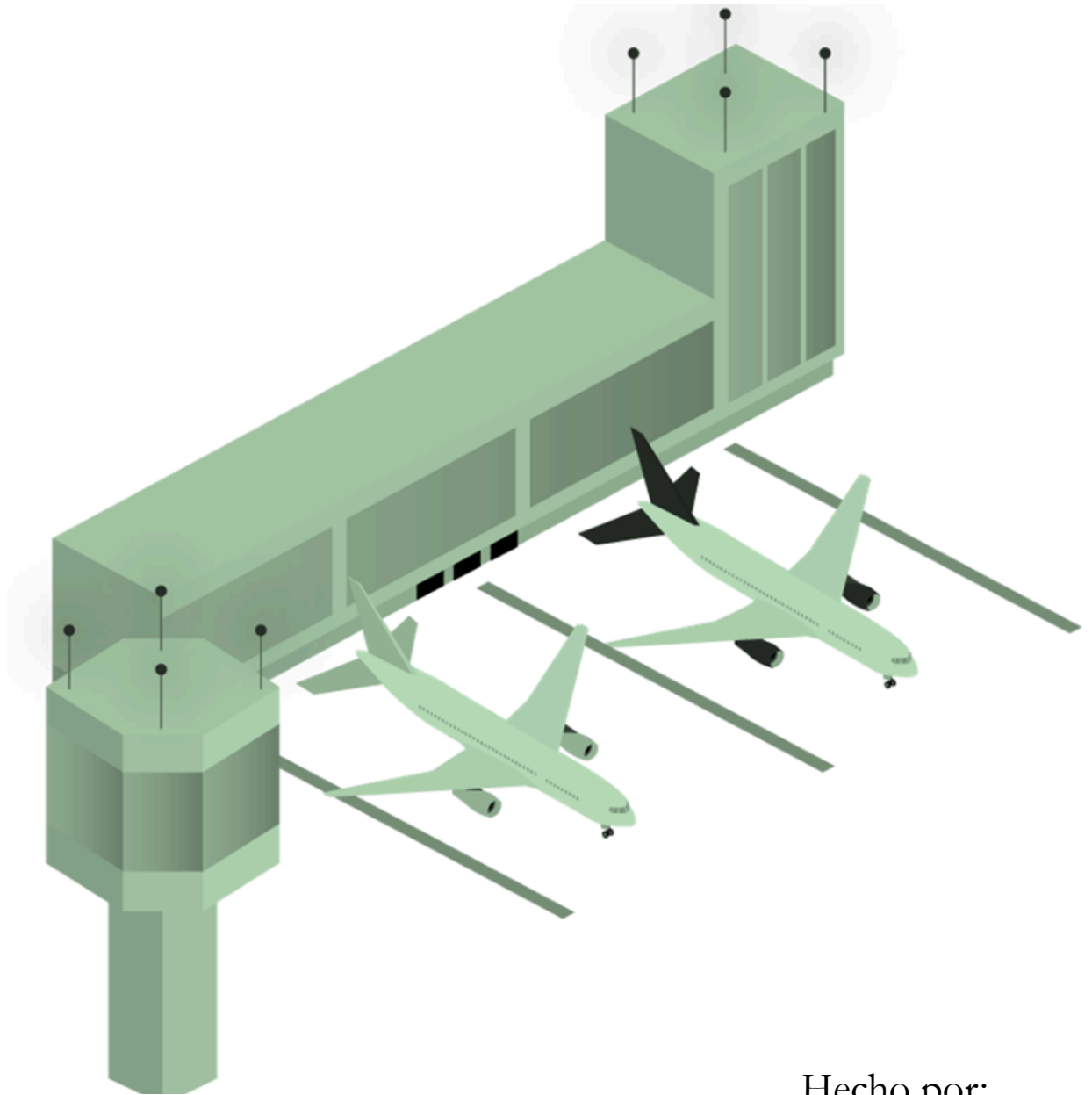


Práctica 2

Programación Evolutiva



Hecho por:

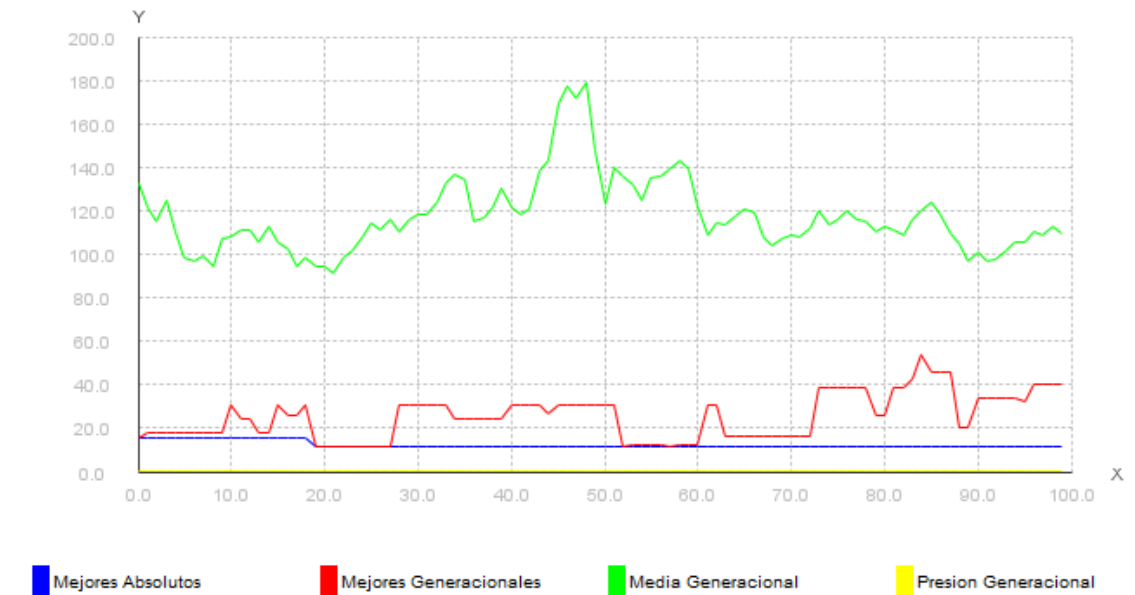
Daniel Couto Seller

Santiago González García

Gráfica representativa de cada caso

- Caso 1(vuelos1.txt y TEL1.txt):

Parámetros: Tamaño Generación (100), Número Generación (100), Porcentaje Cruce (50), Porcentaje Mutación (10), Selección(Torneo Probabilístico), Cruce (PMX), Mutación (Intercambio), Porcentaje Elitismo (0).



Solución:

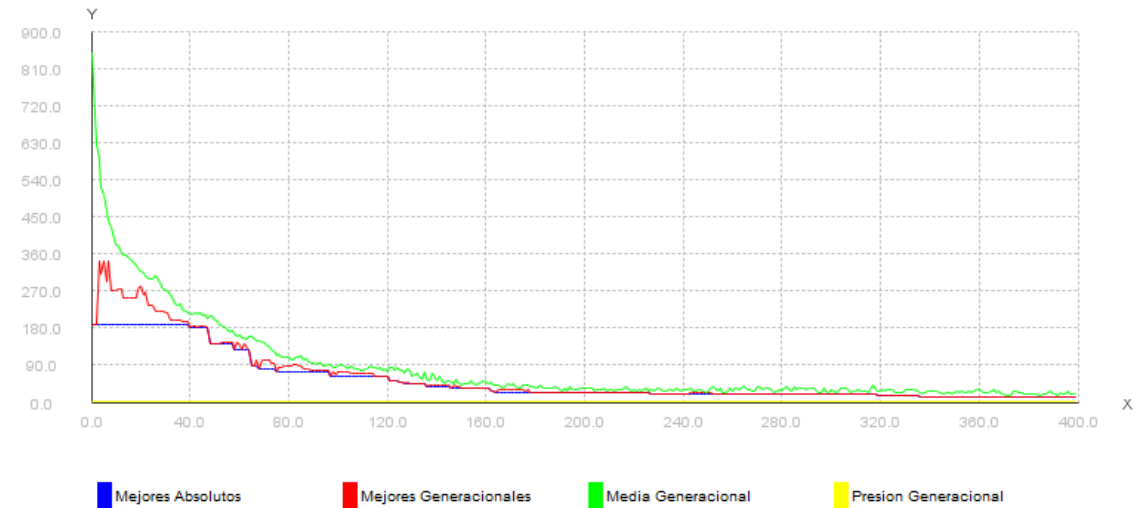
Valor optimo (11.25) con los siguientes vuelos [12 , 9 , 6 , 4 , 11 , 1 , 5 , 10 , 2 , 7 , 3 , 8]

VUELO	Pista1 NOMBRE	TLA	VUELO	Pista2 NOMBRE	TLA	VUELO	Pista3 NOMBRE	TLA
9	AA129	6.0	6	AA128	6.0	12	AA309	5.0
4	NW358	7.0	3	UA599	7.0	11	UA805	6.0
5	UA2897	9.0	8	NW357	8.0	1	UA138	9.0
10	UA2408	10.0						
2	UA532	15.0						
7	UA1482	16.5						

Durante la generacion: 19.0

- Función 2:

Parámetros: Tamaño Generacion (100), Número Generación (400), Porcentaje Cruce (50), Porcentaje Mutación (10), Selección (Ranking), Cruce (OX), Mutación (Inserción), Porcentaje Elitismo(0).



