

### **Introducción:**

Los Algoritmos Genéticos constituyen una rama de la Inteligencia Artificial enfocada en el desarrollo y aplicación de técnicas de optimización inspiradas en la teoría de la evolución biológica. Estos métodos permiten que entidades computacionales, referidas como "individuos" dentro de una población de soluciones, evolucionen a través de procesos de selección, cruce y mutación. Su objetivo principal es optimizar una función de aptitud específica, que, en el contexto de este proyecto, está relacionada con la minimización de la distancia o el tiempo de viaje entre dos puntos en un sistema complejo como lo es la red de metro. A través de iteraciones sucesivas o generaciones, estos algoritmos buscan explorar el espacio de soluciones de manera estocástica pero guiada, permitiendo descubrir soluciones que, aunque no siempre óptimas globalmente, pueden ser extremadamente efectivas bajo ciertas condiciones.

### **Conceptos Fundamentales:**

- Individuo: En el contexto de los algoritmos genéticos, se refiere a una solución potencial al problema planteado, que en este caso es una ruta específica a través de la red del metro. Cada individuo es evaluado por su capacidad para minimizar el tiempo de viaje entre dos estaciones determinadas.
- Población: Conjunto de individuos que representan diferentes soluciones posibles al problema. La población evoluciona generación tras generación hacia soluciones óptimas mediante procesos de selección, cruce y mutación.
- Gen: Componente de un individuo que, en este proyecto, podría interpretarse como una estación específica en la ruta del metro. La combinación de genes (estaciones) define la configuración del individuo (ruta).
- Cromosoma: Secuencia completa de genes en un individuo; en este contexto, es la secuencia completa de estaciones que forman una ruta a través del metro desde El Rosario hasta San Lázaro.
- Genoma: El genoma en un algoritmo genético representa el conjunto total de información genética que puede ser manipulada para explorar el espacio de soluciones del problema, el genoma sería el diseño completo de todas las posibles rutas que pueden ser creadas usando las estaciones disponibles en la red del metro.

Este incluye todos los genes posibles (estaciones) y la manera en que pueden ser organizados (cromosomas) para formar rutas válidas.

- **Función de Aptitud (Fitness):** Métrica que evalúa qué tan buena es una ruta (individuo) basándose en la suma total del tiempo de viaje entre estaciones consecutivas en la ruta. En los algoritmos genéticos, la aptitud de cada individuo influye en su probabilidad de ser seleccionado para reproducción en la siguiente generación.
- **Selección:** Proceso por el cual se eligen individuos de la población actual para reproducirse y generar la próxima generación. Este proceso se basa típicamente en la aptitud de los individuos, favoreciendo aquellos que tienen un mejor desempeño según la función de aptitud.
- **Cruce (Crossover):** Método de reproducción que combina los cromosomas de dos individuos progenitores para crear descendencia, con el objetivo de mezclar y potencialmente mejorar las características de los progenitores en las nuevas rutas generadas.
- **Mutación:** Operación que altera aleatoriamente los genes de un individuo, permitiendo introducir variabilidad en la población y explorar nuevas áreas del espacio de soluciones, lo cual es crucial para evitar la convergencia prematura a soluciones locales subóptimas.

