

# Informe sobre la solución planteada sobre el "Problema del Agente Viajero"

Luis Antonio Mamani Alave

21 de diciembre de 2023

## 1 Introducción

El problema del agente viajero es un desafío clásico en el área de la optimización combinatoria y a su vez con diversas aplicaciones prácticas. El problema consiste en encontrar la ruta más óptima que permita a un visitante visitar un conjunto de ubicaciones y regresar al punto de partida. Aunque parece sencillo su complejidad radica en todas las posibles combinaciones de rutas, que aumenta de manera exponencial con el número de ubicaciones planteadas.

Para dar solución a este problema se han planteado distintos tipos de algoritmos como los algoritmos exactos que buscan encontrar una solución óptima, evaluando todas las posibles combinaciones. Sin embargo, a medida que aumenta el número de ubicaciones, la complejidad computacional se vuelve desproporcionada. Por otro lado, también tenemos los enfoques heurísticos y metaheurísticos, a través de algoritmos genéticos, búsqueda de cuervos entre otros, que ofrecen soluciones aproximadas en tiempos aunque no garantizan la solución óptima pero sí aceptable.

## 2 Objetivos

El objetivo del trabajo fue, plantear un algoritmo que brinde una solución al problema del agente viajero, utilizando enfoques heurísticos que nos permiten obtener una solución aceptable. Para este caso utilizamos un algoritmo genético.

### 3 Algoritmo

---

**Algorithm 2:** Mi algoritmo

---

**Data:** ciudades, poblacion, generaciones  
1 Generamos las rutas aleatorias;  
2 **for**  $i \leftarrow 1$  **to** *generaciones* **do**  
3     Ordenamos las rutas en base a su distancia;  
4     Generamos la elite, padres, descendencia;  
5     **while** *descendencia* < *poblacion* - *elite* **do**  
6         Seleccionamos aleatoriamente dos padres;  
7         Generamos un hijo;  
8         **if** *probabilidad de mutación* **then**  
9             Mutamos al hijo;  
10        **end**  
11        Agregamos al hijo a la descendencia;  
12     **end**  
13     *poblacion* = *padres* + *descendencia*;  
14 **end**  
15 Seleccionamos la mejor ruta;  
**Result:** Devolvemos la mejor ruta

---

### 4 Implementacion

### 5 Resultados

Para poder probar el algoritmo planteado utilizamos 3 grupos de ciudades. El primero formado por 5 ciudades, el segundo formado por 10 ciudades y finalmente el tercero formado por 20 ciudades.

Cuadro 1: Primer Grupo

| Ciudad      | A   | B   | C   | D   | E   |
|-------------|-----|-----|-----|-----|-----|
| Coordenadas | 0,0 | 1,2 | 3,1 | 5,2 | 3,4 |

Cuadro 2: Segundo Grupo

| Ciudad      | A   | B   | C   | D   | E   | F   | G   | H   | I    | J   |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|------|-----|
| Coordenadas | 0,0 | 1,2 | 3,1 | 5,2 | 3,4 | 8,7 | 6,5 | 1,8 | 3,10 | 7,9 |

Cuadro 3: Tercer Grupo

| Ciudad      | A    | B    | C    | D     | E     | F    | G    | H    | I    | J   |
|-------------|------|------|------|-------|-------|------|------|------|------|-----|
| Coordenadas | 0,0  | 1,2  | 3,1  | 5,2   | 3,4   | 8,7  | 6,5  | 1,8  | 3,10 | 7,9 |
| Ciudad      | K    | L    | M    | N     | O     | P    | Q    | R    | S    | T   |
| Coordenadas | 0, 1 | 5, 6 | 3, 9 | 5, 10 | 3, 11 | 8, 2 | 6, 1 | 1, 5 | 2, 6 | 4,7 |

## 5.1 Resultados del primer grupo

Para el primer grupo se realizaron 10 iteraciones sobre un conjunto de 5 ubicaciones, dando como resultados los datos obtenidos en el Cuadro 4.

