



Optimización de rutas de transporte público con Algoritmos Genéticos

Magdalena Genovard Abad

MÀSTER UNIVERSITARI EN ENGINYERIA INFORMÀTICA

Intel·ligència Artificial

Samir Kanaan Izquierdo
Carles Ventura Royo

28/12/2016



Esta obra está sujeta a una licencia de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL PROYECTO FINAL

Título del proyecto:	Optimización de rutas de transporte público con Algoritmos Genéticos
Nombre del autor:	<i>Magdalena Genovard Abad</i>
Nombre del consultor/a:	Samir Kanaan Izquierdo
Nombre del PRA:	<i>Carles Ventura Royo</i>
Fecha de Entrega (mm/aaaa):	12/2016
Titulación o programa:	<i>Màster Universitari en Enginyeria Informàtica</i>
Área del Proyecto Final:	<i>Inteligencia Artificial</i>
Idioma del proyecto:	<i>Castellano</i>
Palabras Clave	<i>Optimización, Rutas, Algoritmos Genéticos</i>
Resumen del Proyecto (máximo 250 palabras):	
<p>El objetivo de este proyecto es la optimización de las líneas de transporte público de una ciudad a través del uso de algoritmos genéticos.</p> <p>Se dispone de una ciudad compuesta por un número determinado de barrios. Según la demanda de pasajeros que desean desplazarse de un barrio a otro y la distancia a la que se encuentran los barrios se buscará la mejor combinación de líneas de transporte posible.</p> <p>Además, se realizará una búsqueda en función de un porcentaje de ecología que debe ser escogido por el usuario. A mayor porcentaje de ecología indicado, la solución intentará encontrar una solución ecológica, es decir con un menor número de líneas. Sin embargo un porcentaje de ecología bajo antepondrá la rapidez con la que viajarán los pasajeros aunque conlleve un gran número de líneas y por tanto más vehículos circulando por la ciudad.</p> <p>Para realizar este proyecto se ha llevado a cabo una aplicación en Python. Dicha aplicación recibe como parámetro el número de barrios que componen la ciudad objeto de estudio y el porcentaje de ecología deseado.</p> <p>La aplicación genera un total de 8 simulaciones, 4 de ellas de 25 generaciones con poblaciones de 25, 50, 75, y 100 individuos, las 4 simulaciones restantes ejecutan 50 generaciones con la misma variabilidad poblacional. Al final se genera un informe en PDF con un estudio de la ciudad y donde se podrá observar la mejor configuración de líneas encontrada y la evolución de la optimización.</p>	

Abstract (in English, 250 words or less):

The goal in this project is the optimization of public transport routes in an ideal city using a genetic algorithm approach.

Disposing a city conformed of neighbourhoods, the passenger travel demand and distance between those neighbours, the most optimized line map is discovered.

Moreover, the algorithm has an ecologic factor parameter, set by the user. The higher the ecologic factor value, the more ecological the solution is. Ecological meaning a few number of transport lines. However, a low ecological factor would prioritize the passenger travel speed, even though it leads to a higher number of lines, so more vehicles are moving in the city.

The project has been implemented in Python, the application receives the number of neighbourhoods in the city and the ecological factor as input.

The application executes 8 simulations for each study, 4 of them using 25 generations using a population of 25, 50, 75, and 100 individual, and the other 4 simulations using 50 generations with the same variability then the first 4 simulations. In the end, a PDF report is generated, containing the city study where the best line configuration and the optimization evolution is exposed.



Índice

1. Introducción.....	1
1.1 Contexto y justificación del Proyecto.....	1
1.2 Objetivos del Proyecto.....	2
1.3 Enfoque y método seguido.....	2
1.4 Planificación del Proyecto.....	3
1.4.1 Recursos del Proyecto.....	3
1.4.2 Calendario del Proyecto.....	4
1.4.3 Etapas del Proyecto.....	5
1.4.4 Planificación del Proyecto.....	6
1.5 Breve resumen de productos obtenidos.....	6
1.6 Breve descripción de los otros capítulos de la memoria.....	6
2. Estudio del Problema.....	7
2.1 Análisis de la Ciudad.....	8
2.2 Análisis del Algoritmo Genético.....	8
2.3 Análisis de la Solución.....	8
3. Diseño.....	9
3.1 Diseño de la ciudad.....	9
3.2 Diseño del Algoritmo Genético.....	10
4. Implementación.....	18
5. Análisis de Resultados.....	22
5.1 Análisis del Impacto del Factor de Ecología.....	22
5.2 Análisis del Impacto del Factor de Mutación Individual y CrossOver.....	23
5.3 Análisis exhaustivo de una ciudad de 8 barrios.....	25
5.3.1 Análisis de la Ciudad de 8 barrios.....	25
5.3.2 Análisis Resultado con FE=0%.....	27
5.3.3 Análisis Resultado con FE=30%.....	31
5.3.4 Análisis Resultado con FE=50%.....	35
5.3.5 Análisis Resultado con FE=70%.....	39
5.3.6 Análisis Resultado con FE=100%.....	43
6. Conclusiones.....	47
6.1 Objetivos conseguidos.....	47
6.2 Observaciones sobre la planificación.....	47
6.3 Observaciones sobre los algoritmos genéticos.....	47
6.4 Observaciones personales.....	47
7. Glosario.....	49
8. Bibliografía.....	51
9. Anexos.....	52

Lista de figuras

Ilustración 1: Representación Problema de Mandl.....	1
Ilustración 2: Diagrama de Gantt.....	6
Ilustración 3: Diagrama de flujo del algoritmo genético.....	17
Ilustración 4: Diagrama de Flujo de la Aplicación.....	18
Ilustración 5: Ciudad de 5 barrios y Factor de Ecología = 50%.....	23
Ilustración 6: Ciudad de 10 barrios y Factor de Ecología = 50%.....	23
Ilustración 7: Ciudad de 15 barrios y Factor de Ecología = 50%.....	23
Ilustración 8: Ciudad de 20 barrios y Factor de Ecología = 50%.....	24
Ilustración 9: Ciudad de 25 barrios y Factor de Ecología = 50%.....	24
Ilustración 10: Ciudad de 30 barrios y Factor de Ecología = 50%.....	24
Ilustración 11: Ubicación de los barrios en una ciudad de 8 barrios.....	25
Ilustración 12: Matriz de Distancia de una ciudad de 8 barrios.....	26
Ilustración 13: Matriz de Demanda de una ciudad de 8 barrios.....	26
Ilustración 14: Solución más óptima encontrada para una ciudad de 8 barrios con FE 0%.....	27
Ilustración 15: Líneas de la solución más óptima encontrada para una ciudad de 8 barrios y FE 0%.....	27
Ilustración 16: Evolución de la optimización para una ciudad de 8 barrios y FE 0%.....	29
Ilustración 17: Evolución del valor de optimización según generacion y población para una ciudad de 8 barrios y un FE 0%.....	30
Ilustración 18: Solución más óptima encontrada para una ciudad de 8 barrios con FE 30%.....	31
Ilustración 19: Líneas de la solución más óptima encontrada para una ciudad de 8 barrios y FE 30%.....	31
Ilustración 20: Evolución de la optimización para una ciudad de 8 barrios y FE 30%.....	34
Ilustración 21:.....	36
Ilustración 22: Solución más óptima encontrada para una ciudad de 8 barrios con FE 50%.....	37
Ilustración 23: Líneas de la solución más óptima encontrada para una ciudad de 8 barrios y FE 50%.....	37
Ilustración 24: Evolución de la optimización para una ciudad de 8 barrios y FE 50%.....	40
Ilustración 25: Evolución del valor de optimización según generacion y población para una ciudad de 8 barrios y un FE 50%.....	41
Ilustración 26: Solución más óptima encontrada para una ciudad de 8 barrios con FE 70%.....	42
Ilustración 27: Líneas de la solución más óptima encontrada para una ciudad de 8 barrios y FE 70%.....	42
Ilustración 28: Evolución de la optimización para una ciudad de 8 barrios y FE 70%.....	45
Ilustración 29: Evolución del valor de optimización según generacion y población para una ciudad de 8 barrios y un FE 70%.....	45
Ilustración 30: Solución más óptima encontrada para una ciudad de 8 barrios con FE 100%.....	46

Ilustración 31: Líneas de la solución más óptima encontrada para una ciudad de 8 barrios y FE 100%.....	46
Ilustración 32: Evolución de la optimización para una ciudad de 8 barrios y FE 100%.....	49
Ilustración 33: Evolución del valor de optimización según generación y población para una ciudad de 8 barrios y un FE 100%.....	49

1. Introducción

1.1 Contexto y justificación del Proyecto

La búsqueda de algoritmos para lograr la optimización de rutas no es una búsqueda actual sino que está presente desde hace varios años. Dicha búsqueda avanza paralelamente al avance de la ingeniería informática, ya que dicho avance, permite crear nuevas soluciones cada vez más óptimas.

A mediados del siglo pasado, en 1930, se planteó el problema del agente viajero (TSP)¹, dicho problema, que consiste en obtener la ruta más óptima para que un viajero pase por todos los puntos de un mapa, ha sido ampliamente estudiado desde entonces, y su análisis y resolución para encontrar la ruta más óptima sigue estando vigente hoy en día. Inicialmente, la resolución se realizaba con algoritmos de fuerza bruta, sin embargo, dichos algoritmos quedaron obsoletos ante la complejidad que se iba añadiendo al problema, por lo que a continuación surgieron soluciones con algoritmos aproximados.

En 1979 Christoph E. Mandl² planteó el problema de la red suiza de transporte y expuso una solución del problema. La representación del problema es una red de 15 nodos y 21 arcos.

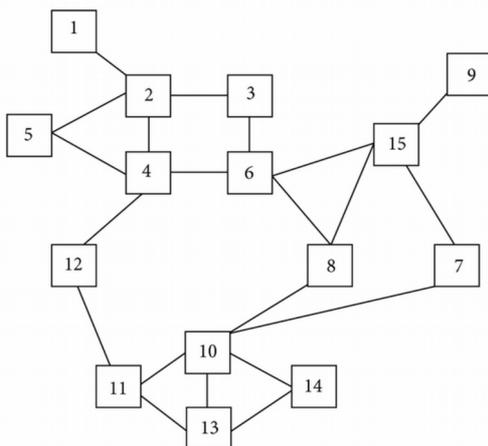


Ilustración 1: Representación Problema de Mandl

Este caso de estudio fue investigado por otros investigadores como Baaj and Mahmassani³ entre otros, cada uno de ellos obteniendo una optimización de rutas para el problema planteado.

La optimización de rutas es siempre un problema presente. En nuestra sociedad el crecimiento de las ciudades ha aumentado considerablemente las últimas décadas dejando insuficientes las rutas de transporte público que existían hasta entonces. Ante la aparición de nuevos núcleos urbanos, los consistorios deben cambiar o añadir nuevas rutas a su servicio de transporte público y ello en condiciones de sostenibilidad y presupuestos, lo cual no siempre es una tarea sencilla.

1 https://es.wikipedia.org/wiki/Problema_del_viajante

2 C. MANDL, *Applied Network Optimization*, Academic Press, London, 1979

3 Baaj and Mahmassani, Hybrid Route Generation Heuristic Algorithm for The Design of Transit Networks, (1995)

Este proyecto se plantea como objetivo obtener una optimización de rutas de transporte de una ciudad usando algoritmos genéticos. De esta manera a medida que la ciudad va cambiando con las nuevas necesidades de transporte solo habría que utilizar de nuevo el algoritmo para conseguir la optimización de las rutas existentes o ver la necesidad de crear nuevas rutas.

Los algoritmos genéticos son algoritmos evolutivos que se inspiran en el principio de la selección natural y como influye la genética en ella. Este tipo de algoritmos simulan la evolución de una población inicial a lo largo de varias generaciones para obtener a los mejores individuos. Por su naturaleza, los algoritmos genéticos son adecuados para resolver problemas de optimización complejos y con gran cantidad de factores.

1.2 Objetivos del Proyecto

Este proyecto se plantea como objetivo la utilización de algoritmos genéticos para encontrar una solución optimizada a las rutas de transporte urbano de una ciudad imaginaria.

A través de la realización de un algoritmo genético se resolverá la cantidad de líneas de transporte necesarias, así como la ruta de cada una de ellas, para que el resultado sea el más eficiente posible. La eficiencia se definirá de acuerdo a los objetivos que se quieran alcanzar. En el caso de Mandl⁴ la eficiencia se definía en el porcentaje de demanda de los viajeros satisfecha sin realizar transbordo así como añadir penalización a las rutas que necesitaban realizarlo.

En este proyecto se pretende añadir una nueva definición de eficiencia y obtener los resultados más óptimos de acuerdo a ella. Una definición de eficiencia, podría ser, como ya se ha comentado, que los viajeros realicen el menor número de trasbordo. Sin embargo una ciudad podría anteponer realizar una línea de transporte urbano sostenible frente al tiempo que se necesita para realizar los desplazamientos.

Los objetivos de este proyecto son los siguientes:

- Obtener la ruta más óptima de transporte público definiendo previamente la eficiencia, con un porcentaje de ecología que determinará si se desea conseguir el menor tiempo de viaje para el ciudadano o el menor número de rutas.
- Realizar comparativa de los resultados obtenidos con las diferentes eficiencias.

1.3 Enfoque y método seguido

Como se ha comentado en el apartado anterior el objetivo de este proyecto es la optimización de una red de transporte público de una ciudad imaginaria. Dicha ciudad será representada en forma de grafo, donde los nodos representarán islas de edificios y los arcos representan la distancia entre ellos. Además se utilizará otro grafo con los mismo nodos pero en este caso los arcos determinarán la cantidad de pasajeros que viajan de una zona a otra de la ciudad.

Para poder trabajar en la solución se representarán los grafos como una matriz donde las filas/columnas son los nodos de la ciudad y el contenido los valores de los arcos.

4 C. MANDL, *Applied Network Optimization*, Academic Press, London, 1979

Por todo ello , la implementación del algoritmo genético se llevará a cabo en un lenguaje de programación con soporte para representar grafos, vectores, matrices y lógicamente algoritmos genéticos.

Para determinar que lenguaje de programación utilizar para la implementación se ha realizado un estudio de los siguientes lenguajes: Python, C++, Perl y Java.

El lenguaje escogido para la realización del proyecto ha sido Python 3.5 por dos razones principales: dispone de librerías necesarias para llevar a cabo la implementación del algoritmo genético y es un lenguaje con el cual el desarrollador del proyecto ya ha trabajado por lo que se evita el coste de tiempo de aprendizaje que conllevaría la elección de un nuevo lenguaje.

A continuación se muestra una tabla con las librerías de Python necesarias para el desarrollo del proyecto.

Librería	
Nombre	Uso
NumPy ⁵	Manipulación de Matrices
Networkx ⁶	Creación de Grafos
Deap ⁷	Algoritmos Genéticos
Collections ⁸	Manipulación de Datos
Random ⁹	Números aleatorios
Matplotlib ¹⁰	Creación de Imágenes
PIL ¹¹	Manipulación de Imágenes
Reportlab ¹²	Creación de PDF

1.4 Planificación del Proyecto

1.4.1 Recursos del Proyecto

El desarrollo del proyecto cuenta con los siguientes recursos:

Recursos de Hardware	
Recurso	Características
Portátil Samsung NP300E5C	Intel core i5, 6GB RAM, 500 GB HDD

5 <http://www.numpy.org/>

6 <https://networkx.github.io/>

7 <http://deap.gel.ulaval.ca/doc/default/>

8 <https://docs.python.org/2/library/collections.html>

9 <https://docs.python.org/3/library/random.html>

10 <http://matplotlib.org/>

11 <http://www.pythonware.com/products/pil>

12 <http://www.reportlab.com/>

Recursos de Software	
Recurso	Uso
Ubuntu 16.04 LTS	Sistema operativo
Python	Lenguaje de programación
Oracle VM VirtualBox	Creación de una máquina virtual
Windows 7 Ultimate	Sistema Operativo
Project 2016	Creación Diagrama de Gantt
LibreOffice 5	Creación de Documentación
Firefox 49.0	Búsqueda de Información
Visor de documentos 3.18.2	Lector de Documentos

Recursos Personal	
Recurso	Características
Ingeniero Informático	Analista: Nivel Medio Desarrollador: Nivel Alto Redacción Documentación: Nivel Bajo

1.4.2 Calendario del Proyecto

La planificación se ha desarrollado según el siguiente calendario:

Calendario Laboral		
Día	Horario	Horas Totales
Lunes	16:00 – 19:00	3
Martes	16:00 – 19:00	3
Miércoles	16:00 – 19:00	3
Jueves	16:00 – 19:00	3
Viernes	16:00 – 19:00	3
Sábado	09:30 - 12:30 16:30 – 18:30	5
Domingo	No Laborable	0
Total Semanal		20

La planificación tiene en cuenta los siguientes días festivos:

Calendario Festivos	
Día	
12 de Octubre	12/10/2016
1 de Noviembre	01/11/2016
Puente de Diciembre	06/12/16 -11/12/2016
Navidad	23/12/16 – 26/12/16

1.4.3 Etapas del Proyecto

El proyecto se desarrolla en las siguiente etapas:

Etapas del Proyecto				
Etapa	Descripción	Tareas	Criticidad	Duración ¹³
Planificación	En esta etapa se realiza la definición del proyecto, se estudia el punto de partida y se realiza la planificación del proyecto	Búsqueda Información Algoritmos Genéticos	Media	5 horas
		Análisis Herramientas	Media	5 horas
		Definición del Proyecto	Muy Alta	10 horas
		Estado del Arte	Media	10 horas
		Planificación del Proyecto	Muy Alta	8 horas
		Documentación Etapa 1	Media	9 horas
Estudio, Diseño e implementación	En esta etapa se realiza el estudio del problema planteado, se realiza el diseño e implementación	Instalación Software	Media	1 horas
		Análisis del Problema	Muy Alta	9,5 horas
		Estudio Representación del Problema	Alta	9 horas
		Estudio Representación de la Solución	Alta	7 horas
		Diseño del Algoritmo Genético	Alta	9 horas
		Implementación del Algoritmo Genético	Alta	11 horas
		Realizar Pruebas Básicas	Media	5 horas
		Corregir Planificación del Proyecto	Media	1 horas
		Documentación Etapa 2	Media	2 horas
Análisis de Resultados	En esta etapa se realiza el análisis de los resultados y se realiza una comparativa	Realizar Pruebas	Media	20 horas
		Analizar Resultados	Muy Alta	20 horas
		Documentación Etapa 3	Media	15 horas
		Corregir posibles errores Implementación	Alta	20 horas
		Realizar Pruebas	Media	
		Corregir Planificación del Proyecto	Media	3 horas
Análisis Documentación Final	En esta etapa se lleva a cabo el análisis final de la aplicación y se desarrolla la documentación final	Documentación Final	Muy Alta	25 horas
		Realizar Presentación	Alta	10 horas
		Realizar Video Presentación	Alta	8 horas
		Corregir Planificación del Proyecto	Media	3 horas
		Entrega Final	Muy Alta	3 horas

¹³ La duración de las tareas se ha determinado según su complejidad y de acorde con el nivel que posee el desarrollador de las mismas

1.4.4 Planificación del Proyecto

A continuación se muestra el diagrama de Gantt con la planificación del proyecto, con una duración total de 220 horas y media.

