tiempo hasta ese momento, el valor de Θ , el valor de adecuación, la vía y las incorporaciones.

7.3.2. Modelo matemático

Con respecto a las restricciones presupuestarias, la herramienta LAHIT utiliza los parámetros L e I en lugar de C_1 y C_2 . L e I simbolizan, respectivamente, la longitud máxima de vía y el número máximo de incorporaciones que se pueden construir. La equivalencia entre los parámetros del modelo matemático (ver ecuaciones (5.2) y (5.3) en la página 39) y los usados por LAHIT se puede ver en las ecuaciones (7.1) y (7.2).

$$L = \frac{C_1}{\hat{c}_1} \quad (7.1)$$

$$I = \frac{C_2}{\hat{c}_2} \quad (7.2)$$

El modelo matemático se ha resumido en varias funciones. La función principal es la función `calcularAdecuacionVia`. Esta función se encarga de calcular el uso de la red, y según los parámetros de entrada calcular la adecuación de la vía. Las funciones más importantes que utiliza `calcularAdecuacionVia` son `calcularModeloLogit`, `calcularCaminoMinimo` y `calcularTiempoOrigenDestino`.

Para dar más eficiencia a los cálculos se usa la estructura de datos **tabla** para almacenar la información de los orígenes, con sus destinos, los nodos en los que se incorpora, los tiempos directos y en la vía, etc.

La estructura tabla tiene los siguientes componentes:

- **origen**: Centroide origen.
- **noi**: Nodo de incorporación/salida de derecha a izquierda del centroide origen a la vía.
- **nod**: Nodo de incorporación/salida de izquierda a derecha del centroide origen a la vía.
- **destinos**: Matriz con la siguiente estructura:
 - **destino**: Centroide destino.
 - **noi**: Nodo de incorporación/salida de derecha a izquierda del centroide destino a la vía.
 - **nod**: Nodo de incorporación/salida de izquierda a derecha del centroide destino a la vía.
 - **demanda**: Demanda de transporte entre el centroide origen y el centroide destino.
 - **tiempo_directo**: Tiempo empleado para ir directamente del centroide origen al centroide destino, sin usar la vía principal.

- **tiempo_via**: Tiempo empleado para ir del centroide origen al centroide destino empleando la vía principal.

Esta estructura se rellena en la función `calcularTiempoOrigenDestino` y se utiliza para calcular la matriz del Modelo Logit y el valor de la adecuación de la vía.

7.3.3. Resolución geométrica

Para la resolución geométrica se ha utilizado la función `calcularRejilla`, que se encarga de calcular los intervalos permitidos de los segmentos y los prohibidos por los obstáculos.

La información acerca de la rejilla se almacena en un vector cuyos elementos son del tipo de la estructura de datos **rejilla**, que tiene los siguientes componentes:

- **c**: Valor del termino de interceptación del segmento.
- **x**: Vector de dos elementos que contiene el intervalo de x del segmento, sin tener en cuenta los obstáculos.
- **obstaculos**: Matriz que para obstáculo indica el índice del obstáculo y el intervalo de x que prohíbe.
- **xPosibles**: Matriz cuyas filas contienen los subintervalos del intervalo x que se permiten para el segmento.

7.3.4. Algoritmo genético

El algoritmo genético se ha concretado en la función `algoritmoGenetico`, que tiene como subfunciones principales las siguientes funciones:

- **generarPoblacionInicial**: Como su nombre indica se encarga de generar los genotipos que forman la población inicial.
- **seleccionarPadres**: Se encarga de seleccionar mediante el *Método de la Ruleta* los padres que se van a recombinar en cada generación.

- **recombinar**: Se encarga de seleccionar una operación de recombinación y ejecutarla.
- **mutar**: Se encarga de seleccionar una operación de mutación y ejecutarla.
- **seleccionarMSoluciones**: Se encarga de seleccionar las M soluciones que tengan un mejor valor de adecuación de la población y los descendientes, y desechar el resto de soluciones.
- **elegirMejor**: Se encarga de seleccionar la solución con un mejor valor de adecuación de la población.

7.3.5. Sección Áurea

El proceso `GenerarAlineamiento` que aparece en los diagramas de estructura queda implementado en la función `seccionAurea`. En dicha función se implementa el algoritmo de la *sección áurea* (ver tabla 6.1 en la página 60).

Capítulo 8

Pruebas Numéricas

-
- 8.1. Ajuste de parámetros
 - 8.2. Estudio del coste computacional
 - 8.3. Análisis de las soluciones obtenidas
-

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

Las pruebas se han realizado con datos de la región de Castilla–La Mancha, que consta de 919 poblaciones. Este número es demasiado elevado para poder considerar todas las poblaciones, por lo que 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 8.1 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 analizada sobre el total de 919 poblaciones de Castilla–La Mancha.

Tabla 8.1: Definición de los problemas

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

En esta experiencia computacional, la matriz de demanda origen–destino se ha estimado mediante encuestas. El Instituto Nacional de Estadística (INE) realizó en el año 2005 una encuesta (censo) a todos los domicilios de España donde se preguntaba en qué población se residía y a qué población se iba a trabajar o estudiar. Esta matriz mide, por tanto, los desplazamientos que se realizan con el propósito de trabajar o estudiar.

Se han tenido en cuenta diferentes tipos de obstáculos:

- O1:** Sin obstáculos
- O2:** Con un obstáculo grande en el trazado de la mejor solución al problema sin obstáculos
- O3:** Con varios obstáculos

Puesto que los *Problemas* 1 y 2 son problemas de una red de autovías, el criterio de diseño que se ha tenido en cuenta es el A.3: *Minimización del tiempo total de transporte en la infraestructura nueva y la existente.*

Las pruebas se han realizado en un ordenador portátil con procesador Intel Pentium M740 1.73Ghz, 1GB de Memoria RAM y 60GB de disco duro.

