Algoritmos genéticos

Gary Kris Edward Chavez Vasquez

Jun 2024

1. Introducción

El trabajo consite en implementar código para dos experimentos con métodos de selección e inicialización de población en algoritmos genéticos. Para cada experimento, generamos un conjunto de 100 ciudades y comparamos los métodos seleccionados en términos de fitness a lo largo de las generaciones.

- Experimento 1: Para un mismo conjunto de 100 ciudades, implementar y comparar la solución obtenida usando los métodos de selección: Roulette wheel selection Rank-based selection, Fitness scaling, y Tournament selection.
- Experimento 2: Para un mismo conjunto de 100 ciudades, implementar y comparar la solución obtenida usando los métodos inicialización de población: random, heuristic y hybrid initialization..

2. Resultados

2.1. Para el experimento 1

Para dicho experimento se inició con los siguientes pasos:

(2.1.1) Generación de Ciudades. Crea 100 ciudades con coordenadas aleatorias.

(2.2.2) Funciones de Selección Desarrolla las siguientes funciones de selección

- Roulette Wheel Selection: Selecciona individuos proporcionalmente a su fitness.
- Rank-based Selection: Asigna probabilidades de selección basadas en el ranking del fitnes.
- Fitness Scaling Selection: Selecciona individuos después de escalar sus valores de fitness.
- Tournament Selection: Selecciona el mejor de un subconjunto aleatorio de la población.

(2.3.3) Función de Fitness. calculate_fitness(route, cities): Calcula la distancia total de una ruta.

(2.4.4) Operadores Genéticos. calculate_fitness(route, cities): Calcula la distancia total de una ruta

- Mutación: mutate(individual, mutation_rate): Realiza intercambio de dos ciudades con una probabilidad definida.
- Cruce: crossover(parent1, parent2): Crea un nuevo individuo a partir de dos padres utilizando un cruce ordenado.

(2.5.5) Algoritmo Genético. run_ga(...): Ejecuta el algoritmo genético por un número de generaciones, usando un método de selección y registra la historia del fitness.

(2.6.6) Visualización.

- plot_route(...): Grafica la mejor ruta obtenida.
- plot_fitness(...): Grafica el progreso del fitness a lo largo de las generaciones.

2.2. Para el experimento 2

Para dicho experimento se inició con los siguientes pasos:

(2.2.1) Generación de Ciudades. Crea 100 ciudades con coordenadas aleatorias.

(2.2.2) Métodos de Inicialización de Población Desarrolla las siguientes funciones de selección

- Random Initialization: Genera permutaciones aleatorias de ciudades.
- Heuristic Initialization: Genera rutas iniciales basadas en una heurística de distancia mínima.
- Hybrid Initialization: Combina los métodos aleatorios y heurísticos para la generación de la población.

(2.2.3) Métodos de Selección. Tournament Selection: Selecciona individuos de torneos de un subconjunto aleatorio de la población.

(2.2.4) Función de Fitness. calculate_fitness(route, cities): Calcula la distancia total de una ruta.

(2.2.5) Operadores Genéticos. calculate_fitness(route, cities): Calcula la distancia total de una ruta.

- Mutación: mutate(individual, mutation_rate): Realiza intercambios de ciudades en un individuo con cierta probabilidad.

- Cruce: crossover(parent1, parent2): Combina dos individuos padres para generar nuevos individuos.

(2.2.6) Algoritmo Genético. run_ga(...): Ejecuta el algoritmo genético utilizando la selección por torneo y registra la historia del fitness.

(2.2.7) Visualización.

- plot_route(...): Grafica la mejor ruta obtenida.
- plot_fitness(...): Grafica el progreso del fitness a lo largo de las generaciones.

3. Discución

Los resultados para los experimentos se muestran a continuación:

3.1. Experimento 1: Métodos de selección

La selección por torneo a menudo proporcionó las rutas de menor distancia debido a su capacidad para equilibrar la exploración y explotación del espacio de soluciones.

