

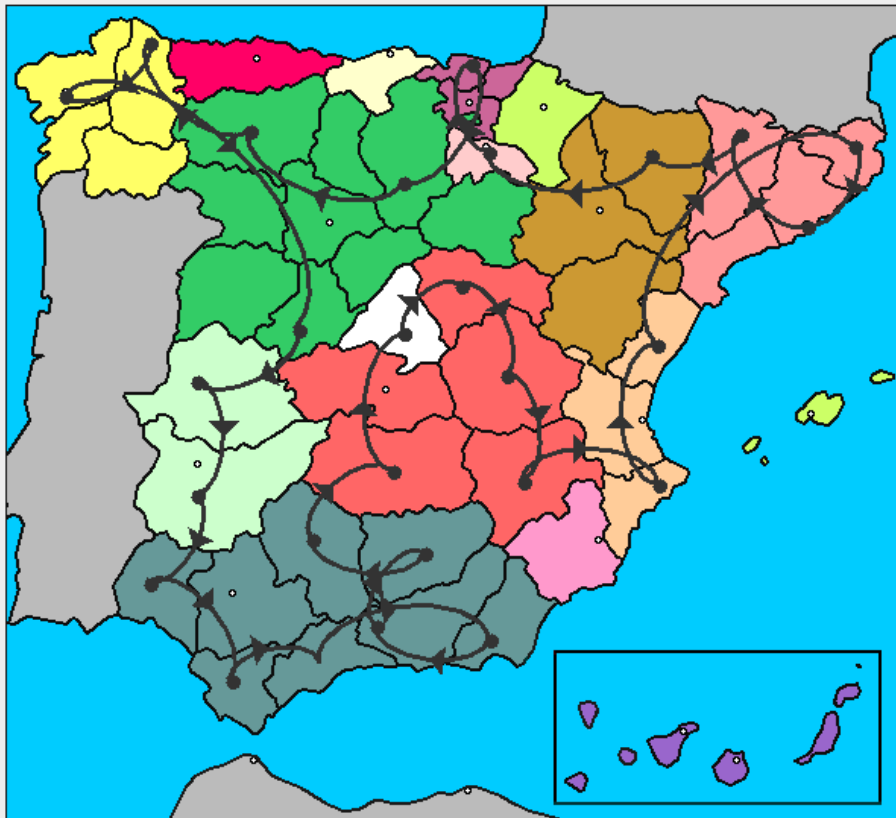
Programación Evolutiva

Practica 2

Ignacio Sánchez Santatecla

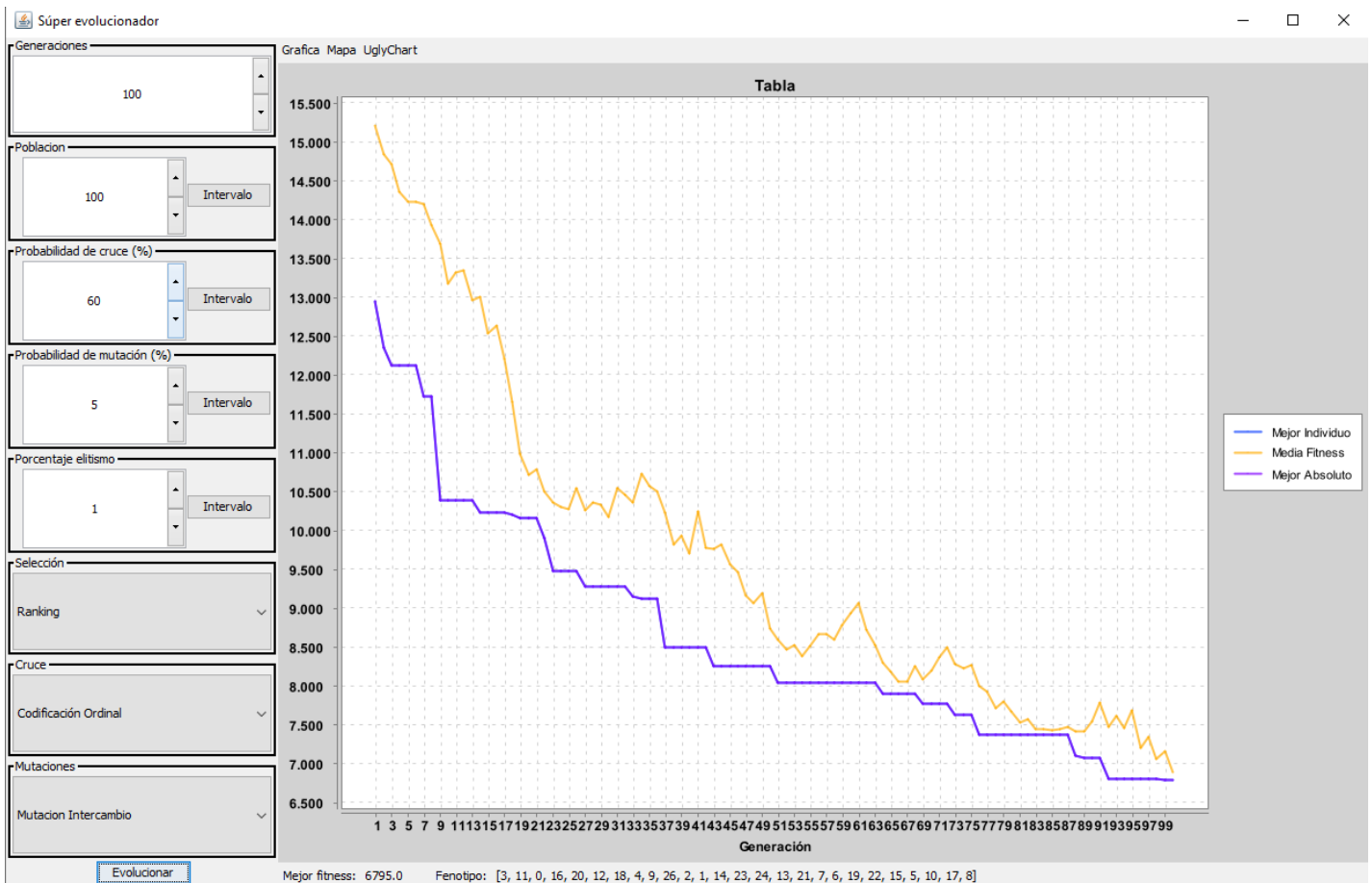
David Montoro Couso

Grupo 7



MADRID -> GUADALAJARA -> CUENCA -> ALBACETE -> ALICANTE -> CASTELLON -> GIRONA ->
BARCELONA -> LLEIDA -> HUESCA -> LOGRONO -> BILBAO -> BURGOS -> LEON -> LUGO ->
CORUÑA -> AVILA -> CACERES -> BADAJOZ -> HUELVA -> CADIZ -> MALAGA -> ALMERIA ->
GRANADA -> JAEN -> CORDOBA -> CIUDAD_REAL -> MADRID

Gráficas:

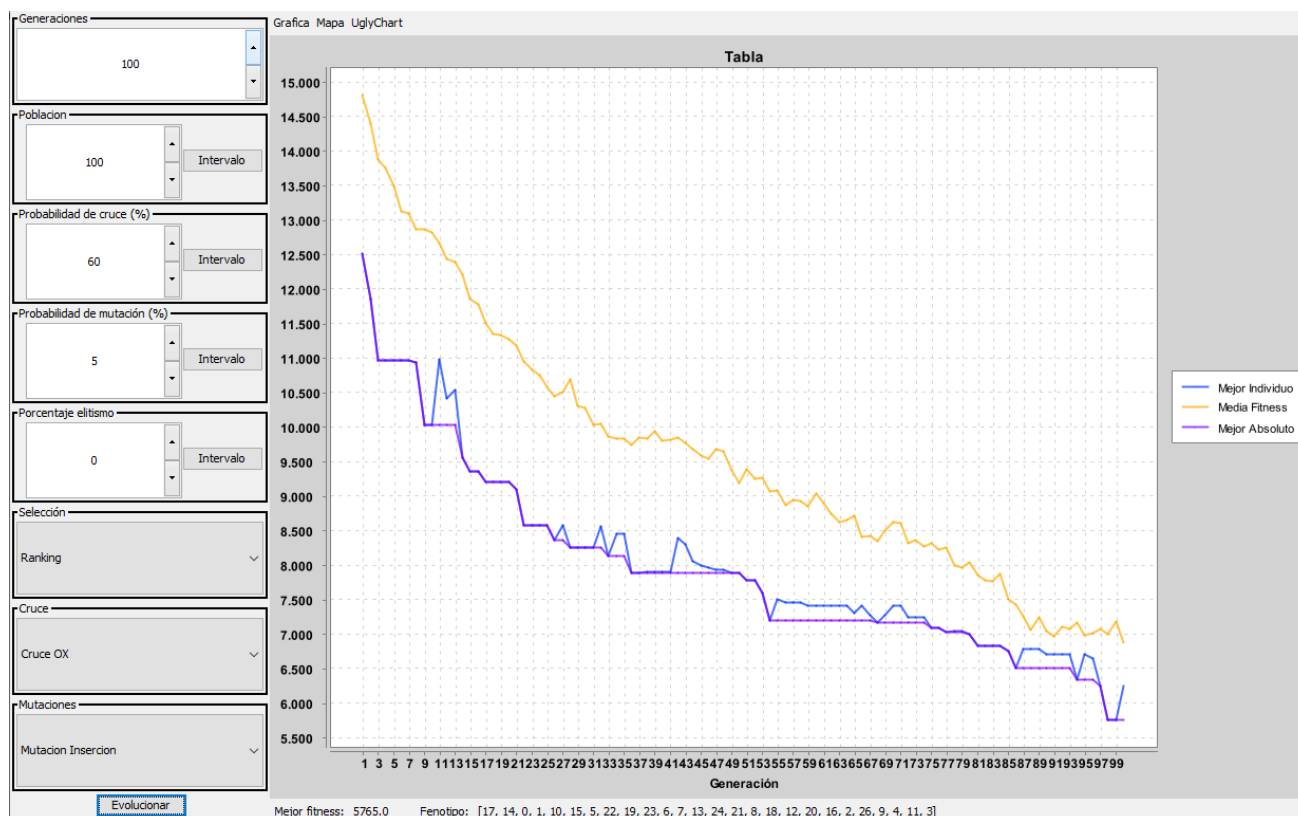


Parámetros estáticos

Generaciones:	100
Individuos:	100
Cruce:	Codificación Ordinal (60%)
Mutación:	Intercambio (5%)
Elitismo:	1%

En este problema, si intentamos asemejar los parámetros a los que solíamos usar en la práctica