### ***Explicación breve del código genetic algorithm***

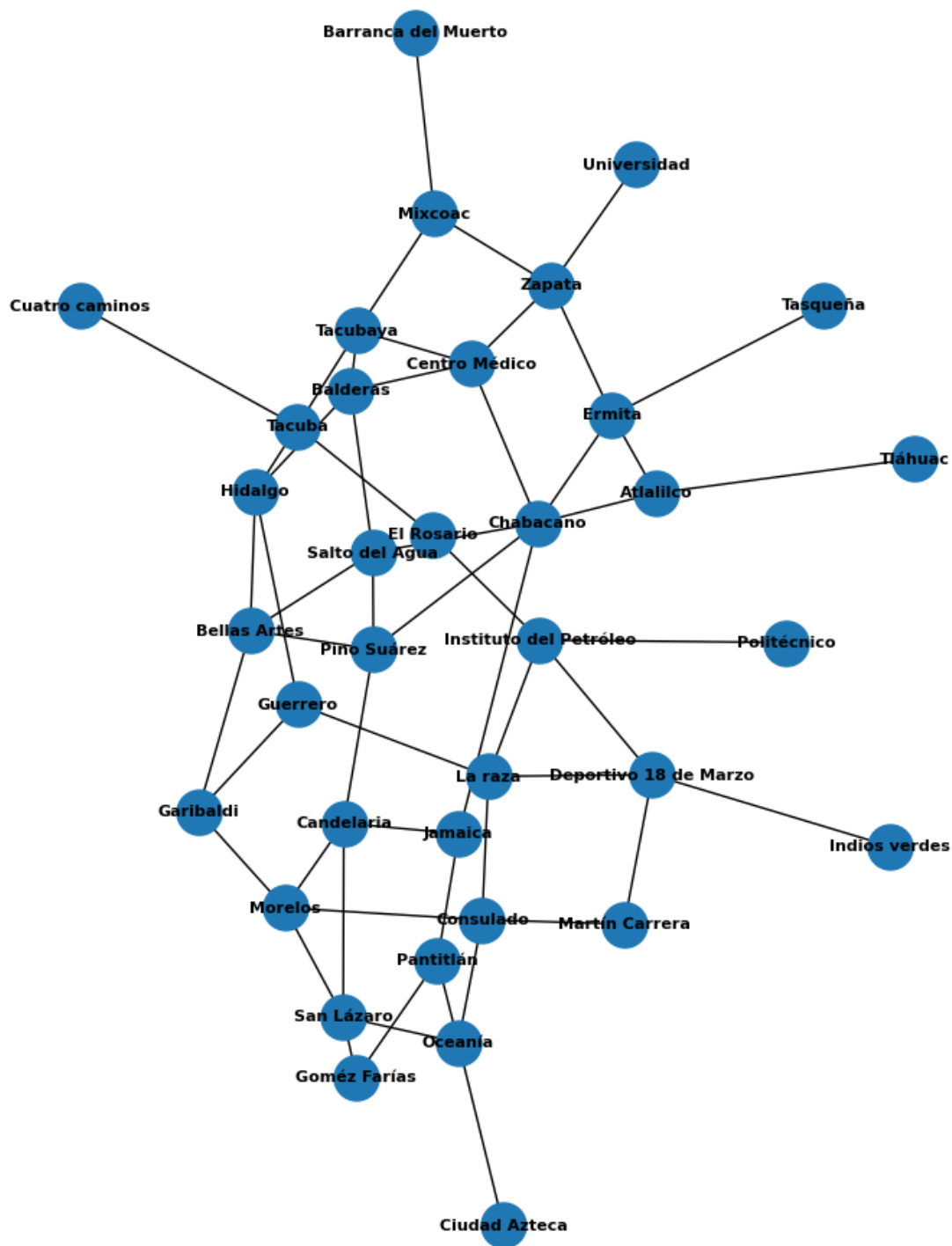
#### **La Metodología de Algoritmos Genéticos**

Los algoritmos genéticos (AG) imitan el proceso de selección natural donde los individuos más aptos sobreviven y se reproducen. Estos individuos son soluciones potenciales al problema en cuestión y sus "genes" son componentes de la solución, que en nuestro caso son las estaciones en una ruta de metro.

#### **Configuración del Problema**

La red de metro es representada como un grafo, donde las estaciones son nodos y las conexiones directas entre ellas son aristas con pesos que indican el tiempo de viaje. El objetivo es encontrar la secuencia de estaciones que minimice el tiempo total desde El Rosario hasta San Lázaro.

## Metro completo



## **Población Inicial**

La población inicial se compone de muchas posibles soluciones (rutas), generadas aleatoriamente. Cada ruta es evaluada para determinar su "aptitud", que en este contexto es el tiempo total de viaje. Las rutas con conexiones no directas reciben una penalización para disuadir su selección.

## **Proceso Evolutivo**

Durante la evolución, se seleccionan rutas para la reproducción basándose en su aptitud; las rutas más cortas tienen una mayor probabilidad de ser seleccionadas. Luego, se aplican operadores genéticos de cruce y mutación:

**Cruce (Crossover):** Este operador combina segmentos de dos rutas progenitoras para crear rutas hijas, con la esperanza de heredar las características deseables de cada progenitor.

