PRÁCTICA 2

programación evolutiva

Grupo 2
Javier Martín-Tesorero Ruiz
Clara Martín Navas

1. Introducción

El objetivo es desarrollar un algoritmo evolutivo para optimizar la ruta de un robot de limpieza en una mansión modelada como una cuadrícula 15×15. El robot debe visitar 20 habitaciones partiendo y regresando a la base (7,7), evitando obstáculos y minimizando el tiempo de recorrido. La solución se representa como una permutación de números del 1 al 20, que indica el orden de visita. Se emplean técnicas de precálculo de rutas mediante A* con diversas penalizaciones (por proximidad a obstáculos, giros, cambios de dirección, tiempo) y se hace uso de caches para evitar cálculos redundantes.

2. Representación de la solución

- **Cromosoma:** Permutación de los números 1 a 20.
- **Punto de partida y retorno:** Base fija en (7,7).
- **Cachés:** Se precalculan y almacenan rutas y distancias entre nodos de interés (base y habitaciones) usando estructuras de caché, lo que mejora la eficiencia durante la evaluación del fitness.

3. Operadores Genéticos

3.1 Selección

Se implementaron métodos como ruleta, torneos determinísticos y probabilísticos, selección estocástica universal, truncamiento, restos y ranking.

- **Mejor resultado:** Los mejores resultados se obtienen con las funciones de torneo determinístico, ranking y truncamiento, debido a su naturaleza más elitista que da prioridad a los mejores individuos de cada generación, consiguiendo las mejores rutas y llegando a la óptima en muchas de las ocasiones.

3.2 Cruce

Se probaron varios cruces específicos para permutaciones:

- **PMX:** Preserva la estructura genética intercambiando segmentos y corrigiendo duplicados.
- **OX y OXPP:** Conservan el orden relativo de las ciudades, permitiendo completar el recorrido en secuencia.
- **Ciclo (CX):** Asigna posiciones mediante ciclos sucesivos.
- **CO (Codificación Ordinal):** Transforma la permutación en una secuencia de desplazamientos sobre una lista dinámica aleatoria, reduciendo el problema a una codificación numérica que varía entre 1 y m – j + 1.

- **ERX:** Combina rutas priorizando conexiones existentes para generar descendientes con patrones eficientes.
- **Invencion Propia:** Hemos optado por hacer un cruce que mezcla el ERX, con una mejora, en vez de tomar entre todas las ciudades conectadas, la que menos conectada esta, se escoge la que está más cerca (menos distancia). Al seleccionar el vecino más cercano, la descendencia probablemente contendrá tramos de ruta que son óptimos en cuanto a la distancia se refiere. Además ayuda a filtrar aquellas conexiones que, aunque puedan ser comúnmente heredadas, no son necesariamente las mejores en cuanto a recorrido se refiere, evitando así rutas con saltos largos que puedan deteriorar la calidad global.

Conclusión: Los métodos ERX y OXPP han mostrado mejores resultados en términos de calidad de solución. El método OXPP y OX son especialmente útiles para problemas del viajante como el nuestro donde las rutas son importantes, ya que mantienen el orden relativo de los elementos generando hijos válidos directamente sin necesidad de hacer correcciones como por ejemplo el PMX. ERX también funciona muy bien para nuestro robot, ya que para la optimización de caminos puede generar hijos con conexiones entre habitaciones que sean eficientes sin tener que romper la estructura, priorizando rutas que ya teníamos, evitando soluciones no tan buenas. Aunque con el resto de cruces también podemos llegar a resultados óptimos, pueden no funcionar tan bien sin una mutación adecuada.

3.3 Mutación

Se aplican diversos operadores:

- **Inserción:** Mueve un elemento a otra posición. No es la mejor opción si buscamos explorar nuevas regiones del espacio de búsqueda.
- **Intercambio:** Intercambia dos genes aleatorios. Puede quedarse estancado en óptimos locales.
- **Inversión:** Invierte un segmento del recorrido. Es muy útil para explorar nuevas regiones y escapar de los óptimos locales.
- **Heurística:** Genera permutaciones parciales sobre un subconjunto seleccionado; aunque es computacionalmente más costosa, favorece la exploración de soluciones de alta calidad.
- **Invención Propia:** Hemos optado por hacer una mutación que funciona con 5 probabilidades, cada una de las cuales tiene una acción asignada. La mutación puede invertir un fragmento del cromosoma (0,2); mover una habitación al inicio o al final del camino (0,1); intercambiar habitaciones cercanas (0,3); acercar una habitación que esté lejos (0,15); o deshacer los cambios si el fitness no mejora (0,05). Aunque esta mutación funciona a base de probabilidades aleatorias, hace que se obtengan resultados buenos.