TFM							
Id	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
1		TFM	220,5 horas	sáb 01/10/16	mié 28/12/16		Magda Genovard
2		Etapa 1 - Planificación	30 horas	sáb 01/10/16	mar 11/10/16		Magda Genovard
3		Búsqueda Información Algoritmos Genéticos	5 horas	sáb 01/10/16	sáb 01/10/16		Magda Genovard
4		Análisis Herramientas	5 horas	lun 03/10/16	mar 04/10/16	3	Magda Genovard
5		Definición del Proyecto	10 horas	mar 04/10/16	vie 07/10/16	4	Magda Genovard
6		Estado del Arte	10 horas	sáb 08/10/16	mar 11/10/16	5	Magda Genovard
7		Planificación del Proyecto	8 horas	sáb 01/10/16	lun 03/10/16		Magda Genovard
8		Documentación Etapa 1	9 horas	sáb 08/10/16	mar 11/10/16	5,7	Magda Genovard
9		Etapa 2 - Estudio, Diseño e Implementación	57 horas	jue 13/10/16	mié 02/11/16	2	Magda Genovard
10		Instalación Software	1 hora	jue 13/10/16	jue 13/10/16	2	Magda Genovard
11		Análisis del Problema	9,5 horas	jue 13/10/16	sáb 15/10/16	2	Magda Genovard
12		Estudio Representación del Problema	9 horas	sáb 15/10/16	mié 19/10/16	11	Magda Genovard
13		Estudio Representación de la Solución	7 horas	mié 19/10/16	sáb 22/10/16	12	Magda Genovard
14		Diseño del Algoritmo Genético	9 horas	sáb 22/10/16	mar 25/10/16	13	Magda Genovard
15		Implementación del Algoritmo Genético	11 horas	mar 25/10/16	sáb 29/10/16	14	Magda Genovard
16		Realizar Pruebas Básicas	5 horas	sáb 29/10/16	lun 31/10/16	15	Magda Genovard
17		Corregir Planificación del Proyecto	1 hora	lun 31/10/16	mié 02/11/16	16	Magda Genovard
18		Documentación Etapa 2	2 horas	mié 02/11/16	mié 02/11/16	17	Magda Genovard
19		Etapa 3 - Análisis	90 horas	vie 04/11/16	vie 02/12/16		Magda Genovard
20		Realizar Pruebas	20 horas	vie 04/11/16	vie 11/11/16	9	Magda Genovard
21		Analizar Resultados	20 horas	vie 11/11/16	vie 18/11/16	20	Magda Genovard
22		Documentación Etapa 3	15 horas	jue 18/11/16	mié 23/11/16	21	Magda Genovard
23		Corregir posibles errores Implementación	20 horas	vie 11/11/16	vie 18/11/16	20	Magda Genovard
24		Realizar Pruebas	10 horas	vie 18/11/16	lun 21/11/16	23	Magda Genovard
25		Corregir Planificación del Proyecto	3 horas	lun 21/11/16	mar 22/11/16	24	Magda Genovard
26		Etapa 4 - Análisis y Documentación Final	59 horas	jue 01/12/16	mié 28/12/16		Magda Genovard
27		Documentación Final	25 horas	sáb 03/12/16	sáb 17/12/16	19	Magda Genovard
28		Realizar Presentación	10 horas	sáb 17/12/16	mié 21/12/16	27	Magda Genovard
29		Realizar Video Presentación	8 horas	mié 21/12/16	mar 27/12/16	28	Magda Genovard
30		Corregir Planificación del Proyecto	3 horas	mié 28/12/16	mié 28/12/16	29	Magda Genovard
31		Entrega Final	3 horas	mié 28/12/16	mié 28/12/16		Magda Genovard

Ilustración 2: Diagrama de Gantt

1.5 Breve resumen de productos obtenidos

Programa realizado en lenguaje Python que utilizará algoritmos genéticos para obtener la configuración más óptima de las líneas de transporte de una ciudad.

Informes en formato PDF, donde se detallará las simulaciones ejecutadas y la mejor solución encontrada para cada una de las ejecuciones del programa realizadas por el usuario.

Memoria del Proyecto: documento en el cual se detallan la justificación, objetivos, planificación, análisis, diseño, implementación y resultados del proyecto.

1.6 Breve descripción de los otros capítulos de la memoria

A continuación se exponen los siguientes capítulos de la memoria:

- Estudio del Problema: donde se realizará el estudio del problema planteado.
- Diseño del Problema: donde se documentará el diseño del Problema, de sus componentes y de la solución encontrada.
- Implementación: donde se muestra la implementación del problemas
- Análisis de Resultados: donde se analizan los resultados obtenidos.
- Conclusiones: donde se realizan la conclusión obtenida tras la realización del proyecto.

2. Estudio del Problema

El objetivo de este proyecto es conseguir la configuración óptima de las líneas de transporte público de una ciudad. Para ello se implementará un algoritmo genético.

Como punto de partida se conocerá el número de barrios que conforman la ciudad, así como la demanda de pasajeros existente que desean viajar de un barrio a otro de la ciudad.

También se conoce que el número mínimo de barrios que componen la ciudad es 4, ya que una ciudad con menos barrios no es objeto de estudio. En ese caso se dispondría únicamente de dos líneas que unirían todos los barrios. Además se restringe que las líneas resultantes no pueden ser líneas circulares ni líneas que dejen barrios aislados, un habitante de un barrio cualquiera debe poder desplazarse a cualquiera de los barrios que componen la ciudad.

Para estudiar el problema se ha decidido identificar el punto de partida del problema, el objetivo del proyecto, sus necesidades y restricciones para poder así, analizar cada concepto por separado, realizando el análisis del problema siguiendo la técnica de "Divide y Vencerás"¹⁴.

Punto de Partida

- a. Se dispone de una ciudad imaginaria compuesta por un número determinado de barrios.

Objetivo

- a. Calcular las líneas necesarias, así como sus paradas, para conseguir optimizar al máximo las líneas de transporte público.

Necesidades

- a. Representar la ciudad, sus barrios y la demanda de pasajeros entre cada uno de ellos.
- b. Definir un algoritmo genético capaz de encontrar la solución con las líneas de transporte más óptimas.
- c. Definir la eficiencia de una línea de transporte.
- d. Representar la solución encontrada.

Restricciones

- a. La solución no debe dejar ningún barrio aislado.
- b. El número de barrios que componen la ciudad es mínimo 4.
- c. El número de líneas que debemos obtener como mínimo son 2, ya que no se aceptan líneas únicas que pasen por todos los barrios.
- d. Las líneas van y vuelven por el mismo camino, es decir cada línea tiene un barrio origen hasta un barrio destino pasando por barrios intermedios. Una vez que se llega al barrio destino vuelve al barrio origen pasando por los barrios intermedios esta vez de forma inversa.

Una vez se han definido los puntos anteriores se procede a analizar cada uno de los componentes implicados de forma individual.

14 https://es.wikipedia.org/wiki/Algoritmo_divide_y_vencer%C3%A1s

2.1 Análisis de la Ciudad

Para llevar a cabo el estudio de la optimización de las líneas de transporte se debe definir la ciudad objeto de estudio.

Se necesita conocer la distancia entre cada barrio así como los ciudadanos que desean viajar de un barrio a otro.

Para conocer la distancia entre cada barrio se utilizará la distancia de Manhattan¹⁵. La razón de utilizar la distancia de Manhattan frente a la distancia Euclidiana se debe a que la distancia más corta entre dos barrios no es la línea recta que los une, ya que en una ciudad existen edificios u otros obstáculos que no hacen posible un trayecto en línea recta¹⁶. Utilizando la distancia de Manhattan se asegura disponer de más de un camino con el mismo peso en distancia simulando así una distancia más real dentro de una ciudad.

2.2 Análisis del Algoritmo Genético

El algoritmo genético deberá ser capaz de evaluar la optimización de una serie de líneas de transporte.

Una línea se considerará óptima si en ella viajan un alto número de pasajeros de un barrio a otro recorriendo la menor distancia posible, de esta manera los pasajeros llegarán a su destino de manera rápida y eficiente. Sin embargo una solución con un número elevado de líneas frente al número de barrios no es sostenible desde el punto de vista de ecología y costes, es por ello, que se debe encontrar un equilibrio entre el número de líneas y el número de barrios que componen la ciudad.

Para hacer frente a este problema se decide añadir un factor de ecología que debe definir el usuario a la hora de empezar la simulación, cuanto mayor sea el factor ecológico más se penalizará la soluciones con elevado número de líneas.

Un Factor de Ecología igual 0 % indica que el usuario premia la rapidez de los pasajeros frente a los costes y factores ecológicos, cuanto mayor sea el % el algoritmo intentará dar una solución más equilibrada entre los dos mundos, ecología y eficiencia.

El algoritmo genético necesitará, además, del diseño de los individuos, definición de un valor de mutación individual y CrossOver, y definición de la función a maximizar.

2.3 Análisis de la Solución

La solución más óptima encontrada tras la realización del algoritmo genético debe proporcionar una cantidad de líneas de transporte que unan de la manera más eficiente posible los barrios de la ciudad.

Esta solución debe indicarnos cada línea de transporte que tendrá la ciudad, las paradas que realizará y el orden de las mismas.

Además, todos los barrios deben estar representados en alguna de estas líneas, ya que todos los barrios deben ser accesibles por los pasajeros desde cualquier barrio, ya sea viajando en la misma línea o realizando transbordo en algún barrio intermedio.

15 https://en.wiktionary.org/wiki/Manhattan_distance

16 https://en.wikipedia.org/wiki/Taxicab_geometry

3. Diseño

Una vez realizado el análisis de los puntos anteriores se lleva a cabo el diseño de cada uno de ellos que se expone a continuación.

3.1 Diseño de la ciudad

Tras el análisis de la ciudad, se conoce que está formada por barrios de los cuales necesitamos conocer la distancia entre ellos y la demanda de pasajeros que desean viajar a cada uno de ellos desde un barrio origen. Para que ello sea posible, se ha decidido que el diseño de la ciudad es el siguiente:

- La ciudad tiene un área finita y delimitada que contendrá los barrios que la componen.
- La localización de los barrios se representa con las coordenadas cartesianas de la posición que ocupan dentro del área de la ciudad.

Ejemplo de la Representación de la localización de los barrios de una ciudad de 8 barrios:

[[-26, -47], [6, 22], [-50, -29], [-31, 24], [-9, -40], [-29, -12], [46, -30], [-6, 43]]

- Los barrios de una ciudad se denominan de forma correlativa empezando desde 0.

Ejemplo de la Representación del nombre de los barrios de una ciudad de 8 barrios:

[0, 1, 2, 3, 4, 5, 6, 7]

- La distancia entre cada barrio es la distancia de Manhattan calculada a partir de sus coordenadas cartesianas. Esta distancia se guarda en forma de matriz cuadrada de dimensión el número de barrios.

Ejemplo de la Representación de la distancia entre barrios de una ciudad de 8 barrios, donde cada fila representa un barrio origen y las columnas un barrio destino. En esta matriz en concreto, la distancia entre el barrio 0 y 4 sería la posición (0,4) que sería 24.

```
[[ 0. 101. 42. 76. 24. 38. 89. 110.]  
 [101. 0. 107. 39. 77. 69. 92. 33.]  
 [ 42. 107. 0. 72. 52. 38. 97. 116.]  
 [ 76. 39. 72. 0. 86. 38. 131. 44.]  
 [ 24. 77. 52. 86. 0. 48. 65. 86.]  
 [ 38. 69. 38. 38. 48. 0. 93. 78.]  
 [ 89. 92. 97. 131. 65. 93. 0. 125.]  
 [110. 33. 116. 44. 86. 78. 125. 0.]]
```

- La demanda de pasajeros que desean ir de un barrio a otro se define como un número entero y se guarda en forma de matriz cuadrada de dimensión el número de barrios.

Ejemplo de la Representación de la demanda de pasajeros entre barrios de una ciudad de 8 barrios, donde, de manera similar a la representación anterior, cada fila representa un barrio origen y las columnas los barrios destino. En esta matriz en concreto, se aprecia que los pasajeros que desean ir del barrio 0 al 4 sería la posición (0,4) que serían 4.

```
[[ 0 9 6 10 4 54 21 51]
 [28 0 16 44 4 41 4 24]
 [26 51 0 50 14 56 16 1]
 [39 14 12 0 25 40 52 8]
 [19 55 60 42 0 37 32 37]
 [ 6 10 59 36 60 0 18 10]
 [35 37 25 5 32 9 0 3]
 [43 4 27 17 6 42 9 0]]
```

3.2 Diseño del Algoritmo Genético

Para definir el diseño del algoritmo genético se debe definir el diseño de sus elementos esenciales: individuo, sus atributos, su método de mutación individual, su método de CrossOver, el método de selección de los individuos y el método de cálculo de evaluación de un individuo.

3.2.1 Diseño del Individuo

Cada individuo es una solución al problema en forma de líneas de transporte de la ciudad, para ello se ha definido a los individuos como un tipo de individuos con un atributo de tipo ruta.

El atributo de tipo líneas se ha definido como una lista de listas donde cada lista es una línea de transporte.

Ejemplo del atributo líneas de una individuo de una ciudad compuesta de 7 barrios.

Atributo líneas = [[4, 6, 5, 1, 3, 2], [2, 3, 1, 6, 4, 5], [3, 5, 2, 0]]

Representa una solución formada por 3 líneas de transporte

1ª Línea: Recorre los barrios 4 - 6 - 5 - 1 - 3 - 2

2ª Línea: Recorre los barrios 2 - 3 - 1 - 6 - 4 - 5

3ª Línea : Recorre los barrios 3 - 5 - 2 - 0

3.2.2 Diseño de la Mutación Individual

Se ha implementado un método propio para la realización de la mutación individual.

El diseño de este método de mutación individual en pseudocódigo es el siguiente:

```
Se recibe un individuo.
Se guarda la longitud original del individuo (tamaño de líneas que lo componen).
Se genera un número aleatorio que decide si el individuo muta o no.
  Si el individuo muta
    Se genera aleatoriamente una nueva longitud para el individuo.
    Si la nueva longitud es menor que la longitud original
      Se recorre las líneas desde 0 hasta la nueva longitud
        Se genera un número aleatorio para decidir si esa línea muta
          Si muta:
            Se crea una nueva línea aleatoria que sustituye a la original
            Se eliminan las rutas sobrantes del individuo para que su longitud se ajuste a la longitud
nueva.
    Si la nueva longitud es igual o mayor que la original
      Se recorre las líneas desde 0 hasta la longitud nueva
        Si estamos en una línea que aun existe en la longitud original
          Se genera un número aleatorio para decidir si esa línea muta
            Si muta:
              Se crea una nueva línea aleatoria que sustituye a
la original
          Si estamos en una posición de línea que no existen en la longitud original
            Se genera aleatoriamente una nueva línea que se añade al individuo.
```

El algoritmo genético tiene configurado una probabilidad de mutación individual que se usará para comparar con el número creado aleatoriamente para decidir si el individuo muta o no.

A continuación se exponen ejemplos de mutación individual que se han producido en la ejecución del algoritmo genético y se analizan para comprobar el funcionamiento del método de mutación.

Ejemplo de mutación individual de un individuo compuesto originalmente por 6 líneas:

Individuo Original : [[4, 7, 3, 5, 0, 2, 1], [2, 3, 0, 5, 4, 7, 1], [3, 7, 6], [0, 5, 6, 4], [7, 5, 4, 2, 6, 0], [3, 1, 5, 2, 0, 6, 7]]

Individuo Mutado: [[4, 7, 3, 5, 0, 2, 1], [2, 3, 0, 5, 4, 7, 1]]

En el ejemplo anterior se puede comprobar como un individuo compuesto por 6 líneas ha mutado para convertirse en un individuo compuesto por tan solo 2 líneas. También se observa que sus dos líneas no han mutado por lo que el individuo conserva sus dos primeras líneas y se eliminan las restantes.