**Mutación:** Este operador realiza cambios aleatorios en una ruta para explorar nuevas áreas del espacio de soluciones, lo que ayuda a mantener la diversidad genética en la población.

## **Selección y Reproducción**

A través de múltiples generaciones, las rutas se seleccionan, cruzan y mutan, con cada nueva generación (iteración) produciendo rutas más eficientes que su predecesora. El proceso continúa hasta que se alcanzan un número máximo de generaciones o hasta que se satisfacen otros criterios de parada, como una ruta que es suficientemente corta.

## **Objetivo del proyecto:**

El objetivo principal de este proyecto es aplicar algoritmos genéticos para encontrar la ruta más corta entre dos estaciones específicas en la red del metro de la Ciudad de México, de El Rosario a San Lázaro. La red de metro, compleja y extensa, presenta un escenario ideal para explorar las capacidades de los algoritmos genéticos en la solución de problemas de optimización en sistemas de transporte urbano.

## **Explicación del código (Primera parte)**

Aquí, en este segmento del código prepara el escenario para el proyecto al construir y visualizar la red del metro como un grafo, utilizando las librerías **networkx** para la manipulación de grafos y **matplotlib** para la visualización. Este es un paso crucial ya que establece la base de datos sobre la cual operará el algoritmo genético.

## **Configuración Inicial y Creación del Entorno:**

Importación de Bibliotecas

Inicialmente, se importan dos bibliotecas esenciales:

**networkx (nx):** Utilizada para crear y manipular estructuras de grafo complejas de manera eficiente. Es fundamental para representar la red del metro donde cada estación es un nodo y cada conexión directa entre estaciones es un borde.

**matplotlib.pyplot (plt):** Empleada para la visualización de datos, aquí se usa para mostrar gráficamente la red del metro, facilitando una interpretación visual del problema a resolver.

### Definición del Grafo

```
G = nx.Graph()
```

Aquí, creo un objeto grafo G utilizando nx.Graph(). Este objeto será el contenedor de toda la información de la red del metro:

### Adición de Estaciones y Conexiones

```
all_stations = [  
    ('Tacubaya', 'Balderas', {'weight': 6}),  
    ...]
```

Luego defino y agrego estas tuplas al grafo G usando el método **add\_edges\_from**, que incorpora los nodos y bordes con sus respectivos pesos al grafo. Este paso es crucial pues define la estructura de la red que el algoritmo intentará optimizar.

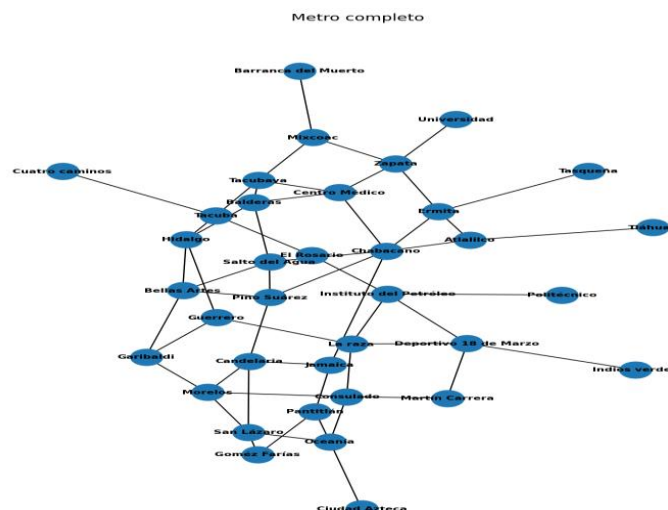
### Dibujo del Grafo

```
pos = nx.spring_layout(G) # Posicionamiento de los nodos según el layout 'spring'  
nx.draw(G, pos, with_labels=True, font_weight='bold', node_size=500, font_size=8)
```

El grafo es dibujado utilizando **nx.draw**, que coloca los nodos y bordes en un layout tipo "spring" para una distribución estética que también intenta minimizar la superposición de bordes:

Este método muestra las conexiones y la estructura de la red del metro y también permite visualizar el 'peso' de cada conexión, representando el tiempo de viaje entre estaciones.