8.1. Ajuste de parámetros

En este apartado se lleva a cabo un ajuste de parámetros para el *Problema* 1, sin tener en cuenta obstáculos (*O1*). Una vez que los parámetros se han ajustado, se realiza una serie de pruebas con los tres tipos de obstáculos descritos anteriormente, *O1*, *O2* y *O3*.

Los datos acerca de los municipios se muestran en la tabla 8.2 y los datos acerca de los traslados se pueden ver en la tabla 8.3. Se han desechado los datos de los traslados de un municipio a él mismo, por no ser relevantes para este estudio.

Tabla 8.2: Datos de entrada de los 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	343490	4425152	82975
Toledo	45168	412492	4412611	75533

Tabla 8.3: Datos de los traslados entre municipios 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	-

Modelo Logit

En primer lugar se van a ajustar los valores de los parámetros de la ecuación del Modelo Logit (ver ecuación (5.5) en la página 41). Hay que determinar los valores de los parámetros α^N , α^E y β . Los valores asignados a dichos parámetros dependerán del tipo de alineamiento, autovía o línea de metro, que se quiera obtener.

Para el problema de la localización del alineamiento de una autovía se tiene que pequeñas variaciones en el coste hacen que los usuarios se decanten por el sistema nuevo o el existente. Por ello el valor del parámetro β debe ser grande. Se ha decidido asignar a α^N y α^E el valor 0, y se han realizado pruebas asignando diferentes valores al parámetro β . Basándose en los resultados, se ha decidido asignar a β el valor $\beta = 4$.

En el problema de la localización de una línea de metro, aunque haya variaciones en el coste, va a haber usuarios que utilicen el metro y otros que utilicen el coche. Los parámetros para este problema se van a ajustar usando los datos de los municipios, puesto que en el momento de realizar estas pruebas no se dispone de los datos del metro de Sevilla. Se ha decidido asignar a α^N el valor 0 y se ha calculado α^E en función de α^N . El valor que se ha obtenido para α^N ha sido $-\ln(1/3 - 1)$. Una vez calculados estos parámetros se han realizado pruebas con diferentes valores de β . Basándose en los resultados, se ha decidido asignar a β el valor $\beta = 0,5$.

Geometría del problema

En relación a la geometría del problema es necesario asignar los valores a los parámetros L (ver ecuación (7.1) en la página 95), I (ver ecuación (7.2) en la página 95) y n (ver ecuación (5.13) en la página 45).

Se ha decidido probar con los valores de los parámetros que se muestran en la tabla 8.4. En el siguiente apartado se muestran los resultados obtenidos con los diferentes valores de L , I y n , así como con los parámetros necesarios para definir completamente el algoritmo genético.

Tabla 8.4: Valores de los parámetros relacionados con la geometría del problema

L (km)	I (incorporaciones)	n (segmentos)
100	20	100
100	40	100
100	20	200
100	40	200
100	20	300
100	40	300

Algoritmo genético

Para definir completamente el algoritmo genético es necesario definir los valores de los siguientes parámetros:

M Tamaño de la población.

n_g Número de generaciones.

n_d Número de descendientes creados en cada generación.

N Número de alineamientos totales.

p_1 Probabilidad de poner una incorporación en un nodo.

p_2 Proporción máxima de incorporaciones permitidas que se pueden poner en una vía de la población inicial.

ζ Determina el grado de no uniformidad en la mutación no uniforme.

P_G Probabilidad de algún cambio en las incorporaciones en la mutación de incorporaciones.

El parámetro p_1 determina cómo se asignan las incorporaciones a los nodos. Si su valor es grande, se asignarán la mayor parte de las incorporaciones a los primeros nodos de la vía, pudiendo dejar el resto de nodos sin la posibilidad de ser una incorporación. Por otra parte, el valor de p_1 tampoco puede ser muy pequeño, ya que se corre el riesgo de que se asignen muy pocas incorporaciones. Después de realizar diversas pruebas generando incorporaciones para diferentes vías, se ha asignado el valor $p_1 = 0,4$.

El parámetro p_2 se usa para determinar el número de incorporaciones que tendrán las soluciones iniciales. Se debe asignar a p_2 un valor no demasiado grande, ya que si se ponen demasiadas incorporaciones en la solución inicial, la operación de mutación de incorporaciones apenas podrá añadir nuevas incorporaciones. Además, el valor de p_2 no debe ser demasiado pequeño, ya que en ese caso los alineamientos tendrían bastantes incorporaciones menos de las que permite el presupuesto. Se ha decidido asignar el valor $p_2 = 0,8$.

El parámetro ζ es arbitrario. Una manera de estimarlo consiste en suponer que cuando se han creado un 50 % de las generaciones, $t = 0,5 \cdot n_g$, el valor de $\left(1 - \frac{t}{n_g}\right)^\zeta$ es 0,7 (se admite un 70 % de variabilidad inicial en el intervalo), por lo que el valor de ζ es $\zeta = \frac{\ln 0,7}{\ln 0,5}$.

El parámetro P_G sirve para determinar una probabilidad P_l (ver ecuación (6.16) en la página 76) que disminuya a medida que aumente el número de generaciones realizadas, t . La ecuación que define P_l está definida de forma que en la primera generación ($t = 1$) el valor de P_l sea $P_l = P_G$, y vaya disminuyendo conforme aumenta t . Se ha asignado a P_G el valor 0,7, permitiendo una probabilidad grande de realizar modificaciones en las incorporaciones durante las primeras generaciones y consiguiendo unos

valores muy pequeños de P_l en las siguientes (en la iteración $t = 10$, el valor de P_l ha disminuido hasta 0,1487).

El resto de parámetros, a excepción de L al que se ha asignado un valor de $L = 100$, se va a asignar dependiendo de los resultados de las pruebas. En la tabla 8.5 se muestran los tiempos totales de viaje en la infraestructura nueva y la existente, \mathcal{Z}_3 (Véase ecuación (5.9) en la página 5.9), así como el tiempo de ejecución, t , según los diferentes valores de los parámetros.