Cuadro 4: Resultados de procesamiento del Primer Grupo

| Nro | P. Inicio | Orden de Visita | Distancia | Tiempo (s) |
|-----|-----------|-----------------|-----------|------------|
| 1   | B: [1, 2] | B, E, D, C, A   | 13.2913   | 0.8596     |
| 2   | E: [3, 4] | E, B, A, C, D   | 13.2913   | 0.8812     |
| 3   | D: [5, 2] | D, E, B, A, C   | 13.2913   | 0.8755     |
| 4   | A: [0, 0] | A, B, E, D, C   | 13.2913   | 0.8788     |
| 5   | D: [5, 2] | D, C, A, B, E   | 13.2913   | 0.9102     |
| 6   | A: [0, 0] | A, C, D, E, B   | 13.2913   | 0.9030     |
| 7   | D: [5, 2] | D, C, A, B, E   | 13.2913   | 0.9091     |
| 8   | E: [3, 4] | E, D, C, A, B   | 13.2913   | 0.9461     |
| 9   | E: [3, 4] | E, D, C, A, B   | 13.2913   | 0.8779     |
| 10  | E: [3, 4] | E, B, A, C, D   | 13.2913   | 1.1982     |

Como podemos observar en el Cuadro 4, al ser el número de ubicaciones a visitar solo 5 todas las iteraciones muestran la misma distancia aceptable, a pesar de que el recorrido se realice de distinta manera, en cuanto al tiempo de procesamiento este tiene una media de 0.9240 segundos, mientras que la distancia óptima en general es de 13.2913. Así como se muestra en la Figura 1 donde podemos ver la evolución de la mejor distancia en la quinta iteración, en donde la distancia no varía.

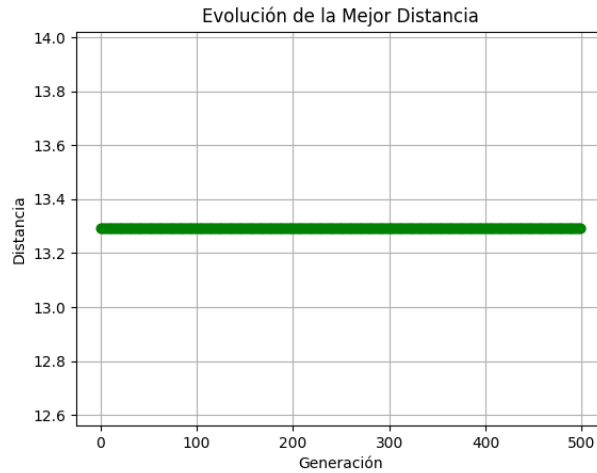


Figura 1: Evolución de la mejor distancia en la quinta iteración.

En cuanto a la ruta óptima como apreciamos en la Figura 2, esta no muestra ninguna variación con la de las demás iteraciones a pesar de iniciar de un punto distinto en cada caso.

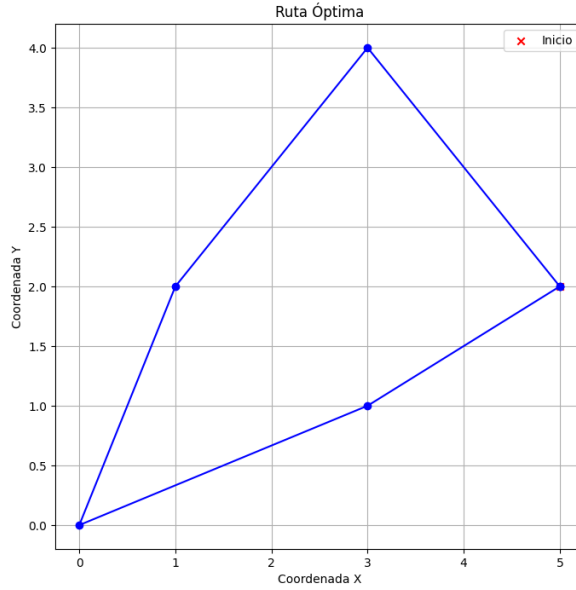


Figura 2: Ruta óptima de la quinta iteración.

## 5.2 Resultados de procesamiento del Segundo Grupo

Como observamos en el Cuadro 5, para el segundo grupo se realizaron 10 iteraciones, en donde si encontramos variaciones, ya que el conjunto de datos es de 10 ciudades, haciendo mas compleja la busqueda de una ruta aceptable haciendo el resultado variable.