Solución:

VUELO	Pista1 NOMBRE	TLA	VUELO	Pista2 NOMBRE	TLA	VUELO	Pista3 NOMBRE	TLA	VUELO	Pista4 NOMBRE	TLA
5	UA2897	1.0	13	UA960	3.0	24	AA878	1.0	14	UA980	0.0
7	UA1482	3.0	8	NW357	4.0	6	AA128	2.0	15	UA1012	1.0
11	UA805	4.0	9	AA129	5.0	21	AA109	3.0	20	NW369	2.0
19	UA1474	5.5	22	UA2442	8.0	17	UA2897	5.0	18	AA1260	3.0
2	UA532	7.0	3	UA599	13.0				4	NW358	5.0
									1	UA138	6.0
									10	UA2408	8.0
									25	UA1098	15.0

Durante la generacion: 336.0

VUELO	Pista5 NOMBRE	TLA
12	AA309	1.0
23	UA895	3.0
16	NW384	4.0

Valor optimo (14.25) con los siguientes vuelos

[13 , 14 , 5 , 8 , 15 , 24 , 20 ,
7 , 6 , 12 , 23 , 18 , 4 , 9 , 1 ,
22 , 10 , 25 , 11 , 3 , 16 , 19 , 21 , 17 , 2]

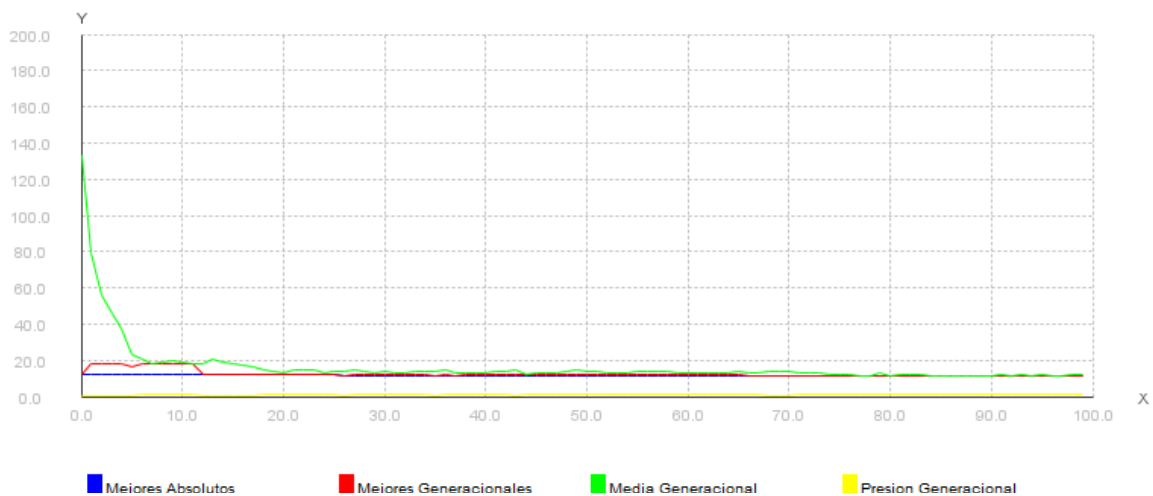
Durante la generacion: 336.0

Conclusiones de los resultados

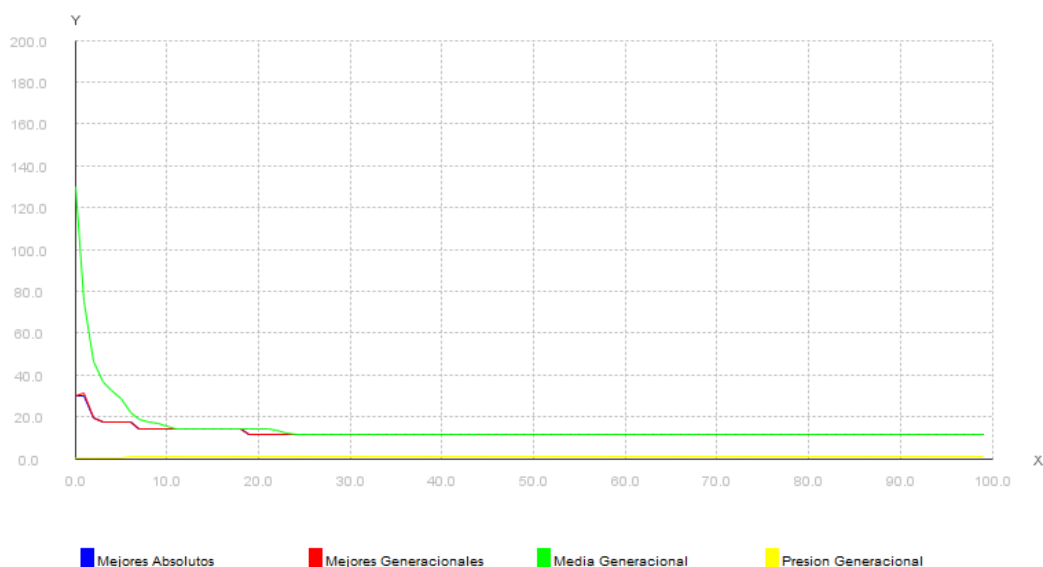
Tras muchas pruebas con todas las funciones hemos llegado a varias conclusiones.

Análisis de convergencia

Al observar las gráficas, podemos notar la diferencia entre el mejor individuo en cada generación y el mejor individuo de todas las generaciones hasta el momento. Esta diferencia disminuye gradualmente a lo largo de las generaciones, indicando que el algoritmo está convergiendo hacia una solución óptima como en el siguiente ejemplo:



La convergencia puede estar influenciada por diversos factores como pueden ser los métodos de selección, mutación y cruce. Muchas de las funciones de selección pueden ser muy elitistas como pudimos ver en la práctica 1. El elitismo favorecerá en gran medida a la convergencia de los mejores generacionales y los mejores absolutos. Por otro lado, tenemos que el método de mutación Heurística, al seleccionar el mejor entre varios individuos generados mediante la permutación de n posiciones, ayuda a que la búsqueda de una solución óptima sea más rápida de lo normal. A continuación se muestra una gráfica del caso 1 con este método.



Diferencias notables entre los casos

