

Traveling Salesman Problem

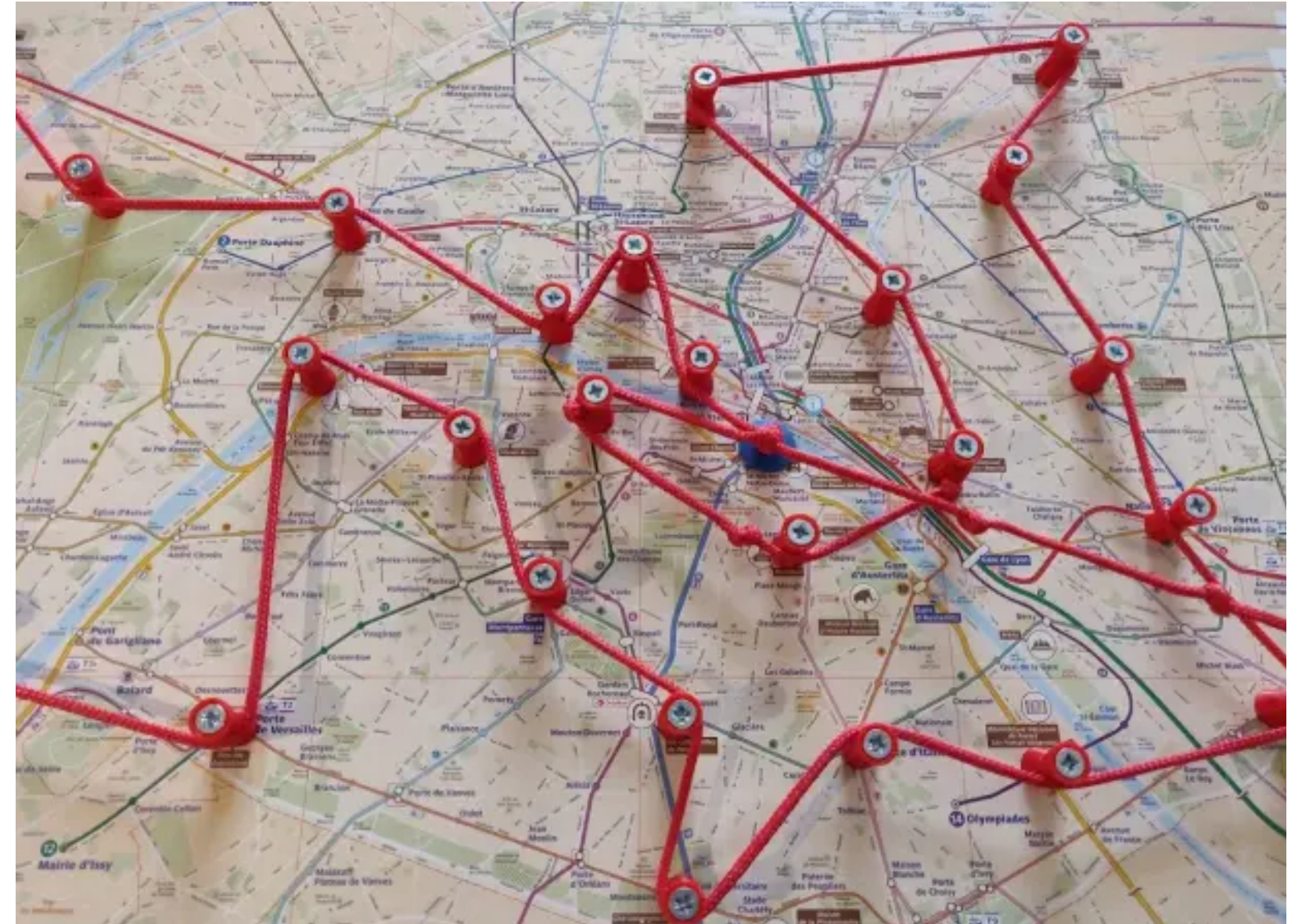
Sebastián Reyes
Aaron Beltran

Tabla de contenido

I	Definición
II	Complejidad
III	Posibles Soluciones
IV	Enfoque de algoritmos genéticos
V	Distancia Haversine
VI	Cálculo de matriz de distancia
VII	Partes del algoritmo genético
VIII	Resultados
IX	Conclusiones