Tabla 8.5: Valores de σ , \mathcal{Z}_3 y tiempo de ejecución en función de los parámetros

L (km)	I	n	M	n_g	n_d	N	σ	\mathcal{Z}_3 (h)	t (s)
100	20	100	500	50	100	1000	$2,3786 \cdot 10^{-4}$	4204,2	670
100	20	100	1000	100	200	2000	$2,5998 \cdot 10^{-4}$	3846,4	4103
100	20	200	500	50	100	1000	$2,375 \cdot 10^{-4}$	4210,5	1238
100	20	200	1000	100	200	2000	$2,5565 \cdot 10^{-4}$	3911,6	8058
100	20	300	500	50	100	1000	$2,3192 \cdot 10^{-4}$	4311,8	1871
100	20	300	1000	100	200	2000	$2,668 \cdot 10^{-4}$	3748,1	12139
100	40	100	500	50	100	1000	$2,3863 \cdot 10^{-4}$	4190,6	666
100	40	100	1000	100	200	2000	$2,6355 \cdot 10^{-4}$	3794,3	4143
100	40	200	500	50	100	1000	$2,425 \cdot 10^{-4}$	4123,7	1232
100	40	200	1000	100	200	2000	$2,5022 \cdot 10^{-4}$	3996,5	8183
100	40	300	500	50	100	1000	$2,4507 \cdot 10^{-4}$	4080,4	1861
100	40	300	1000	100	200	2000	$2,6665 \cdot 10^{-4}$	3750,2	12701

A continuación se detallan los valores asignados a los parámetros basándose en los resultados mostrados en la tabla 8.5. Estos valores son los que se utilizarán en los siguientes apartados para resolver el *Problema 2*.

Se observa que la variación del parámetro I apenas influye en los valores de la adecuación ni en el tiempo de ejecución, por lo que se le ha asignado el valor $I = 20$.

Con respecto al parámetro n , se observa que su aumento no produce grandes mejoras en el valor de Z_3 y sin embargo incrementa considerablemente el tiempo de ejecución del programa. Por ello se ha decidido asignarle el valor $n = 100$.

Los parámetros M , n_g y n_d afectan en gran medida al tiempo de ejecución. Se puede observar que al aumentar el valor de estos tres parámetros, se incrementa significativamente el tiempo de ejecución. Sin embargo, la mejora en el valor de Z_3 es muy pequeña, por lo que se ha decidido asignar a estos tres parámetros los valores $M = 500$, $n_g = 50$ y $n_d = 100$.

El valor del parámetro N influye en el número de vías que se generan para la población inicial del algoritmo genético, cuyo tamaño M debe mantenerse a lo largo de la ejecución. Debido a los obstáculos, una parte de las vías generadas para la población inicial no son factibles, con lo que se deberá generar un número total de vías suficientemente grande para evitar que el número de vías factibles sea menor que M . Por ello se ha asignado a N el valor $N = 100$, es decir, el doble del parámetro M .

8.2. Estudio del coste computacional

En este apartado se resuelve el *Problema 2* para los tres tipos de obstáculos, $O1$, $O2$ y $O3$, y se realiza un estudio de los resultados obtenidos. Los valores que se han asignado a los parámetros son los calculados en el apartado anterior (ver tabla 8.5).

En la tabla 8.6 se muestran los valores de adecuación, σ , tiempos totales de viaje, Z_3 , y tiempo de ejecución, t , para los diferentes tipos de obstáculos a lo largo de la ejecución de cada una de las pruebas realizadas en este apartado. En dicha tabla se puede observar como se incrementan significativamente los tiempos de ejecución a medida que aumenta el número de obstáculos considerados. También se puede observar que los valores de adecuación para los casos $O2$ y $O3$ son mejores que para el caso $O1$.

Esto puede ser debido a que en el caso $O1$ se haya alcanzado un óptimo local no global.

Tabla 8.6: Resultados de la ejecución del *Problema 2*

Obstáculos	σ	\mathcal{Z}_3 (h)	t (s)
O1	$2,8981 \cdot 10^{-5}$	34505,4	15609
O2	$3,0842 \cdot 10^{-5}$	32446,5	37428
O3	$3,2011 \cdot 10^{-5}$	31239	53040

En la figura 8.1 se muestra una gráfica con la evolución del valor de \mathcal{Z}_3 a lo largo de la ejecución del algoritmo de la sección áurea para los tres casos de prueba. En el caso $O1$ se puede observar como el valor de \mathcal{Z}_3 disminuye conforme avanza el tiempo de ejecución, lo que hace suponer que el valor de \mathcal{Z}_3 pueda seguir disminuyendo hasta alcanzar el óptimo global.

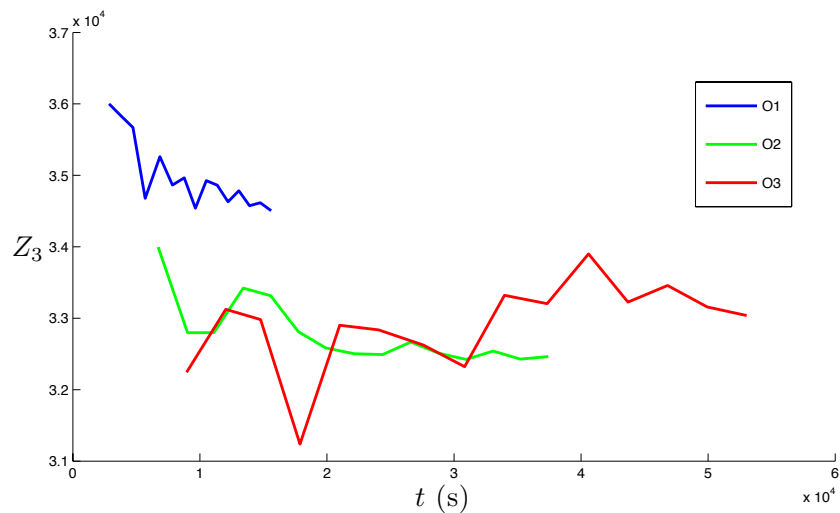


Figura 8.1: Valores de \mathcal{Z}_3 en función del tiempo de ejecución

8.3. Análisis de las soluciones obtenidas

En este apartado se analizan los resultados obtenidos para el *Problema 2* en el apartado anterior.

