

# Algoritmos geneticos

Luis Trucios  
Max Tresierra  
Maicol Mendo

Junio 8<sup>th</sup> 2024

## 1 Introducción

El problema del viajero (TSP) ha resultado ser un desafio habitual en los temas de optimizacion. Este, busca determinar la ruta mas corta para visitar un conjunto de ciudades una sola vez y que se pueda regresar al punto de inicio. Ademas, se conoce que es un problema NP-hard, es por esto que tratar de resolverlo implica grandes cantidades de tiempo y esfuerzo. Es por eso, que el uso de metodos heuristicos y metaheuristicos, uo de ellos los algoritmos geneticos. Estos se inspiran e la seleccion natural y simulan la evolucion biologica

## 2 Optimizacion de hiperparametros

Consideraciones a tener en cuenta dependiendo del tipo de inicializacion:

- **Aleatoria:** Se encuentra que es mejor tener una cantidad grande de poblacion, un grupo de elite tambien grande ( $> 20$ ) y un ratio de mutacion pequeno ( $< 0.1$ ), esto debido a que al tener un grupo de elite grande, no se necesita una mutacion grande.
- **Heuristica:** Tambien se encuentra que es mejor tener una cantidad grande de poblacion, pero con un grupo de elite mas pequeno ( $> 10$ ) y un ratio de mutacion alta ( $< 0.2$ ), esto debido a que al partir de una poblacion mas optimizada, se puede elegir un grupo de elite mas corto y una mutacion grande.

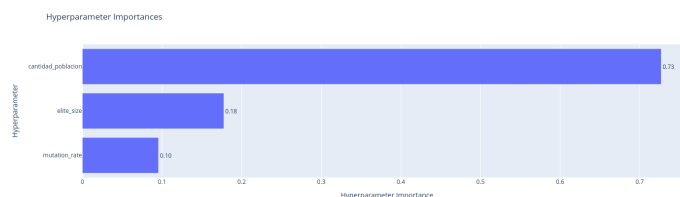


Figure 1: Importancia de optimizar los parametros

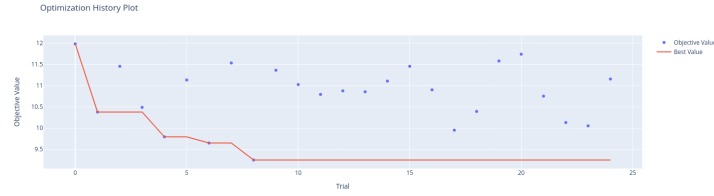


Figure 2: Optimizacion

### 3 Comparación de métodos de selección

Para el presente experimento, vamos a comparar cuatro métodos de selección al problema del viajero (TSP). Este tendrá 100 ciudades que fueron generadas aleatoriamente

1. Roulette Wheel Selection
2. Rank-Based Selection
3. Fitness Scaling Selection
4. Tournament Selection

#### 3.1 Resultados

Se ha hecho dos análisis para el experimento de comparación de métodos de selección. Usando 5000 generaciones se decidió algunas condiciones para tener un ambiente controlado para la experimentación, estos son los que se detallan a continuacion.

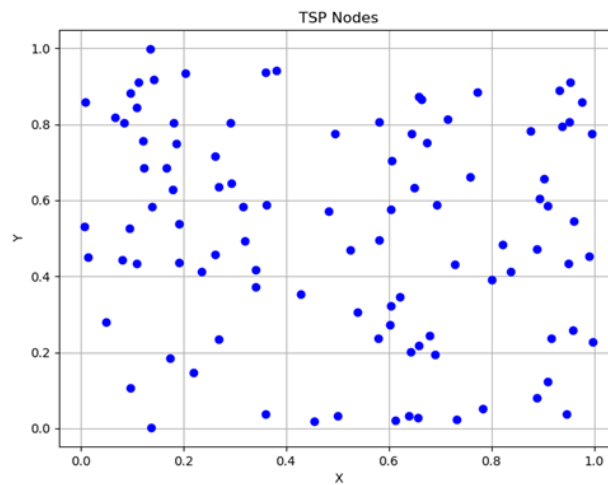


Figure 3: Nodos del problema (TSP)