Cuadro 5: Resultados de procesamiento del Segundo Grupo

| Nro | P. Inicio  | Orden de Visita              | Distancia | Tiempo (s) |
|-----|------------|------------------------------|-----------|------------|
| 1   | D: [5, 2]  | D, G, F, J, I, H, E, B, A, C | 30.1133   | 1.4898     |
| 2   | F: [8, 7]  | F, G, D, C, A, B, E, H, I, J | 30.1133   | 1.4713     |
| 3   | J: [7, 9]  | J, I, H, B, A, C, D, E, G, F | 31.6411   | 1.3995     |
| 4   | B: [1, 2]  | B, A, C, D, E, G, F, J, I, H | 31.6411   | 1.3410     |
| 5   | F: [8, 7]  | F, J, I, H, E, B, A, C, D, G | 30.1133   | 1.3133     |
| 6   | E: [3, 4]  | E, B, A, C, D, G, F, J, I, H | 30.1133   | 1.4520     |
| 7   | A: [0, 0]  | A, C, D, G, F, J, I, H, E, B | 30.1133   | 1.4499     |
| 8   | B: [1, 2]  | B, E, H, I, J, F, G, D, C, A | 30.1133   | 1.3315     |
| 9   | I: [3, 10] | I, H, G, E, B, A, C, D, F, J | 34.4746   | 1.4359     |
| 10  | A: [0, 0]  | A, C, D, G, F, J, I, H, E, B | 30.1133   | 1.3199     |

En el cuadro anterior observamos como la distancia optima para realizar el recorrido, asi como el tiempo de procesamiento han variado, ya que esta vez las ciudades a recorrer son 10, siento la mejor distancia aceptable 30.1133 y la maxima distancia aceptable 34.4746, con una media de procesamiento de 1.4004 segundos.

En cuanto a la evolucion de la mejor distancia podemos observar como esta va cambiando en cada generacion hasta llegar a un punto donde ya no mejora como podemos ver en la Figura 3.

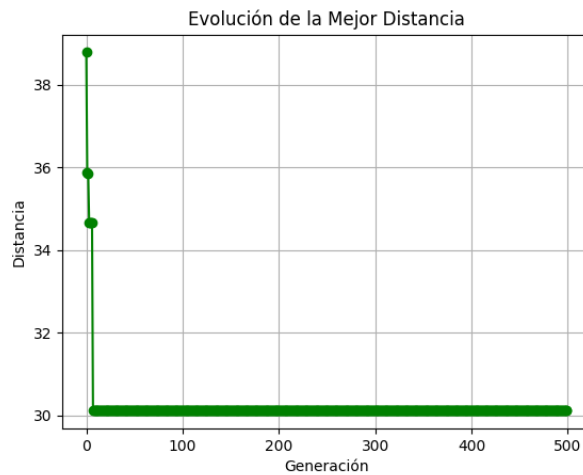


Figura 3: Evolución de la mejor distancia en la primera iteración.

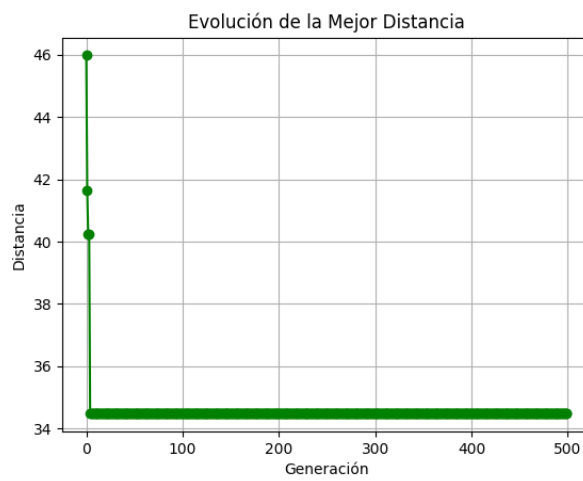


Figura 4: Evolución de la mejor distancia en la novena iteración.

Por ultimo en cuanto a la creacion de la ruta óptima podemos apreciar como esta al aumentar el numero de ciudades a visitar sufre variaciones en busqueda de la mejor ruta posible.

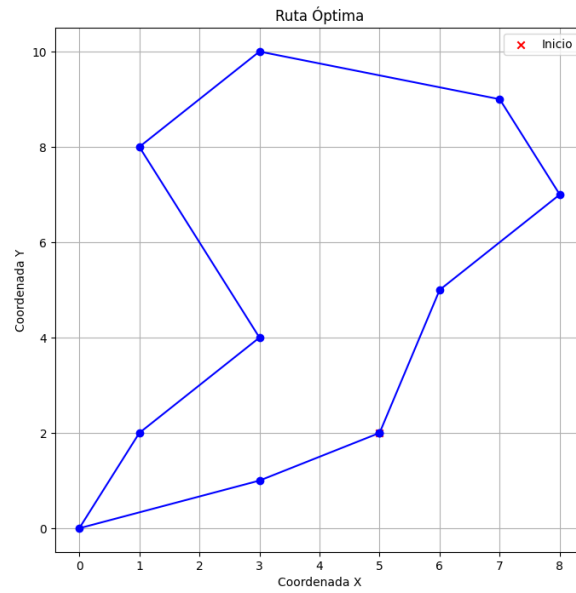


Figura 5: Ruta óptima de la primera iteración.