En la figura 8.2 se muestra una gráfica con los valores de Z_3 en función de los valores de Θ , es decir, en función de las diferentes inclinaciones de la rejilla.

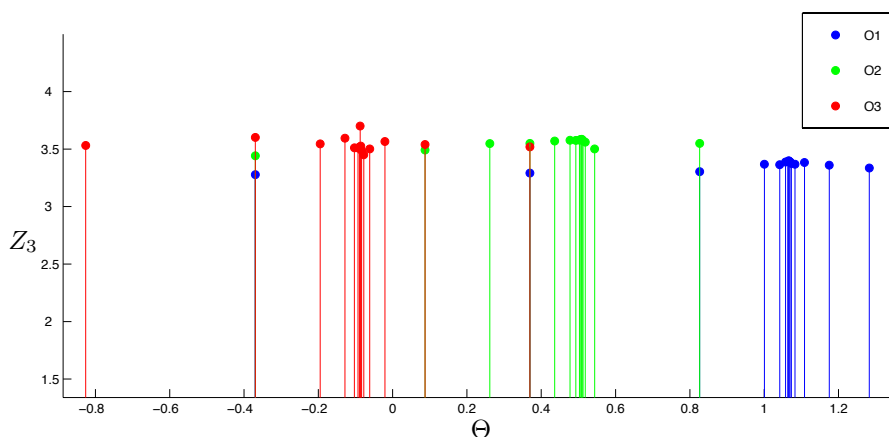


Figura 8.2: Valores de Z_3 en función del valor de Θ

La figura 8.3 muestra la vía obtenida para el *Problema 2* en el caso O1.

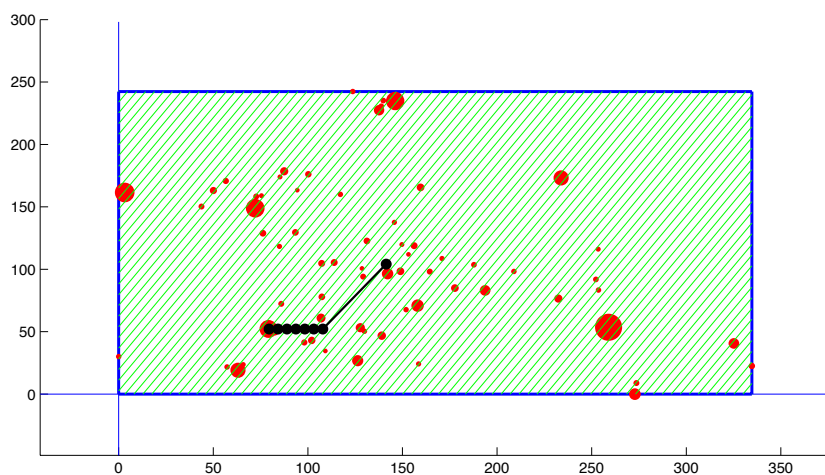


Figura 8.3: Solución al *Problema 2* con O1

La figura 8.4 muestra la vía obtenida para el *Problema 2* en el caso *O2*.

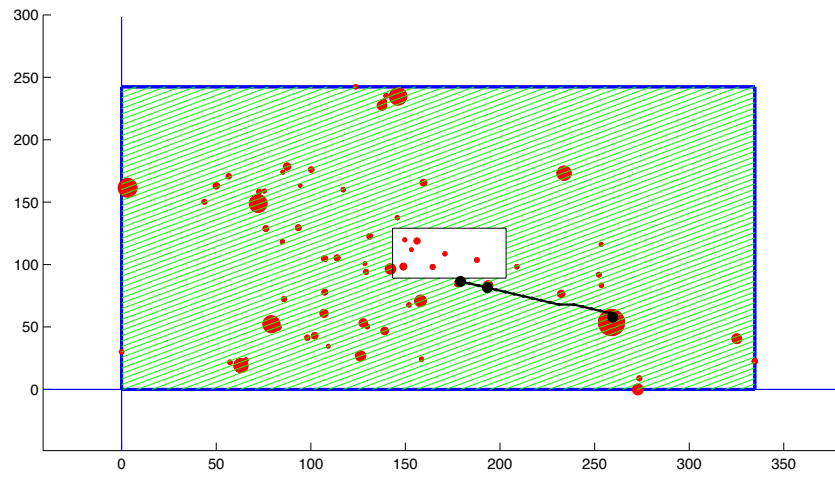


Figura 8.4: Solución al *Problema 2* con *O2*

La figura 8.5 muestra la vía obtenida para el *Problema 2* en el caso *O3*.