Poblacion	Elites iniciales	Mutacion
75	30	0.1

Table 1: Hiperparametros optimizados

### 3.2 Primer analisis

Usando hiperparámetros optimizados, se obtuvieron los siguientes resultados

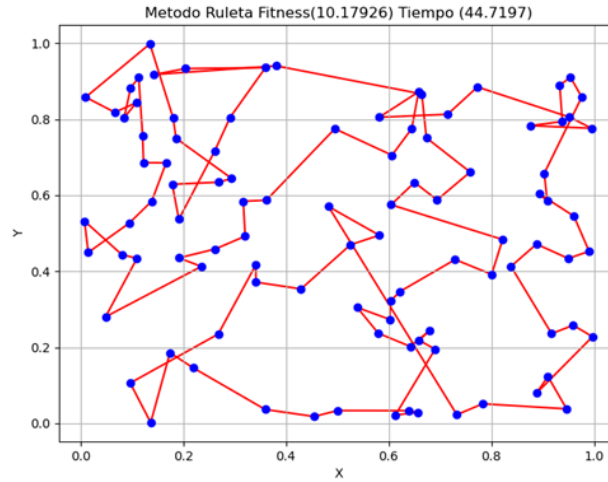


Figure 4: Metodo ruleta

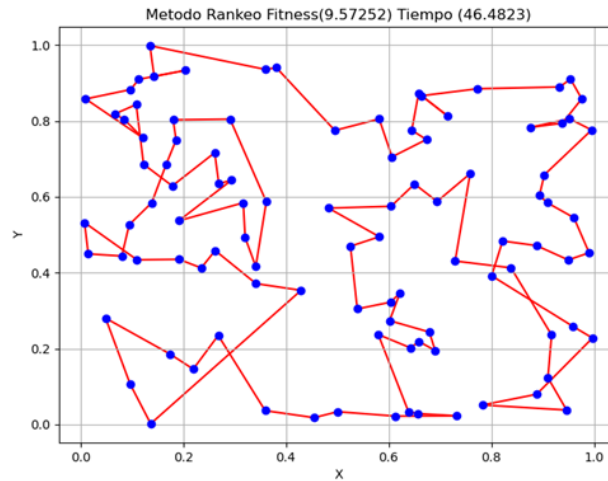


Figure 5: Metodo rank

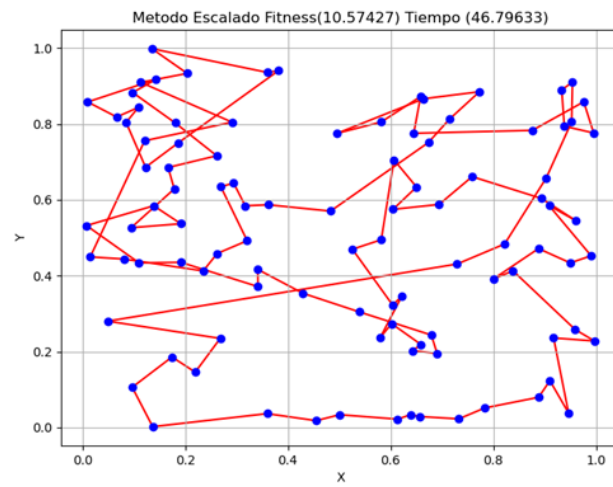


Figure 6: Metodo escalado

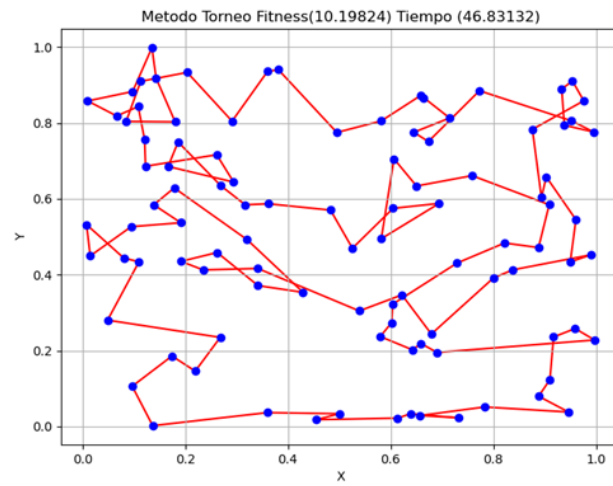


Figure 7: Metodo torneo