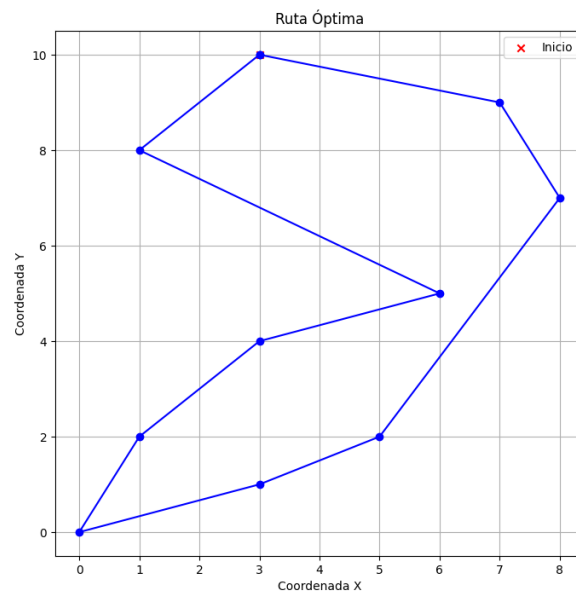


Figura 6: Ruta óptima de la novena iteración.

### 5.3 Resultados de procesamiento del Tercer Grupo

Por ultimo, para el tercer grupo de 20 ciudades tambien se realizaron 10 iteraciones en donde la dificultad para encontrar la ruta optima entre las generaciones se elevo, dando los resultados obtenidos en el Cuadro 6.

Cuadro 6: Resultado de procesamiento del Tercer Grupo

| Nro | P. Inicio  | Orden de Visita  | Distancia | Tiempo (s) |
|-----|------------|--|-----------|------------|
| 1   | T: [4, 7]  | T, D, Q, C, A, K, B, E, R, S, H, N, J, F, P, G, L, O, I, M | 52.3898   | 2.6521     |
| 2   | M: [3, 9]  | M, S, R, H, O, I, T, L, G, P, Q, D, C, A, K, B, E, F, J, N | 48.6087   | 2.7606     |
| 3   | H: [1, 8]  | H, M, I, O, N, G, P, Q, D, C, B, R, S, T, J, F, L, E, A, K | 54.8528   | 2.7292     |
| 4   | Q: [6, 1]  | Q, C, A, K, B, R, S, H, T, L, G, F, J, N, O, I, M, E, D, P | 45.8186   | 2.6659     |
| 5   | R: [1, 5]  | R, B, K, A, C, G, L, T, M, I, O, N, J, F, P, Q, D, E, S, H | 46.3000   | 2.6411     |
| 6   | S: [2, 6]  | S, H, O, I, M, T, N, J, F, G, L, E, B, K, A, C, Q, P, D, R | 47.8384   | 2.5960     |
| 7   | T: [4, 7]  | T, L, G, Q, P, F, J, N, O, I, M, H, S, R, E, D, C, A, K, B | 47.3671   | 2.6121     |
| 8   | S: [2, 6]  | S, T, L, G, P, Q, D, C, A, K, B, E, F, J, N, O, I, M, H, R | 44.1508   | 2.5910     |
| 9   | F: [8, 7]  | F, J, N, O, I, M, L, G, P, Q, D, C, A, K, B, E, R, S, H, T | 44.6734   | 2.5761     |
| 10  | I: [3, 10] | I, O, N, J, F, L, G, P, Q, D, C, A, K, B, E, R, S, H, T, M | 42.4662   | 2.6297     |

Como observamos en el cuadro anterior, la busqueda de la mejor ruta varia notablemente frente a los grupos anteriores, siendo la mejor distancia optima la de 42.4662, mientras que por otro lado tenemos 54.8528 como la distancia mas larga, con una media de procesamiento de 2.6609 segundos.

En cuanto a la evolucion de la mejor distancia en la Figura podemos apreciar como esta varia al paso de cada generacion en busqueda de una distancia aceptable como solucion al problema.