(3.1.1) Roulette Wheel Selection. Tendió a explorar ampliamente el espacio de soluciones, pero a veces se quedó atrapado en óptimos locales, especialmente si había una gran disparidad en los valores de fitness.

(3.2.2) Rank-based Selection. Ofreció una mejor estabilidad al evitar sesgos excesivos hacia los individuos de alta calidad, pero su convergencia fue más lenta.

(3.3.3) Fitness Scaling. Similar al método Roulette Wheel pero con menos riesgo de concentración en óptimos locales debido a la escalación del fitness.

(3.4.4) Tournament Selection. Mostró una robustez superior en la explotación de soluciones óptimas y fue eficaz en mantener la diversidad genética, logrando a menudo la mejor solución en menos generaciones.

3.2. Experimento 2: Métodos de Inicialización

La inicialización híbrida proporcionó consistentemente las mejores soluciones finales en términos de distancia, aprovechando tanto la diversidad como la calidad inicial.

(3.2.1) Random Initialization. Proporcionó una amplia diversidad inicial, pero a menudo requirió más generaciones para alcanzar una solución óptima debido a la aleatoriedad.

(3.2.2) Heuristic Initialization. Generó soluciones iniciales de alta calidad, lo que aceleró la convergencia inicial pero a veces limitó la exploración del espacio de soluciones.

(3.2.3) Hybrid Initialization. Combinó la diversidad de la inicialización aleatoria con la calidad de la inicialización heurística, logrando un equilibrio óptimo entre exploración y explotación..

3.3. Comparación General

(3.3.1) Eficiencia en la Convergencia.

- Expeimento 1. La selección por torneo fue más eficiente en encontrar soluciones óptimas rápidamente, aprovechando su balance entre selección competitiva y mantenimiento de la diversidad. La selección basada en el ranking tuvo una convergencia más estable, pero más lenta, mientras que la selección por ruleta y escalado de fitness a veces se estancaron en óptimos locales.
- Experimento 2. La inicialización híbrida ofreció la mejor convergencia inicial y final, combinando la rápida mejora inicial de la heurística con la exploración amplia de la inicialización aleatoria. La inicialización aleatoria fue la más lenta en alcanzar óptimos debido a la alta variabilidad inicial, mientras que la heurística proporcionó soluciones de alta calidad rápidamente pero limitó la exploración.

(3.3.2) Calidad de las Soluciones.

- Expeimento 1. La selección por torneo produjo las mejores soluciones finales, debido a su capacidad para mantener una población de alta calidad a lo largo de las generaciones. La selección basada en el ranking también proporcionó buenas soluciones, aunque la calidad variaba según el equilibrio de las probabilidades de selección.
- Experimento 2. La inicialización híbrida proporcionó soluciones de alta calidad, ya que combinaba la diversidad necesaria para la exploración y la calidad inicial que facilitaba una rápida convergencia. La inicialización heurística también produjo buenas soluciones rápidamente, aunque a veces no exploró suficientemente el espacio de soluciones debido a su naturaleza más dirigida.

(3.3.3) Robustez y Estabilidad.

- Expeimento 1. La selección por torneo demostró ser la más robusta frente a variaciones en la población inicial, proporcionando soluciones consistentes. La selección basada en el ranking fue estable, pero su eficiencia dependía en gran medida de la distribución de los valores de fitness.
- Experimento 2. La inicialización híbrida mostró la mayor robustez, combinando ventajas de ambos extremos de la exploración y la explotación. La inicialización aleatoria fue menos robusta debido a la alta variabilidad en las soluciones iniciales, mientras que la heurística fue más estable pero a veces demasiado conservadora.

(3.3.3) Implicaciones Prácticas.

- Selección por Torneo y Inicialización Híbrida. Son los métodos más efectivos para la optimización del TSP con algoritmos genéticos, combinando exploración eficiente y rápida convergencia hacia soluciones óptimas.
- Inicialización Aleatoria. Es útil para mantener la diversidad genética, aunque puede requerir más generaciones para alcanzar soluciones óptimas.
- Inicialización Heurística. proporciona soluciones iniciales de alta calidad que pueden acelerar la convergencia inicial, pero debe complementarse con métodos que aseguren una exploración adecuada del espacio de soluciones.