### Primeros resultados



El resultado que se observa en la visualización adjunta es un grafo que representa la red del metro de la Ciudad de México. Cada nodo, señalado con un punto, corresponde a una estación dentro de la red, y las aristas, las líneas que conectan los puntos, representan las rutas directas entre estaciones. Las conexiones entre los nodos reflejan las posibles transiciones de una estación a otra, y la ausencia de una conexión directa entre dos nodos indica que no hay una ruta directa de metro entre esas dos estaciones..

### **Explicación del código (segunda parte)**

Ahora, la siguiente implementación como parte del proyecto, se establecen las constantes y parámetros que regirán el comportamiento del algoritmo genético, se definen las funciones clave para su operación y se detalla el flujo principal del algoritmo. A continuación, desglosamos cada parte:

### **Configuración de los Parámetros del Algoritmo Genético**

Se definen varias constantes esenciales que servirán como los ajustes iniciales para el algoritmo:

- **POPULATION\_SIZE:** Representa el número de individuos, o rutas posibles, que se generarán y mantendrán en la población. Aquí se elige 100,000, lo que indica una población bastante grande para proporcionar una diversidad sustancial en el espacio de búsqueda.
- **P\_CROSSOVER:** La probabilidad de cruce es del 80%. Esto significa que hay una alta posibilidad de que dos individuos se mezclen para crear descendencia, lo cual es vital para la combinación de buenas características entre rutas.
- **P\_MUTATION:** Con un 20%, esta probabilidad de mutación asegura que habrá una oportunidad significativa de variación aleatoria en la población, lo que ayuda a explorar nuevas posibles soluciones y mantiene la diversidad genética de la población.
- **MAX\_GENERATIONS:** Se realizarán 100 generaciones en el proceso evolutivo, lo que establece un límite en el número de ciclos de evolución que el algoritmo ejecutará.
- **HALL\_OF\_FAME\_SIZE:** El tamaño del salón de la fama es de 5, lo que significa que los cinco mejores individuos se guardarán a lo largo de las generaciones.

### **Inicialización y Creación de Objetos Genéticos**

- **RANDOM\_SEED:** Se establece una semilla aleatoria para garantizar resultados consistentes y reproducibles en cada ejecución del algoritmo.
- **Objetos de DEAP:** Utilizando la biblioteca DEAP, se crea un objeto de "FitnessMin" para definir que se trata de un problema de minimización, y un objeto "Individual" para representar a cada ruta posible en la población.
- **Toolbox:** Es una estructura que permite registrar las diferentes funciones genéticas y parámetros que utilizará el algoritmo.

## Maapeo de Estaciones a Índices y Funciones Genéticas

- **Maapeo de Estaciones:** Se convierten las estaciones en índices numéricos para facilitar el proceso de cruce y mutación dentro del algoritmo.
- **Crossover de Rutas Conectadas:** Es una función de cruce diseñada para mantener la validez de las rutas, asegurándose de que solo se cruzan en estaciones que son adyacentes en ambos padres.
- **Reparación de Rutas:** Una función de reparación corrige las rutas no válidas resultantes del cruce, garantizando que las rutas resultantes sigan siendo posibles dentro del contexto de la red del metro.

## Generación de Rutas y Evaluación de Aptitud

- **Generación Aleatoria de Rutas:** Se implementa una función para crear permutaciones aleatorias de las estaciones, con puntos de inicio y final fijos, representando diferentes rutas potenciales a través de la red del metro.
- **Evaluación de Fitness:** Se establece la función de aptitud que mide la calidad de cada ruta en función de su tiempo total de viaje, penalizando fuertemente las rutas con transiciones no válidas.

## Selección y Operadores Genéticos

- **Operadores Genéticos:** Se registra la función de selección y los operadores de cruce y mutación que manipularán la población a lo largo del algoritmo genético.

## Ejecución del Algoritmo Genético y Presentación de Resultados

- **Flujo del Algoritmo Genético:** La función **main** encapsula todo el proceso de ejecución del algoritmo genético, desde la inicialización de la población hasta la evolución a través de las generaciones y la presentación de las mejores rutas encontradas.
- **Resultados:** Una función específica imprime los resultados de manera amigable, destacando las rutas más eficientes en términos de tiempo de viaje.

