

Optimización del Problema del Viajante mediante Algoritmos Genéticos: Comparación de Métodos de Selección y Inicialización

Silupú Peñaranda Rodrigo

June 8, 2024

Abstract

El Problema del Viajante (TSP) es un desafío clásico en la optimización combinatoria, donde se busca la ruta más corta que permita visitar todas las ciudades una vez y regresar al punto de partida. Este trabajo detalla la implementación de un GA para resolver el TSP. La implementación incluye inicialización de la población, cálculo de aptitud, selección, cruce y mutación, y la ejecución del algoritmo a lo largo de múltiples generaciones. Se comparan métodos de selección como ruleta, rango y torneo, encontrando que el método de torneo ofrece un mejor rendimiento en términos de optimización de rutas. Esta investigación demuestra la eficacia de los GA en la resolución de problemas complejos de optimización.

1 Introducción

En el ámbito de la optimización y búsqueda de soluciones, los algoritmos genéticos han demostrado ser herramientas efectivas debido a su capacidad para explorar vastos espacios de solución y encontrar óptimos globales. Estos algoritmos se inspiran en los principios de la evolución natural y utilizan mecanismos como la selección, cruce y mutación para generar nuevas soluciones a partir de las existentes.

El presente trabajo se enfoca en la implementación y comparación de diferentes métodos de selección y inicialización de población en algoritmos genéticos aplicados a un problema de optimización de rutas en un conjunto de 100 ciudades. En particular, se llevarán a cabo dos experimentos principales:

- Experimento 1: Implementación y comparación de la solución obtenida usando los métodos de selección: *Roulette wheel selection*, *Rank-based selection*, *Fitness scaling* y *Tournament selection*.
- Experimento 2: Implementación y comparación de la solución obtenida usando los métodos de inicialización de población: *random*, *heuristic* y *hybrid initialization*.

Para cada experimento, se guardarán los valores de aptitud en intervalos específicos de generaciones y se generarán gráficos que mostrarán cómo cada método afecta la búsqueda de soluciones.

2 Implementación

En esta sección se detallará la implementación del código, el cual puede encontrarse en el siguiente [enlace](#).

El Problema del Viajante (TSP) es un problema clásico en el campo de la optimización combinatoria, donde el objetivo es encontrar la ruta más corta que permita a un viajante visitar todas las ciudades exactamente una vez y regresar al punto de partida. Este problema es NP-difícil, lo que lo hace un candidato ideal para ser abordado mediante Algoritmos Genéticos (GA).

La implementación del GA para el TSP se basa en la clase **'GeneticAlgorithmTSP'**, que encapsula los componentes principales del algoritmo. Esta clase se inicializa con una lista de ciudades, el tamaño de la población, el número de generaciones, la tasa de mutación, el método de selección y el método de inicialización de la población. Los componentes del algoritmo son los siguientes:

- Inicialización de la Población: La función **'initialize_population'** genera la población inicial. En el primer experimento, cada individuo se crea mediante permutaciones aleatorias de las ciudades. En el segundo experimento, se introducen métodos de inicialización adicionales: aleatoria (*random*), heurística (*heuristic*) y una combinación de ambas (*hybrid*).
- Cálculo de Aptitud: La función **'fitness'** calcula la distancia total de un recorrido, sumando las distancias euclidianas entre ciudades consecutivas y cerrando el ciclo al regresar a la ciudad de origen.

- Selección: La función **'select'** elige a los padres para la generación siguiente. Se implementan métodos de selección como la ruleta, el rango y el torneo, con el torneo siendo el método utilizado en el segundo experimento.
- Cruce y Mutación: El método **'crossover'** combina partes de dos padres para generar un nuevo individuo, mientras que la función **'mutate'** introduce variaciones aleatorias en los individuos para mantener la diversidad genética.
- Ejecución del Algoritmo: El método **'run'** ejecuta el GA a lo largo de múltiples generaciones. En cada generación, se evalúan las aptitudes de los individuos, se seleccionan padres, se generan nuevos individuos mediante cruce y mutación, y se registra la mejor aptitud alcanzada.