Conclusión: Los mejores resultados se obtienen con el método de inversión, al ser una mutación simple que consigue no quedarse estancada en óptimos locales a la vez que explora soluciones de manera muy efectiva. A su vez, la mutación que nos hemos inventado también funciona muy bien, ya que tiene un buen equilibrio de exploración y explotación debido a su mezcla de operaciones y es flexible.

4. Fitness

Se diseñaron múltiples funciones de fitness que evalúan la calidad de cada solución basándose en criterios:

- **Distancia total recorrida**: Se penalizan rutas inviables (sin camino o con altos costes).
- **Penalización por proximidad a obstáculos:** Se aumenta el coste de rutas que pasan cerca de obstáculos, favoreciendo trayectorias seguras.
- **Penalización por giros o cambios de dirección**: Se suman penalizaciones por cada giro que supere un umbral, incentivando caminos rectos y fluidos.
- **Heurística mejorada**: Se utiliza una heurística que incorpora la densidad de obstáculos en la dirección del destino, permitiendo encontrar rutas intermedias de mejor calidad.
- **Distancia euclidiana**: Esta función calcula la distancia que hay en línea recta entre dos puntos haciendo uso de la fórmula de la distancia euclidiana. Si la distancia real que existe entre esos dos puntos es mayor que la euclidiana, se inflige un castigo proporcional.
- **Costo de tiempo de recorrido:** Se modela el tiempo como función de la distancia recorrida y los giros, penalizando cambios de dirección excesivos.
- **Optimización multiobjetivo:** Se combinan varios criterios (distancia, giros, proximidad a obstáculos, tiempo) con ponderaciones ajustadas para equilibrar eficiencia, seguridad y fluidez.

Pese a que los fitness aparecen más altos en algunas funciones, no quiere decir que la ruta sea así de larga, simplemente que al añadirse las penalizaciones, el valor real del fitness se ve aumentado. Las rutas siguen alcanzando los valores deseables.

5. Cachés y Precálculo de Caminos

La clase Casa implementa precálculo de rutas entre la base y cada habitación, y entre todas las habitaciones, utilizando el algoritmo A* modificado.

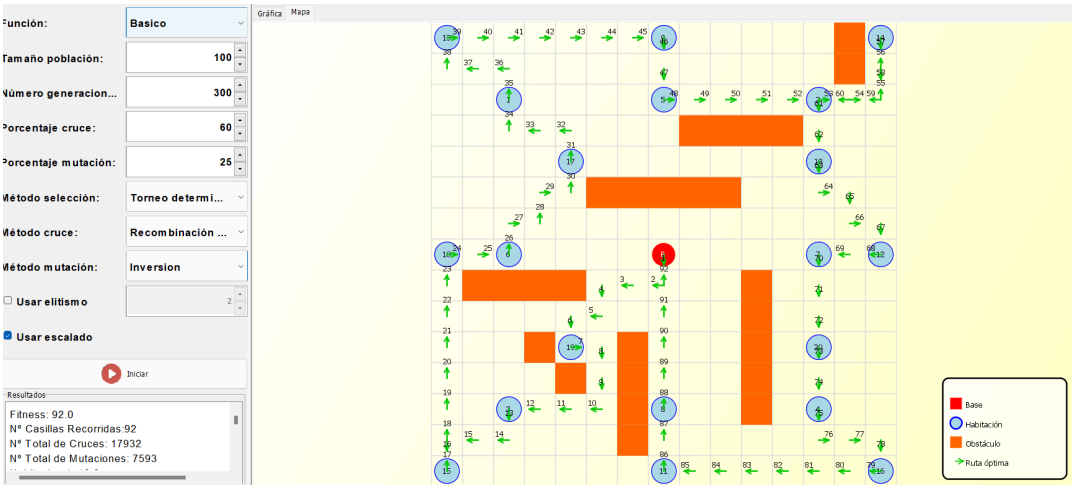
- Se almacenan en caché tanto la ruta óptima (lista de Points) como la distancia.
- La actualización de la cache se realiza solo cuando los valores de fitness cambian, minimizando cálculos redundantes

Para cada función de fitness tenemos una función (que es una variante del A*) que hace el cálculo de los mejores caminos y distancias entre habitaciones en base a lo que estemos queriendo buscar. Por ejemplo, si tratamos de evitar obstáculos o giros, la función que precalcula los caminos tendrá más en cuenta esas restricciones, no limitándose solamente a encontrar el camino más corto, y aplica penalizaciones en función de lo que se intenta evitar.