Ejemplo de mutación individual de un individuo compuesto originalmente por 2 líneas:

Individuo Original : [[4, 7, 3, 5, 0, 2, 1], [2, 3, 0, 5, 4, 7, 1]]

Individuo Mutado: [[1, 0, 7, 2, 6, 4, 5], [0, 4, 1, 7, 5, 2, 6], [0, 4], [2, 6, 7, 3, 5, 4], [1, 7, 2, 6, 0]]

En el ejemplo anterior se puede comprobar como un individuo compuesto por 2 líneas ha mutado para convertirse en un individuo compuesto por 5 líneas. También se observa que sus dos primeras líneas han mutado por lo que el individuo no conserva ninguna línea original, ha mutado tanto en longitud como en las líneas que lo conforman.

Ejemplo de mutación individual de un individuo compuesto originalmente por 2 líneas:

Individuo Original : [[2, 3, 1, 4, 6], [1, 4, 2, 5, 6, 3]]

Individuo Mutado: [[0, 5, 1, 3, 7], [6, 7, 2, 5, 1, 0, 4]]

En el ejemplo anterior se puede comprobar como un individuo compuesto por 2 líneas ha mutado pero en este caso conservando su longitud original. Se observa que sus dos primeras líneas han mutado por lo que el individuo no conserva ninguna línea original, ha mutado tan solo en las líneas que lo conforman.

Ejemplo de mutación individual de un individuo compuesto originalmente por 6 líneas:

Individuo Original : [[3, 0, 6, 4, 7, 5, 1], [6, 1, 5, 3, 7], [1, 3, 6, 4, 0, 5, 2], [7, 4, 5, 3, 6], [4, 2, 0, 5], [0, 4, 2, 1, 5, 6]]

Individuo Mutado: [[0, 5, 6, 2, 7, 1, 3], [6, 1, 5, 3, 7]]

En el ejemplo anterior se puede comprobar como un individuo compuesto por 6 líneas ha mutado su longitud para convertirse en un individuo compuesto por 2 líneas. Se observa que su primera línea muta no así la segunda línea la cual conserva igual que la original, este individuo ha mutado en longitud y parcialmente en alguna de sus líneas.

Con los ejemplos analizados anteriormente queda de manifiesto que con el método de mutación individual definido para el caso concreto de estudio, un individuo puede mutar en longitud y a su vez conservar las líneas que lo componen o perderlas y obtener nuevas líneas.

Con ello se pretende conseguir una mayor diversidad en las líneas para conseguir que el algoritmo trabaje con mayor número de probabilidades de líneas y poder obtener así la mejor solución posible.

3.2.3 Diseño del CrossOver

En el diseño del algoritmo genético también es necesario definir el diseño del CrossOver que ocurrirá entre dos individuos.

En este caso, igual que se ha realizado para la mutación individual, se ha desarrollado un método de CrossOver propio, adaptado al caso de estudio para intentar conseguir la mayor diversidad de individuos.

El diseño de este método de CrossOver en pseudocódigo es el siguiente:

```
Se reciben los dos individuos
Se guarda la mínima longitud de los dos individuos.
Se elige la línea del individuo 1 que va a formarse de nuevo con el CrossOver
Se elige la línea del individuo 2 que va a formarse de nuevo con el CrossOver
Se crea la nueva ruta para el individuo 1
    Se elige aleatoriamente parte de la línea del individuo 1
    Se elige aleatoriamente parte de la línea del individuo 2
    Se crea la nueva línea con los dos puntos anteriores
    Se eliminan las paradas repetidas
    La nueva línea se introduce en el individuo 1
Se crea la nueva ruta para el individuo 2
    Se elige aleatoriamente parte de la línea del individuo 1
    Se elige aleatoriamente parte de la línea del individuo 2
    Se crea la nueva línea con los dos puntos anteriores
    Se eliminan las paradas repetidas
    La nueva línea se introduce en el individuo 2
```

El algoritmo genético tiene configurado una probabilidad de CrossOver que determinará la probabilidad de que se produzca el CrossOver entre dos individuos.

A continuación se exponen ejemplos de CrossOver entre dos individuos.

Ejemplo de CrossOver entre dos individuos de diferente longitud, se observa que han mutado su segunda línea:

Individuo 1 :[[1, 3, 0], [0, 1, 4, 3], [0, 4, 2, 3]]

Individuo 2: [[1, 4, 0, 3], [0, 2, 3, 1]]

Resultado del CrossOver

Individuo 1 : [[1, 3, 0], [0, 1, 2], [0, 4, 2, 3]]

Individuo 2: [[1, 4, 0, 3], [0, 1, 4, 2]]

Ejemplo de CrossOver entre dos individuos de diferente longitud, se observa que el individuo 1 ha mutado su segunda línea y el individuo 2 la tercera:

Individuo 1 : [[4, 3, 2], [2, 4, 0], [3, 4, 1, 0]]

Individuo 2: [[1, 3, 4, 2], [2, 3, 0], [3, 1, 4], [4, 2]]

Resultado del CrossOver

Individuo 1 : [[4, 3, 2], [2, 4, 3], [3, 4, 1, 0]]

Individuo 2: [[1, 3, 4, 2], [2, 3, 0], [3, 4, 1], [4, 2]]

3.2.4 Diseño de la función Objetivo

El objetivo de este proyecto es encontrar la combinación de las líneas de transporte más eficiente para conectar los barrios de una ciudad, para encontrar dicha combinación a través de un algoritmo genético debemos definir la función objetivo a maximizar.

Una solución será más eficiente cuando sus líneas transporten mayor número de pasajeros de un barrio a otro, recorriendo la menor distancia posible.

Sin embargo una solución con una proporción de número de líneas respecto al número de barrios elevada puede no ser deseada ya que su ejecución conllevaría un gran número de vehículos de transporte recorriendo gran parte de la ciudad con el gasto y efecto contaminante que ello conlleva, sin embargo los pasajeros no esperarían demasiado tiempo para poder viajar al barrio deseado.

Encontrar el equilibrio ideal entre esta relación es difícil, es por ello que en este proyecto se ha incluido un porcentaje de ecología que debe ser indicado al algoritmo, de esta manera cuanto mayor es el porcentaje se entiende que no se desea una solución que incluya demasiadas líneas mientras que cuanto menor sea ese porcentaje se antepone la comodidad de los viajeros, que viajen rápido de un punto a otro, frente al efecto contaminante y gasto de tener más vehículos en la ciudad en circulación.

La función objetivo debe poder calcular la optimización de una solución teniendo en cuenta el factor de ecología determinado para poder aplicar una penalización según el número de líneas.

Además a la hora de realizar la valoración de un individuo se debe tener en cuenta que pueden aparecer individuos no válidos. Los individuos inválidos son aquellos en los que su atributo representa unas líneas de transporte que no son deseadas ya que en su conjunto cumplen al menos uno de los siguientes puntos:

- Tienen rutas repetidas
- Dejan barrios aislados a los cuales es imposible que un ciudadano pueda viajar.
- Alguna de sus líneas quedan aisladas siendo imposible que los ciudadanos de ciertos barrios se puedan desplazar a otros de la ciudad.

Ejemplo de Individuos Inválidos.

Individuo 1 : [[0, 3], [0, 2, 1, 5, 6]], en este caso es inválido ya que con esta combinación de líneas el barrio 4 queda aislado y sin posibilidad de llegar a él.

Individuo 2: [[2, 7], [0, 4, 5, 2, 7, 1, 3], [2, 7]], en este caso existen dos líneas repetidas

Individuo 3: [[1, 2], [0, 3, 4, 7, 5, 6]] todos los barrios aparecen y no hay rutas repetidas, sin embargo desde los barrios 1 y 2 solo se puede viajar entre ellos y no sería posible viajar a los otros barrios.

Para poder calcular el valor de optimización de la solución que representa un individuo, se realizan los siguientes pasos:

1) Se crea la matriz de adyacencias, es decir, el número de conexiones que hay entre cada uno de los diferentes barrios. La matriz de adyacencias se crea como una matriz cuadrada de dimensiones número de barrios, donde se representa la distancia que hay que recorrer para desplazarse de un barrio a otro, los barrios que no están conectados se representan con el valor 0.

Ejemplo de la Matriz de Adyacencias representada por una solución de una ciudad de 5 barrios. En ella se observa que el barrio 0 no tienen conexión directa con el barrio 1, el barrio 1 no tiene conexión directa con el barrio 0 ni el 2, el barrio 2 no tiene conexión directa con el barrio 1 ni el 4 y el barrio 4 no tiene con el barrio 2. Se observa que el barrio 3 tiene conexión directa con todos los barrios.

```
[[ 0.  0. 42. 76. 24.]
 [ 0.  0.  0. 39. 77.]
 [42.  0.  0. 72.  0.]
 [76. 39. 72.  0. 86.]
 [24. 77.  0. 86.  0.]]
```

2) A partir de la matriz de adyacencias se genera un grafo con la representación de los diferentes barrios y sus conexiones. El objetivo de la creación de este grafo es poder calcular la distancia más corta para ir de un barrio a otro que nos permite las rutas que componen el individuo, para realizar este cálculo se utiliza el algoritmo de Dijkstra's¹⁷. Como en la matriz de adyacencias se ha guardado la distancia entre los barrios, al buscar el camino más corto los nodos del grafo no tienen el mismo peso, permitiendo así que el algoritmo de Dijkstra's encuentre el camino más corto en distancia y no en número de saltos.

3) Se crea la matriz de distancia donde se muestra la distancia más corta que debe realizar un pasajero para ir de un barrio a otro. A la hora de calcular la matriz de distancia por cada parada que debe realizarse se aumenta en 5 el valor de distancia para penalizar rutas con muchas paradas que retrasarán la llegada de un pasajero a un barrio.

Ejemplo de la Matriz de Distancia representada por el individuo [[4, 5, 1, 6, 7, 3, 0], [3, 1, 2, 5, 0, 7], [5, 3, 4, 6, 0, 1]], en ella se observa que para ir del barrio 0 al 1 el coste es 101, para ir al 2 es 86...

```
[[ 0. 101. 86. 76. 96. 38. 89. 110.]
 [ 101. 0. 107. 39. 127. 69. 92. 93.]
 [ 86. 107. 0. 86. 96. 38. 181. 150.]
 [ 76. 39. 86. 0. 86. 38. 141. 44.]
 [ 96. 127. 96. 86. 0. 48. 65. 140.]
 [ 38. 69. 38. 38. 48. 0. 123. 92.]
 [ 89. 92. 181. 141. 65. 123. 0. 125.]
 [ 110. 93. 150. 44. 140. 92. 125. 0.]]
```

4) Una vez se ha obtenido la matriz de distancia de la solución, se calcula la matriz de optimización de un individuo como la división entre la matriz de pasajeros de la ciudad y la matriz de distancia que representa ese individuo calculada en el paso anterior.

5) Se suma los componentes de la matriz de optimización definiendo así el valor de optimización del individuo.

6) A continuación se calcula el factor de ecología que se debe aplicar. Como se ha explicado anteriormente, si solo se tiene en cuenta para calcular la optimización de una solución la cantidad de pasajeros que desplaza una ruta entre la distancia que recorre, las soluciones con mayor número de rutas se valorarán por encima de las soluciones con menor número de rutas. Debido a ello, según el factor de ecología definido en este paso se aplica una penalización a la solución.

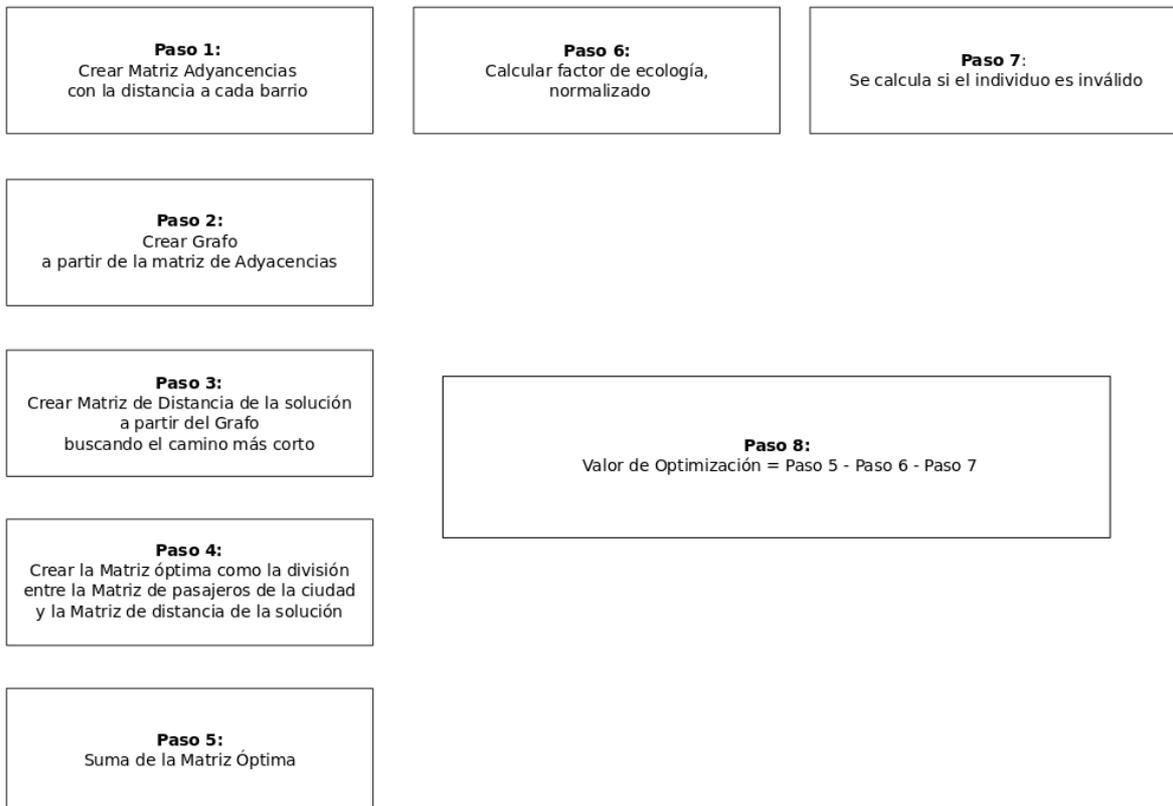
Se conoce que el mínimo de rutas posible de una solución es 2 y el mayor número el número de barrios -1. Por ello el factor de ecología se normaliza para obtener un valor entre 0 y 1 que se multiplica al valor obtenido en el paso 5, obteniendo la penalización de ecología.

17 https://en.wikipedia.org/wiki/Dijkstra's_algorithm

7) Se comprueba si el individuo es inválido según los preceptos explicados anteriormente y si es así se añade una penalización suficientemente grande para evitar que estos individuos sean escogidos.

8) El valor de optimización de una solución se calcula como el valor de optimización encontrado en el paso 5 menos la penalización de ecología calculado en el paso 6 menos la penalización por individuo inválido calculado en el paso 7.

A continuación se muestra un esquema con los pasos a realizar para evaluar un individuo:



A continuación se muestra el diagrama de flujo del algoritmo genético

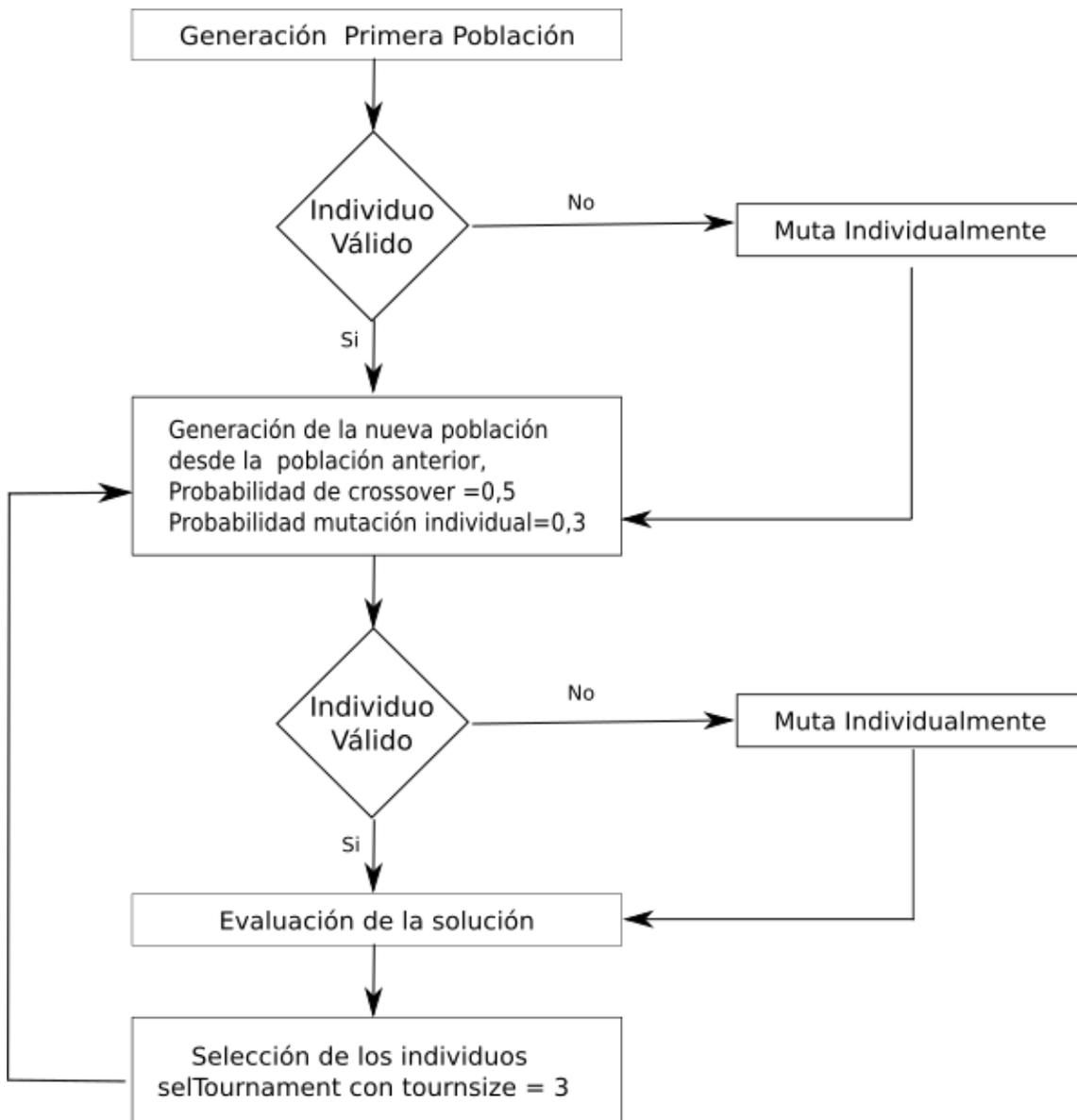


Ilustración 3: Diagrama de flujo del algoritmo genético

4. Implementación

El primer paso de la aplicación es pedir al usuario que introduzca el número de barrios que componen la ciudad y el porcentaje de ecología que desea, dando así la oportunidad al usuario que decida el porcentaje de penalización sobre el número de líneas de la solución.