En esta práctica nos enfrentamos a 2 casos distintos. En el primer problema tenemos que asignar 12 vuelos a 3 pistas distintas, mientras que en el segundo problema los números incrementan a 25 vuelos y 5 pistas. Hemos podido observar una notable diferencia entre cuántas generaciones hacen falta para llegar a la solución óptima en ambos casos. En el primer problema en la gran mayoría de las pruebas con 100 generaciones llega a una solución óptima. Por otro lado, para el segundo problema al ser más compleja, se necesita un número de generaciones bastante mayor, hasta el punto de necesitar unas 400 generaciones.

Mejoras

En esta práctica hemos implementado una serie de mejoras al algoritmo genético para obtener mejores resultados y tener más información sobre las diferentes generaciones de los individuos.

Presión selectiva

En esta práctica hemos implementado la mejora de la presión selectiva. Para esto la hemos ido calculando generación tras generación y la hemos representado en la gráfica como una línea amarilla que sigue la siguiente fórmula.

$$PresSel[t] \equiv \frac{AptMax[t]}{AptMed[t]}$$

Desplazamiento de la aptitud

Igual que en la anterior práctica en esta también hemos aplicado el desplazamiento de la aptitud de los individuos. Donde la aptitud adaptada de cada individuo i en una generación será dado por el 105% de la máxima aptitud de la generación menos la aptitud real del individuo. De esta forma nos aseguramos que todas las aptitudes son positivas, transformando así el problema de minimización en uno de maximización, y no simplemente multiplicamos sus valores por -1.

Cruce OXOP

Aunque no se nos pedía implementarlo, hemos decidido implementar también el cruce OXOP. Este cruce, también conocido como cruce por orden con orden prioritario, consiste en seleccionar unas posiciones al azar, después miro en el padre 1 a que aviones corresponden esas posiciones y en qué posiciones están esos aviones en el segundo padre. Sucesivamente creó el primer hijo con los elementos del segundo padre menos las posiciones anteriores y los huecos se rellenan con las

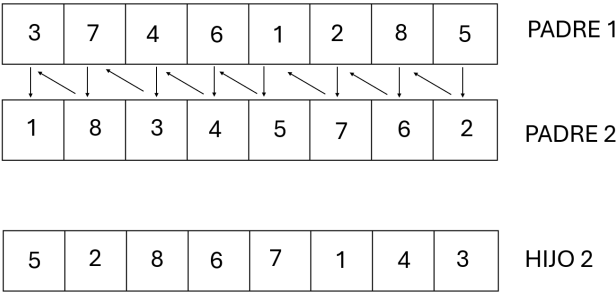
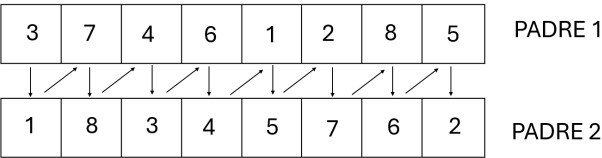
ciudades que había se habían seleccionado previamente. Esto se repite para el segundo hijo pero intercambiando los padres.