anterior, no conseguimos acercarnos a resultados decentes.

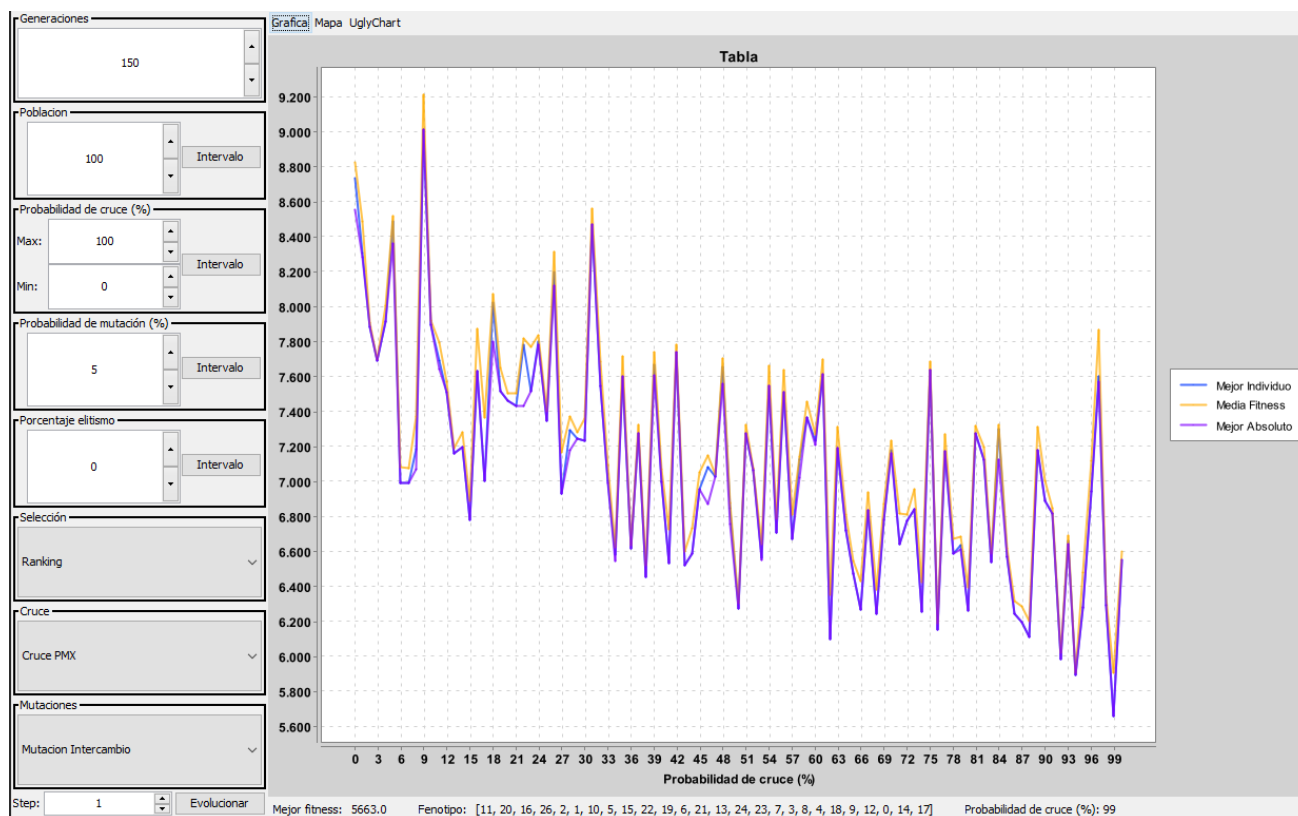
Hemos notado que la selección por ranking da los mejores resultados con la mayoría de cruces y mutaciones nuevas. A sí que será la que utilicemos para la mayoría de pruebas.