A continuación la aplicación crea la ciudad y configura las variables necesarias y prepara el informe inicial.

La aplicación realiza 8 simulaciones, 4 de ellas de 25 generaciones con poblaciones de 25, 50, 75, y 100 individuos, las 4 simulaciones restantes ejecutan 50 generaciones con la misma variabilidad poblacional.

Al final se crea un informe final que se encuentra en la misma ruta de archivos donde se ejecuta la aplicación y una carpeta que contiene todas las imágenes generadas indicando el número de generaciones y de población que han logrado esa solución.

A continuación se expone el diagrama de flujo de la aplicación:

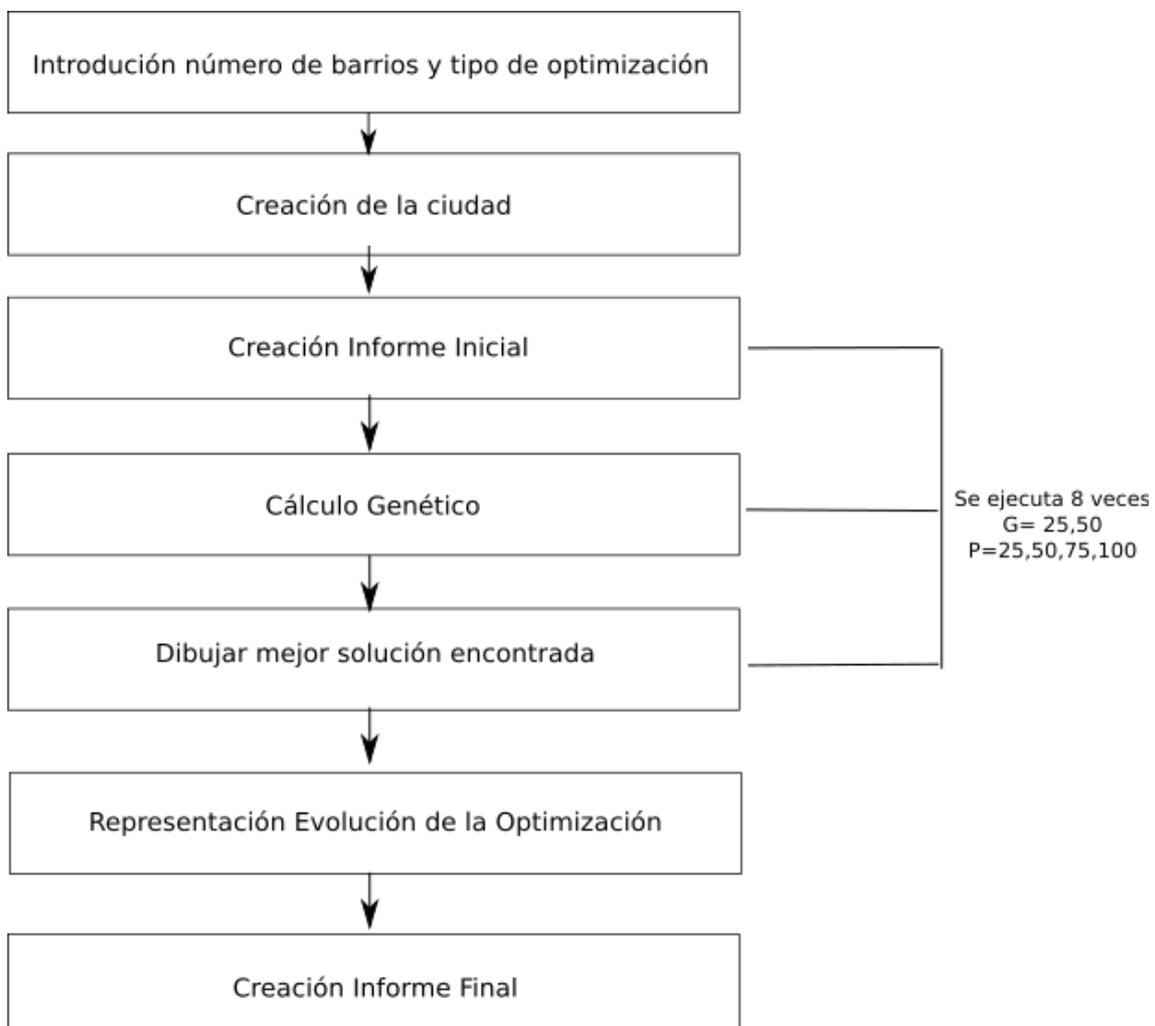


Ilustración 4: Diagrama de Flujo de la Aplicación

La implementación de la aplicación se ha desarrollado en 6 ficheros en Python:

Fichero <i>TFM.py</i>	
Es el Archivo principal y el que debe ejecutarse, compuesto de cuatro partes diferenciadas:	
Recogida y definición de variables	Número de barrios que componen la ciudad, porcentaje de factor de ecología deseado, número de generaciones y población, probabilidad de Crossover y mutación individual.
Creación de la ciudad	Llama al fichero ciudad.py para crear la ciudad y sus barrios.
Llamada al algoritmo genético.	Llama al fichero genetico.py que es donde se realizará las simulaciones para la búsqueda de la mejor solución.
Creación del informe.	Llama al fichero pdf.py y gráficos.py, ya que crea un informe final en formato PDF con los resultados de todas las simulaciones que se encuentra en la misma ruta de ficheros.

Fichero ciudad.py	
Archivo que contiene la definición de la clase ciudad y su implementación.	
Crea la ciudad y sus dos matrices, de pasajeros y de distancia entre barrios.	En la implementación inicial de la clase se crea aleatoriamente un vector con las coordenadas de los barrios y a continuación se crean las dos matrices, la de distancia, donde se guarda la distancia de Manhattan entre los diferentes barrios, y la matriz de pasajeros, donde se guarda los pasajeros que desean ir de un barrio a otro.
Dispone de 2 funciones.	<p>coste_ruta(): que devuelve el coste en distancia entre dos barrios de la ciudad.</p> <p>pasajeros_ruta(): que devuelve el número de pasajeros que desean ir de un barrio a otro.</p>

Fichero genetico.py	
Es el archivo que contiene todo el algoritmo genético y todas sus funciones.	
Se crea el individuo, y se define los siguientes rasgos necesarios para el funcionamiento del algoritmo	<p>Los individuos tienen como atributos el tipo ruta</p> <p>Una población es una lista de individuos.</p> <p>El método de selección de los individuos es <code>selTournament</code> con <code>toursize = 3</code></p> <p>El método de <code>CrossOver</code> es una función propia <code>CrossOver</code></p> <p>Se crea un decorator, para controlar los individuos inválidos que llama a la función propia <code>comprobar()</code></p> <p>La mutación individual se realiza con una función propia <code>mutacion_individual()</code>.</p>
Dispone de las siguientes funciones:	<p><code>líneas()</code>: que crea el atributo de un individuo, devuelve una lista con las líneas que conforman el individuo.</p> <hr/> <p><code>calcular_penalizacion()</code>: donde se comprueba si el individuo es inválido por cumplir al menos una de las 3 razones comentadas en el capítulo anterior.</p> <hr/> <p><code>crear_matriz_adyacencias()</code>: donde se crea la matriz de adyacencias entre barrios que representa el individuo.</p> <hr/> <p><code>crear_matriz_distancias()</code>: donde se crea la matriz de distancia entre barrios según el camino más corto posible con las rutas que componen el individuo.</p> <hr/> <p><code>crear_grafo()</code>: donde se crea el grafo que representa el individuo.</p> <hr/> <p><code>calcular_optimizacion_solucion()</code>: donde se calcula la optimización del individuo.</p> <hr/> <p><code>mutacion_individual()</code>: donde se define la mutación individual de un individuo.</p> <hr/> <p><code>CrossOver()</code>: donde se define el <code>CrossOver</code> entre dos individuos.</p> <hr/> <p><code>comprobar()</code>: función que comprueba si el individuo es inválido y en tal caso lo obliga a mutar individualmente.</p>

Fichero <i>graficos.py</i>	
En este archivo se realizan los gráficos que ayudarán a realizar el informe final.	
Dispone de 3 funciones.	Representar_evolucion(): crea una imagen que representa la evolución de la optimización de la simulación.
	Representar_informe_final(): crea la imagen de optimización final y escribe el informe.
	Dibujar_solucion() : crea la imagen con la representación de la solución encontrada y escribe dicha parte en el informe.

Fichero <i>pdf.py</i>	
En este archivo se realiza la creación del informe en PDF con la información obtenida tras la realización de la simulación.	
Dispone de 3 funciones.	crear_informe_inicial(): función que crea el informe inicialmente.
	crear_pagina_informe(): función que añade una página al informe.
	informe_final(): función que realiza el informe final.

Fichero <i>normalizar.py</i>	
En este fichero se lleva a cabo una función de normalización del valor de la optimización de la mejor solución encontrada.	
Dispone de la función.	normalizar_optimizacion_solucion(): función que normaliza el valor.

5. Análisis de Resultados

Una vez se ha concluido la implementación del algoritmo se han llevado a cabo diversas ejecuciones variando el número de barrios que componen la ciudad así como el factor de ecología deseado.

Todos los resultados, que en este apartado se analizan, se han obtenido tras realizar la ejecución del algoritmo con una configuración de 25 y 50 generaciones y indicando el número de población en 25, 50, 75 y 100.

5.1 Análisis del Impacto del Factor de Ecología

A continuación se muestra una tabla con el número de líneas de la solución encontrada tras la ejecución de diversas configuraciones en función del número de barrios y del factor de ecología indicado.

Número de Líneas de la Solución Más Óptima					
Número de Barrios	FE 0 %	FE 30 %	FE 50%	FE 70%	FE 100%
4	3	2	2	2	2
5	4	2	2	2	2
6	5	2	2	2	2
7	6	3	2	2	2
8	7	3	2	2	2
9	8	4	2	2	2
10	9	4	2	3	2
15	14	5	4	3	2
20	18	11	8	4	2
25	22	11	10	6	4
30	27	12	9	7	6
40	39	23	14	13	6
50	48	27	14	13	11

Como se puede observar el número de líneas más alto se obtiene con porcentajes de ecología más bajos, esto es un comportamiento esperado, ya que si se le indica un factor de ecología bajo el usuario está indicando que no desea penalizar la relación del número de líneas alto respecto al número de barrios. Sin embargo, si se aumenta el factor de ecología se indica que se desea una solución más sostenible aunque haya mayores tiempos de espera por lo que el número de líneas se reduce.

5.2 Análisis del Impacto del Factor de Mutación Individual y CrossOver

Otra parte importante del algoritmo genético es definir el valor de Mutación Individual y de CrossOver. Ajustarlos a valores muy bajos no es óptimo ya que no favorece la aparición de nuevos individuos. Tampoco es óptimo ajustarlos a valores muy altos, debido a que los individuos degeneran aleatoriamente en otros que no tienen porque ser mejores. Es por eso que debe encontrarse un equilibrio entre los dos factores que favorezca la búsqueda de un buen resultado.

A continuación se han realizado diversas simulaciones con diferentes valores de Mutación Individual y de CrossOver para comprobar el valor máximo de optimización encontrado, se han realizado las pruebas variando el tamaño de la ciudad. A continuación se muestran los resultados obtenidos en forma de tabla:

NB=5 FE 50%					
Mutpb \ Cxpb	0,1	0,3	0,5	0,7	0,9
0,1	1,304	1,334	1,313	1,334	1,332
0,3	1,361	1,353	1,361	1,353	1,353
0,5	1,361	1,345	1,348	1,361	1,361
0,7	1,361	1,345	1,353	1,332	1,361
0,9	1,353	1,353	1,353	1,361	1,361

Ilustración 5: Ciudad de 5 barrios y Factor de Ecología = 50%

NB=10 FE 50%					
Mutpb \ Cxpb	0,1	0,3	0,5	0,7	0,9
0,1	1,079	1,103	1,135	1,078	1,130
0,3	1,195	1,222	1,174	1,174	1,120
0,5	1,120	1,184	1,184	1,184	1,220
0,7	1,187	1,187	1,195	1,235	1,157
0,9	1,183	1,183	1,190	1,161	1,179

Ilustración 6: Ciudad de 10 barrios y Factor de Ecología = 50%

NB=15 FE 50%					
Mutpb \ Cxpb	0,1	0,3	0,5	0,7	0,9
0,1	0,950	0,976	1,020	0,998	0,956
0,3	1,050	0,992	1,046	1,035	1,001
0,5	1,053	1,030	1,103	1,094	1,015
0,7	1,081	1,060	1,037	1,045	1,045
0,9	1,075	1,075	1,035	1,095	1,079

Ilustración 7: Ciudad de 15 barrios y Factor de Ecología = 50%

NB=20 FE 50%					
Mutpb \ Cxpb	0,1	0,3	0,5	0,7	0,9
0,1	1,003	0,952	1,000	0,986	1,001
0,3	1,009	1,037	1,009	1,032	1,039
0,5	1,025	1,086	1,040	1,021	1,044
0,7	1,069	1,046	1,062	1,051	1,039
0,9	1,032	1,048	1,095	1,080	1,068

Ilustración 8: Ciudad de 20 barrios y Factor de Ecología = 50%

NB=25 FE 50%					
Mutpb \ Cxpb	0,1	0,3	0,5	0,7	0,9
0,1	0,957	0,974	0,990	0,992	0,991
0,3	0,985	0,989	1,019	1,031	1,025
0,5	0,989	1,038	1,030	1,032	1,027
0,7	1,014	1,035	1,021	1,047	1,018
0,9	1,014	1,042	1,051	1,013	1,040

Ilustración 9: Ciudad de 25 barrios y Factor de Ecología = 50%

NB=30 FE 50%					
Mutpb \ Cxpb	0,1	0,3	0,5	0,7	0,9
0,1	0,938	0,965	0,973	0,982	0,970
0,3	0,998	0,993	0,980	0,999	0,993
0,5	0,997	1,001	0,995	1,014	0,990
0,7	1,001	1,022	0,993	1,024	0,988
0,9	1,028	0,990	1,020	1,015	1,022

Ilustración 10: Ciudad de 30 barrios y Factor de Ecología = 50%

Tras el estudio se decide realizar las simulaciones con un factor de Mutación Individual de 0,3 y un factor de CrossOver en 0,5. Si bien no es la configuración más óptima para todos los casos, tras analizar los resultados anteriores se han decidido estos valores ya que se acercan al equilibrio entre producir nuevas rutas que aportan diversidad a la población pero sin crear un caos de aleatoriedad que podría conllevar una pérdida de la selección natural.

A continuación se van a analizar las soluciones concretas encontradas para una ciudad de 8 barrios. Se ha escogido esta cantidad de barrios para realizar el análisis detallado ya que es un número significativo y no demasiado extenso en número de líneas para poder realizar un análisis.

5.3 Análisis exhaustivo de una ciudad de 8 barrios

En este apartado se va a analizar los resultados obtenidos para una ciudad compuesta por 8 barrios y un factor de ecología determinado en 0%, 30%, 50%, 70% y 100%.

Todas las simulaciones se han realizado con un factor de Mutación Individual = 0,3 y un factor de CrossOver = 0,5.

5.3.1 Análisis de la Ciudad de 8 barrios

La configuración de la ciudad de 8 barrios, la posición de sus barrios y la demanda de pasajeros entre barrios es independiente del factor de ecología deseado por el usuario. Debido a ello, el análisis de la ciudad se analizará generalmente en este apartado ya que servirá como punto de partida para analizar los resultados concretos por factor de ecología.