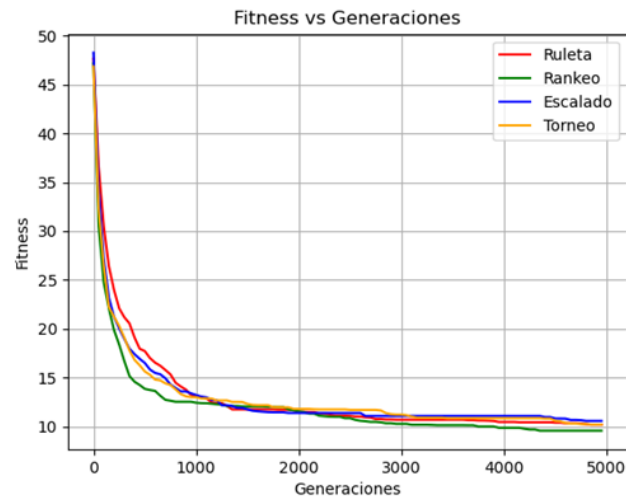


Figure 8: Fitness - Generaciones

Método de Selección	Fitness	Tiempo (s)
Ruleta	10.17926	44.7197
Rankeo	9.57252	46.4823
Escalado	10.57427	46.79633
Torneo	10.19824	46.83132

Table 2: Comparación de métodos de selección en algoritmos genéticos

### 3.3 Segundo analisis

El segundo experimento se decidió dejar la optimización al algoritmo de selección, por lo que se cambiaron los hiperparámetros:

Poblacion	Elites iniciales	Mutacion
75	1	0.0001

Table 3: Comparación de métodos de selección en algoritmos genéticos

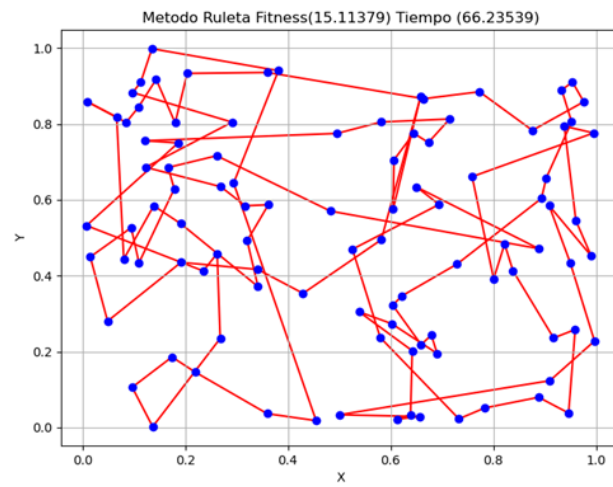


Figure 9: Metodo ruleta

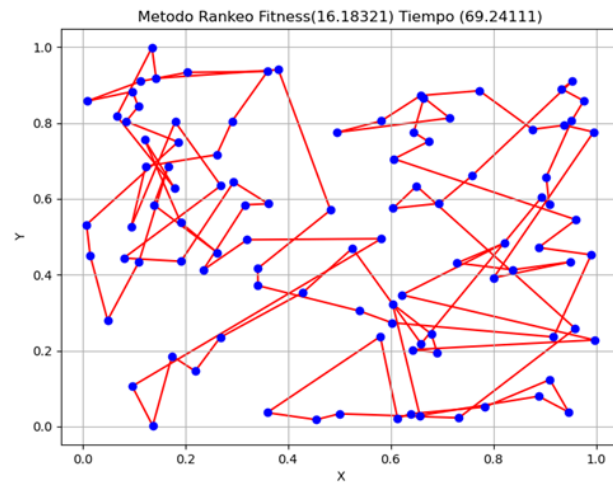


Figure 10: Metodo rank

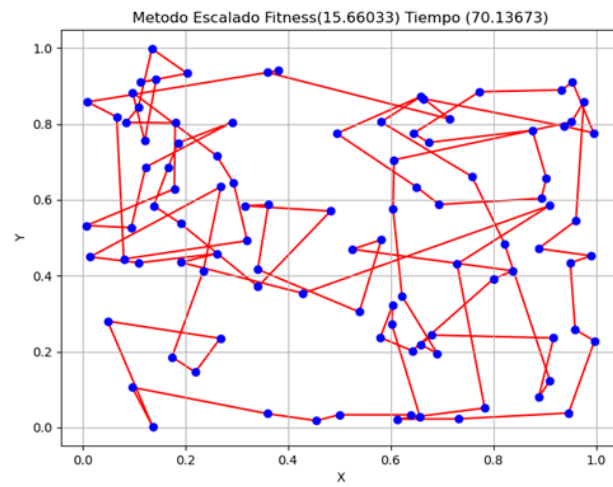


Figure 11: Metodo escalado

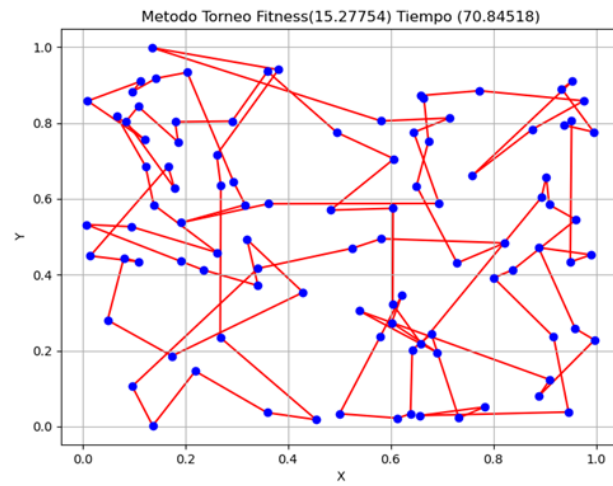


Figure 12: Metodo torneo

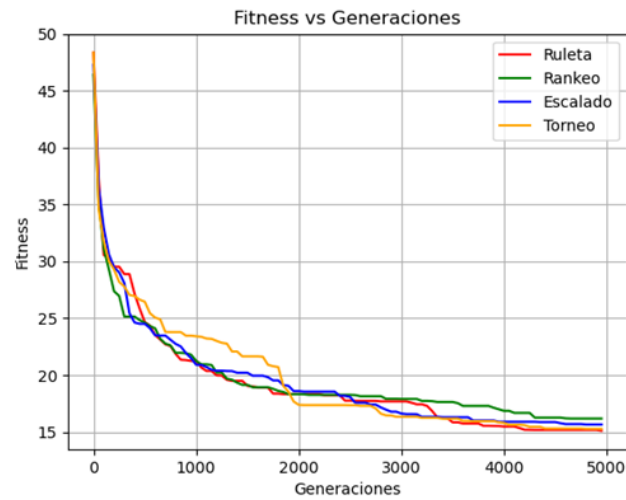


Figure 13: Fitness - Generaciones

Método de Selección	Fitness	Tiempo (s)
Ruleta	15.11379	66.23539
Rankeo	16.18321	69.24111
Escalado	15.66033	70.13673
Torneo	15.27754	70.84518

Table 4: Comparación de métodos de selección en algoritmos genéticos

### 3.4 Analisis adicional

Usamos las mismas condiciones del experimento 1 y 2 pero aumentaremos las genraciones a 10000.