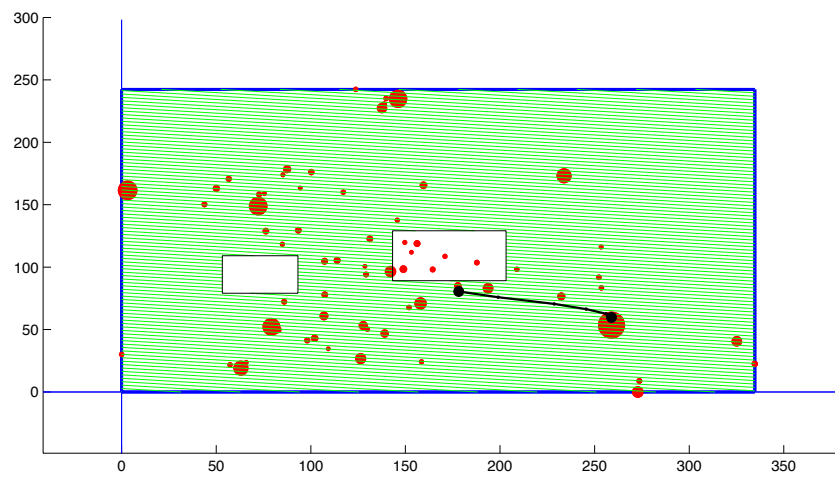


Figura 8.5: Solución al *Problema 2* con *O3*

Capítulo 9

Conclusiones y propuestas futuras

9.1. Conclusiones

9.2. Propuestas de mejora

9.1. Conclusiones

Se ha demostrado computacionalmente que el modelo matemático y el algoritmo de resolución propuestos son aplicables a problemas de tamaño medio. Para resolver problemas de mayor complejidad, que requieren un tiempo de computación mucho más elevado que los problemas resueltos en la experiencia numérica del capítulo anterior, se deberá utilizar el servicio de Supercomputación de la UCLM.

El objetivo perseguido en este PFC es la creación de un producto software, en el que nuevos objetivos (modelo en espiral) se van añadiendo en la vida del producto, como las mejoras aquí propuestas. El objetivo del PFC no es cubrirlas de antemano sino sentar las bases para que esto sea posible. En este sentido la elección estratégica de MATLAB como lenguaje, de por sí multiplataforma, permite la paralelización de los procesos (Stara-p/matlab) y posibilita la mejora b) que se expone en siguiente apartado. La segunda elección estratégica de un algoritmo genético como motor de resolución del modelo permite la incorporación de funciones objetivos complejas como las propuestas en la alternativa a) del siguiente apartado.

Finalmente decir que el paradigma de la programación estructurada y la buena práctica de implementación y documentación permite a personas diferentes a la autora de la herramienta llevar a cabo las mejoras que se describen a continuación, dotando a la herramienta de vida propia e independencia.

9.2. Propuestas de mejora

La herramienta LAHIT puede ser mejorada en dos direcciones fundamentalmente:

- a) Implementación de mejoras en el modelo matemático. La herramienta implementa un modelo matemático que puede ser modificado con el objeto de dar respuesta a

otras necesidades de los planificadores de las infraestructuras de transporte. Dos mejoras posibles son:

a.1) **Extensión estocástica de la demanda: diseño robusto.**

Infraestructuras como las estudiadas en este proyecto requieren largos periodos de tiempo para su implantación, por lo que existe un gran desfase entre la fase de estudio y la aplicación. Es por esto que asumir una matriz $o-d$ dada puede conducir a grandes sesgos. Esto motiva la necesidad de introducir un diseño que sea robusto frente a distintas matrices $o-d$ verosímiles, con lo que una posible mejora de la herramienta sería considerar el flujo como una variable aleatoria y trabajar con matrices $o-d$ aleatorias.

a.2) **Alineamientos verticales.**

La estructura del modelo permite implementar funciones de costes más sofisticadas con el objeto de considerar alineamientos verticales, obteniéndose como posibles salidas no sólo el alineamiento sino también movimientos de tierra, túneles, puentes a localizar con objeto de salvar las dificultades del terreno, etc. Esta línea consiste en añadir una etapa posterior del diseño de autovías a la considerada en este PFC, y en general, se podría incorporar a todas las etapas hasta la última del trazado.

b) **Incrementar la eficiencia computacional.**

Las pruebas numéricas han puesto de manifiesto la complejidad computacional del problema y la necesidad de mejorar los tiempos de resolución del modelo. En este sentido una posible mejora es la paralelización del algoritmo. Este problema es masivamente paralelizable sobre un cluster de ordenadores, en el que cada uno de ellos analice diferentes rejillas.

Anexo A

Manual de Usuario

A.1. Requisitos de instalación

A.2. Ejecución

A.1. Requisitos de instalación

Para que la aplicación funcione correctamente es necesario que el equipo en el que se ejecute tenga instalado el entorno de desarrollo MATLAB 7.1.

Para instalar LAHIT el usuario únicamente tiene que copiar la carpeta *fuentes* del CD adjunto al disco duro del ordenador en el que desea ejecutar la aplicación.

A.2. Ejecución

Para ejecutar LAHIT hay que ejecutar MATLAB y cambiar el directorio actual del programa al directorio *fuentes* que se ha copiado al ordenador. Una vez hecho esto, se ejecuta el comando LAHIT y aparece la ventana que se muestra en la figura A.1.

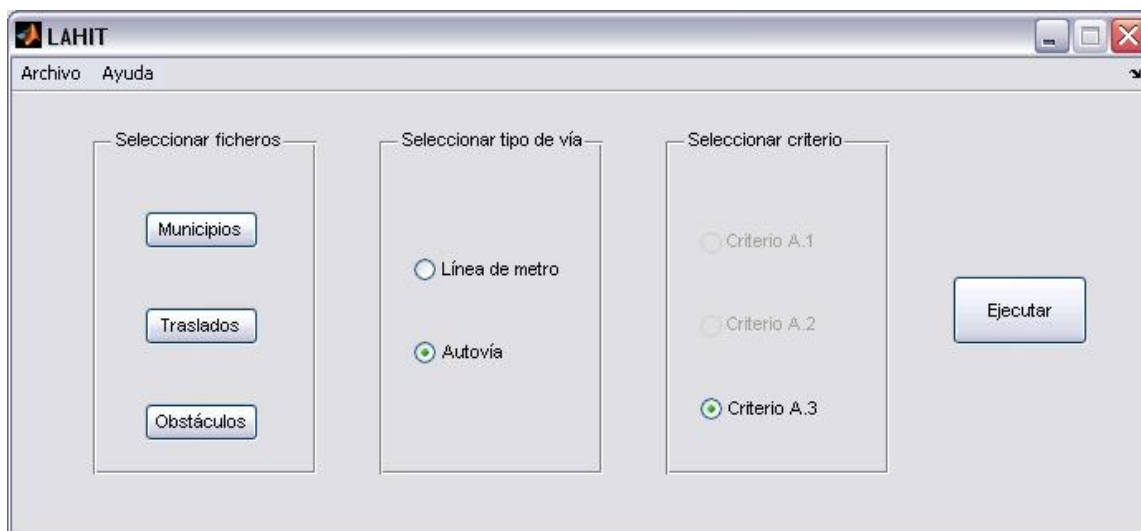


Figura A.1: Ventana principal de LAHIT

En dicha ventana, el usuario debe indicar a la aplicación qué ficheros de entrada desea utilizar. Para ello deberá pulsar cada uno de los botones correspondientes a los ficheros de municipios, traslados y obstáculos (ver tabla 7.3 en la página 93).