Como se puede observar en la segunda página de cada informe generado, tras la simulación con una ciudad de 8 barrios encontramos la siguiente información importante:

Los barrios ordenados de mayor a menor demanda son [4 6 3 2 1 7 5 0]

Las rutas con mayor demanda son del barrio 4 al 2 y del 5 al 4.

Las rutas con menor demanda es la que va del barrio 2 al 7.

La ubicación cartesiana de los 8 barrios:

[[[-26, -47], [6, 22], [-50, -29], [-31, 24], [-9, -40], [-29, -12], [46, -30], [-6, 43]]

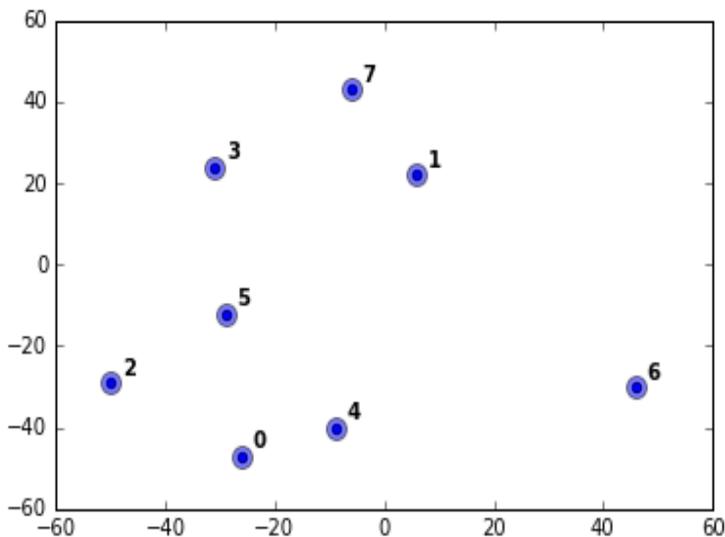


Ilustración 11: Ubicación de los barrios en una ciudad de 8 barrios.

La matriz de distancia de la ciudad de 8 barrios

-	0	1	2	3	4	5	6	7
0	0	101	42	76	24	38	89	110
1	101	0	107	39	77	69	92	33
2	42	107	0	72	52	38	97	116
3	76	39	72	0	86	38	131	44
4	24	77	52	86	0	48	65	86
5	38	69	38	38	48	0	93	78
6	89	92	97	131	65	93	0	125
7	110	33	116	44	86	78	125	0

Ilustración 12: Matriz de Distancia de una ciudad de 8 barrios.

A continuación se muestra la matriz de demanda de pasajeros. Se han resaltado los valores por encima de la media para que se aprecien los barrios con mayor demanda de pasajeros.

-	0	1	2	3	4	5	6	7
0	0	9	6	10	4	54	21	51
1	28	0	16	44	4	41	4	24
2	26	51	0	50	14	56	16	1
3	39	14	12	0	25	40	52	8
4	19	55	60	42	0	37	32	37
5	6	10	59	36	60	0	18	10
6	35	37	25	5	32	9	0	3
7	43	4	27	17	6	42	9	0

Ilustración 13: Matriz de Demanda de una ciudad de 8 barrios

5.3.2 Análisis Resultado con FE=0%

Se ha generado una simulación de una ciudad de 8 barrios y un factor de ecología definido en 0%.

Como podemos ver en el informe generado, la mejor solución encontrada ha sido una solución formada por 7 líneas [[5, 3, 7, 4, 0, 2, 6], [6, 5, 0, 3, 4, 2], [7, 1, 5, 2, 6, 0], [2, 3, 1, 4, 5], [2, 7, 5, 0], [3, 0, 1, 2, 4, 6], [0, 6, 1, 2, 7, 3,5]], obtenida en una simulación de 25 generaciones y 75 población.

A continuación la imagen de representación de las líneas de la solución:

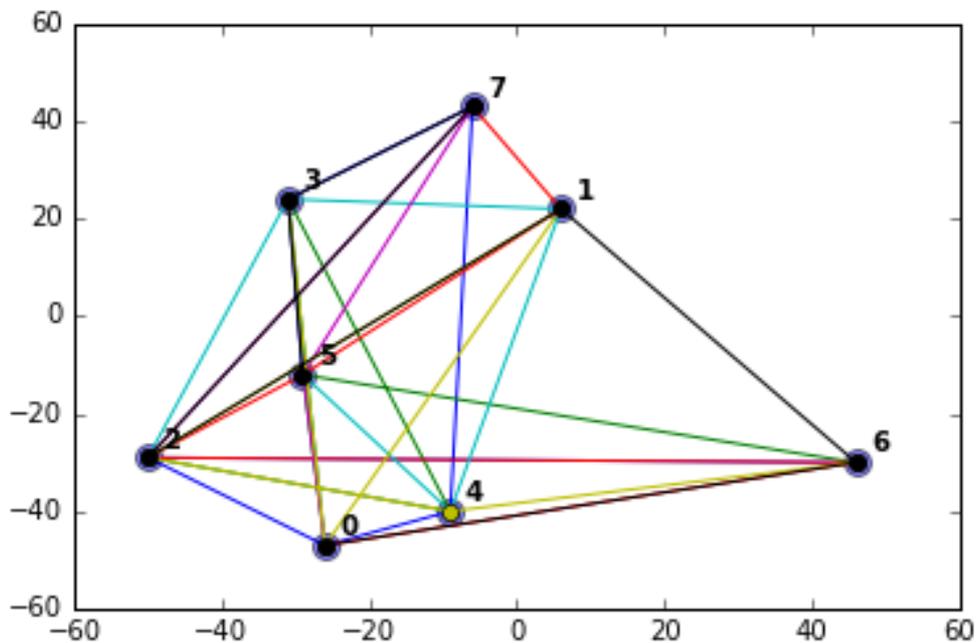


Ilustración 14: Solución más óptima encontrada para una ciudad de 8 barrios con FE 0%

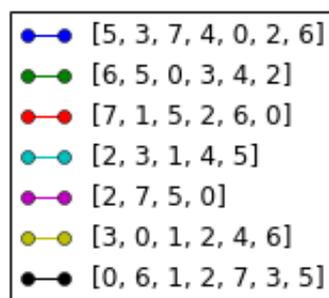


Ilustración 15: Líneas de la solución más óptima encontrada para una ciudad de 8 barrios y FE 0%

El número de líneas de la solución es el máximo permitido, siendo el número de barrios menos 1. Obtener un número elevado de líneas era lo esperado ya que se ha indicado un factor de ecología a 0%, con esto el usuario indica que desprecia el factor ecológico y antepone la comodidad de los viajeros y transportar el mayor número de pasajeros recorriendo la menor distancia posible aunque para ello tenga un número elevado de líneas recorriendo la ciudad.

A continuación se muestra, para poder comparar fácilmente, la matriz de distancia de la ciudad, la matriz de distancia de la mejor solución encontrada y la matriz de demanda de la ciudad.

Matriz de Distancia de la ciudad

-	0	1	2	3	4	5	6	7
0	0	101	42	76	24	38	89	110
1	101	0	107	39	77	69	92	33
2	42	107	0	72	52	38	97	116
3	76	39	72	0	86	38	131	44
4	24	77	52	86	0	48	65	86
5	38	69	38	38	48	0	93	78
6	89	92	97	131	65	93	0	125
7	110	33	116	44	86	78	125	0

Matriz de Distancia de la Solución

-	0	1	2	3	4	5	6	7
0	0	101	42	76	24	38	89	120
1	101	0	107	39	77	69	92	33
2	42	107	0	72	52	38	97	116
3	76	39	72	0	86	38	141	44
4	24	77	52	86	0	48	65	86
5	38	69	38	38	48	0	93	78
6	89	92	97	141	65	93	0	135
7	120	33	116	44	86	78	135	0

Matriz de Demanda de la Ciudad

-	0	1	2	3	4	5	6	7
0	0	9	6	10	4	54	21	51
1	28	0	16	44	4	41	4	24
2	26	51	0	50	14	56	16	1
3	39	14	12	0	25	40	52	8
4	19	55	60	42	0	37	32	37
5	6	10	59	36	60	0	18	10
6	35	37	25	5	32	9	0	3
7	43	4	27	17	6	42	9	0

Se puede observar que la matriz de distancia de la solución es muy parecida a la matriz de distancia de la ciudad, esto es lógico ya que la solución dispone de 7 líneas por lo que se puede ir en pocos transbordo de un barrio a otro debido a un gran número de conexiones existentes.

En la solución se observa que se ha creado dos líneas, una entre los barrios 2, 7, 5, 0 y otra entre los barrios 2, 3, 1, 4, 5. Es un resultado esperado ya que son barrios que están relativamente cerca.

Si se observa la matriz de distancia de la solución por filas se aprecia lo siguiente:

En la fila del barrio 0, se percibe que todas las conexiones son directas con todos los barrios excepto con el barrio 7 ya que es un barrio que se encuentra lejano al barrio 0.

En las filas de los barrios 1, 2, 4 y 5, se observa que todas las conexiones son directas con todos los barrios. Los ciudadanos se pueden desplazar directamente a cualquier barrio de la ciudad.

En la fila del barrio 3, se ve que todas las conexiones son directas con todos los barrios excepto con el barrio 6 ya que es un barrio que se encuentra lejano al barrio 3.

En la fila del barrio 6, se destaca que todas las conexiones son directas con todos los barrios excepto con los barrios 3 y 7. El barrio 6 es el barrio más alejado del resto de los barrios que componen la ciudad.

En la fila del barrio 7, se advierte que todas las conexiones son directas con todos los barrios excepto con los barrios 0 y 6 ya que son barrios muy alejados.

Estos resultados son lógicos ya que al tener un número elevado de líneas la mayoría de los barrios están conectados directamente.

A continuación, en la imagen, se aprecia la evolución del valor de optimización a través del desarrollo de las generaciones:

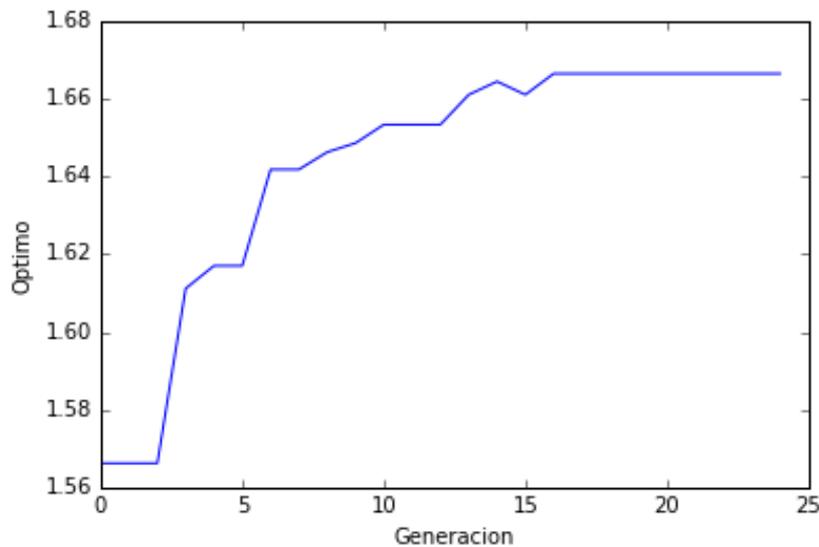


Ilustración 16: Evolución de la optimización para una ciudad de 8 barrios y FE 0%

En la imagen se aprecia que el valor de optimización va en aumento de forma proporcional al número de generaciones, ha aumentado más rápidamente hasta la generación número 8. Después sigue aumentando ligeramente hasta la generación 17, para seguir estable hasta la generación 25.

En la siguiente imagen se muestra la evolución del valor de optimización según el número de generaciones y poblaciones simuladas. Se observa, que en este caso, con un factor de ecología igual al 0% lo que realmente influye en la búsqueda de una solución es el número de población que debe ser cercano al 75, y el número de generaciones no influye tanto en el resultado.

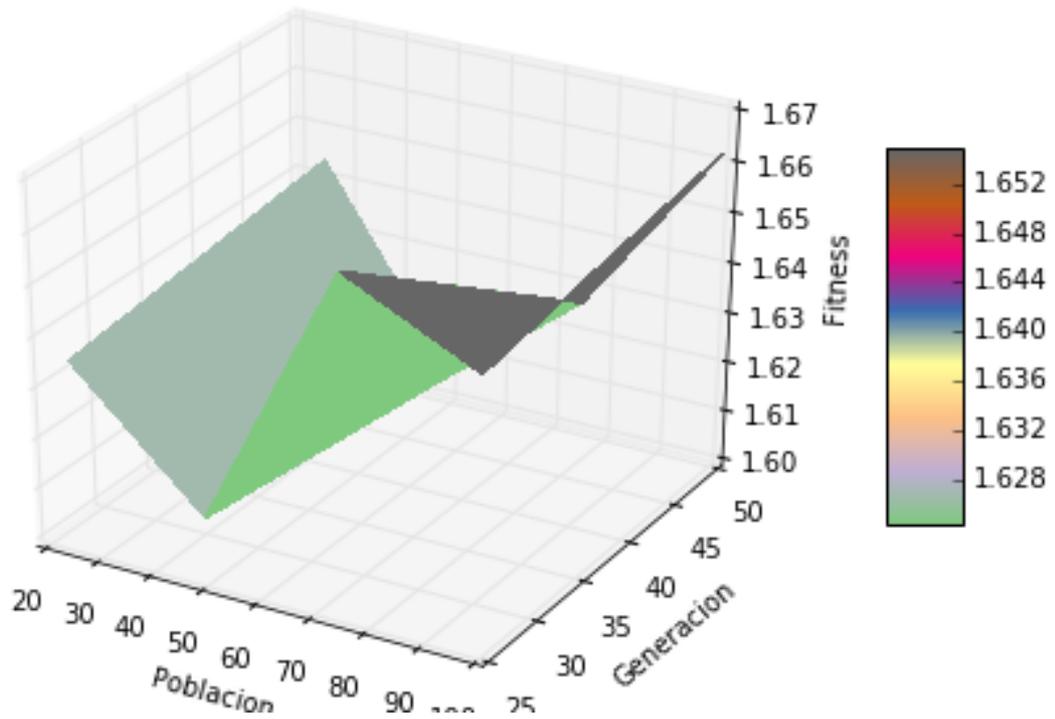


Ilustración 17: Evolución del valor de optimización según generacion y población para una ciudad de 8 barrios y un FE 0%

5.3.3 Análisis Resultado con FE=30%

Como podemos ver, en el informe generado, la mejor solución encontrada ha sido una solución formada por 3 líneas, [[2, 5, 3, 1, 6, 0, 7], [7, 6, 4, 5, 0], [0, 2, 4, 1, 7, 5]], obtenida en una simulación de 25 generaciones y 50 de población.

Al haber indicado ahora un factor de ecología del 30%, la solución encontrada tiene un número de líneas bastante menor que en el apartado anterior, 3 frente a 7, pero que en realidad es un valor con mayor equilibrio respecto al número de barrios que componen la ciudad.

A continuación la imagen de representación de las líneas de la solución:

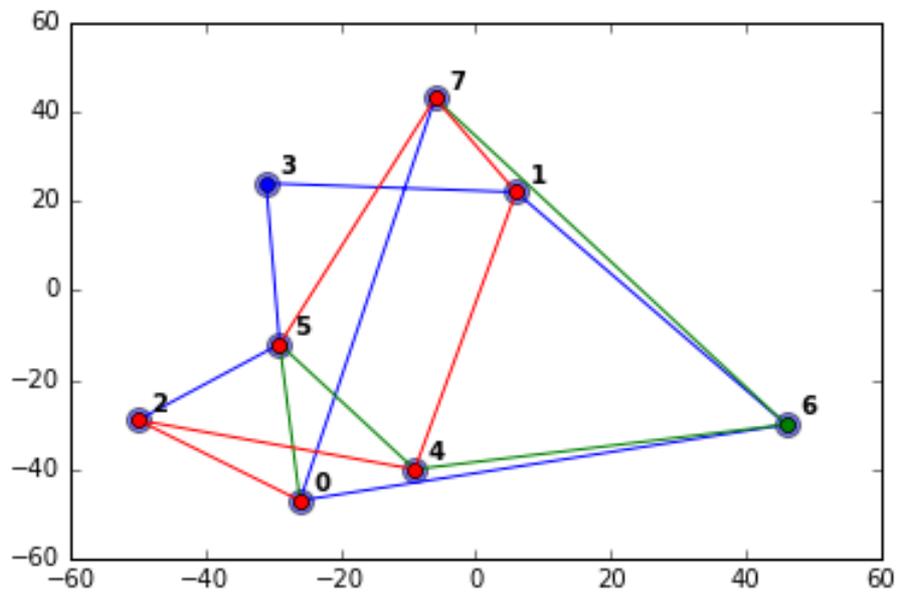


Ilustración 18: Solución más óptima encontrada para una ciudad de 8 barrios con FE 30%

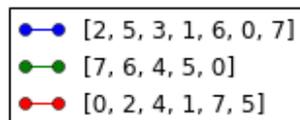


Ilustración 19: Líneas de la solución más óptima encontrada para una ciudad de 8 barrios y FE 30%

A continuación se muestra, la matriz de distancia de la ciudad, la matriz de distancia de mejor solución encontrada y la matriz de demanda de la ciudad.