## Segundos resultados

gen	nevals	min	avg
0	10000	2.5e+07	3.1953e+07
1	8412	2.4e+07	3.11424e+07
2	8485	2.4e+07	3.06691e+07
3	8434	2.4e+07	3.03753e+07
4	8393	2.3e+07	3.01386e+07
5	8434	1.9e+07	2.99399e+07
6	8351	1.9e+07	2.98087e+07
7	8394	2.2e+07	2.97059e+07
8	8316	2.2e+07	2.95026e+07
9	8429	2e+07	2.94848e+07
10	8432	2e+07	2.9423e+07
11	8358	2.2e+07	2.93859e+07
12	8341	2.1e+07	2.93011e+07
13	8435	2e+07	2.92886e+07
14	8404	2e+07	2.92289e+07
15	8380	2e+07	2.92234e+07
16	8404	2e+07	2.92055e+07
17	8383	2e+07	2.91858e+07
32	8420	2e+07	2.90257e+07
33	8420	1.9e+07	2.90092e+07
34	8369	1.9e+07	2.90405e+07
35	8360	2.1e+07	2.90228e+07
36	8474	2.1e+07	2.90519e+07
37	8450	2e+07	2.9064e+07
38	8416	2.1e+07	2.90955e+07
39	8397	2.1e+07	2.91111e+07
40	8359	2.1e+07	2.9083e+07
41	8389	2e+07	2.90848e+07
42	8426	2.1e+07	2.90927e+07
43	8429	2.1e+07	2.90623e+07
44	8301	2.1e+07	2.90321e+07
45	8366	2e+07	2.90681e+07
46	8370	2.1e+07	2.9087e+07
47	8372	2.1e+07	2.90721e+07
48	8299	2.1e+07	2.91137e+07
49	8315	2.1e+07	2.90518e+07
50	8567	2.1e+07	2.91221e+07

# Inteligencia artificial

## Gerardo Irazabal Monroy 171809

```
----- Mejores Rutas Encontradas -----

Ruta 1 con el Tiempo Más Corto (18000035.0 minutos):
El Rosario -> Atlalilco -> Centro Médico -> Balderas -> Guerrero -> Garibaldi -> Morelos -> Candelaria -> Jamaica -> Consulado -> Oceanía -> Tacubaya -> Instituto del Petróleo -> Ciudad Azteca -> La raza -> Deportivo 18 de Marzo -> Indios verdes -> Martín Carrera -> Mixcoac -> Zapata -> Ermita -> Tasqueña -> Universidad -> Politécnico -> Cuatro caminos -> Tacuba -> Tláhuac -> Gómez Farías -> Pantitlán -> Hidalgo -> Bellas Artes -> Pino Suárez -> Salto del Agua -> Chabacano -> Barranca del Muerto -> San Lázaro

Ruta 2 con el Tiempo Más Corto (19000032.0 minutos):
Ciudad Azteca -> Oceanía -> Indios verdes -> Centro Médico -> Candelaria -> San Lázaro -> Cuatro caminos -> Tacuba -> Hidalgo -> Garibaldi -> Guerrero -> La raza -> Pino Suárez -> Mixcoac -> Atlalilco -> Ermita -> Zapata -> Bellas Artes -> Tasqueña -> Politécnico -> Instituto del Petróleo -> Deportivo 18 de Marzo -> Martín Carrera -> Consulado -> Morelos -> Jamaica -> Chabacano -> Salto del Agua -> Balderas -> Universidad -> Tacubaya -> Tláhuac -> Barranca del Muerto -> Pantitlán -> El Rosario -> Gómez Farías

Ruta 3 con el Tiempo Más Corto (19000034.0 minutos):
Ciudad Azteca -> Guerrero -> Garibaldi -> Oceanía -> Tasqueña -> Tacubaya -> Mixcoac -> Salto del Agua -> Pino Suárez -> Candelaria -> Morelos -> Consulado -> La raza -> Deportivo 18 de Marzo -> Indios verdes -> Ermita -> Hidalgo -> Balderas -> Tláhuac -> Universidad -> Bellas Artes -> Zapata -> Chabacano -> Jamaica -> Cuatro caminos -> Tacuba -> Barranca del Muerto -> Martín Carrera -> Atlalilco -> El Rosario -> Instituto del Petróleo -> Politécnico -> Centro Médico -> Pantitlán -> Gómez Farías -> San Lázaro

Ruta 4 con el Tiempo Más Corto (19000036.0 minutos):
Guerrero -> Jamaica -> Cuatro caminos -> Tacuba -> El Rosario -> Universidad -> Pino Suárez -> Candelaria -> Salto del Agua -> Tláhuac -> Oceanía -> Pantitlán -> Garibaldi -> Hidalgo -> Bellas Artes -> Ermita -> Balderas -> Tacubaya -> Centro Médico -> Zapata -> Gómez Farías -> Atlalilco -> La raza -> Instituto del Petróleo -> Politécnico -> Ciudad Azteca -> Chabacano -> Tasqueña -> Barranca del Muerto -> Mixcoac -> Indios verdes -> Deportivo 18 de Marzo -> Martín Carrera -> Consulado -> Morelos -> San Lázaro

Ruta 5 con el Tiempo Más Corto (19000039.0 minutos):
El Rosario -> Centro Médico -> Balderas -> Universidad -> Indios verdes -> Ermita -> Atlalilco -> Tláhuac -> Zapata -> Ciudad Azteca -> Cuatro caminos -> Tacuba -> Hidalgo -> Martín Carrera -> Pino Suárez -> Bellas Artes -> Garibaldi -> Salto del Agua -> Tasqueña -> Politécnico -> Instituto del Petróleo -> Deportivo 18 de Marzo -> Candelaria -> Jamaica -> Chabacano -> Guerrero -> Morelos -> Consulado -> Oceanía -> Pantitlán -> Mixcoac -> Tacubaya -> La raza -> Barranca del Muerto -> Gómez Farías -> San Lázaro

Out[7]: ([[2,
34,
16,
35,
4,
15,
30,
31,
7,
9,
17,
28,
22,
18,
0,
32,
25,
24,
6,
```

