



# TECNOLÓGICO NACIONAL DE MÉXICO



## Tópicos de IA

### Modulo III TAREA DE VALIDACIÓN

08/11/2025

Repository: <https://github.com/DiesukaLoL10/TOPICOS-DE-IA>

Gomez Gastelum Diego Alejandro  
Meza Manjarrez Marco Josue

## 1. Descripción del Proyecto

El Problema del Vendedor Viajero (TSP) pregunta: Dada una lista de ciudades y las distancias entre cada par de ciudades, ¿cuál es la ruta más corta posible que visita cada ciudad exactamente una vez y regresa a la ciudad de origen?

El TSP pertenece a la clase de problemas **NP-hard** (No-determinista Polinomialmente difícil).

- Para ciudades, el número de posibles rutas únicas es .
- Este valor crece factorialmente, lo que significa que a partir de aproximadamente 20 o 30 ciudades, el tiempo necesario para encontrar la solución óptima mediante fuerza bruta se vuelve inviable incluso para las supercomputadoras más potentes. Por ejemplo, para 50 ciudades, el número de rutas supera el .

## 2. Justificación del Uso del AG

Se elige el Algoritmo Genético (AG) porque es una metaheurística de optimización robusta, inspirada en la selección natural y la genética. Está diseñado para encontrar soluciones de alta calidad (cercanas al óptimo global) en un tiempo razonable para problemas NP-hard donde la solución exacta es computacionalmente prohibitiva.

- El AG trabaja bien con problemas de ordenación (permutación) como el TSP.
- A diferencia de los métodos constructivos (como la heurística del vecino más cercano), el AG explora un gran espacio de búsqueda simultáneamente (a través de la población) y es menos propenso a quedarse atascado en óptimos locales

## 3. Diseño del Algoritmo

### Representación

- **Genotipo:** Un individuo (una ruta) se representa como una permutación (una lista ordenada) de las ciudades. Por ejemplo, si hay 5 ciudades (), una ruta válida es .

### Evaluación de Aptitud

- **Cálculo de Distancia:** La distancia total de una ruta se calcula sumando la Distancia Euclíadiana entre ciudades adyacentes, incluyendo la distancia entre la última ciudad y la primera para cerrar el ciclo (ruta cíclica).

$$\text{Distancia Total} = \sum_{i=0}^{N-1} \text{Distancia}(\text{Ciudad}_i, \text{Ciudad}_{(i+1) \text{ (mod } N)})$$

- **Función de Aptitud:** Dado que el objetivo es minimizar la distancia, la aptitud se establece para maximizar su inverso, ya que los AG buscan maximizar la función objetivo.

$$\text{Aptitud} = \frac{1}{\text{Distancia Total}}$$

## Operadores Genéticos Clave

Selección por Elitismo y Ruleta:

- **Elitismo:** Los mejores individuos de la generación actual (definidos por tam\_elite) pasan directamente a la siguiente generación para asegurar que la mejor solución no se pierda.
- **Ruleta:** El resto de la población para el apareamiento se selecciona probabilísticamente: cuanto mayor es la aptitud de una ruta, mayor es la porción que ocupa en la "ruleta" y, por lo tanto, mayor es su probabilidad de ser seleccionada como padre.

**Crossover Ordenado (OX):** Este es el operador de recombinación estándar para el TSP.

1. Se elige aleatoriamente un segmento de la ruta del Padre 1.
2. Este segmento se copia directamente al hijo.
3. Los genes restantes del hijo se llenan con las ciudades del Padre 2 en el orden en que aparecen en el Padre 2, omitiendo las ciudades que ya fueron copiadas del Padre 1.

*Propósito:* Garantiza que el hijo siempre sea una permutación válida (cada ciudad aparece solo una vez).

## Mutación por Intercambio (*Swap Mutation*):

- Con una probabilidad baja (tasa\_mutacion), el operador selecciona dos posiciones aleatorias en la ruta de un individuo y simplemente intercambia las ciudades en esas posiciones.

## 4. Conclusiones

El proyecto ha demostrado la implementación exitosa de un Algoritmo Genético (AG) como una metaheurística efectiva para abordar el Problema del Vendedor Viajero (TSP).