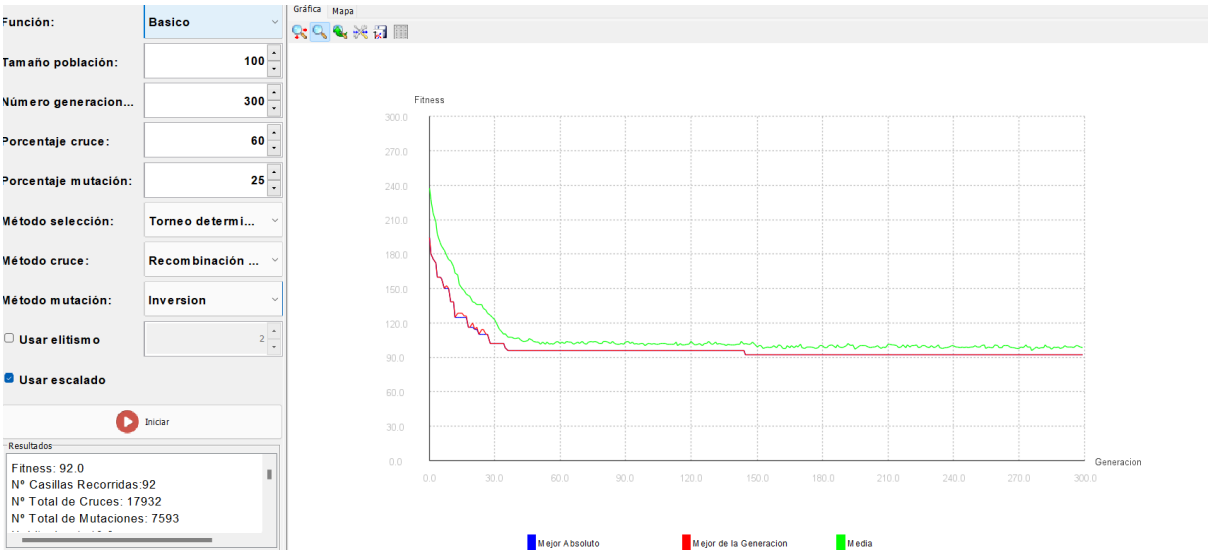
6. Resultados y Gráficas

A continuación vamos a poner los resultados de 5 de las posibles mejores ejecuciones:

- **Individuo solución**: fitness básico
- **Aptitud máxima**: 92
- **Aptitud mínima**: 194
- **Selección**: torneo determinístico
- **Cruce:** ERX (recombinación de rutas)
- **Total cruces:** 17932
- **Mutación:** Inversión
- **Total mutaciones:** 7593
- **Gráfica recorrido:**