Las pruebas realizadas en el caso del experimento 1 dieron los siguientes resultados:

Método de Selección	Fitness	Tiempo (s)
Ruleta	8.72213	89.6963
Rankeo	9.7197	96.86751
Escalado	9.33449	98.7132
Torneo	9.98314	98.80568

Table 5: Comparación de métodos de selección en algoritmos genéticos



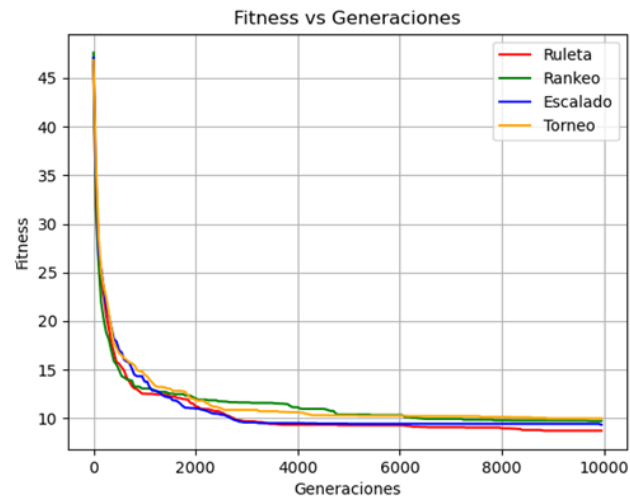


Figure 14: Fitness - Generaciones

Las pruebas realizadas en el caso del experimento 2 dieron los siguientes resultados:

Método de Selección	Fitness	Tiempo (s)
Ruleta	18.96109	133.00771
Ranqueo	13.45453	139.18093
Escalado	12.20539	142.42733
Torneo	12.55272	142.14268

Table 6: Comparación de métodos de selección en algoritmos genéticos

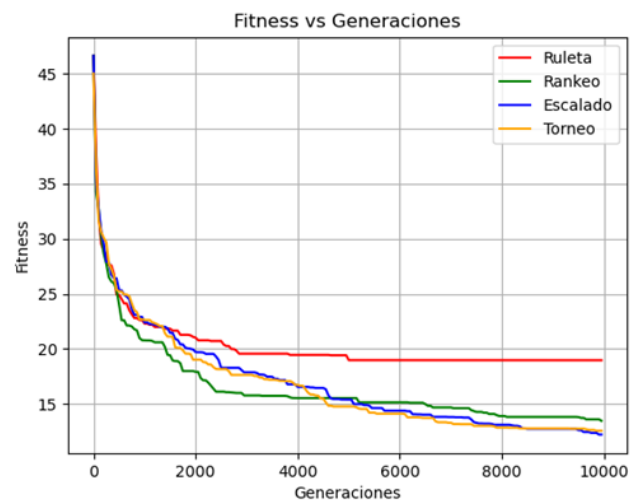


Figure 15: Fitness - Generaciones

### 3.5 Discusion

Vemos que todos los algoritmos son útiles al momento de optimizar valores pues, de escoger de forma adecuada los Hiperparámetros, mantienen un balance entre diversificación e intensificación y se aseguran de encontrar una solución óptima. Si no se escoge adecuadamente los hiperparámetros puede haber inconvenientes, por ejemplo: al agregar solamente 1 elemento optimo a la nueva población, por cuestiones de suerte, el algoritmo de ruleta, se quedo en un valor demasiado alto y es muy probable que necesite una mayor cantidad de generaciones para encontrar la solución óptima. En todos los casos (con o sin optimización de hiperparámetros) el método del Torneo siempre se asegura de ir intensificando la solución a partir de una alta diversificación.