Para aplicaciones prácticas, se recomienda el uso de inicialización híbrida combinada con selección por torneo para lograr una combinación óptima de calidad y eficiencia en la resolución de problemas complejos con algoritmos genéticos. Esta combinación aprovecha tanto la exploración amplia como la explotación de soluciones prometedoras, resultando en mejores soluciones en menos generaciones.

- Expeimento 1. La selección por torneo demostró ser la más robusta frente a variaciones en la población inicial, proporcionando soluciones consistentes. La selección basada en el ranking fue estable, pero su eficiencia dependía en gran medida de la distribución de los valores de fitness.
- Experimento 2. La inicialización híbrida mostró la mayor robustez, combinando ventajas de ambos extremos de la exploración y la explotación. La inicialización aleatoria fue menos robusta debido a la alta variabilidad en las soluciones iniciales, mientras que la heurística fue más estable pero a veces demasiado conservadora.

4. Anexo

4.1. Link github

Link github: `https://github.com/GKECV/Clases.git`

4.2. Resultados Generales

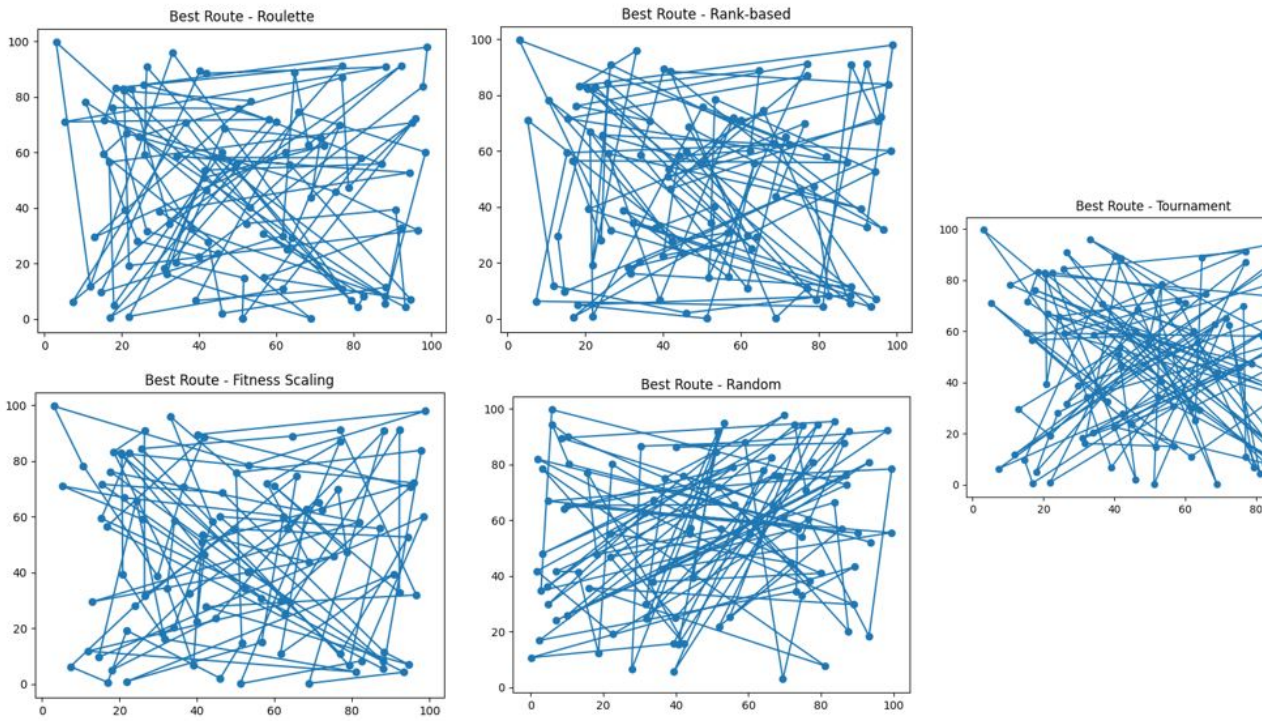

Figura 1: Experimento 1 - Mejores rutas

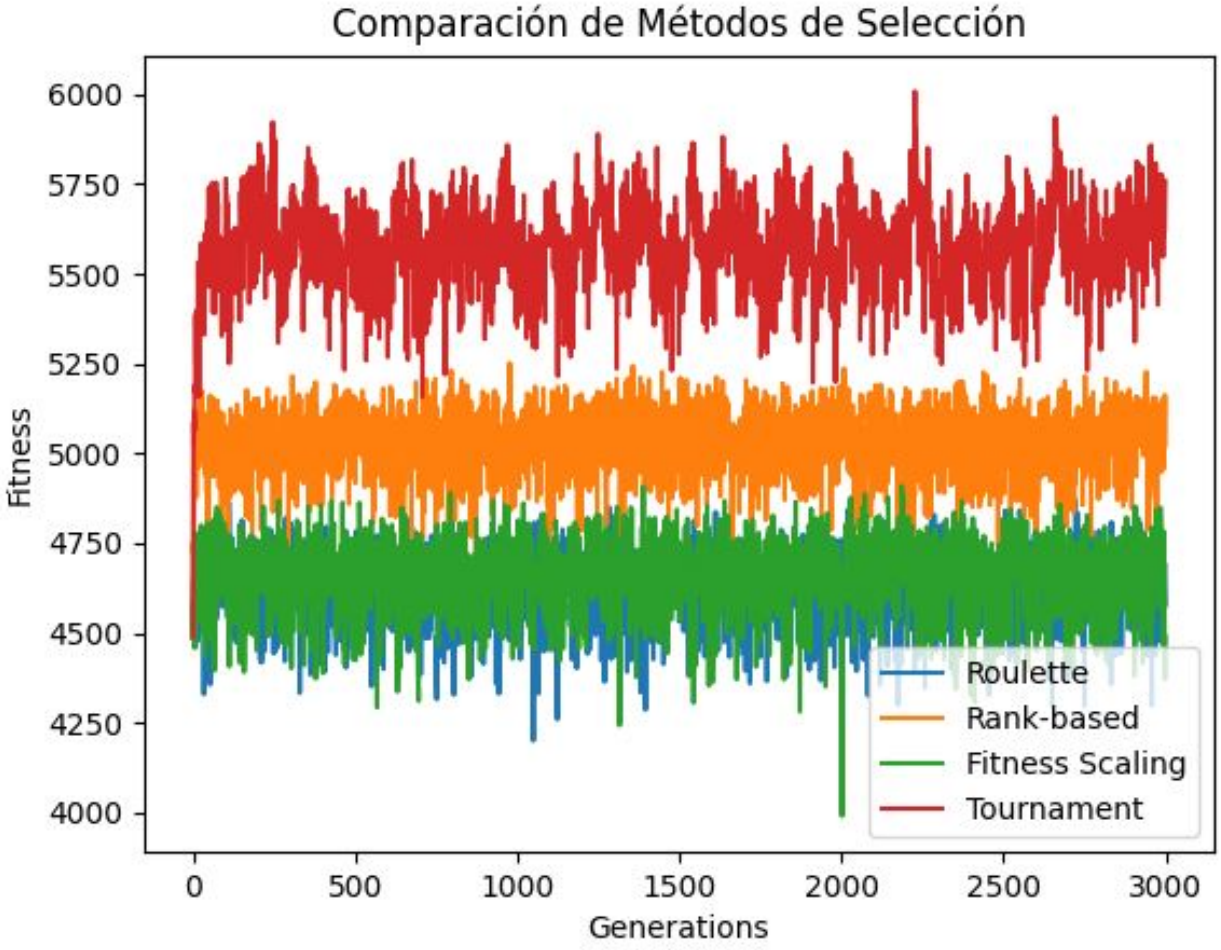

Figura 2: Experimento 1

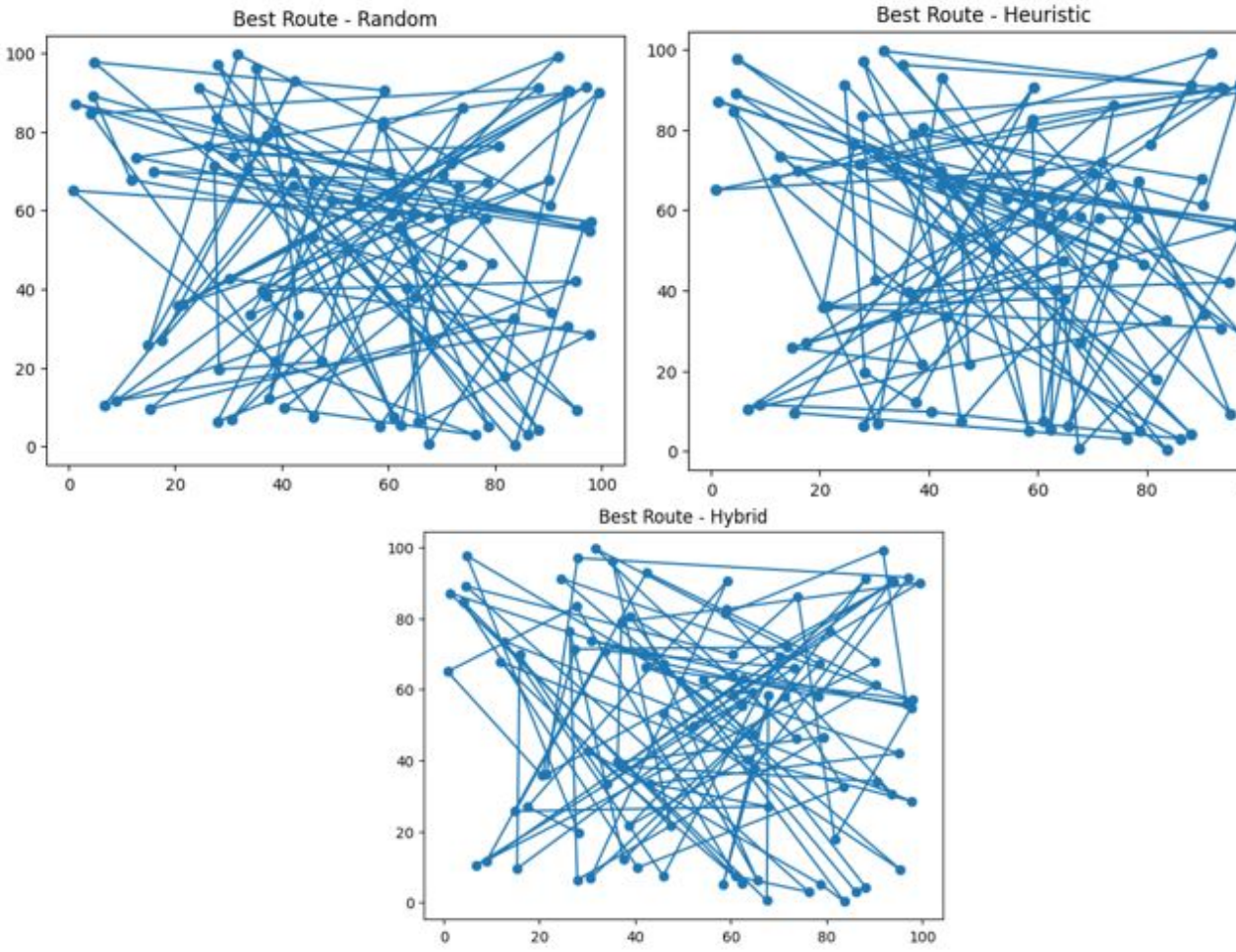

Figura 3: Experimento 2 - Mejores rutas

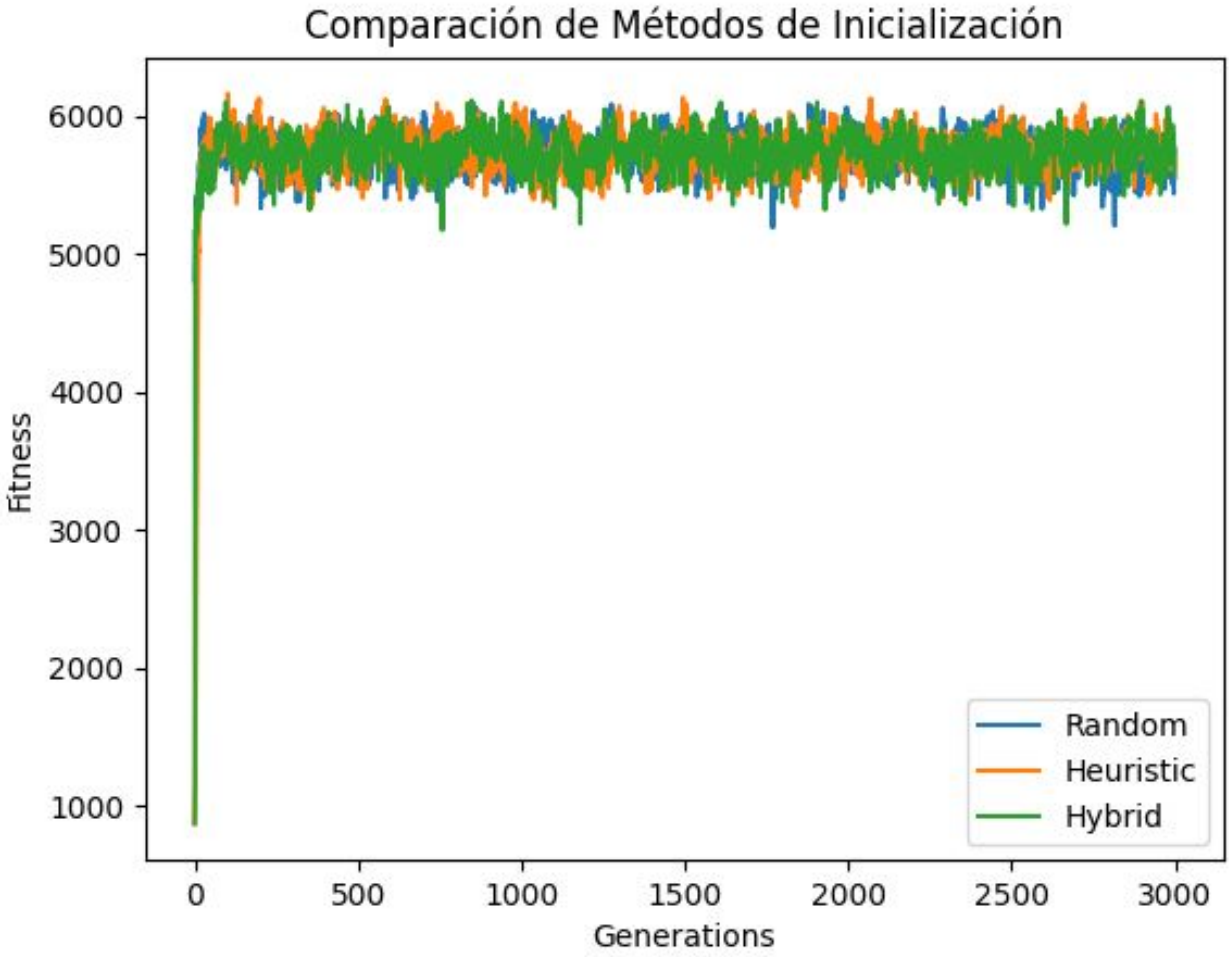

Figura 4: Experimento 2