Después el usuario debe seleccionar el tipo de vía que desea generar entre las dos opciones que proporciona LAHIT: *línea de metro* o *autovía*.

A continuación el usuario debe seleccionar el criterio de diseño que desea emplear (ver apartado 5.2.3). Si en el paso anterior seleccionó la opción *línea de metro*, podrá elegir entre los criterios A.1 y A.2. Si seleccionó la opción *autovía*, solamente podrá seleccionar el criterio A.3. Al pasar el ratón por encima de los botones de cada criterio la interfaz muestra una pequeña ayuda con la explicación del criterio correspondiente (ver figura A.2).

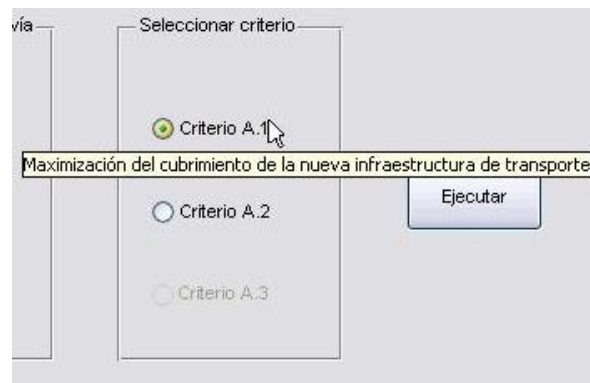


Figura A.2: Ayuda contextual

Una vez que el usuario ha seleccionado los ficheros de entrada, el tipo de vía y el criterio de diseño, debe pulsar el botón “Ejecutar” para que comience la ejecución del algoritmo de optimización. Si el usuario no ha seleccionado todos los ficheros de entrada, aparece una ventana de error indicándolo (ver figura A.3).



Figura A.3: Ventana de error

Cuando el algoritmo de optimización termina su ejecución, aparece otra ventana (ver figura A.4) que muestra la vía generada, así como los datos de la ejecución.

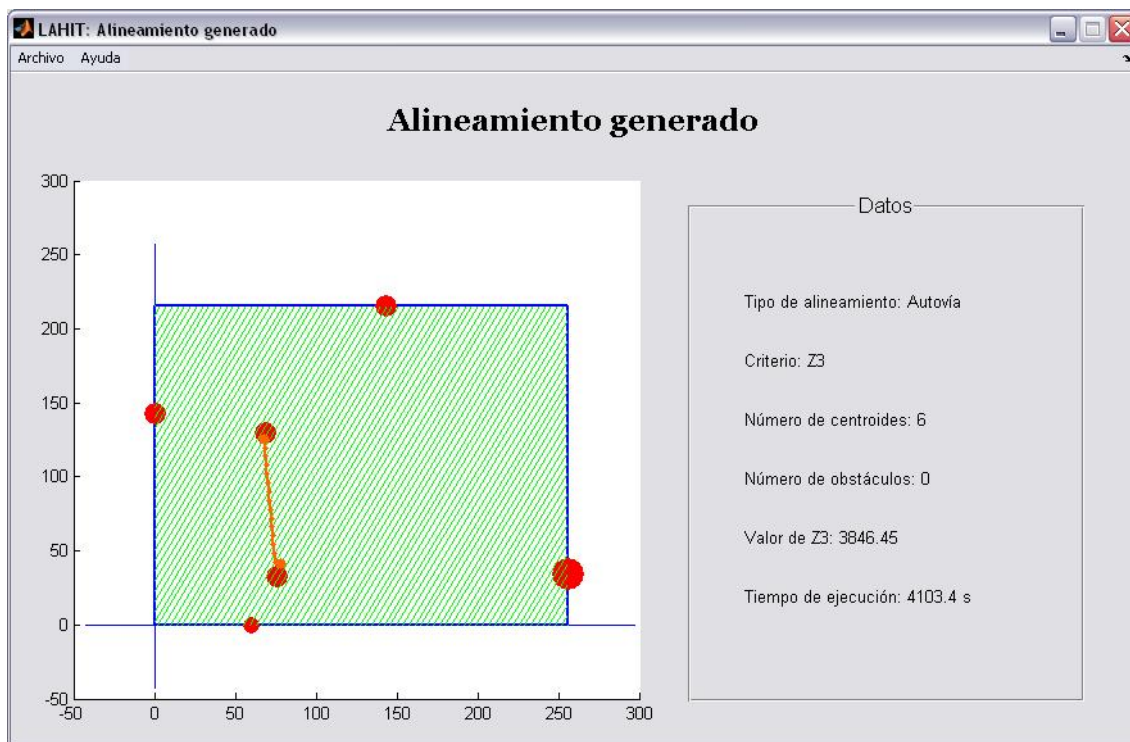


Figura A.4: Ventana de solución

En el menú de esta ventana se da la opción al usuario de abrir los ficheros de log de la herramienta para visualizarlos (ver figura A.5).



Figura A.5: Menú de la ventana de solución

Tanto el menú de la ventana principal como el de la ventana de solución dan la

opción de mostrar una ventana *Acerca de* con información de LAHIT (ver figura A.6).