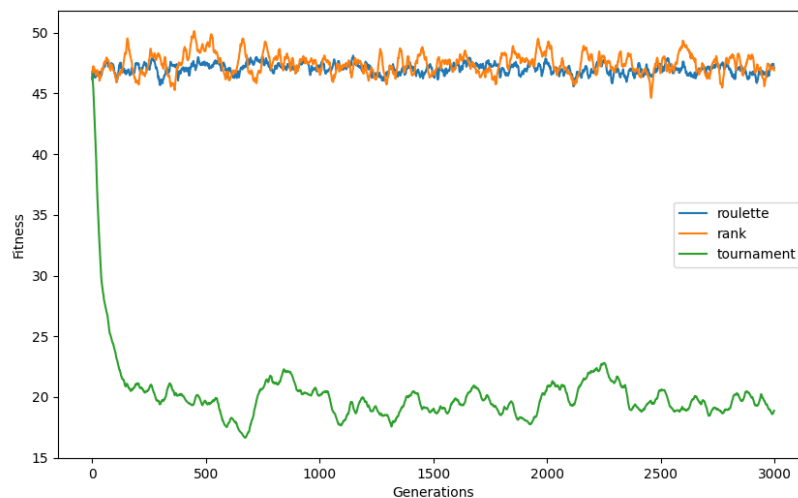
3 Resultados

3.1 Experimento 1: Comparación de Métodos de Selección

En el primer experimento, se comparan tres métodos de selección: ruleta, rango y torneo. Cada método se evalúa utilizando una población de 100 individuos, a lo largo de 3000 generaciones y con una tasa de mutación del 1%. La función **'plot_fitness'** grafica la evolución de la aptitud a lo largo de las generaciones, permitiendo comparar visualmente el desempeño de cada método. Además, la función **'plot_route'** muestra la mejor ruta encontrada por cada método.

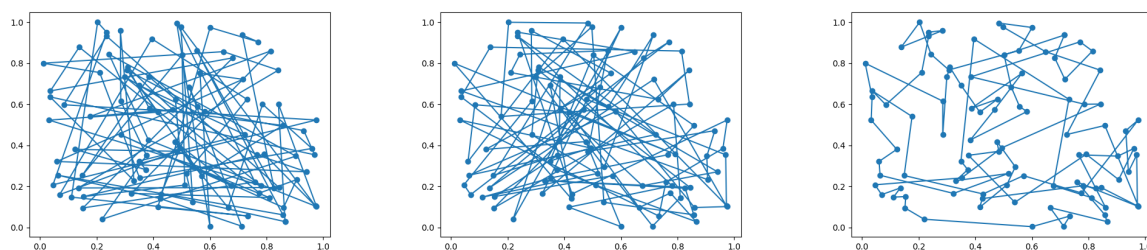
La siguiente gráfica muestra la evolución de la aptitud a lo largo de las generaciones para cada método de selección:

Figure 1: Comparación de la evolución de la aptitud para los métodos Ruleta, Rango y Torneo



Las mejores rutas encontradas por cada método son las siguientes:

Figure 2: Mejores rutas encontradas por los métodos Ruleta, Rango y Torneo



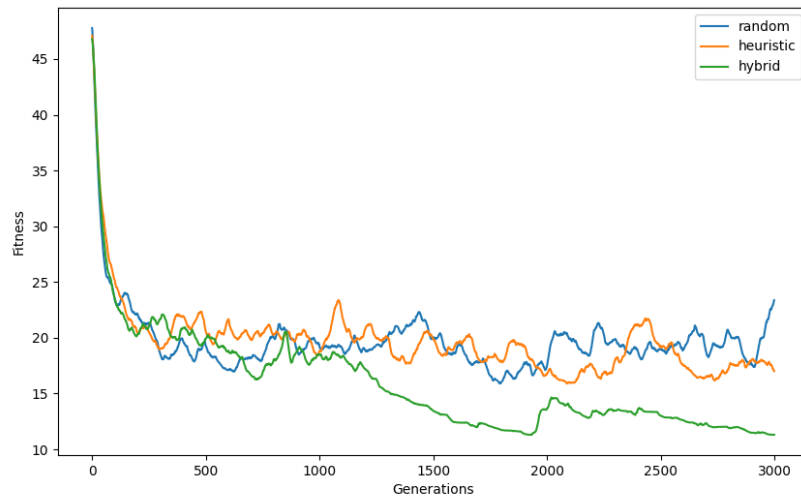
(a) Mejor ruta para el método Ruleta (b) Mejor ruta para el método Rango (c) Mejor ruta para el método Torneo