Parámetros estáticos

Generaciones:	100
Individuos:	100
Cruce:	Orden (OX) (60%)
Mutación:	Inserción (5%)
Elitismo:	0%

Al tratarse del problema del viajante, donde la codificación es el orden de las ciudades en el que se recorren, el cruce por orden obtiene mejores resultados, gracias a que mantiene el orden relativo entre genes para los padres. De esta manera los cruces no son tan destructivos y, a pesar de no tener nada de elitismo, se obtienen resultados mucho más cercanos al óptimo.



Parámetros dinámicos

Generaciones: 150

Individuos: 100

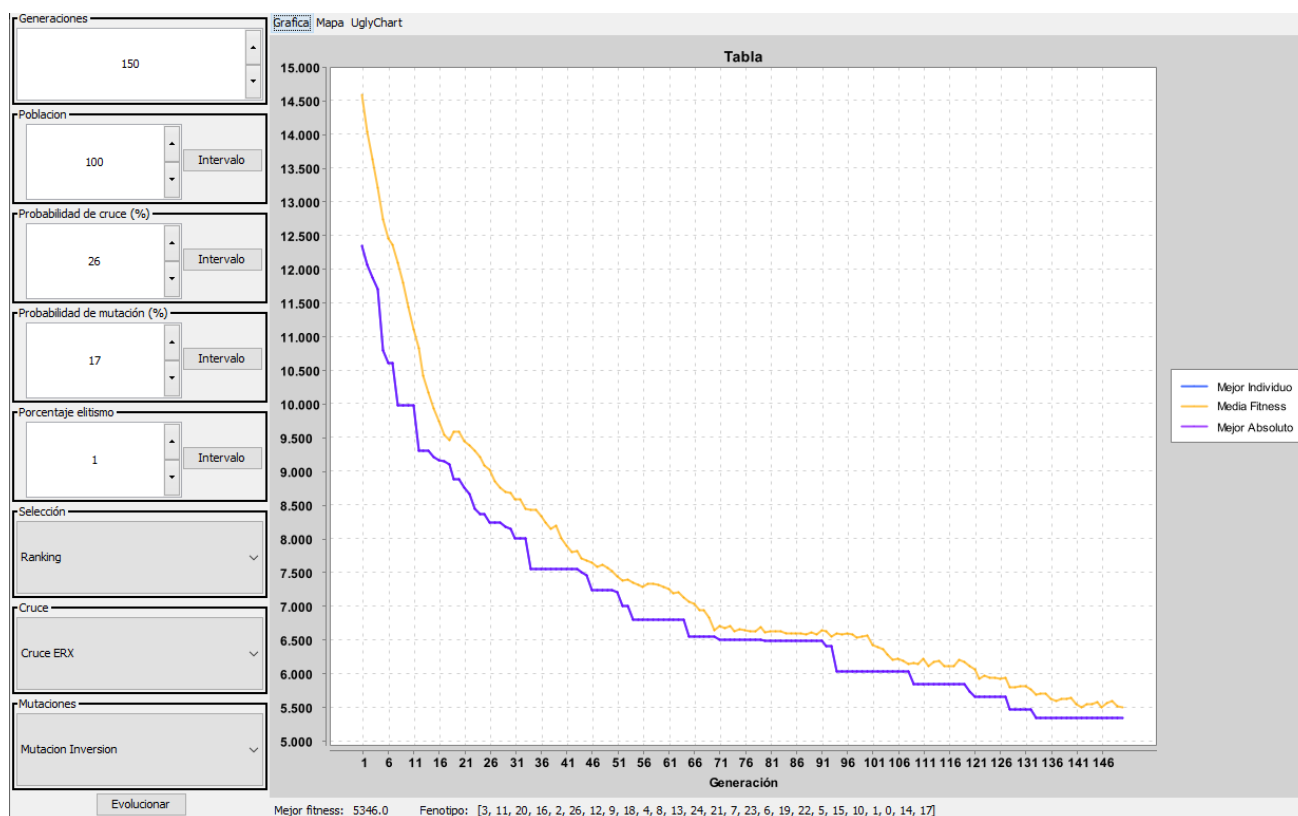
Cruce: Emparejamiento parcial (PMX) (de 0% hasta 100%)