Métodos propios

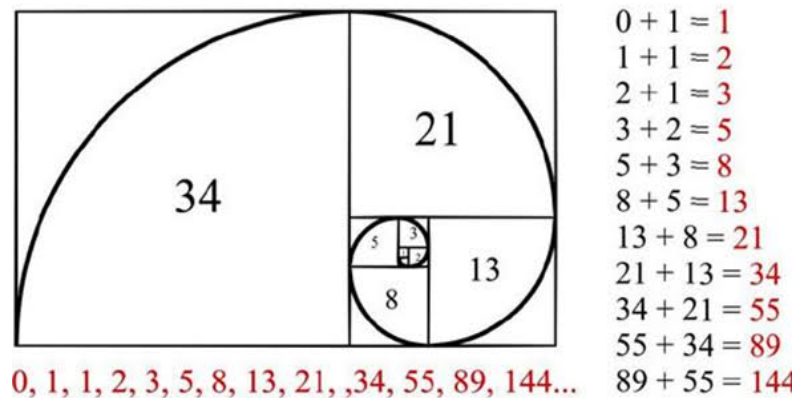
Hemos desarrollado un método propio de cruce y otro de mutación. Hemos intentado ser lo más originales posibles, y hemos acabado ideando el Cruce ZigZag y la Mutación Fibonacci

Cruce ZigZag

El cruce zigzag consiste en ir recorriendo los dos cromosomas a la vez escogiendo de cada uno los genes que no estén del hijo. Primero se ve la posición 0 del padre 1, luego la posición 0 del padre 2, posición 1 del padre 1, posición 1 del padre 2... Se añadirán al hijo los genes que no se hayan añadido aún. Para poder generar 2 hijos con este cruce lo que hacemos es recorrer los padres de izquierda a derecha para un hijo, y de derecha a izquierda para el otro hijo. De esta forma ambos descendientes serán distintos y no habrá elementos repetidos en ninguno.



Mutación Fibonacci



Como mutación propia hemos ideado la mutación Fibonacci. La mutación Fibonacci consiste en mutar al individuo sirviéndonos de los primeros números de la famosa sucesión de Fibonacci, y en grandes rasgos consiste en desplazar cada gen de un individuo siguiendo los números de la sucesión.

Una vez seleccionado el individuo que va a ser mutado se siguen los siguientes pasos:

1. Primero hay que determinar los elementos de la sucesión que vamos a usar, en nuestro caso hemos decidido usar 1,2,3,5.
2. Después hay que crear nuevas combinaciones del individuo, tantas como elementos vayamos a usar.
3. Posteriormente, cada combinación nueva se rellenará desplazando cada gen hasta llegar a su posición más el elemento de la sucesión que estamos considerando.
4. Una vez hayamos creado todas las variaciones del individuo, nos quedaremos con la variación que mejor adaptación tenga.

Aquí un ejemplo con el individuo [9,3,2,12,4,11,8,7,10,1,6,5]:

- Usando 1 \rightarrow [5 9 3 2 12 4 11 8 7 10 1 6].
- Usando 2 \rightarrow [6 5 9 3 2 12 4 11 8 7 10 1].
- Usando 3 \rightarrow [1 6 5 9 3 2 12 4 11 8 7 10].
- Usando 5 \rightarrow [7 10 1 6 5 9 3 2 12 4 11 8].

Reparto de tareas

-Santiago González García

- Desarrollo de la función de selección de ranking.
- Desarrollo del individuo
- Desarrollo de los cruces

- Desarrollo de la salida de la solución por pantalla
- Realización de la mutación propia Fibonacci
- Implementación de mejoras
- Corrección de errores
- Realización de la memoria

- Daniel Couto Seller

- Desarrollo de la función de selección de ranking.
- Desarrollo del individuo
- Desarrollo de las mutaciones
- Desarrollo de una nueva interfaz gráfica con WindowBuilder
- Realización del cruce propio ZigZag
- Corrección de errores
- Realización de la memoria