Lo que observamos aquí, primero, es la salida del algoritmo genético que proporciona información sobre el proceso de evolución y optimización que ha tenido lugar. Se presenta en forma de registros generacionales que resumen el rendimiento de la población a lo largo del tiempo, así como las rutas más cortas que ha descubierto el algoritmo. Vamos a desglosar y explicar estos resultados.

### Registro Generacional

La tabla proporcionada es un registro del proceso evolutivo del algoritmo generación por generación, donde cada fila representa una generación diferente y las columnas representan:



- **gen:** El número de la generación actual.
- **nevals:** El número de individuos (rutas) que han sido evaluados durante esa generación.
- **min:** El valor mínimo de la función de aptitud en la población actual, que en este contexto representa la ruta más corta encontrada en esa generación.
- **avg:** El valor promedio de la función de aptitud para toda la población.

A lo largo de las generaciones, podemos observar una tendencia hacia la disminución tanto en el valor mínimo como en el promedio de la función de aptitud, lo que indica que el algoritmo está afinando su búsqueda y encontrando rutas más eficientes con el tiempo. Por ejemplo, el valor mínimo de la función de aptitud disminuye de  $2.5e+07$  en la generación 0 a  $1.8e+07$  en la generación 20, lo que sugiere una mejora significativa en la calidad de las rutas encontradas por el algoritmo.

### Mejores Rutas Encontradas

La sección "Mejores Rutas Encontradas" lista las rutas más eficientes que el algoritmo ha identificado hasta el momento, en términos del tiempo total de viaje (minimizado). Cada ruta se muestra con su respectiva secuencia de estaciones, y el tiempo asociado indica la suma total de los pesos (tiempos de viaje) de las aristas (conexiones directas) que componen la ruta.

Las rutas están clasificadas según su eficiencia, con la "Ruta 1" siendo la que tiene el tiempo más corto encontrado. Estos tiempos son extremadamente grandes (por ejemplo, 18000035.0 minutos) debido a la penalización aplicada por el algoritmo en caso de rutas inválidas durante el proceso de evaluación. En la realidad, las rutas serían mucho más cortas y las penalizaciones sirven para guiar al algoritmo genético hacia rutas válidas y óptimas.

Los números en los corchetes al final representan una codificación de la ruta en términos de índices de estaciones, proporcionando una forma concisa de referirse a las estaciones involucradas en cada ruta.