Si hablamos en temas de tiempo de ejecución. De menor a mayor tiempo

1. Roulette Wheel selection
2. Rank-based selection
3. Fitness scaling selection
4. Tournament selection

Definimos que con los hiperparámetros adecuados la solución para el problema se encuentra entre una distancia total entre 9 y 10 unidades. Tomando en cuenta lo anterior nos hemos dado cuenta que quienes llegan en menor cantidad de generaciones a este resultado óptimo es el algoritmo de la ruleta y de ranqueo. Si se desea sacrificar un poco de tiempo por una mayor diversificación aumentando un poco más las generaciones el método del ranqueo puede ser apropiado para este caso. Además, definimos que el algoritmo de Ranqueo si bien no llega a las soluciones tan pequeñas, aun así, se encuentra dentro del rango de solución, es más es el método que más probablemente llegue a la solución optima en menor cantidad de generaciones, esto debido a que sin importar que tan cerca se encuentren entre sí los fitness, balancea bien la probabilidad de los fitness más altos. Es por todo esto, que nuestra conclusión es que el método de Ranqueo es el más adecuado para encontrar una solución para este tipo de problema.

## 4 Comparación de métodos de inicialización de población

En este segundo experimento, comparamos tres métodos de inicialización de la población para el algoritmo genético al mismo problema del experimento uno. Los métodos fueron:

1. Random Initialization
2. Heuristic Initialization
3. Hybrid Initialization

Se usaron los siguientes hiperparametros:

Inicializacion	textbfPoblacion	Elites iniciales	Mutacion
Random Initialization	75	30	0.10
Heuristic and Hybrid Initialization	75	10	0.2

Table 7: Hiperparametros optimizados

## 4.1 Analisis

- **Random Initialization:** Proporciona una amplia diversidad genética inicial, pero puede requerir más generaciones para converger a una solución óptima.

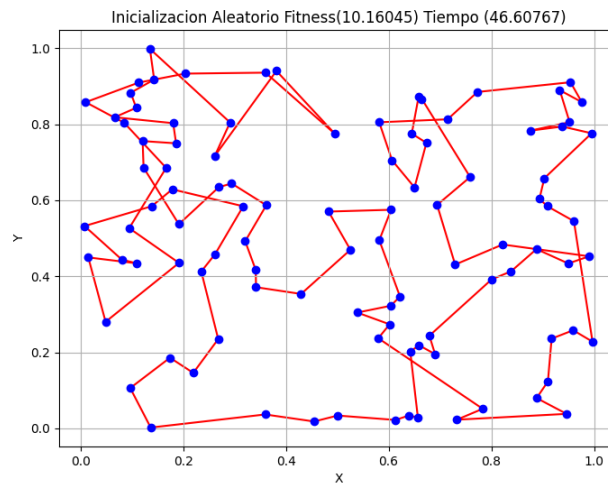


Figure 16: Random Initialization

- **Heuristic Initialization:** Nos da soluciones cercanas a las óptimas, pero nos quita la diversidad y la capacidad de explorar un espacio de soluciones más grande.

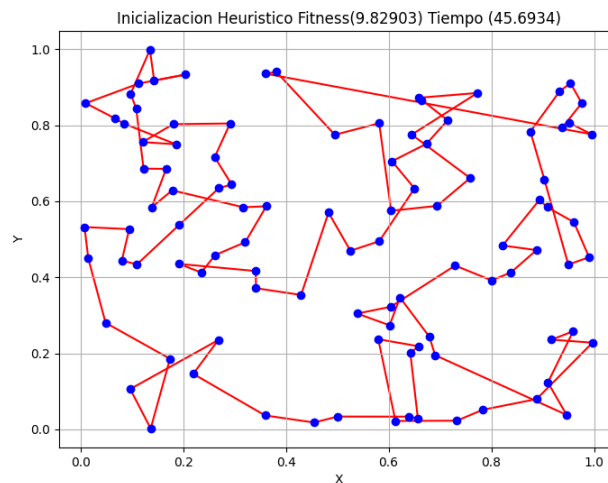


Figure 17: Heuristic Initializatio

- **Hybrid Initialization:** Realiza la combinación de ambos métodos, esto nos asegura que proporciona la diversidad inicial y luego realizará mejoras heurísticas