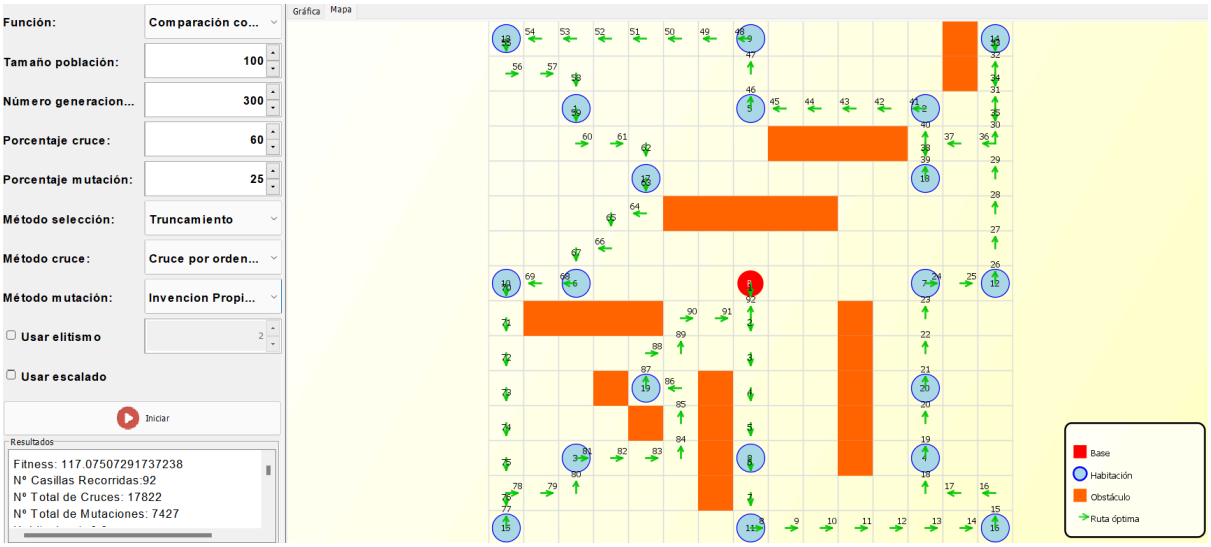


- **Gráfica evolución:**

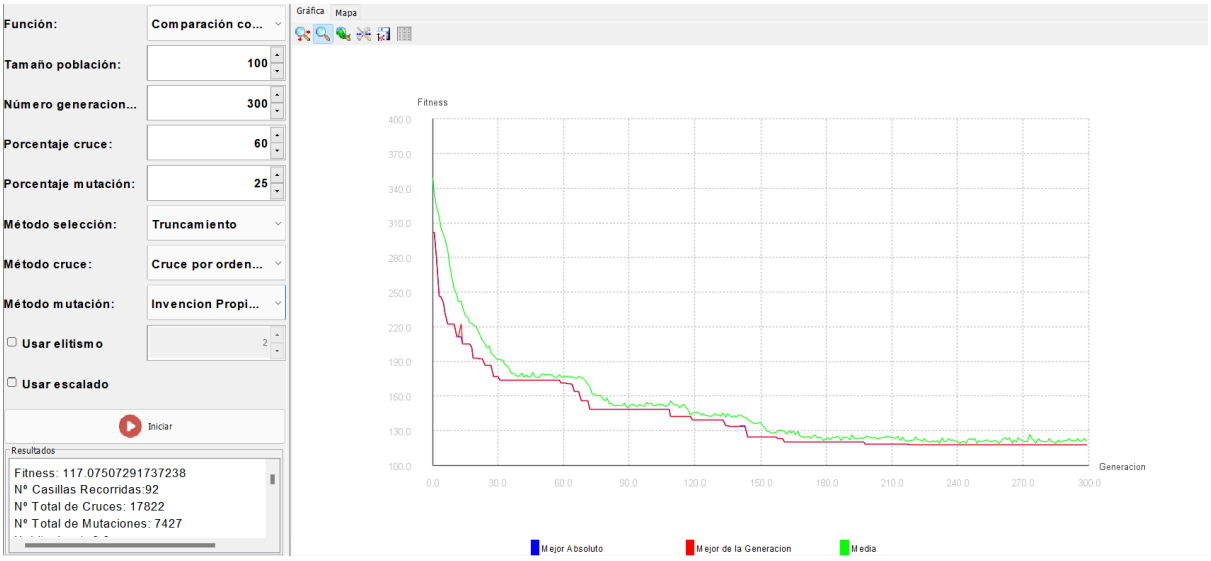


Esta función suma uno al fitness por cada casilla recorrida, así que cuanto más corta sea la ruta que se use, mejor fitness obtiene el individuo. Simplemente se limita a elegir los mejores caminos entre habitaciones ya precalculados que existen en el orden dado por el cromosoma. Presenta buenos resultados, y con ella se puede encontrar la ruta mínima. Funciona mejor con el método de selección de truncamiento o torneo determinístico. El torneo determinístico hace que la selección sea bastante elitista.

- **Individuo solución**: comparación distancia euclidiana
- **Aptitud máxima**: 92
- **Aptitud mínima**: 301
- **Selección**: truncamiento
- **Cruce:** OXPP
- **Total cruces:** 17822
- **Mutación:** invención propia
- **Total mutaciones:** 7427
- **Gráfica recorrido:**