Definición

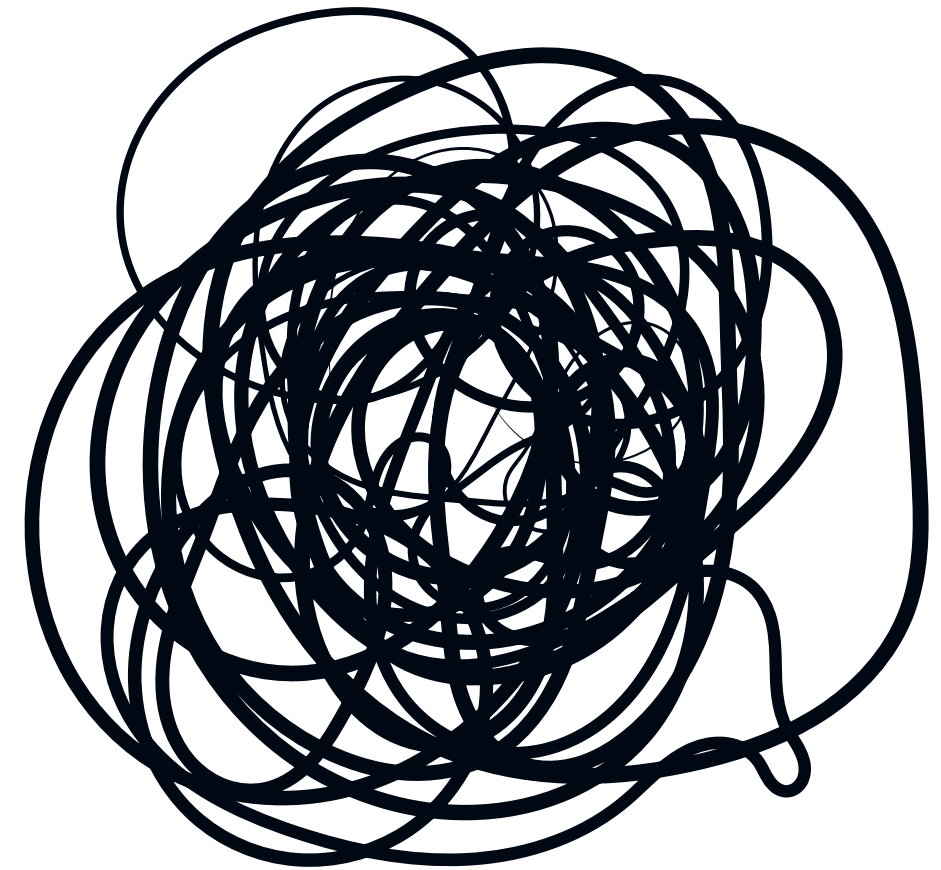
Traveling Salesman Problem consiste en encontrar la ruta más corta que permite a un vendedor visitar una serie de ciudades, pasando una sola vez por cada una y regresando a su ciudad de origen



Complejidad

El TSP es un problema NP-difícil, lo que implica que no existe un algoritmo eficiente que lo resuelva en tiempo polinomial para todos los casos.

La complejidad NP-difícil describe problemas que son tan complicados como los más difíciles en la clase NP. Estos problemas no tienen un método conocido para ser resueltos rápidamente en todos los casos, y el tiempo para encontrar una solución aumenta exponencialmente con el tamaño del problema.



Posibles Soluciones

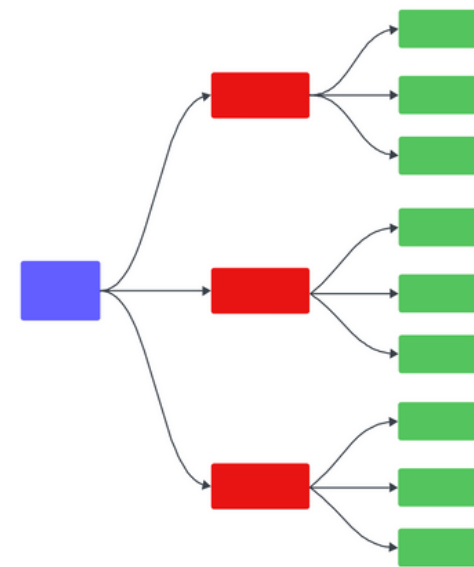
Enfoque de fuerza bruta

Acá se calculan y comparan todas las posibles rutas, según la cantidad de destinos que se tengan que visitar, para luego establecer una única solución que en este caso, sería la más corta y por ende la más óptima.



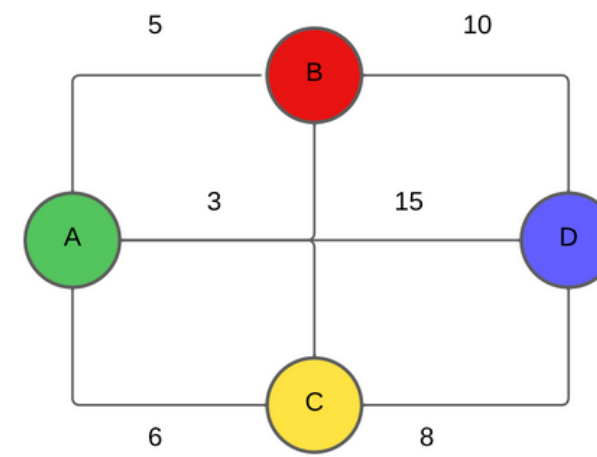
Método de ramificación y atadura

En este método se divide el problema en múltiples subproblemas donde cada uno de éstos tiene varias soluciones posibles.



