

DESCRIPCION DEL PROBLEMA

El Problema del Agente Viajero (TSP) consiste en encontrar la ruta más corta que pasa por todas las ciudades exactamente una vez y regresa al punto de partida. En esta versión del problema, se busca resolver la "Ruta de las Metrópolis", donde las ciudades están representadas como nodos en el mapa y las distancias entre ellas se definen en la Tabla 1 (distancias en millas).



El problema será realizado tomando en cuenta la matriz de distancias proporcionada, siendo un problema simétrico ya que cuesta lo mismo ir que regresar:

Ciudad	New York	Los Angeles	Chicago	Houston	Phoenix	Philadelphia	San Diego	Dallas	San Francisco	Austin	Las Vegas
1. New York	-	3091	927	1876	2704	94	2999	1641	3471	1838	3013
2. Los Angeles	3091	-	2542	1681	375	2994	138	1442	389	1407	290
3. Chicago	927	2542	-	1337	2169	930	2464	1100	2935	1168	2465
4. Houston	1876	1681	1337	-	1308	1778	1603	240	2075	163	1604
5. Phoenix	2704	375	2169	1308	-	2603	366	1069	767	1034	296
6. Philadelphia	94	2994	930	1778	2603	-	2898	1543	3369	1740	2915
7. San Diego	2999	138	2464	1603	366	2898	-	1364	531	1329	338
8. Dallas	1641	1442	1100	240	1069	1543	1364	-	1836	201	1365
9. San Francisco	3471	389	2935	2075	767	3369	531	1836	-	1801	607
10. Austin	1838	1407	1168	163	1034	1740	1329	201	1801	-	1330
11. Las Vegas	3013	290	2465	1604	296	2915	338	1365	607	1330	-

## **EXPLICACION DEL PROBLEMA**

El problema que abordaremos es conocido como el Problema del Agente Viajero (TSP), específicamente en su versión denominada de la tarea es: "Ruta de las Metrópolis". En este escenario, las ciudades se representan como nodos en un mapa, y las distancias entre ellas están predefinidas en una tabla, que se encuentra en la parte de arriba. Estas distancias se expresan en millas y son cruciales para calcular la longitud total del recorrido.

El objetivo principal es encontrar la ruta óptima que minimice la distancia total recorrida por el agente viajero, permitiéndole visitar cada ciudad una sola vez y finalmente regresar al punto de inicio. La solución a este problema tiene una amplia gama de aplicaciones en la optimización de rutas de transporte, logística, planificación urbana y más.

## **ALGORITMO GENETICO HIBRIDO (HGA)**

Para resolver este desafío, implementaremos un Algoritmo Genético Híbrido (HGA, por sus siglas en inglés). Este enfoque combina elementos de los algoritmos genéticos, que se inspiran en la evolución biológica, con heurísticas y estrategias de mejora local. Utilizaremos conceptos como la generación de poblaciones de rutas, la selección de padres, el cruce por "Edge Recombination", la remoción de abruptos y la evaluación de la aptitud de las rutas para guiar la búsqueda hacia soluciones cada vez mejores.

En el Paso 1, se genera aleatoriamente una población inicial de rutas que representan posibles soluciones al problema. Cada ruta es una permutación aleatoria de las ciudades disponibles. Luego, se evalúa la aptitud de cada individuo en la población inicial, es decir, se calcula la distancia total de cada ruta.

En el Paso 2, se aplica la heurística de "Remoción de Abruptos" a todas las rutas en la población inicial. Esta heurística busca mejorar las soluciones eliminando la conexión no óptima entre ciudades, utilizando el parámetro  $m$ .

En el Paso 3, se seleccionan aleatoriamente dos padres de la población actual. Luego, se aplica el cruce por "Edge Recombination (ER)" entre los padres seleccionados para generar un descendiente. El cruce por ER utiliza la información de los vecinos de las ciudades en los padres para generar una nueva ruta que combina características de ambos padres. Luego, se evalúa la aptitud del descendiente y se aplica la heurística de "Remoción de Abruptos" para mejorar su calidad.

En el Paso 4, los padres y el descendiente se ordenan según su aptitud, y los dos mejores individuos se seleccionan para conformar la siguiente generación.

En el Paso 5 se utiliza el parámetro de probabilidad de mutación que se aplicará a cada uno de los individuos, en caso de cumplirse dicho individuo se reemplazará completamente por otro nuevo. Después simplemente se repiten los pasos relevantes en el bucle general al final de código una cantidad definida de veces (200 en este caso).