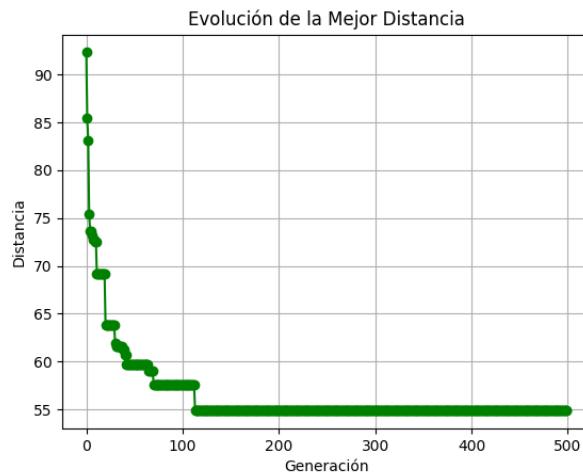
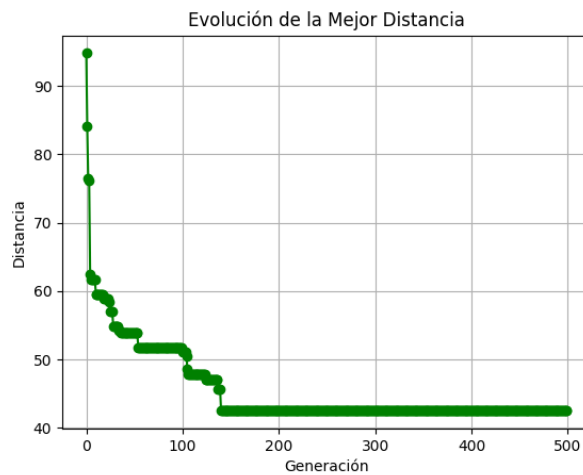


Figura 7: Evolución de la mejor distancia en la tercera iteración.





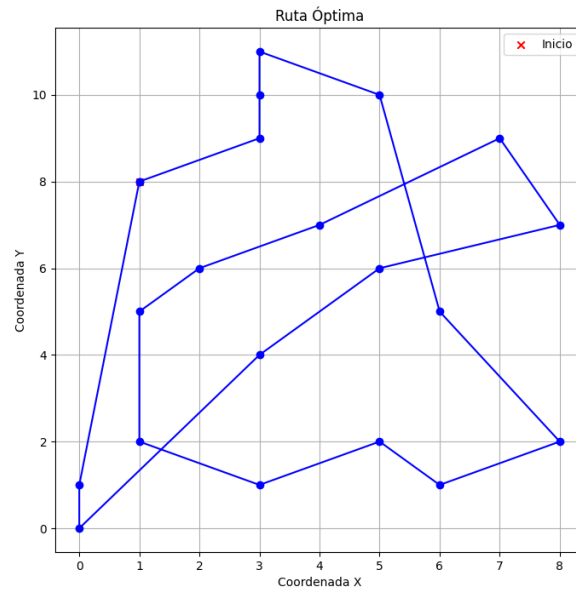


Figura 9: Evolución de la mejor distancia en la tercera iteración.

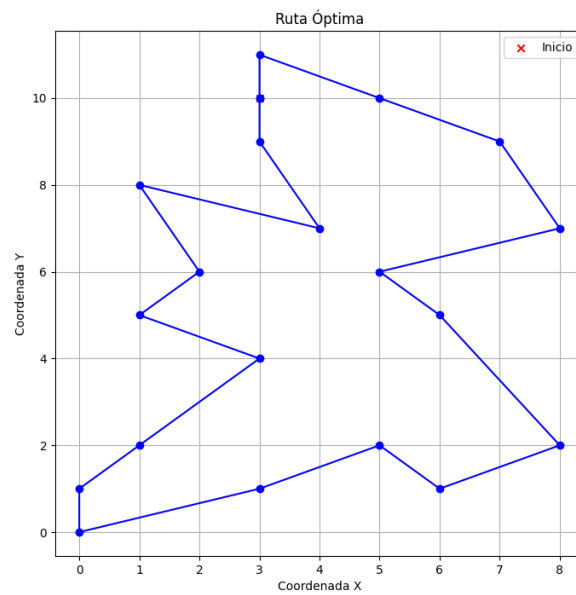


Figura 10: Evolución de la mejor distancia en la decima iteración.

## 6 Conclusiones

- Como podemos apreciar de los resultados obtenidos en cada uno de los grupos de prueba es distinto, siendo que los grupos de con mas ciudades los que muestran mayor opciones rutas aceptables como solución al problema.
- El aumentar el numero de generacion al procesamiento del algoritmo no garantiza encontrar la mejor solución.
- La aplicación de un algoritmo genético logra brindar una solucion al problema con un costo computacional aceptable, a diferencia de un algoritmo exacto.