Figura A.6: Ventana Acerca de LAHIT

Finalmente, cuando el usuario intenta cerrar la ventana principal de LAHIT, la aplicación le muestra una ventana solicitando confirmación (ver figura A.7).



Figura A.7: Ventana de confirmación

Anexo B

Estructura del Código

En este capítulo se describe la estructura del código fuente de LAHIT. Este se puede consultar en el directorio **fuentes** del CD adjunto.

El código fuente de LAHIT está estructurado en los siguientes ficheros **.m** y **.fig**:

- algoritmoGenetico.m
- calcularAdecuacionVia.m
- calcularRejilla.m
- calcularTiemposDirectosCentroides.m
- dibujarGrafica.m
- esFactibleTramo.m
- LAHIT.fig
- LAHIT.m
- mdAcercaDe.fig
- mdAcercaDe.m
- mostrarSolucion.fig
- mostrarSolucion.m
- localizacionDeAlineamiento.m
- localizacionDeAlineamiento2.m
- procesarFicherosDeEntrada.m

El fichero `algoritmoGenetico.m` contiene el código del algoritmo genético.

El fichero `calcularAdecuacionVia.m` contiene el código fuente del modelo matemático.

El fichero `calcularRejilla.m` contiene el código necesario para calcular los segmentos que forman la rejilla, con sus intervalos permitidos y sus intervalos prohibidos por los obstáculos.

El fichero `calcularTiemposDirectosCentroides.m` contiene el código que se encarga de calcular los tiempos de viaje directo entre centroides. Este cálculo se hace una única vez después de procesar los datos de entrada, no siendo necesario repetir los cálculos con cada alineamiento.

El fichero `dibujarGrafica.m` contiene el código que se utiliza para dibujar las gráficas de las vías. Dichas gráficas muestran la rejilla, los centroides y las vías generadas.

El fichero `esFactibleTramo.m` contiene el código necesario para comprobar si un tramo de vía intersecta con algún obstáculo.

Los ficheros `LAHIT.m`, `LAHIT.fig`, `mdAcercaDe.m`, `mdAcercaDe.fig`, `mostrarSolucion.m` y `mostrarSolucion.fig` contienen la interfaz de usuario de LAHIT.

El fichero `localizacionDeAlineamiento.m` contiene el código necesario para ejecutar el algoritmo de optimización y almacenar los resultados de dicha ejecución en los ficheros de log. Se utiliza desde la interfaz de LAHIT.

El fichero `localizacionDeAlineamiento2.m` contiene casi el mismo código que el fichero `localizacionDeAlineamiento.m`, con la diferencia de que el primero es el que se emplea para utilizar la aplicación desde línea de comandos. Este fichero es el que se ha empleado para ejecutar la aplicación en el servicio de Super Computación de la UCLM.

El fichero `procesarFicherosDeEntrada.m` contiene el código necesario para leer los ficheros con los datos necesarios para la ejecución de LAHIT y procesar los datos que contienen dichos ficheros.

Anexo C

CD adjunto

El CD adjunto consta de los siguientes directorios:

- **documentacion:** Contiene este documento en formato PDF.
- **fuentes:** Contiene el código fuente de LAHIT.
- **pruebas:** Contiene los datos utilizados para ejecutar las pruebas numéricas del capítulo 8.

CD adjunto

Referencias

- [AC73] G.C. Ahanassoulis and V. Calogero. Optimal location of a new highway from a to b - a computer technique for route planning. In *PTRC Seminar Proceedings on Cost Models and Optimization in Highways*, 1973.
- [ALM03] Enrique Alba, Manuel Laguna, and Rafael Martí. Métodos evolutivos. *Ingeniería UC*, 10(3), 2003.
- [AT02] Enrique Alba and Marco Tomassini. Parallelism and evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 6:443–462, Oct 2002.
- [BGI98] G. Bruno, G. Ghianini, and G. Improta. A multi-modal approach to the location of a rapid transit line. *European Journal of Operational Research*, 104:321–332, 1998.
- [BGL02] G. Bruno, M Gendreau, and G. Laporte. A heuristic for the location of a rapid transit line. *Computer and Operations Research*, 21, 2002.
- [CC88] S.C. Chapra and R.P. Canale. *Numerical Methods for Engineers*. McGraw-Hall, Inc., 1988.
- [CGF89] 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.
- [CRC85] J. R. Current, C.S. ReVelle, and J.L. Cohon. The maximum covering/shortest path problems: A multiobjective network design and routing formulation. *European Journal of Operational Research*, 21:189–199, 1985.
- [DGL96] H. Dufourd, M. Gendreau, and G. Laporte. Locating a transit line using tabu search. *Location Science*, 4:1–19, 1996.
- [DP04] Eduardo Domínguez and Mario Piattini. *Material docente de la asignatura Ingeniería del Software I*, 2004.
- [Eas98] S.M. Easa. Selection of roadway grades that minimize earthwork cost using linear programming. *Transportation Research Part A*, 22:121–136, 1998.