El AG es una solución práctica y eficiente para problemas de optimización de ordenación (NP-hard) donde la búsqueda de la solución óptima exacta es computacionalmente inviable debido al crecimiento factorial de las posibilidades.

## 5. Resultados:

```
ejecutar.py
Distancia inicial: 581.7112
Generación 50 / 500 - Mejor distancia: 317.7234
Generación 100 / 500 - Mejor distancia: 317.7234
Generación 150 / 500 - Mejor distancia: 317.7234
Generación 200 / 500 - Mejor distancia: 317.7234
Generación 250 / 500 - Mejor distancia: 317.7234
Generación 300 / 500 - Mejor distancia: 317.7234
Generación 350 / 500 - Mejor distancia: 317.7234
Generación 400 / 500 - Mejor distancia: 317.7234
Generación 450 / 500 - Mejor distancia: 317.7234
Generación 500 / 500 - Mejor distancia: 317.7234
Distancia final: 317.723405260005
Mejor distancia encontrada: 317.723405260005
```

```
Distancia inicial: 684.1154
Generación 50 / 500 - Mejor distancia: 391.8115
Generación 100 / 500 - Mejor distancia: 391.8115
Generación 150 / 500 - Mejor distancia: 391.8115
Generación 200 / 500 - Mejor distancia: 391.8115
Generación 250 / 500 - Mejor distancia: 391.8115
Generación 300 / 500 - Mejor distancia: 391.8115
Generación 350 / 500 - Mejor distancia: 391.8115
Generación 400 / 500 - Mejor distancia: 391.8115
Generación 450 / 500 - Mejor distancia: 391.8115
Generación 500 / 500 - Mejor distancia: 391.8115
Distancia final: 391.8115
```

Distancia inicial: 684.1154  
Generación 6 / 200 - Mejor distancia: 568.3101  
Generación 12 / 200 - Mejor distancia: 502.4464  
Generación 18 / 200 - Mejor distancia: 502.4464  
Generación 24 / 200 - Mejor distancia: 420.0708  
Generación 30 / 200 - Mejor distancia: 447.0615  
Generación 36 / 200 - Mejor distancia: 425.5214  
Generación 42 / 200 - Mejor distancia: 409.4624  
Generación 48 / 200 - Mejor distancia: 391.8115  
Generación 54 / 200 - Mejor distancia: 391.8115  
Generación 60 / 200 - Mejor distancia: 391.8115  
Generación 66 / 200 - Mejor distancia: 391.8115  
Generación 72 / 200 - Mejor distancia: 391.8115  
Generación 78 / 200 - Mejor distancia: 391.8115  
Generación 84 / 200 - Mejor distancia: 391.8115  
Generación 90 / 200 - Mejor distancia: 391.8115  
Generación 96 / 200 - Mejor distancia: 391.8115  
Generación 102 / 200 - Mejor distancia: 391.8115  
Generación 108 / 200 - Mejor distancia: 391.8115  
Generación 114 / 200 - Mejor distancia: 391.8115  
Generación 120 / 200 - Mejor distancia: 391.8115  
Generación 126 / 200 - Mejor distancia: 391.8115  
Generación 132 / 200 - Mejor distancia: 391.8115  
Generación 138 / 200 - Mejor distancia: 391.8115  
Generación 144 / 200 - Mejor distancia: 391.8115  
Generación 150 / 200 - Mejor distancia: 391.8115  
Generación 156 / 200 - Mejor distancia: 391.8115  
Generación 162 / 200 - Mejor distancia: 391.8115  
Generación 168 / 200 - Mejor distancia: 391.8115  
Generación 174 / 200 - Mejor distancia: 391.8115  
Generación 180 / 200 - Mejor distancia: 391.8115  
Generación 186 / 200 - Mejor distancia: 391.8115  
Generación 192 / 200 - Mejor distancia: 391.8115  
Generación 198 / 200 - Mejor distancia: 391.8115  
Distancia final: 391.8115

Programado en Java con el algoritmo ACO de la librería JACOPE.