## PARAMETROS USADOS

```
#===== PARAMETROS =====
m = 6
Nciudades = 11
MATRIZ_de_distancias = [
    [0, 3091, 927, 1876, 2704, 94, 2999, 1641, 3471, 1838, 3013],
    [3091, 0, 2542, 1681, 375, 2994, 138, 1442, 389, 1407, 290],
    [927, 2542, 0, 1337, 2169, 930, 2464, 1100, 2935, 1168, 2465],
    [1876, 1681, 1337, 0, 1308, 1778, 1603, 240, 2075, 163, 1604],
    [2704, 375, 2169, 1308, 0, 2603, 366, 1069, 767, 1034, 296],
    [94, 2994, 930, 1778, 2603, 0, 2898, 1543, 3369, 1740, 2915],
    [2999, 138, 2464, 1603, 366, 2898, 0, 1364, 531, 1329, 338],
    [1641, 1442, 1100, 240, 1069, 1543, 1364, 0, 1836, 201, 1365],
    [3471, 389, 2935, 2075, 767, 3369, 531, 1836, 0, 1801, 607],
    [1838, 1407, 1168, 163, 1034, 1740, 1329, 201, 1801, 0, 1330],
    [3013, 290, 2465, 1604, 296, 2915, 338, 1365, 607, 1330, 0]
]
Nindividuos = 50
genMax = 50
probabilidad_mutacion = 0.05
```

## TABLAS DE RESULTADOS

EJECUCION	VALOR FUNCION OBJ.	INDIVIDUO (RUTA)
1	7989	[8, 7, 2, 9, 11, 5, 6, 1, 3, 10, 4]
2	8442	[9, 2, 7, 5, 8, 10, 3, 6, 1, 4, 11]
3	8037	[5, 8, 3, 1, 6, 4, 10, 11, 2, 9, 7]
4	8130	[1, 6, 8, 10, 4, 7, 2, 9, 11, 5, 3]
5	7886	[8, 4, 10, 5, 11, 9, 2, 7, 6, 1, 3]
6	7895	[3, 1, 6, 8, 4, 10, 5, 11, 9, 2, 7]
7	7890	[11, 9, 2, 7, 10, 4, 6, 1, 3, 8, 5]
8	8345	[2, 11, 7, 8, 3, 6, 1, 4, 10, 5, 9]
9	8073	[8, 6, 1, 3, 10, 4, 7, 9, 2, 11, 5]
10	7961	[4, 10, 11, 9, 2, 7, 5, 8, 3, 1, 6]

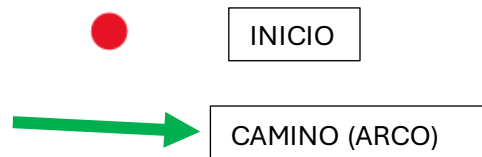
## TABLA DE ESTADISTICAS

MEJOR VALOR	VALOR PROMEDIO	DESVIACION ESTANDAR	PEOR VALOR
7886 (ejecución 5)	8064.8	182.69088647220474	8442

## MEJOR INDIVIDUO Y RUTA GRAFICA

Mejor individuo: [8, 4, 10, 5, 11, 9, 2, 7, 6, 1, 3]

Representación gráfica:



## DISCUSIÓN

En lo que se puede observar en cuanto a las estadísticas se puede asumir que el código es bastante estable dada la poca diferencia entre las soluciones dadas, todas dando soluciones bastante favorables juzgando por la función objetivo.

El peor resultado a pesar de que si se diferencia un poco del mejor, no representa una solución tan negativa ya que se parece al valor promedio (en este caso 8064.8). Con esta información puedo decir con confianza que el código da resultados estables con los parámetros establecidos al inicio del reporte.

En cuanto a la representación grafica de la mejor solución se puede deducir visualmente que la solución es bastante buena, ya que evita conexiones no optimas con la remoción de abruptos, cuyo parámetro  $m$  ayudo demasiado al código a obtener la mejor solución y evitar el estancamiento en óptimos locales, cumpliendo esta última función la mutación también.

También se puede deducir su efectividad al mirar las conexiones, donde en promedio siempre escoge la ruta mas corta posible y solo hay un camino que se consideraría largo, pero necesario dada la acumulación de ciudades lejanas en la parte superior izquierda (New York, Philadelphia y Detroit), por lo que me atrevo a decir que el algoritmo tuvo un buen desempeño.