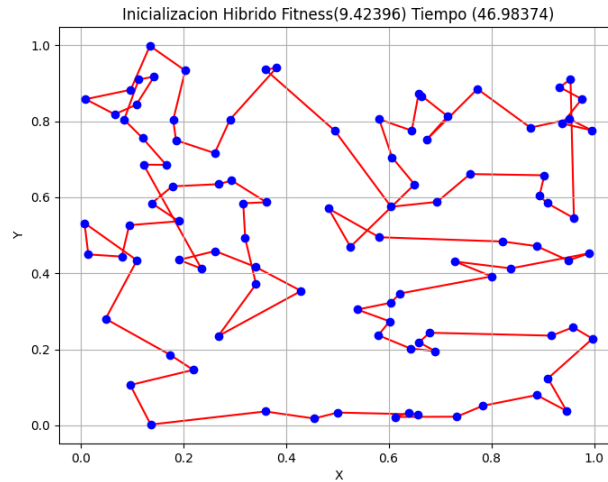


Figure 18: Hybrid Initialization

## 4.2 Discusion

Se observa que la inicializacion heuristica es la que converge mas rapido, pero a la larga la inicializacion hibrida es la que obtiene mejores resultados, esto debido a que una inicializacion hibrida aprovecha lo mejor de ambos mundos, y no requiere una configuracion mas precisa de los parametros. Ademas, nos dimos cuenta que la inicializacion heuristica es muy sensible a la inicializacion de parametros, ya que al converger mas rapido, se podria agregar un scheduler que pueda modificar el ratio de mutacion a medida vamos aumentando las generaciones.

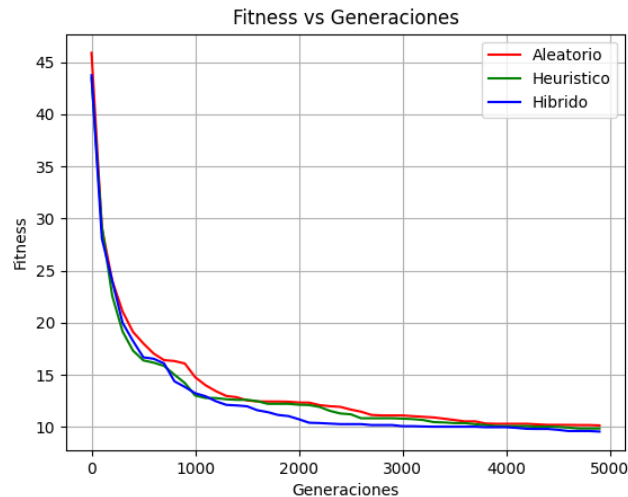


Figure 19: Fitness - generaciones

Método de Selección	Fitness	Tiempo (s)
Random Initialization	10.1645	46.60767
Heuristic Initialization	9.82903	45.6934
Hybrid Initialization	9.42396	46.98374

Table 8: Comparación de métodos de selección en algoritmos genéticos

## 5 Conclusiones

1. Se observa que la aleatoriedad en un principio puede ayudar a diversificar soluciones pero si nos quedamos solo con eso, llegamos a caer a un optimo local. Es por esto que es bueno combinar las diferentes formas de inicializacion, ya que si iniciamos por una muy buena solucion, perdemos la posibilidad de explorar el espacio de soluciones
2. El numero de la poblacion a seleccionar en los parametros iniciales tiene un papel importante en el algoritmo genetico, ya que determinara la diversidad, tiempo de ejecucion, convergencia. Ademas, a mayores candidatos, estaremos explorando el espacio de busqueda, mientras que por el contrario, se estara buscando una intensificacion.
3. Nos hemos percatado que el metodo del torneo evalua las soluciones en funcion de su rendimiento relativo. Esto ayuda a eliminar la presencia de multiples optimos locales

## 6 Repositorio

En este apartado se deja el link del repositorio en GitHub.

[https://github.com/Maicoll1591/algoritmos\\_geneticos.git](https://github.com/Maicoll1591/algoritmos_geneticos.git)