3.2 Experimento 2: Comparación de Métodos de Inicialización

En el segundo experimento, se comparan tres métodos de inicialización de la población: aleatoria, heurística y una combinación híbrida de ambos. Se utiliza una población de 100 individuos, una tasa de mutación del 1% y se ejecuta el algoritmo durante 3000 generaciones con selección por torneo. El método de inicialización heurística introduce un componente de conocimiento previo en la generación de los individuos, mientras que el método híbrido busca combinar los beneficios de la aleatoriedad y la heurística.

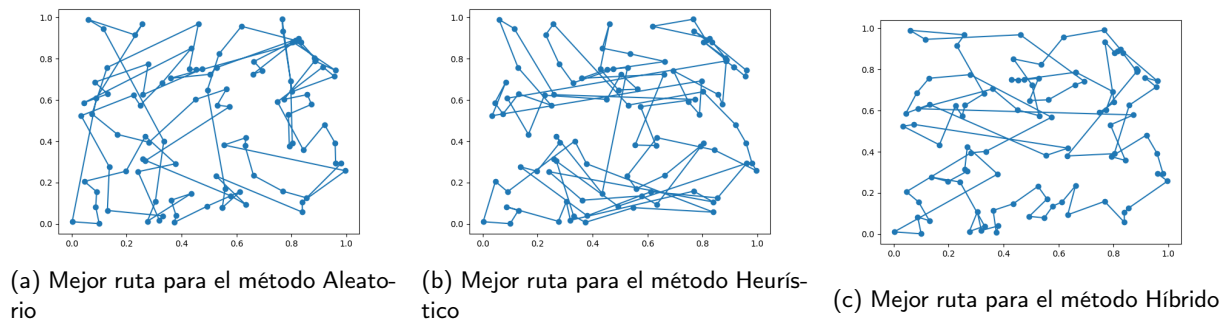
La siguiente gráfica muestra la evolución de la aptitud a lo largo de las generaciones para cada método de inicialización:

Figure 3: Comparación de la evolución de la aptitud para los métodos de inicialización Aleatorio, Heurístico e Híbrido



Las mejores rutas encontradas por cada método son las siguientes:

Figure 4: Mejores rutas encontradas por los métodos Aleatorio, Heurístico y Híbrido



4 Discusión

A continuación, se presenta un análisis detallado de ambos experimentos, destacando las características y el desempeño de cada método.

4.1 Experimento 1: Comparación de Métodos de Selección

A comparación de los modelos ruleta y rango, el método torneo presenta una convergencia más rápida y alcanza una aptitud final más baja en comparación con los métodos de ruleta y rango. Esto puede deberse a que los métodos de selección por ruleta y rango, aunque efectivos en ciertos contextos, pueden ser más susceptibles a problemas de diversidad y fluctuaciones en la aptitud. Por otro lado, el método de torneo, selecciona un subconjunto aleatorio (en este caso, de tamaño 5) y elige el individuo con la mejor aptitud dentro de ese

subconjunto. Esto proporciona una presión selectiva controlada, ya que solo los mejores individuos dentro de cada subconjunto tienen la oportunidad de ser seleccionados.

4.2 Experimento 2: Comparación de Métodos de Inicialización

Se observa que el método híbrido logra la aptitud más baja, seguido por el heurístico y finalmente el aleatorio. Esto puede deberse a que el método híbrido combina la diversidad introducida por la aleatoriedad con la dirección proporcionada por las heurísticas. En este sentido, el método híbrido puede explotar de manera más efectiva el espacio de búsqueda, logrando una convergencia eficiente y robusta.