Matriz de Distancia de la ciudad

-	0	1	2	3	4	5	6	7
0	0	101	42	76	24	38	89	110
1	101	0	107	39	77	69	92	33
2	42	107	0	72	52	38	97	116
3	76	39	72	0	86	38	131	44
4	24	77	52	86	0	48	65	86
5	38	69	38	38	48	0	93	78
6	89	92	97	131	65	93	0	125
7	110	33	116	44	86	78	125	0

Matriz de Distancia de la Solución

-	0	1	2	3	4	5	6	7
0	0	145	42	86	96	38	89	110
1	145	0	145	39	77	87	92	33
2	42	145	0	86	52	38	127	126
3	86	39	86	0	96	38	141	82
4	96	77	52	96	0	48	65	120
5	38	87	38	38	48	0	123	78
6	89	92	127	141	65	123	0	125
7	110	33	126	82	120	78	125	0

Matriz de Demanda de la Ciudad

-	0	1	2	3	4	5	6	7
0	0	9	6	10	4	54	21	51
1	28	0	16	44	4	41	4	24
2	26	51	0	50	14	56	16	1
3	39	14	12	0	25	40	52	8
4	19	55	60	42	0	37	32	37
5	6	10	59	36	60	0	18	10
6	35	37	25	5	32	9	0	3
7	43	4	27	17	6	42	9	0

Se puede observar que contrariamente al apartado anterior, la matriz de distancia de la solución no es tan parecida a la matriz de distancia de la ciudad. Esto es lógico ya que la solución dispone de 3 líneas por lo que los barrios no están directamente tan conectados como en el apartado anterior.

Si se observa la matriz de distancia de la solución por filas se aprecia lo siguiente:

En la fila del barrio 0 se percibe que todas las rutas entre barrios han aumentado, excepto las rutas con los barrios 2, 5, 6 y 7. En el caso de las rutas con los barrios 5, 6 y 7 tienen una gran demanda de pasajeros. En el caso de la ruta con el barrio 2 no es así pero es un barrio cercano. La ruta que más ha aumentado es la del barrio 0 al 4, pasando de 24 a 96. En la distribución del mapa se observa que son barrios cercanos pero la demanda de pasajeros es mínima(4).

En la fila del barrio 1, se observa que todas las rutas han aumentado, menos las rutas con los barrios 3, 4, 6 y 7. En el caso de los barrios 3 y 7 son rutas con una gran demanda de pasajeros, como se observa en la matriz de demanda. En el caso del barrio 4 la gran demanda es de la ruta inversa del barrio 4 al 1(55). La ruta que más ha aumentado es la del barrio 1 al 0, pasando de 101 a 145. En la distribución del mapa se observa que son barrios alejados y la demanda de pasajeros de esta ruta no es elevada.

En la fila del barrio 2, se aprecia que todas las rutas han aumentado la distancia a recorrer, menos las rutas con los barrios 0, 4 y 5. En el caso del barrio 5 la demanda de esta ruta es alta, justificando el enlace directo y en el caso de los barrios 0 y 4 son barrios muy cercanos. La ruta que más ha aumentado es la del barrio 2 al 6, pasando de 97 a 127. En la distribución del mapa se observa que son barrios alejados y además es una ruta con baja demanda.

En la fila del barrio 3, se percibe que todas las rutas han aumentado la distancia a recorrer, menos las rutas con los barrios 1 y 5. En el caso del barrio 1 se observa que la demanda de pasajeros es baja(14) pero son barrios cercanos y la demanda del barrio 1 al 3 es elevada(44). En el caso del barrio 5 la demanda es elevada(40), además de ser barrios muy cercanos. La ruta que más ha aumentado es la del barrio 3 al 7, pasando de 44 a 82. En la matriz de demanda se observa que la demanda es baja(8).

En la fila del barrio 4, se percibe que todas las rutas han aumentado la distancia a recorrer, menos las conexiones con los barrios 1, 2, 5 y 6. En todos estos barrios la demanda de pasajeros es elevada. El enlace que más ha aumentado del barrio 4 al 7, pasando de 86 a 120.

En la fila del barrio 5, se percibe que todas las rutas han aumentado la distancia a recorrer, menos las conexiones con los barrios 0, 2, 3, 4 y 7. En el caso de los barrios 2, 3 y 4 la demanda es elevada además de ser barrios cercanos y en el caso del barrios 0 y 7 están muy cercanos al barrio 5. El enlace que más ha aumentado del barrio 5 al 6, pasando de 93 a 123, en la distribución del mapa se observa que son barrios relativamente lejanos y la demanda de pasajeros no es elevada(18).

En la fila del barrio 6, se percibe que todas las rutas han aumentado la distancia a recorrer, menos las rutas con los barrios 0, 1, 4 y 7. El barrio 6 está bastante alejado del resto de barrios, pero las rutas con los barrios 0, 1 y 4 son demandadas. La ruta que más ha aumentado es la del barrio 6 al 5, pasando de 93 a 123. En la distribución del mapa se observa que son barrios relativamente lejanos y la demanda de pasajeros no es elevada(9).

En la fila del barrio 7, se percibe que todas las rutas han aumentado la distancia a recorrer, menos las rutas con los barrios 0, 1, 5 y 6. En el caso de los barrios 0 y 6 tienen una demanda elevada. En el caso de los barrios 1 y 5 son barrios cercanos al barrio 7. La ruta que más ha aumentado es la del barrio 7 al 4, pasando de 86 a 120. En la distribución del mapa se observa que son barrios relativamente lejanos y la demanda de pasajeros no es elevada(6).

A continuación, en la imagen, se aprecia la evolución del valor de la optimización a través del desarrollo de las generaciones:

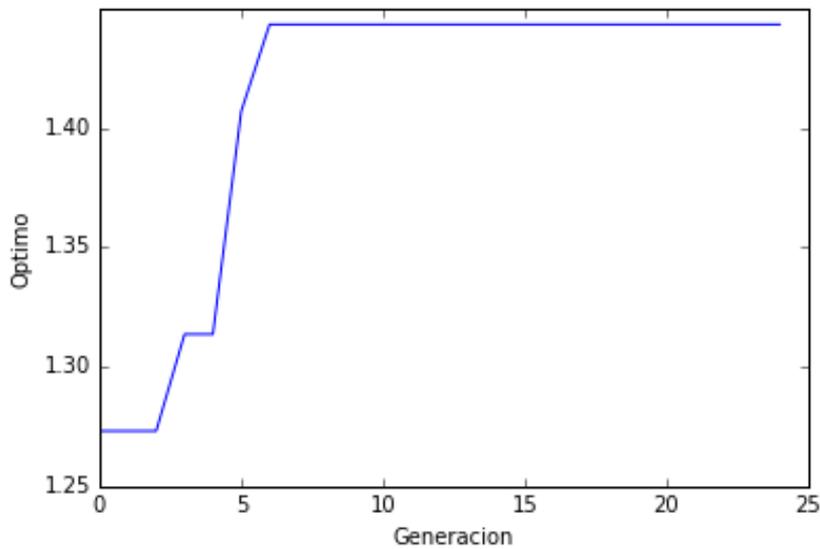


Ilustración 20: Evolución de la optimización para una ciudad de 8 barrios y FE 30%

En la imagen, se aprecia que el valor de optimización va en aumento rápidamente hasta la generación número 6, para después mantenerse estable hasta la generación 25.

En la siguiente imagen se muestra la evolución del valor de optimización según el número de generaciones y poblaciones simuladas. Se observa que, en este caso, con un factor de ecología igual al 30% lo que realmente influye en la búsqueda de una solución es el número de población, que debe ser cercano al 50, y el número de generaciones no influye tanto en el resultado.

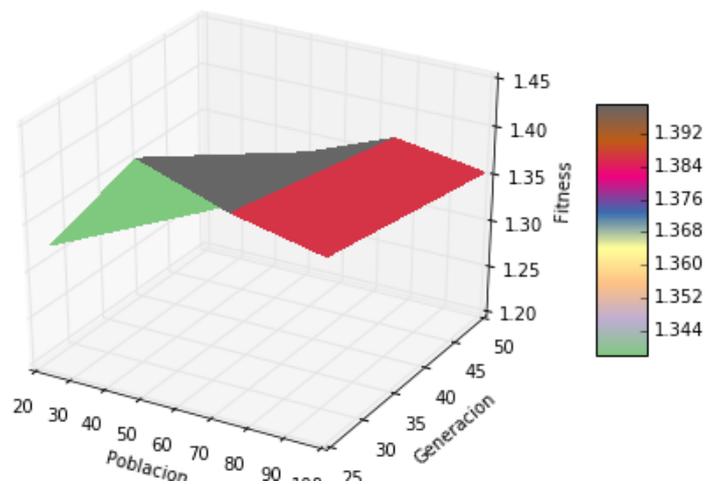


Ilustración 21:

Evolución del valor de optimización según generación y población para una ciudad de 8 barrios y un FE 30%

5.3.4 Análisis Resultado con FE=50%

Como podemos ver en el informe generado, la mejor solución encontrada ha sido una solución formada por 2 líneas, $[[6, 1, 5, 2, 4, 0], [2, 0, 5, 3, 1, 7, 4]]$, obtenida en una simulación de 50 generaciones y 100 población.

Al haber indicado ahora un factor de ecología del 50%, la solución encontrada tiene el número de líneas mínimo, 2, una menos que en el apartado anterior. Esto es debido al incremento del porcentaje del factor de ecología.

A continuación la imagen de representación de las líneas de la solución:

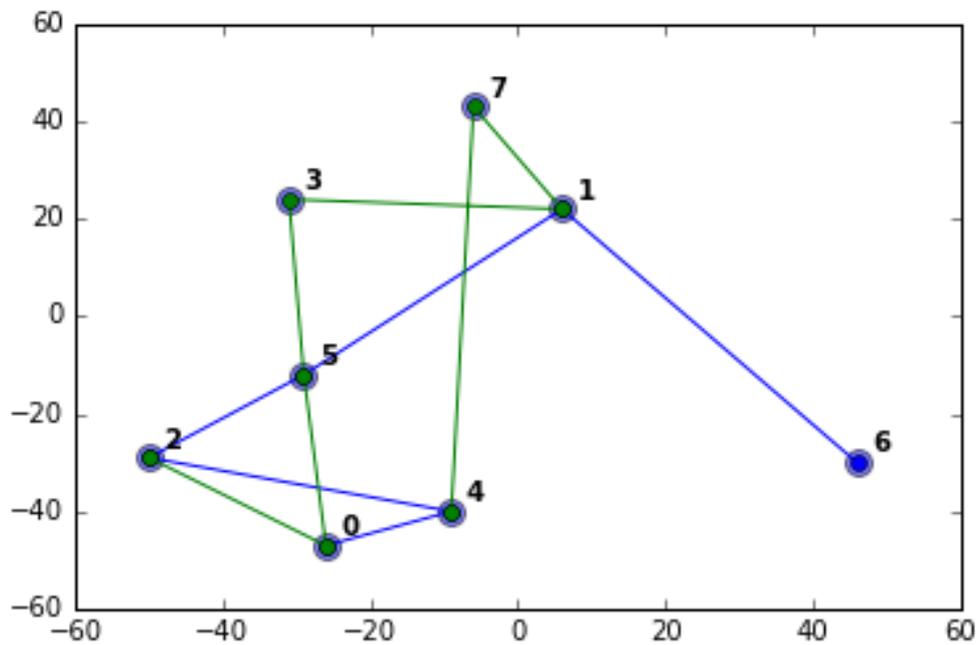


Ilustración 22: Solución más óptima encontrada para una ciudad de 8 barrios con FE 50%

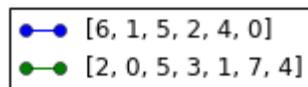


Ilustración 23: Líneas de la solución más óptima encontrada para una ciudad de 8 barrios y FE 50%

A continuación se muestra la matriz de distancia de la ciudad, la matriz de distancia de mejor solución encontrada y la matriz de demanda de la ciudad.

Matriz de Distancia de la ciudad

-	0	1	2	3	4	5	6	7
0	0	101	42	76	24	38	89	110
1	101	0	107	39	77	69	92	33
2	42	107	0	72	52	38	97	116
3	76	39	72	0	86	38	131	44
4	24	77	52	86	0	48	65	86
5	38	69	38	38	48	0	93	78
6	89	92	97	131	65	93	0	125
7	110	33	116	44	86	78	125	0

Matriz de Distancia de la Solución

-	0	1	2	3	4	5	6	7
0	0	117	42	86	24	38	229	120
1	117	0	117	39	129	69	92	33
2	42	117	0	86	52	38	229	148
3	86	39	86	0	130	38	141	82
4	24	129	52	130	0	72	241	86
5	38	69	38	38	72	0	171	112
6	229	92	229	141	241	171	0	135
7	120	33	148	82	86	112	135	0

Matriz de Demanda de la Ciudad

-	0	1	2	3	4	5	6	7
0	0	9	6	10	4	54	21	51
1	28	0	16	44	4	41	4	24
2	26	51	0	50	14	56	16	1
3	39	14	12	0	25	40	52	8
4	19	55	60	42	0	37	32	37
5	6	10	59	36	60	0	18	10
6	35	37	25	5	32	9	0	3
7	43	4	27	17	6	42	9	0

Se puede observar que al tener tan solo dos líneas la solución la matriz de distancia de la solución es mayor a la matriz de distancia de la ciudad.

Si se observa la matriz de distancia de la solución por filas se aprecia lo siguiente:

En la fila del barrio 0, se percibe que todas las rutas han aumentado la distancia a recorrer, menos las rutas con los barrios 2, 4 y 5. En el caso de los barrios 2 y 4 no tienen una gran demanda pero son barrios cercanos al barrio 0. En el caso del barrio 5 si tiene una gran demanda de pasajeros(54). La ruta que más ha aumentado es la del barrio 0 al 6, pasando de 89 a 229. En la distribución del mapa se observa que son barrios alejados y la demanda de pasajeros no es elevada(21).

En la fila del barrio 1, se observa que todas las rutas han aumentado la distancia a recorrer, menos las rutas con los barrios 3, 5, 6 y 7. En el caso de los barrios 3, 5 y 7 son rutas con una gran demanda de pasajeros, como se aprecia en la matriz de demandas. En el caso del barrio 6 no hay gran demanda de pasajeros pero es la única vía de enlace para llegar al barrio 6 y no dejarlo aislado del resto. La ruta que más ha aumentado es la del barrio 1 al 4, pasando de 77 a 129. En la distribución del mapa se observa que son barrios alejados y la demanda de pasajeros no es elevada(4).

En la fila del barrio 2, se percibe que todas las rutas han aumentado la distancia a recorrer, menos las rutas con los barrios 0, 4 y 5. En el caso del barrio 4 es una ruta con baja demanda desde el barrio 2 al 4, sin embargo de demanda máxima de modo inverso, de 4 al 2 (60). En el caso del barrio 5 la demanda de esta ruta es alta, justificando el enlace directo. Y en el caso del barrio 0 es un barrio muy cercano al barrio 2. La ruta que más ha aumentado es la

del barrio 2 al 6, pasando de 97 a 229. En la distribución del mapa se observa que son barrios muy alejados y la demanda de pasajeros no es elevada(16).

En la fila del barrio 3, se aprecia que todas las rutas han aumentado la distancia a recorrer, menos las rutas con los barrios 1 y 5. En el caso del barrio 1 se observa que la demanda de pasajeros es baja(14) pero son barrios cercanos y la demanda del barrio 1 al 3 es elevada(44). En el caso del barrio 5 la demanda es elevada(40) además de ser barrios muy cercanos. La ruta que más ha aumentado es la del barrio 3 al 4, pasando de 86 a 130. En la distribución del mapa se observa que son barrios relativamente lejanos y la demanda de pasajeros no es elevada(21).

En la fila del barrio 4, se percibe que todas las rutas han aumentado la distancia a recorrer, menos las rutas con los barrios 0, 2 y 7. En el caso de los barrios 2 y 7 son barrios con alta demanda de pasajeros. En el caso del barrio 0 la demanda es baja(19) pero son barrios que están muy cercanos. La ruta que más ha aumentado es la del barrio 4 al 6, pasando de 65 a 241. En la distribución del mapa se observa que son barrios relativamente lejanos y la demanda de pasajeros no es elevada(32).

En la fila del barrio 5, se percibe que todas las rutas han aumentado la distancia a recorrer, menos las rutas con los barrios 0, 1, 2 y 3. En el caso de los barrios 2 y 3 la demanda es elevada y en el caso de los barrios 0 y 1 son barrios muy cercanos al barrio 5. La ruta que más ha aumentado es la del barrio 5 al 6, pasando de 93 a 171. En la distribución del mapa se observa que son barrios relativamente lejanos y la demanda de pasajeros no es elevada(18).

En la fila del barrio 6, se observa que todas las rutas han aumentado la distancia a recorrer, menos las rutas con el barrio 1. El barrio 6 es el barrio que queda menos comunicado con los restantes ya que tan solo está conectado directamente con el barrio 1, sus pasajeros deben pasar por este barrio antes de poder desplazarse a cualquier otro.

En la fila del barrio 7, se percibe que todas las rutas han aumentado la distancia a recorrer, menos las rutas con los barrios 1 y 4. En el caso del barrio 4 la ruta 7 al 4 tiene baja demanda sin embargo la ruta del barrio 4 al 7 tiene una demanda elevada. En el caso del barrio 1 la demanda es baja pero los barrios están muy cerca. La ruta que más ha aumentado es la del barrio 7 al 3, pasando de 44 a 82. En la distribución del mapa se observa que son barrios relativamente lejanos y la demanda de pasajeros no es elevada(17).