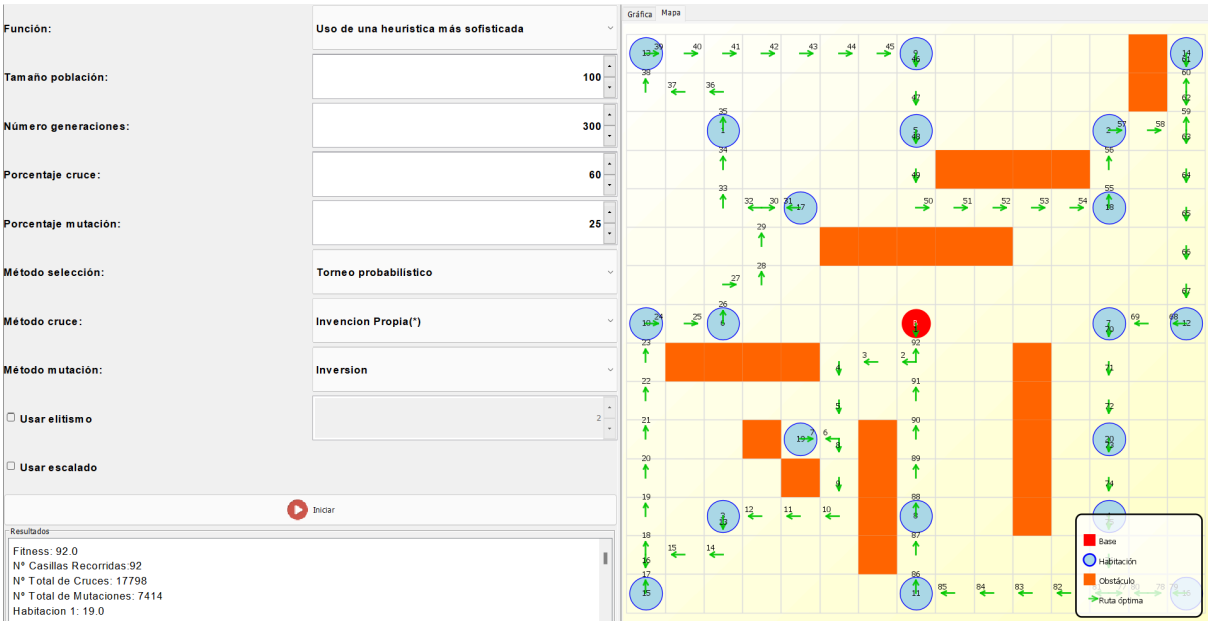
- **Gráfica evolución:**



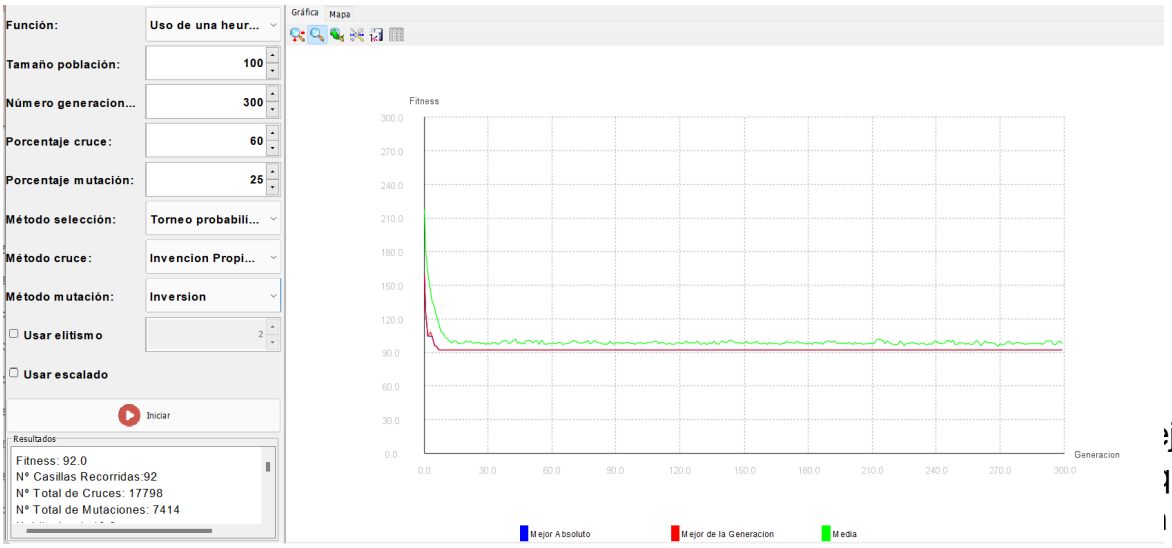
Tal y como hemos explicado anteriormente, esta función, que penaliza el individuo según lo distante que sea la distancia real entre dos habitaciones de la distancia euclidiana, tiene un fitness más alto de lo normal que no representa las casillas recorridas, ya que a ese número se le añaden las consecuentes penalizaciones. El número real de pasos dados por el robot aparece especificado debajo del fitness. Observamos también que el método de truncamiento es muy elitista.