Mutación: Intercambio (5%)

Elitismo: 0%

El cruce por emparejamiento parcial es uno de los que mejores resultados nos ha dado junto con el de recombinación de rutas. En esta gráfica puede verse cómo a medida que aumenta el porcentaje de cruce, el resultado final de cada ejecución disminuye, confirmando la eficacia de este cruce.

Una posible causa de que este cruce mejora el cruce por orden puede ser que las ciudades fuera del punto de cruce no mantienen tanto el orden parcial como en el de orden, lo que genera más variedad y un menor estancamiento.



Parámetros estáticos

Generaciones: 150

Individuos: 100

Cruce: Emparejamiento parcial (PMX) (26%)

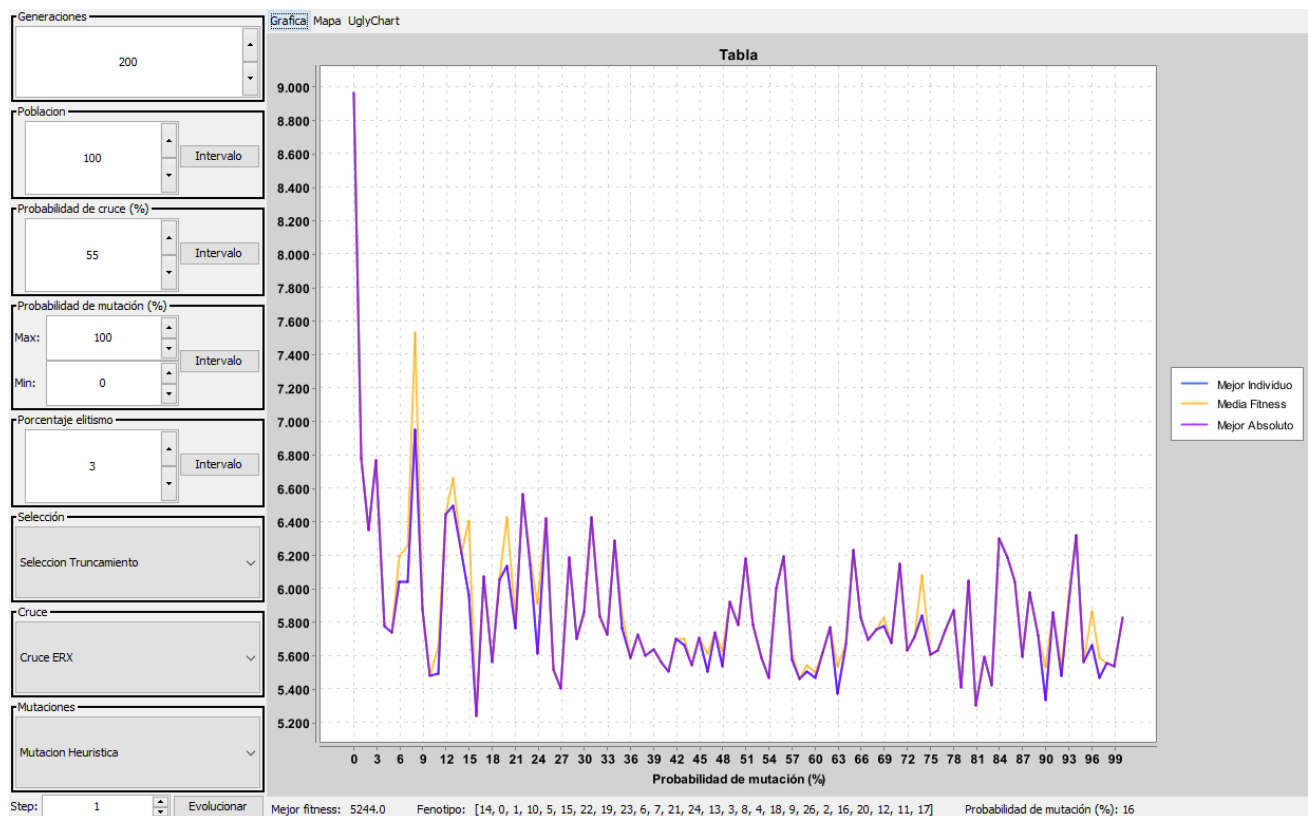
Mutación: Inversión (17%)