### Desafíos e importante leer del resultado final

Los algoritmos genéticos son técnicas de búsqueda heurística que se inspiran en la teoría de la evolución natural para resolver problemas de optimización. Aunque son herramientas poderosas y flexibles, tienen limitaciones inherentes, especialmente cuando se aplican a problemas con un espacio de soluciones muy grande y restricciones específicas, como es el caso de la búsqueda de la ruta más corta en una red de metro. Veamos por qué:

### Exploración vs. Explotación

Los algoritmos genéticos trabajan con un equilibrio entre exploración (buscando en nuevas áreas del espacio de soluciones) y explotación (mejorando las soluciones existentes). Para un problema con un espacio de búsqueda vasto como el de una red de metro extensa:

- **Exploración:** El algoritmo necesita explorar suficientemente el espacio de soluciones para encontrar la óptima, lo que puede requerir un número prohibitivamente grande de iteraciones debido a la cantidad masiva de posibles rutas.
- **Explotación:** En un espacio de soluciones grande y complejo, es fácil quedarse atrapado en mínimos locales, donde el algoritmo no logra encontrar la solución óptima global sino solamente la mejor solución dentro de una región limitada del espacio de búsqueda.

### Complejidad del Espacio de Soluciones

El problema específico de encontrar la ruta más corta en una red de metro tiene las siguientes complejidades:

- **Tamaño del espacio de soluciones:** Hay una cantidad astronómica de posibles rutas, especialmente si consideramos que una ruta podría teóricamente pasar varias veces por la misma estación.
- **Restricciones de conectividad:** La solución óptima no solo debe ser la más corta en términos de tiempo o distancia, sino también una ruta válida que siga las conexiones reales del metro.

### Convergencia Prematura y Diversidad Genética

- **Convergencia Prematura:** El algoritmo genético puede converger prematuramente a una solución subóptima si no mantiene suficiente diversidad genética en la población. Esto es aún más probable cuando las restricciones del problema son estrictas, como es el caso de las rutas de metro que deben seguir caminos válidos específicos.
- **Diversidad Genética:** Mantener una diversidad genética en la población es clave para evitar la convergencia prematura, pero es un desafío en problemas con restricciones fuertes, donde muchas mutaciones o cruces resultarán en soluciones no válidas que deben ser penalizadas y, por lo tanto, descartadas.

### Limitaciones Específicas de la Red de Metro

El problema de la ruta más corta en la red del metro de la Ciudad de México tiene particularidades que aumentan la dificultad para un algoritmo genético:

- **Rutas Obligadas:** Hay estaciones por donde obligatoriamente se debe pasar para ir de El Rosario a San Lázaro, como "Instituto del Petróleo" o "Tacuba". Si el algoritmo genético no incorpora esta lógica en su proceso de generación y evaluación de rutas, puede generar muchas soluciones no viables. (y aún así si las ponemos como restricciones, ya no será la naturaleza del algoritmo genético, porque, si es así, nos vamos restringiendo hasta dejar San Lázaro)
- **Espacio de Búsqueda Discreto:** A diferencia de problemas continuos donde pequeñas mutaciones pueden llevar a mejoras incrementales, cambiar una estación en la ruta puede llevar a una ruta completamente diferente y posiblemente no válida.

### Tiempo y Recursos Computacionales

- Recursos Computacionales: Para explorar completamente el espacio de soluciones y encontrar la solución óptima, el algoritmo podría necesitar un número excesivo de generaciones e individuos, lo cual requeriría un tiempo y recursos computacionales impracticables.

### Conclusión

aunque el algoritmo genético puede proporcionar soluciones buenas y cercanas al óptimo, encontrar la ruta absolutamente más corta en una red de metro complicada con un algoritmo genético sería computacionalmente muy costoso y probablemente inviable en la práctica. Por tanto, para este problema, propongo otros algoritmo más determinístico y específico para este tipo de problemas, como el algoritmo de Dijkstra, que pueden garantizar encontrar la ruta más corta en un tiempo computacional razonable y con la certeza de cumplir con todas las restricciones del sistema de transporte.

### Referencias

Sheppard, C. (2017). *Genetic algorithms with Python*. Austin, TX, USA: Smashwords Edition.

Lee, W., & Kim, H. Y. (2005, July). Genetic algorithm implementation in Python. In Fourth Annual ACIS International Conference on Computer and Information Science (ICIS'05) (pp. 8-11). IEEE.