- **Individuo solución**: uso de una heurística más sofisticada
- **Aptitud máxima**: 92
- **Aptitud mínima**: 160
- **Selección**: Torneo probabilístico

- **Cruce**: invención propia
- **Total cruces:** 17798
- **Mutación:** inversión
- **Total mutaciones:** 7414
- **Gráfica recorrido:**



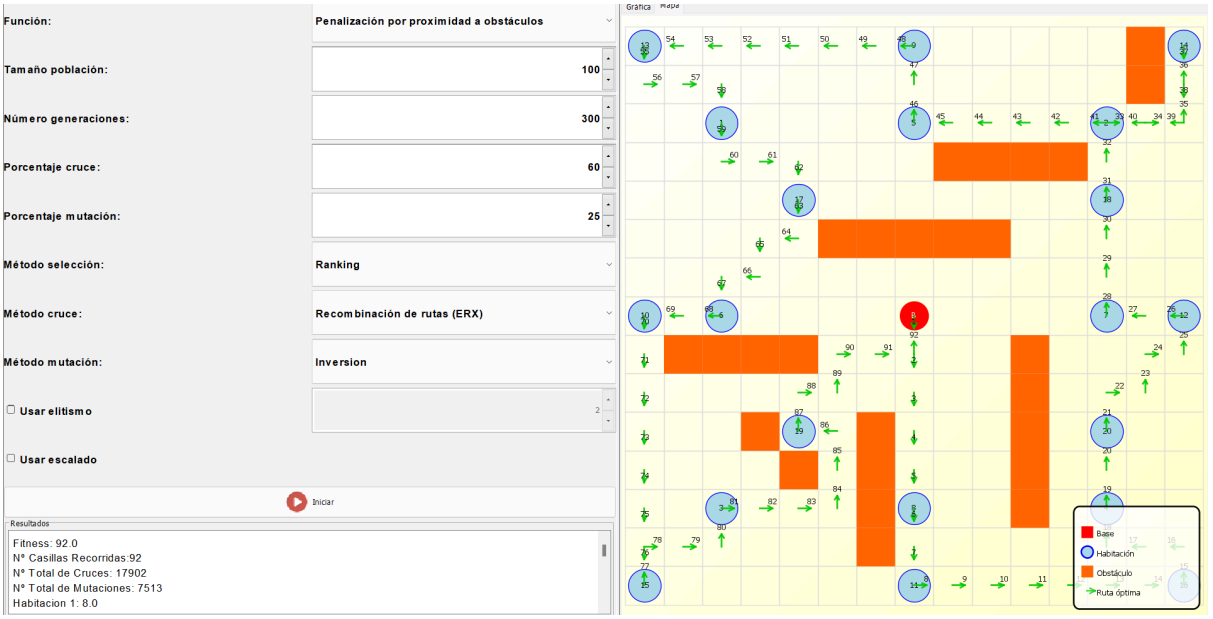
- **Gráfica evolución:**



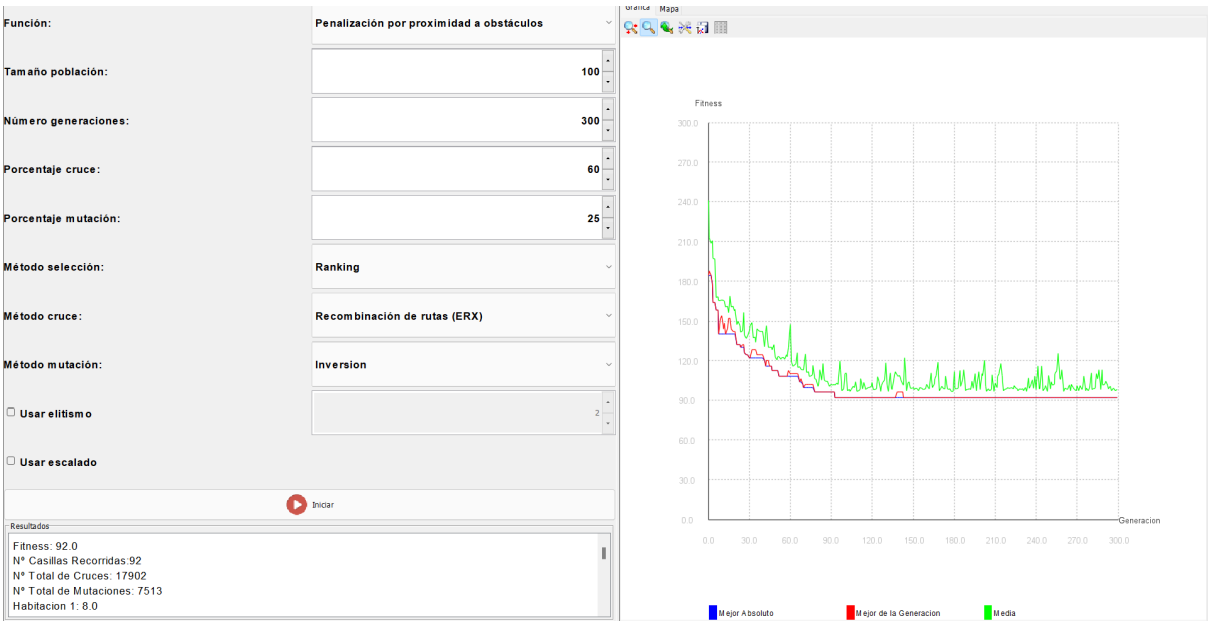
ejor
que
el
torneo probabilístico hace que la gráfica tenga también aspecto elitista.