Elitismo: 1%

El cruce por emparejamiento parcial ha sido el que mejor resultados nos ha dado cuando lo aplicábamos con la mutación por inversión.

La conclusión a la que hemos llegado ha sido que como en el cruce estamos eligiendo siempre la ciudad menos conectada a la anterior, estamos creando individuos más variados, dentro de la coherencia del problema ya que no estamos creando conexiones entre ciudades completamente nuevas.

Cuando lo combinamos con la mutación por inversión, en la que solo estamos alterando las conexiones entre las ciudades separadas por los puntos de corte elegidos y el resto se mantienen sus caminos originales, ya sea con el mismo orden o el inverso, se consigue una evolución que progresa de una manera muy fructífera y continua.



Parámetros dinámicos

Generaciones: 200

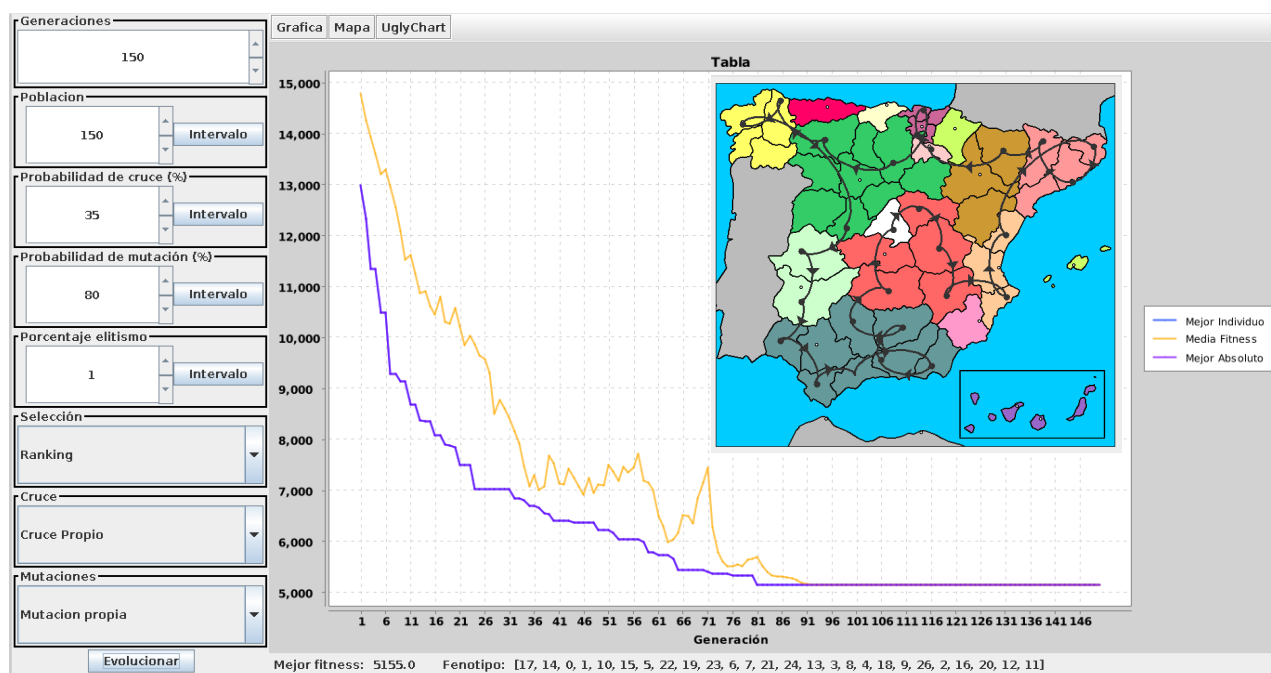
Individuos: 100

Cruce: Recombinación de rutas (ERX) (55%)

Mutación: Heurística (de 0 hasta 17%)

Elitismo: 3%

En esta gráfica se aprecia muy bien el comportamiento de la mutación heurística, muy diferente a las demás. El resto de mutaciones tienen un comportamiento parecido, cuando el porcentaje aumenta por encima del 50% la media comienza a crecer y el ruido introducido empieza a ser insostenible para la evolución. En el caso de la mutación Heurística, al mutar el individuo solo en el caso de que este mejore, podemos poner valores muy altos de mutación que no nos vamos a ver afectados por el ruido ocasionado. Sin embargo, esto fomenta un comportamiento elitista que acelera la convergencia y, en algunos casos, no es capaz de llegar al mejor resultado por falta de variedad en la población.



Parámetros estáticos

Generaciones: 150

Individuos: 150

Cruce: Propio (35%)

Mutación: Propia (80%)

Elitismo: 1%

Hemos implementado un cruce y una mutación inventados por nosotros.

El cruce sigue un esquema parecido al cruce uniforme, manteniendo un orden parcial entre progenitores. Para crear a un hijo se va seleccionando de manera alterna el gen de un padre y el de otro, y cuando uno de los seleccionados ya está en el genotipo del hijo, avanzamos a la siguiente posición y lo intentamos de nuevo con el otro padre. Si ocurre de nuevo se repite el proceso hasta completar el cromosoma.

La mutación es una combinación de la mutación por inversión y la mutación Heurística, ya que han sido las que mejores resultados han dado. El procedimiento consiste en realizar la inversión de genes al individuo un número determinado de veces (entre 3 y 7) para después elegir el mejor de esos resultados y aplicárselo al individuo. Al ser una mutación con naturaleza parecida a la heurística, también tiene la desventaja de una posible convergencia prematura al no dejar cabida a mutaciones que empeoren el fitness pero que a largo plazo puedan generar mejores resultados.

Reparto de Tareas

Para la realización de esta entrega, hemos continuado trabajando de manera conjunta y simultánea a la hora de implementar los nuevos cruces, mutaciones y selección aprendidas, además de las modificaciones estructurales necesarias para el nuevo problema a resolver.

Fuera de la teoría de la asignatura, hemos trabajado por separado para hacer cambios en lo que se refiere a la interfaz gráfica del programa. Nacho se ha encargado de reestructurar el panel de control de los parámetros y adaptar la ventana principal a la gráfica que ya teníamos. Y David ha añadido la gráfica recomendada en el campus virtual para poder hacer comparaciones más fácilmente, y ha añadido un mapa de España con las provincias representadas en el problema, que dibuja el teórico recorrido a seguir por nuestro viajante de comercio en su viaje por la península.

Detalles de la implementación

Hemos reestructurado toda la vista del programa, olvidándonos de la biblioteca anterior para crear formularios, ya que de esta manera teníamos más libertad para implementar la problemática de los intervalos.

Hemos creado un nuevo individuo que contiene un array de enteros para representar el orden en el que se recorren las ciudades y hemos añadido un .jar al zip para poder ejecutar y hacer pruebas más fácilmente sin importar el proyecto en ningún IDE.