- [Fwa89] T.F. Fwa. Highway vertical alignment analysis by dynamic programming. *Transportation Research Record*, 1239:1–9, 1989.
- [GC99] Mitsuo Gen and Runwei Cheng. *Genetic Algorithms & Engineering Optimization*. John Wiley & Sons, Inc., 1999.
- [GCF88] C.J. Goh, E.P. Chew, and T.F. Fwa. Discrete and continuous model for computation of optimal vertical highway alignment. *Transportation Research Part B*, 22:399–409, 1988.
- [GL97] Fred Glover and Manuel Laguna. *Tabu Search*. Kluwer Academic Publishers, 1997.
- [Hay70] R.W. Hayman. Optimization of vertical alignment for highways through mathematical programming. *Highway Research Record, Highway Research Board*, 306:1–9, 1970.
- [HBS68] B.E. Howard, Z. Brammick, and J.F.B. Shaw. Optimum curvature principle in highway routing. *Journal of the Highway Division, ASCE*, 94:61–82, 1968.
- [HLS⁺01] H.W. Hamacher, A. Liebers, A. Schöbel, D. Wagner, and F. Wagner. Locating new stops in a railway network. *Electronic Notes in Theoretical Computer Science*, 50, 2001.
- [Hog73] 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.
- [Hol75] John H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
- [Jha00] M.K. Jha. *A geographic information systems-based model for highway design optimization*. PhD thesis, University of Maryland, College Park, 2000.
- [Jha02] M.K. Jha. Optimizing highway networks: a genetic algorithm and swarm intelligence based approach. *American Society of Civil Engineers*, pages 76–89, 2002.
- [Jha03] M.K. Jha. Criteria-based decision support system for selecting highway alignments. *Journal of Transportation Engineering*, 129:33–41, 2003.
- [JJLS00] 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.
- [Jon98] J.C. Jong. *Optimizing highway alignments with genetic algorithms*. PhD thesis, University of Maryland, College Park, 1998.

- [JS03] J.C. Jong and P. Schonfeld. An evolutionary model for simultaneously optimizing three-dimensional highway alignments. *Transportation Research Part B*, 37:107–128, 2003.
- [JS04] Manoj K. Jha and Paul Schonfeld. A highway alignment optimization model using geographic information systems. *Transportation Research Part A*, 38:339–360, 2004.
- [Kim01] E. Kim. *Modeling intersections and other structures in highway alignment optimization*. PhD thesis, University of Maryland, College Park, 2001.
- [KJLS04] 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.
- [KJS05] Eungcheol Kim, Manoj K. Jha, and Bongsoo Son. Improving the computational efficiency of highway alignment optimization models through a stepwise genetic algorithms approach. *Transportation Research Part B*, 39:339–360, 2005.
- [LLRM98] Martine Labbé, Gilbert Laporte, and Inmaculada Rodríguez-Martín. *Fleet Management and Logistics*, chapter Path tree and cycle location, pages 187–204. Kluwer, 1998.
- [LMMO04] G. Laporte, A. Marín, J. Mesa, and F. Ortega. An integrated methodology for the rapid transit network design problem. *Lectures Notes in Computer Science*, 2004.
- [LMO98] G. Laporte, J. Mesa, and F. Ortega. Locating stations on rapid transit. *Technical report, Centre for Research on Transportation, Publication CRT*, 22, 1998.
- [LMOS05] G. Laporte, J.A. Mesa, F.A. Ortega, and I. Sevillano. Maximizing trip coverage in location of a single rapid transit alignment. *Annals of Operations Research*, 136:49–63, 2005.
- [MB96] Juan Antonio Mesa and Brian T. Boffey. A review of extensive facility location in networks. *European Journal of Operational Research*, 95:592–603, 1996.
- [MG07] A. Marín and R. García. Location of infrastructure in urban railway networks. *Computers & Operations Research*, 2007.
- [Mur73] J.D. Murchland. Methods of vertical profile optimization for an improvement to an existing road. *PTRC Seminar Proceedings on Cost Models and Optimization in Highways*, Session L12, 1973.

- [NEW76] A.J. Nicholson, D.G. Elms, and A. Williman. A variational approach to optimal route location. *The Journal of the Institution of Highway Engineers*, 23:22–25, 1976.
- [OEC73] Optimization of road alignment by the use of computers. Organization of Economic Cooperation and Development, Paris, 1973.
- [Par77] N.A. Parker. Rural highway route corridor selector. *Transportation Planning and Technology*, 3:247–256, 1977.
- [Puy73] J. Puy. Opygar: Optimization and automatic design of highway profiles. *PTRC Seminar proceedings on Cost Models and Optimization in Highways*, L13, 1973.
- [RG06] M^a Carmen Rafael and Ricardo García. Modelización y resolución numérica del problema de localización de curvas en el plano. aplicación al diseño de redes suburbanas de transporte privado. Septiembre 2006.
- [Rob73] R. Robinson. Automatic design of the road vertical alignment. *PTRC Seminar Proceedings on Cost Models and Optimization in Highways*, Session L9, 1973.
- [RWW97] C.S. Revelle, E.E. Whitlatch, and J.R. Wright. *Civil and Environmental Systems Engineering*. Prentice Hall, 1997.
- [SH81] J.F.B. Shawn and B.E. Howard. Comparison of two integration methods in transportation routing. *Transportation Research Record*, 806:8–13, 1981.
- [SH82] J.F.B. Shawn and B.E. Howard. Expressway route optimization by ocp. *Proceedings of the American Society of Swokowski*, 108:227–243, 1982.
- [SY99] Sadiq M. Sait and Habib Youssef. *Iterative Computer Algorithms with Applications in Engineering. Solving Combinational Optimization Problems*. IEEE Computer Society, 1999.
- [TM71] A.K. Turner and R.D. Miles. A computer assisted method of regional route location. *Highway Research Record*, 348:1–15, 1971.
- [Tri87a] D. Trietsch. Comprehensive design of highway networks. *Transportation Science*, 21:26–35, 1987.
- [Tri87b] D. Trietsch. A family of methods for preliminary highway alignment. *Transportation Science*, 21:17–25, 1987.
- [TS88] N.R. Thomson and J.F. Sykes. Route selection through a dynamic ice field using the maximum principle. *Transportation Research Part B*, 22:339–356, 1988.
- [Ven02] P. Venkataraman. *Applied Optimization with MATLAB Programming*. Wiley-Interscience, 2002.

-
- [Wan95] F.Y.M Wan. *Introduction to the Calculus of Variations and its Applications*. Chapman & Hall, 1995.
- [WC97] Benjamin W. Wah and Yao-Jen Chang. Trace-based methods for solving nonlinear global optimization and satisfiability problems. *J. of Global Optimization*, 10(2):107–141, 1997.