- **Individuo solución**: penalización por proximidad a obstáculos
- **Aptitud máxima**: 92
- **Aptitud mínima**: 184
- **Selección**: Ranking
- **Cruce**: ERX

- **Total cruces:** 17902
- **Mutación:** inversión
- **Total mutaciones:** 7513
- **Gráfica recorrido:**



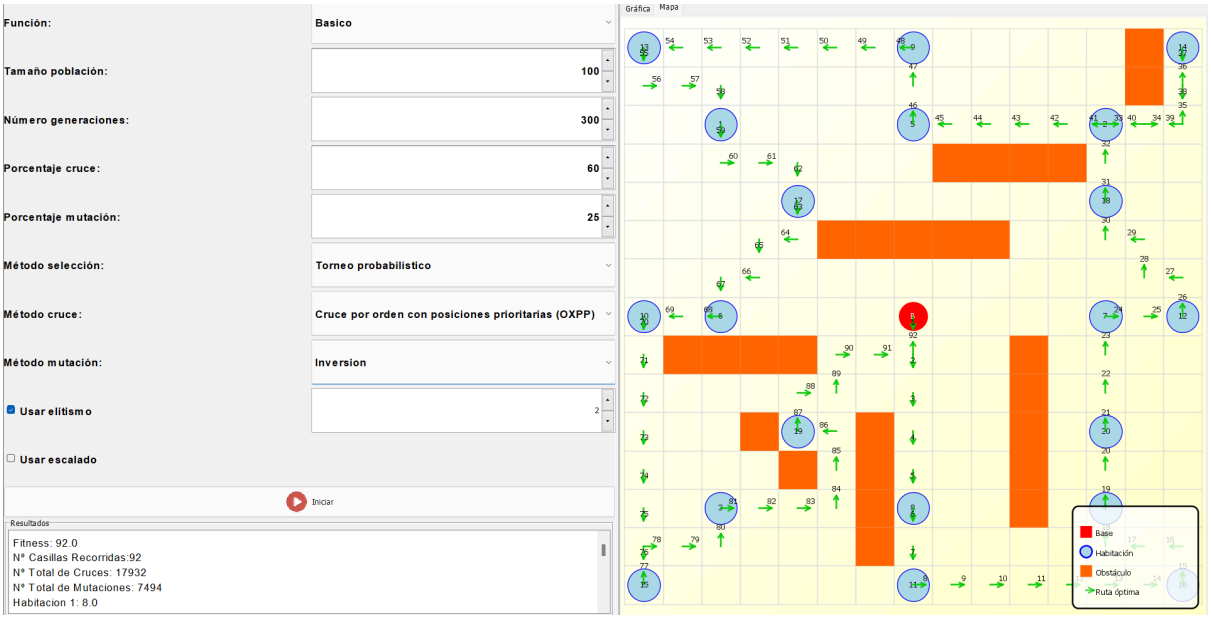
- **Gráfica evolución:**



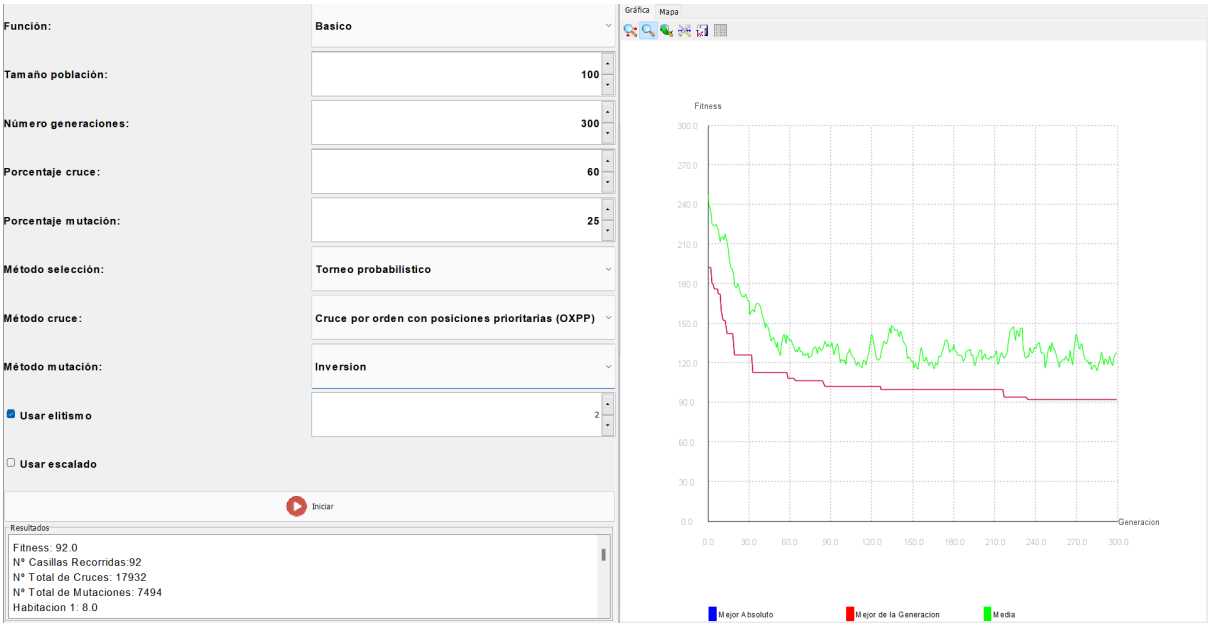
Haciendo uso del ranking y de ERX, a la función que penaliza por obstaculos no le es difícil encontrar la mejor ruta, puesto que vemos que tras un descenso con algún altibajo, la estabilidad se encuentra rápidamente sin necesitar todas las generaciones.

- **Individuo solución**: fitness básico con elitismo del 2%
- **Aptitud máxima**: 92

- **Aptitud mínima**: 192
- **Selección**: Ranking
- **Cruce**: OXPP
- **Total cruces:** 17932
- **Mutación:** inversión
- **Total mutaciones:** 7494
- **Gráfica recorrido:**



- **Gráfica evolución:**



Esta función implementa elitismo junto con OXPP e inversión. La forma de la gráfica es parecida a la del torneo determinístico, el funcionamiento del elitismo sigue siendo el mismo que en la práctica anterior.

7. Conclusiones

- **Eficiencia:** El uso de caches y el precálculo de rutas reducen significativamente el tiempo de evaluación.