A continuación, en la imagen, se aprecia la evolución del valor de la optimización a través del desarrollo de las generaciones:

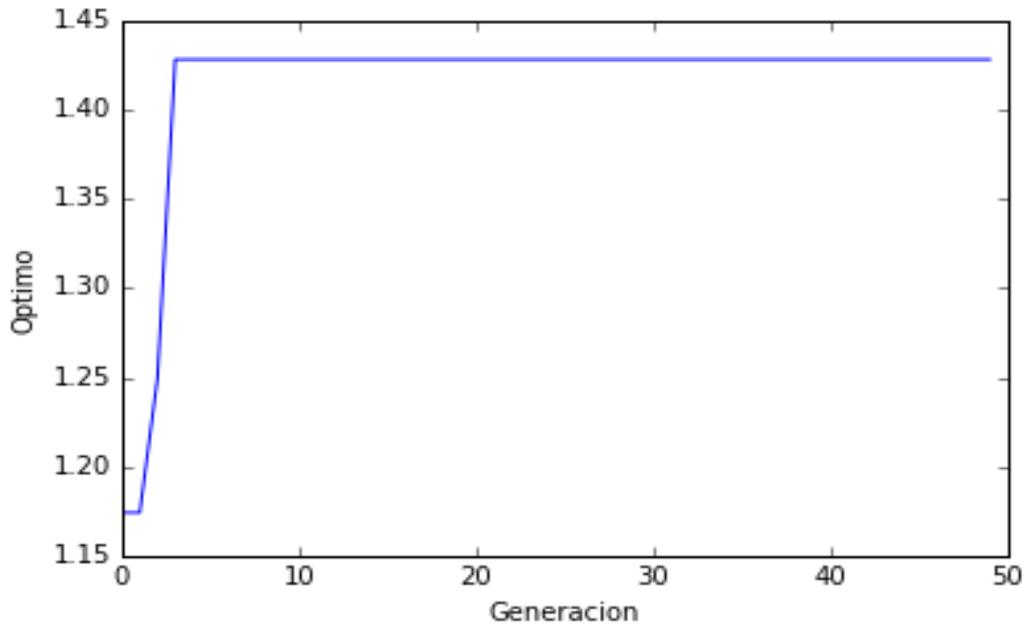


Ilustración 24: Evolución de la optimización para una ciudad de 8 barrios y FE 50%

En la imagen se observa que el valor de optimización se satura muy rápidamente, encontrando ya, en la generación 3 el valor máximo. Esto es debido a que el número de líneas es mínimo y las combinaciones posibles se encuentran rápidamente.

En la siguiente imagen se muestra la evolución del valor de optimización según el número de generaciones y poblaciones simuladas. Se observa que en este caso con un factor de ecología igual al 50% ya influye tanto el número de población como el número de generaciones, necesitando un valor alto en ambos parámetros.

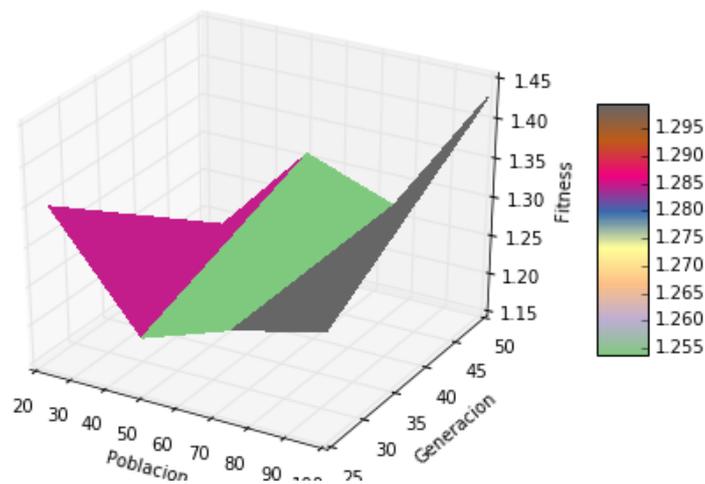


Ilustración 25: Evolución del valor de optimización según generación y población para una ciudad de 8 barrios y un FE 50%

5.3.5 Análisis Resultado con FE=70%

Como podemos ver en el informe generado, la mejor solución encontrada ha sido una solución formada por 2 líneas, $[[1, 3, 4, 0, 2, 5, 6], [1, 7, 0, 5, 3]]$, obtenida en una simulación de 50 generaciones y 50 población.

Al haber indicado ahora un factor de ecología del 70%, la solución encontrada tiene el número de líneas mínimo, 2, el mismo número que en el apartado anterior. En el apartado anterior ya se disponía de una solución que tenía el menor número de líneas posible. Por lo que el número de líneas a partir de ahora con el incremento del factor ecología siempre habrá el mínimo número de líneas.

La razón por la cual con un factor de ecología igual a 50% se haya llegado al menor número posible de líneas, se debe a que la ciudad que estamos analizando dispone de tan solo 8 barrios, que es un número relativamente bajo. Si se estudiara una ciudad con mayor número de barrios se tardaría más en obtener una solución con el menor número posible en relación al aumento del factor de ecología.

A continuación la imagen de representación de las líneas de la solución:

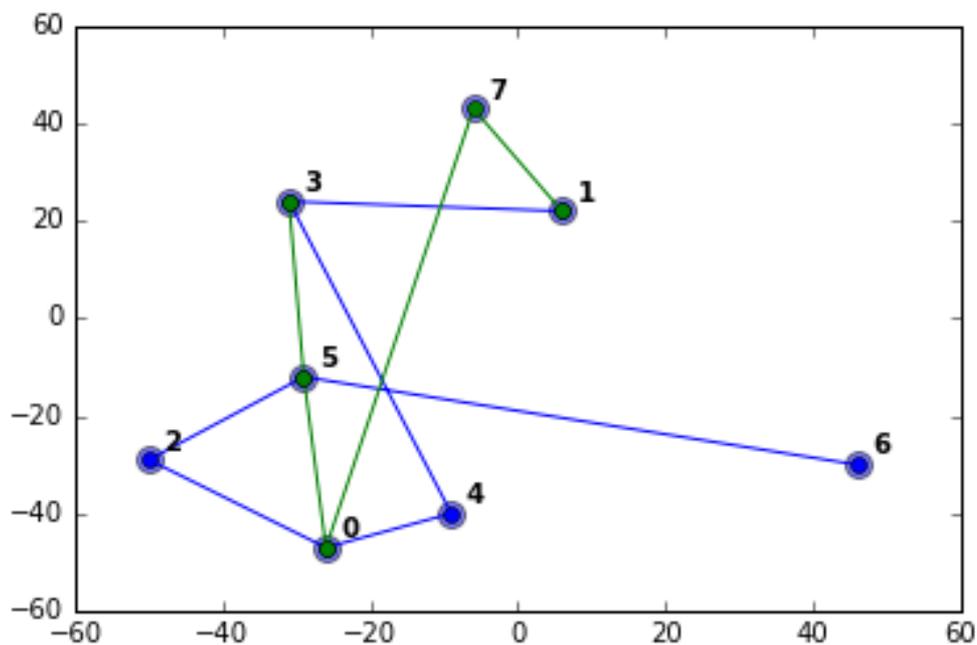


Ilustración 26: Solución más óptima encontrada para una ciudad de 8 barrios con FE 70%

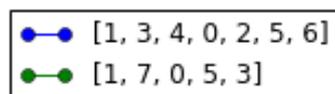


Ilustración 27: Líneas de la solución más óptima encontrada para una ciudad de 8 barrios y FE 70%

A continuación se muestra la matriz de distancia de la ciudad, la matriz de distancia de mejor solución encontrada y la matriz de demanda de la ciudad.

Matriz de Distancia de la ciudad

-	0	1	2	3	4	5	6	7
0	0	101	42	76	24	38	89	110
1	101	0	107	39	77	69	92	33
2	42	107	0	72	52	38	97	116
3	76	39	72	0	86	38	131	44
4	24	77	52	86	0	48	65	86
5	38	69	38	38	48	0	93	78
6	89	92	97	131	65	93	0	125
7	110	33	116	44	86	78	125	0

Matriz de Distancia de la Solución

-	0	1	2	3	4	5	6	7
0	0	145	42	86	24	38	141	110
1	145	0	145	39	135	87	200	33
2	42	145	0	86	76	38	141	208
3	86	39	86	0	86	38	141	82
4	24	135	76	86	0	72	185	144
5	38	87	38	38	72	0	93	140
6	141	200	141	141	185	93	0	263
7	110	33	208	82	144	140	263	0

Matriz de Demanda de la Ciudad

-	0	1	2	3	4	5	6	7
0	0	9	6	10	4	54	21	51
1	28	0	16	44	4	41	4	24
2	26	51	0	50	14	56	16	1
3	39	14	12	0	25	40	52	8
4	19	55	60	42	0	37	32	37
5	6	10	59	36	60	0	18	10
6	35	37	25	5	32	9	0	3
7	43	4	27	17	6	42	9	0

Si se observa la matriz de distancia de la solución por filas se aprecia lo siguiente:

En la fila del barrio 0, se percibe que todas las rutas han aumentado la distancia a recorrer, menos las rutas con los barrios 2, 4, 5 y 7. En el caso de los barrios 2 y 4 no tienen una gran demanda pero son barrios cercanos al barrio 0. En el caso de los barrios 5 y 7 si tienen una gran demanda de pasajeros. La ruta que más ha aumentado es la del barrio 0 al 6, pasando de 89 a 141. En la distribución del mapa se observa que son barrios alejados y la demanda de pasajeros no es elevada(21).

En la fila del barrio 1, se observa que todas las rutas han aumentado la distancia a recorrer, menos las rutas con los barrios 3 y 7. En el caso de los barrios 3 y 7 son rutas con una gran demanda de pasajeros, como se aprecia en la matriz de demanda. La ruta que más ha aumentado del barrio 1 al 6, pasando de 92 a 200. En la distribución del mapa se observa que son barrios alejados y la demanda de pasajeros no es elevada(6).

En la fila del barrio 2, se aprecia que todas las rutas han aumentado la distancia a recorrer, menos las rutas con los barrios 0 y 5. En el caso del barrio 5 la demanda de esta ruta es alta justificando el enlace directo y en el caso del barrio 0 es un barrio muy cercano al barrio 2. La ruta que más ha aumentado del barrio 2 al 6, pasando de 116 a 208. En la distribución del mapa se observa que son barrios muy alejados y la demanda de pasajeros no es elevada(16).

En la fila del barrio 3, se percibe que todas las rutas han aumentado la distancia a recorrer, menos las rutas con los barrios 1, 4 y 5. En el caso del barrio 1 se observa que la demanda de pasajeros es baja(14) pero son barrios cercanos y la demanda del barrio 1 al 3 es elevada(44). En el caso del barrio 5 la demanda es elevada(40) además de ser barrios muy cercanos. La ruta que más ha aumentado es la del barrio 3 al 7, pasando de 44 a 82. En la

distribución del mapa se observa que son barrios aunque no estén lejanos la demanda de pasajeros no es elevada(21).

En la fila del barrio 4, se percibe que todas las rutas han aumentado la distancia a recorrer, menos las rutas con los barrios 0 y 3. En el caso del barrio 3 es una ruta con gran demanda(42). En el caso del barrio 0 la demanda es baja(19) pero son barrios que están muy cercanos. La ruta que más ha aumentado es la del barrio 4 al 6, pasando de 65 a 185, donde la demanda de pasajeros no es elevada(32).

En la fila del barrio 5, se percibe que todas las rutas han aumentado la distancia a recorrer, menos las rutas con los barrios 0, 2, 3 y 6. En el caso de los barrios 2 y 3 la demanda es elevada además de ser barrios cercanos y en el caso del barrios 0 es muy cercano al barrio 5. En el caso del barrio 6 no hay gran demanda de pasajeros pero es la única vía de enlace para llegar al barrio 6 y no dejarlo aislado del resto. La ruta que más ha aumentado es la del barrio 5 al 7, pasando de 78 a 140. En la distribución del mapa se observa que son barrios relativamente lejanos y la demanda de pasajeros no es elevada(10).

En la fila del barrio 6, se percibe que todas las rutas han aumentado la distancia a recorrer, menos las rutas con el barrio 5. El barrio 6 es el barrio que queda menos comunicado con los restantes ya que tan solo está conectado directamente con el barrio 1, sus pasajeros deben pasar por este barrio antes de poder desplazarse a cualquier otro.

En la fila del barrio 7, se aprecia que todas las rutas han aumentado la distancia a recorrer, menos las rutas con los barrios 0 y 1. En el caso del barrio 0 la ruta la ruta tiene una demanda elevada(43). En el caso del barrio 1 la demanda es baja pero los barrios están muy cerca. La ruta que más ha aumentado es la del barrio 7 al 6, pasando de 125 a 263. En la distribución del mapa se observa que son barrios relativamente lejanos y la demanda de pasajeros no es elevada(9).

A continuación, en la imagen, se aprecia la evolución del valor de la optimización a través del desarrollo de las generaciones:

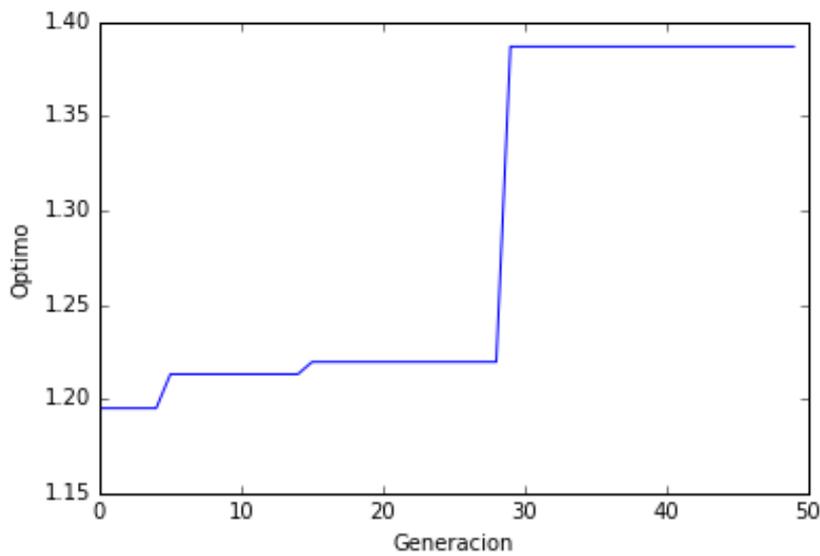


Ilustración 28: Evolución de la optimización para una ciudad de 8 barrios y FE 70%

En la imagen se aprecia que el valor de optimización crece muy lentamente hasta la generación 28, donde se produce un cambio radical, en el cual aumenta para mantenerse estable hasta la generación 50. Este cambio tan brusco puede ser debido a que tan solo disponemos de 2 líneas en la solución y una mutación individual o un Crossover provoca un cambio brusco en su valor máximo.

En la siguiente imagen se muestra la evolución del valor de optimización según el número de generaciones y poblaciones simuladas. Se observa que en este caso con un factor de ecología igual al 70% se necesita un número elevado de generaciones y no tanto de población.

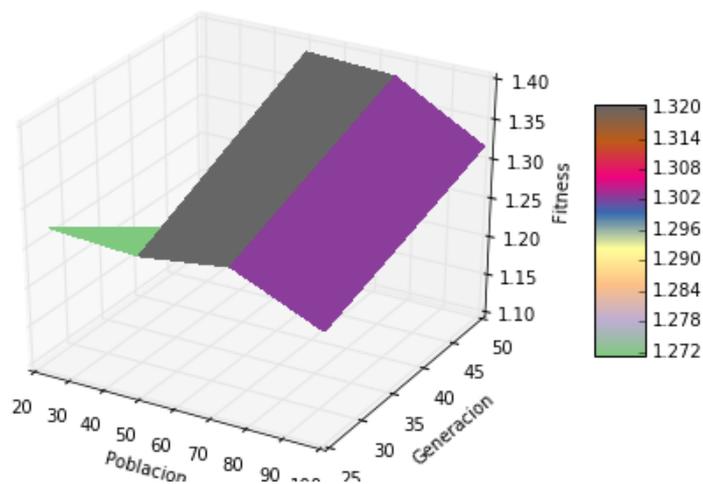


Ilustración 29: Evolución del valor de optimización según generación y población para una ciudad de 8 barrios y un FE 70%

5.3.6 Análisis Resultado con FE=100%

Como podemos ver en el informe generado, la mejor solución encontrada ha sido una solución formada por 2 líneas, $[[4, 6, 2, 5, 3, 7, 0], [0, 5, 4, 1, 3]]$, obtenida en una simulación de 50 generaciones y 75 de población.

Lógicamente y al igual que pasaba en el apartado anterior el número de líneas se mantiene en el mínimo número posible.