- **Calidad de las Soluciones:** La combinación de operadores (ERX, OXPP, cruces ordinales) y las funciones de fitness multiobjetivo permiten obtener rutas competitivas y seguras.
- **Robustez:** La incorporación de penalizaciones por giros, proximidad a obstáculos y tiempo de recorrido mejora la viabilidad de las soluciones en entornos complejos.
- **Balance:** La selección por torneo y los parámetros ajustados (población, probabilidad de cruce y mutación) favorecen la convergencia sin caer en estancamientos prematuros.

Esta práctica integra diversas técnicas avanzadas de algoritmos evolutivos aplicadas al problema del TSP y la planificación de rutas, demostrando la aplicabilidad de estos métodos en problemas de optimización combinatoria y en entornos con restricciones físicas.

Otras observaciones: El uso de escalado lineal modifica muy ligeramente los resultados, no implica diferencias drásticas, pero aun así hace que los caminos finales sean ligeramente mejores.

8. REPARTO DE TAREAS

Javier Martín-Tesorero Ruiz: implementación de algoritmos de fitness, mutación, cruce y cambios en algoritmo genético, actualización de la GUI, lógica del mapa, invención del cruce nuevo, depuración y trabajo en la memoria.
Clara Martín Navas: implementación de algoritmos de fitness, mutación, cruce y cambios en algoritmo genético, creación de ranking, lógica del mapa, invención de la mutación nueva, depuración y trabajo en la memoria.