A continuación la imagen de representación de las líneas de la solución:

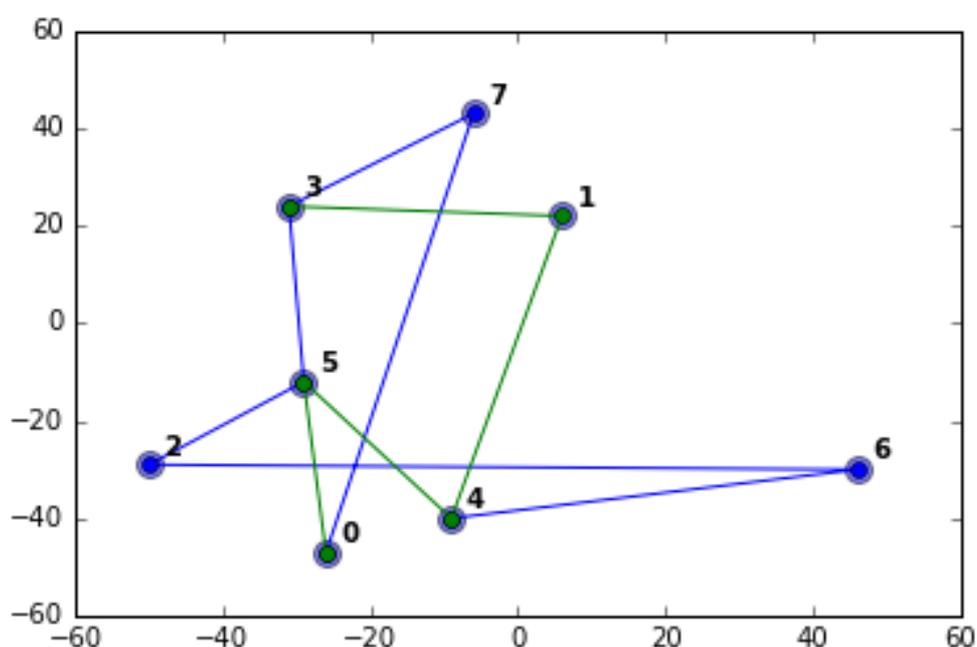


Ilustración 30: Solución más óptima encontrada para una ciudad de 8 barrios con FE 100%

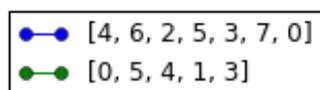


Ilustración 31: Líneas de la solución más óptima encontrada para una ciudad de 8 barrios y FE 100%

A continuación se muestra la matriz de distancia de la ciudad, la matriz de distancia de mejor solución encontrada y la matriz de demanda de la ciudad.

Matriz de Distancia de la ciudad

-	0	1	2	3	4	5	6	7
0	0	101	42	76	24	38	89	110
1	101	0	107	39	77	69	92	33
2	42	107	0	72	52	38	97	116
3	76	39	72	0	86	38	131	44
4	24	77	52	86	0	48	65	86
5	38	69	38	38	48	0	93	78
6	89	92	97	131	65	93	0	125
7	110	33	116	44	86	78	125	0

Matriz de Distancia de la Solución

-	0	1	2	3	4	5	6	7
0	0	145	86	86	96	38	181	110
1	145	0	145	39	77	87	152	93
2	86	145	0	86	96	38	97	150
3	86	39	86	0	96	38	181	44
4	96	77	96	96	0	48	65	160
5	38	87	38	38	48	0	123	92
6	181	152	97	181	65	123	0	255
7	110	93	150	44	160	92	255	0

Matriz de Demanda de la Ciudad

-	0	1	2	3	4	5	6	7
0	0	9	6	10	4	54	21	51
1	28	0	16	44	4	41	4	24
2	26	51	0	50	14	56	16	1
3	39	14	12	0	25	40	52	8
4	19	55	60	42	0	37	32	37
5	6	10	59	36	60	0	18	10
6	35	37	25	5	32	9	0	3
7	43	4	27	17	6	42	9	0

Si se observa la matriz de distancia de la solución por filas se aprecia lo siguiente:

En la fila del barrio 0, se percibe que todas las rutas han aumentado la distancia a recorrer, menos las rutas con los barrios 5 y 7. En el caso de los barrio 5 y 7 tiene una gran demanda de pasajeros. La ruta que más ha aumentado es la del barrio 0 al 6, pasando de 89 a 181. En la distribución del mapa se observa que son barrios alejados y la demanda de pasajeros no es elevada(21).

En la fila del barrio 1, se aprecia que todas las rutas han aumentado la distancia a recorrer, menos las rutas con los barrios 3 y 4. En el caso del barrio 3 es una ruta con una gran demanda de pasajeros, como se observa en la matriz de demanda. En el caso del barrio 4 la gran demanda es de la ruta inversa del barrio 4 al 1(55). La ruta que más ha aumentado es la del barrio 1 al 6, pasando de 92 a 152. En la distribución del mapa se observa que son barrios alejados y la demanda de pasajeros no es elevada(6).

En la fila del barrio 2, se percibe que todas las rutas han aumentado la distancia a recorrer, menos las rutas con los barrios 5 y 6. En el caso del barrio 5 la demanda de esta ruta es alta justificando el enlace directo y en el caso del barrio 6 no hay gran demanda de pasajeros pero es la única vía de enlace para llegar al barrio 6 y no dejarlo aislado del resto. La ruta que más ha aumentado es la del barrio 2 al 1, pasando de 107 a 145. En la distribución del mapa se observa que son barrios alejados.

En la fila del barrio 3, se percibe que todas las rutas han aumentado la distancia a recorrer, menos las rutas con los barrios 1, 5 y 7. En el caso del barrio 1 se observa que la demanda de pasajeros es baja(14) pero son barrios cercanos y la demanda del barrio 1 al 3 es elevada(44). En el caso del barrio 5 la demanda es elevada(40) además de ser barrios muy cercanos. En el caso del barrio 7 son barrios muy cercanos. La ruta que más ha aumentado es la del barrio 3 al 6, pasando de 131 a 181. En la distribución del mapa se observa que son barrios relativamente lejanos.

En la fila del barrio 4, se percibe que todas las rutas han aumentado la distancia a recorrer, menos las rutas con los barrios 1, 5 y 6. En todos estos barrios la demanda de pasajeros es elevada. La ruta que más ha aumentado es la del barrio 4 al 7, pasando de 86 a 160.

En la fila del barrio 5, se percibe que todas las rutas han aumentado la distancia a recorrer, menos las rutas con los barrios 0, 2, 3 y 4. En el caso de los barrios 2, 3 y 4 la demanda es elevada además de ser barrios cercanos y en el caso del barrios 0 es muy cercano al barrio 5. La ruta que más ha aumentado es la del barrio 5 al 6, pasando de 93 a 123. En la distribución del mapa se observa que son barrios relativamente lejanos y la demanda de pasajeros no es elevada(18).

En la fila del barrio 6, se percibe que todas las rutas han aumentado la distancia a recorrer, menos las rutas con los barrios 2 y 4. El barrio 6 está bastante alejado del resto de barrios, pero las rutas con los barrios 2 y 4 son demandadas. La ruta que más ha aumentado es la del barrio 6 al 7, pasando de 125 a 255. En la distribución del mapa se observa que son barrios relativamente lejanos y la demanda de pasajeros no es elevada(3).

En la fila del barrio 7, se percibe que todas las rutas han aumentado la distancia a recorrer, menos las rutas con los barrios 0 y 3. En el caso del barrio 0 la ruta la ruta tiene una demanda elevada(43). En el caso del barrio 3 la demanda es baja pero los barrios están muy cerca. La ruta que más ha aumentado es la del barrio 7 al 6, pasando de 125 a 255. En la distribución del mapa se observa que son barrios relativamente lejanos y la demanda de pasajeros no es elevada(9).

A continuación, en la imagen, se aprecia la evolución del valor de la optimización a través del desarrollo de las generaciones:

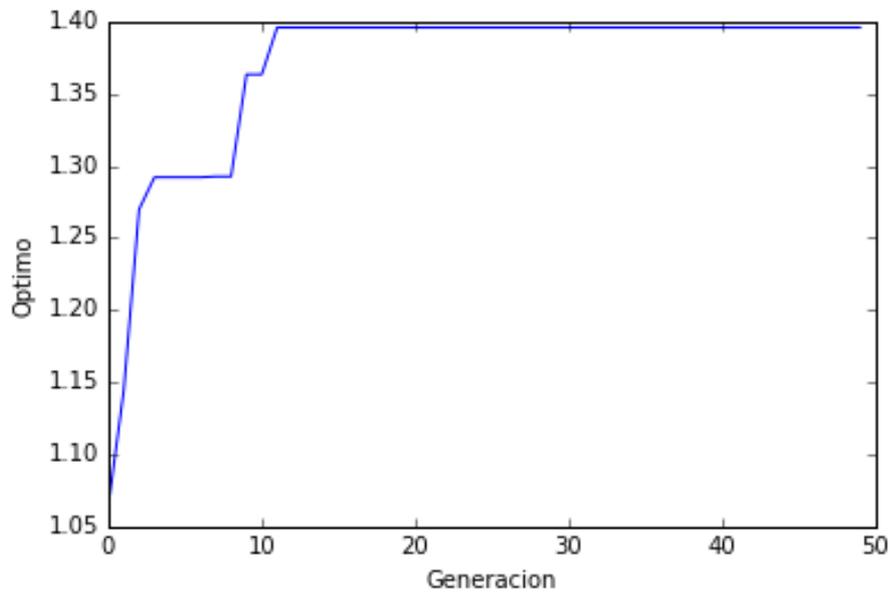


Ilustración 32: Evolución de la optimización para una ciudad de 8 barrios y FE 100%

En la imagen se aprecia que el valor de optimización se satura muy pronto, obteniendo el valor máximo en la generación 11.

En la siguiente imagen se muestra la evolución del valor de optimización según el número de generaciones y poblaciones simuladas. Se observa que en este caso con un factor de ecología igual al 100% se necesita un valor alto de generaciones y población para obtener los mejores resultados.

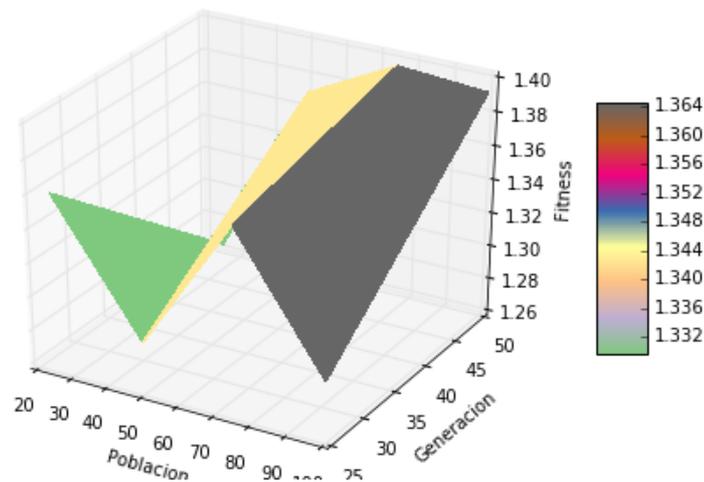


Ilustración 33: Evolución del valor de optimización según generacion y población para una ciudad de 8 barrios y un FE 100%

6. Conclusiones

6.1 Objetivos conseguidos

Tras la realización del análisis de los resultados, se puede afirmar que este proyecto ha logrado cumplir con los objetivos definidos en el apartado 1.2.

A través de la implementación de un algoritmo genético se ha logrado obtener un informe que contiene la mejor configuración de las líneas de transporte.

6.2 Observaciones sobre la planificación

Al comienzo de la realización de este proyecto se realizó una planificación del mismo para poder cumplir con los plazos establecidos.

Aunque la experiencia previa con otros proyectos indica que ninguna planificación inicial es real e intocable, quizá no se tuvo en cuenta los cambios y errores que podrían aparecer en etapas finales por lo que la planificación inicial se ha visto alterada.

6.3 Observaciones sobre los algoritmos genéticos

Tras la realización de este proyecto y el posterior análisis de los resultados obtenidos se han obtenido las siguientes conclusiones respecto a los algoritmos genéticos:

- Los algoritmos genéticos son de gran utilidad y eficientes ante problemas de optimización en entornos reales.
- Uno de los puntos críticos a la hora de implementar un algoritmo genético es definir la función a evaluar. En este caso de estudio ha sido difícil definir la eficiencia de una línea de transporte y conseguir un equilibrio entre el número de líneas y el número de barrios.
- Realizar simulaciones con valores de mutación individual y CrossOver elevados no obtiene mejores resultados. Al contrario las poblaciones degeneran aleatoriamente y las generaciones posteriores no garantizan mejores soluciones.
- En este proyecto, el factor de ecología creado influye en el valor máximo de optimización, cuanto mayor es el factor menor optimización se consigue.
- Un factor de ecología elevado requiere de un mayor número de generaciones y poblaciones para obtener resultados más óptimos

6.4 Observaciones personales

En general estoy satisfecha con el trabajo realizado y con los resultados obtenidos. La realización de este proyecto me ha permitido ampliar mis conocimientos sobre algoritmos genéticos y realizar un extenso trabajo de documentación. Además me ha permitido ampliar mis conocimientos en lenguaje Python.

Otra lección aprendida en el área de planificación de un proyecto. Aprender a tener en cuenta que ninguna planificación inicial es ideal y que siempre hay que tener espacio para imprevistos de última hora.

7. Glosario

Algoritmo: Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.

Algoritmo de Dijkstra's : también llamado algoritmo de caminos mínimos, es un algoritmo para la determinación del camino más corto dado un vértice origen al resto de los vértices en un grafo con pesos en cada arista. Su nombre se refiere a Edsger Dijkstra, quien lo describió por primera vez en 1959.

Algoritmos Evolutivos: métodos de optimización y búsqueda de soluciones basados en los postulados de la evolución biológica. En ellos se mantiene un conjunto de entidades que representan posibles soluciones, las cuales se mezclan, y compiten entre sí, de tal manera que las más aptas son capaces de prevalecer a lo largo del tiempo, evolucionando hacia mejores soluciones cada vez.

Algoritmo Genético: se inspiran en la evolución biológica y su base genético-molecular. Estos algoritmos hacen evolucionar una población de individuos sometiéndola a acciones aleatorias semejantes a las que actúan en la evolución biológica (mutaciones y recombinaciones genéticas), así como también a una selección de acuerdo con algún criterio, en función del cual se decide cuáles son los individuos más adaptados, que sobreviven, y cuáles los menos aptos, que son descartados.

CrossOver: intercambio de material genético entre dos individuos.

Diagrama de Flujo : representación gráfica del algoritmo o proceso.

Generación: Sucesión de descendientes en línea recta.

Genético: Parte de la biología que trata de la herencia y de lo relacionado con ella.

Grafo: Representación simbólica de los elementos constituidos de un sistema o conjunto, mediante esquemas gráficos.

Línea: Servicio regular de vehículos que recorren un itinerario determinado.

Mutación: Alteración en los atributos de un organismo, que se transmite por herencia.

Optimización: método para determinar los valores de las variables que intervienen en un proceso o sistema para que el resultado sea el mejor posible.

Población: Conjunto de individuos de la misma especie que ocupan determinada área geográfica.

Pseudocódigo: es una descripción de alto nivel compacta e informal del principio operativo de un programa informático u otro algoritmo.

Python: es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.

Ruta: Itinerario para el viaje para desplazarse de un punto a otro.

8. Bibliografía

- C. MANDL, *Applied Network Optimization*, Academic Press, London, 1979
- Baaj and Mahmassani, *Hybrid Route Generation Heuristic Algorithm for The Design of Transit Networks*, (1995)

Páginas Web – Consultadas entre Septiembre y Diciembre 2016

- <https://es.wikipedia.org/wiki/NumPy>
- <http://matthiaseisen.com/articles/graphviz/>
- https://es.wikipedia.org/wiki/Problema_del_viajante
- <http://dle.rae.es>
- <https://www.techopedia.com>
- <http://www.numpy.org/>
- <https://networkx.github.io/>
- <http://deap.gel.ulaval.ca/doc/default/>
- <https://docs.python.org/2/library/collections.html>
- <https://docs.python.org/3/library/random.html>
- <http://matplotlib.org/>
- <http://www.pythonware.com/products/pil>
- <http://www.reportlab.com/>
- https://es.wikipedia.org/wiki/Algoritmo_divide_y_vencer%C3%A1s
- https://es.wikipedia.org/wiki/Algoritmo_de_Dijkstra
- http://es.wikipedia.org/wiki/Algoritmo_genético
- https://es.wikipedia.org/wiki/Algoritmo_evolutivo
- <https://es.wikipedia.org/wiki/Pseudoc%C3%B3digo>
- https://es.wikipedia.org/wiki/Diagrama_de_flujo
- <https://es.wikipedia.org/wiki/Grafo>
- <https://es.wikipedia.org/wiki/Python>

9. Anexos

- Informes generados en PDF para una ciudad de 8 barrios con factor de mutación Individual = 0,3 y CrossOver=0,5 utilizados para el análisis del apartado 5.4

Se encuentran en la carpeta Resultados/NB=8

NB=8_mutpb=0,3cxpb=0,5_Informe_0.pdf
NB=8_mutpb=0,3cxpb=0,5_Informe_30.pdf
NB=8_mutpb=0,3cxpb=0,5_Informe_50.pdf
NB=8_mutpb=0,3cxpb=0,5_Informe_70.pdf
NB=8_mutpb=0,3cxpb=0,5_Informe_100.pdf

- Imágenes generadas en la simulación para una ciudad de 8 barrios con factor de mutación Individual = 0,3 y CrossOver=0,5 utilizadas para el análisis del apartado 5.4

Se encuentran en la carpeta Resultados/NB=8

- Informes generados tras la realización de diferentes pruebas con distinto número de barrios y factor de ecología.

Se encuentran en la carpeta Resultados/NB='número de barrios'.