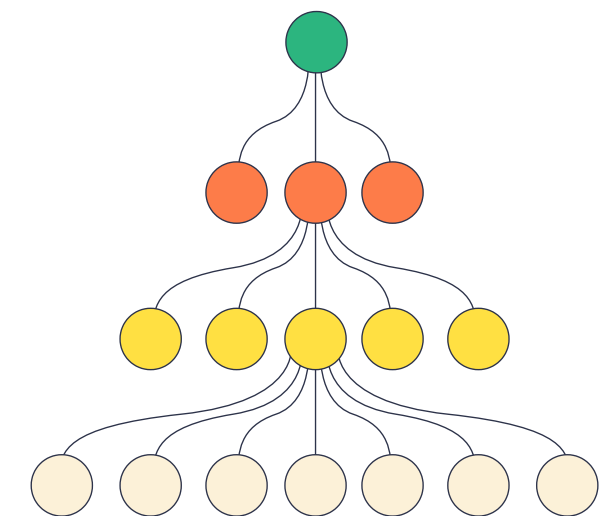
Enfoque de la ruta más corta

Selecciona siempre la ciudad más cercana a la actual hasta completar el recorrido.

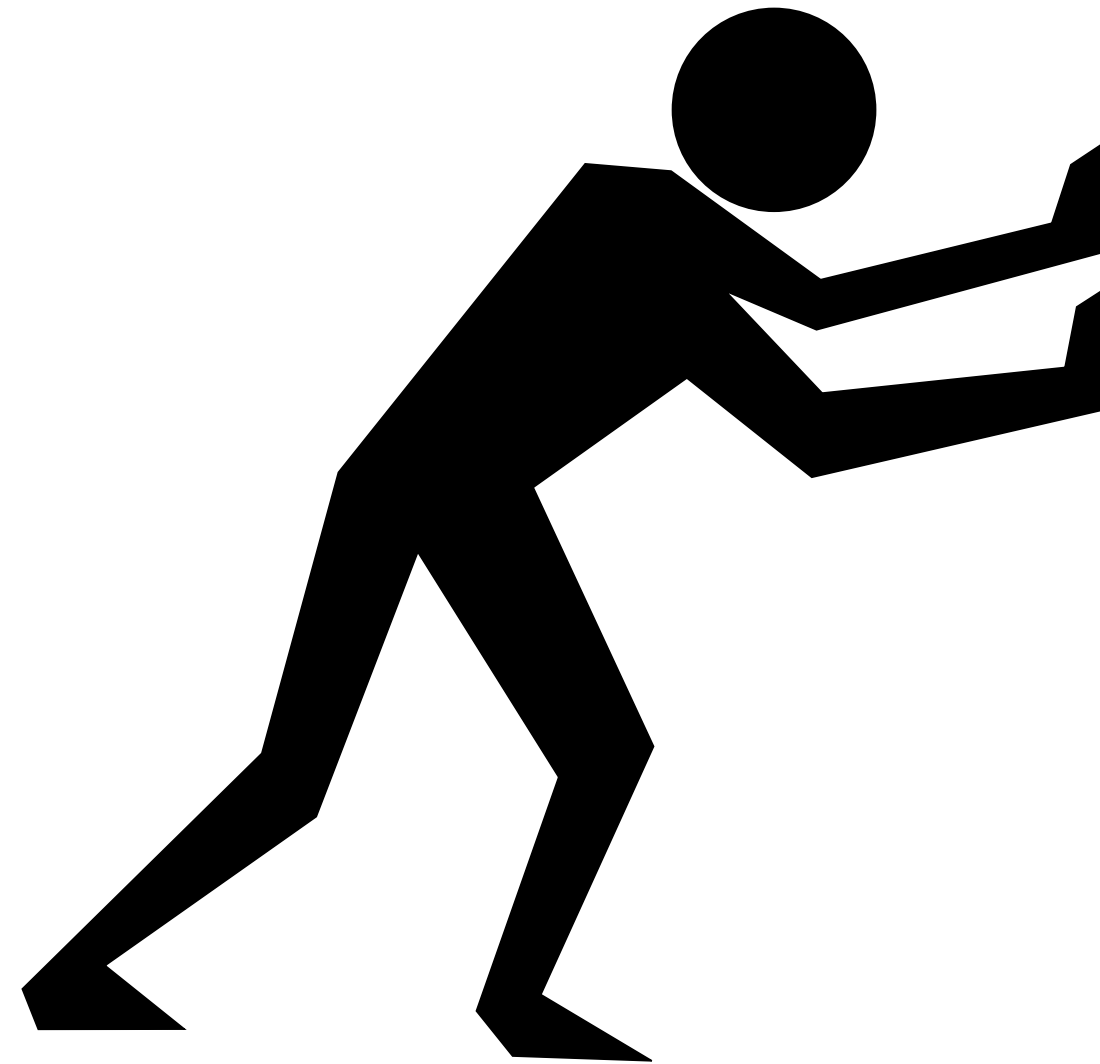


Enfoque de algoritmos genéticos

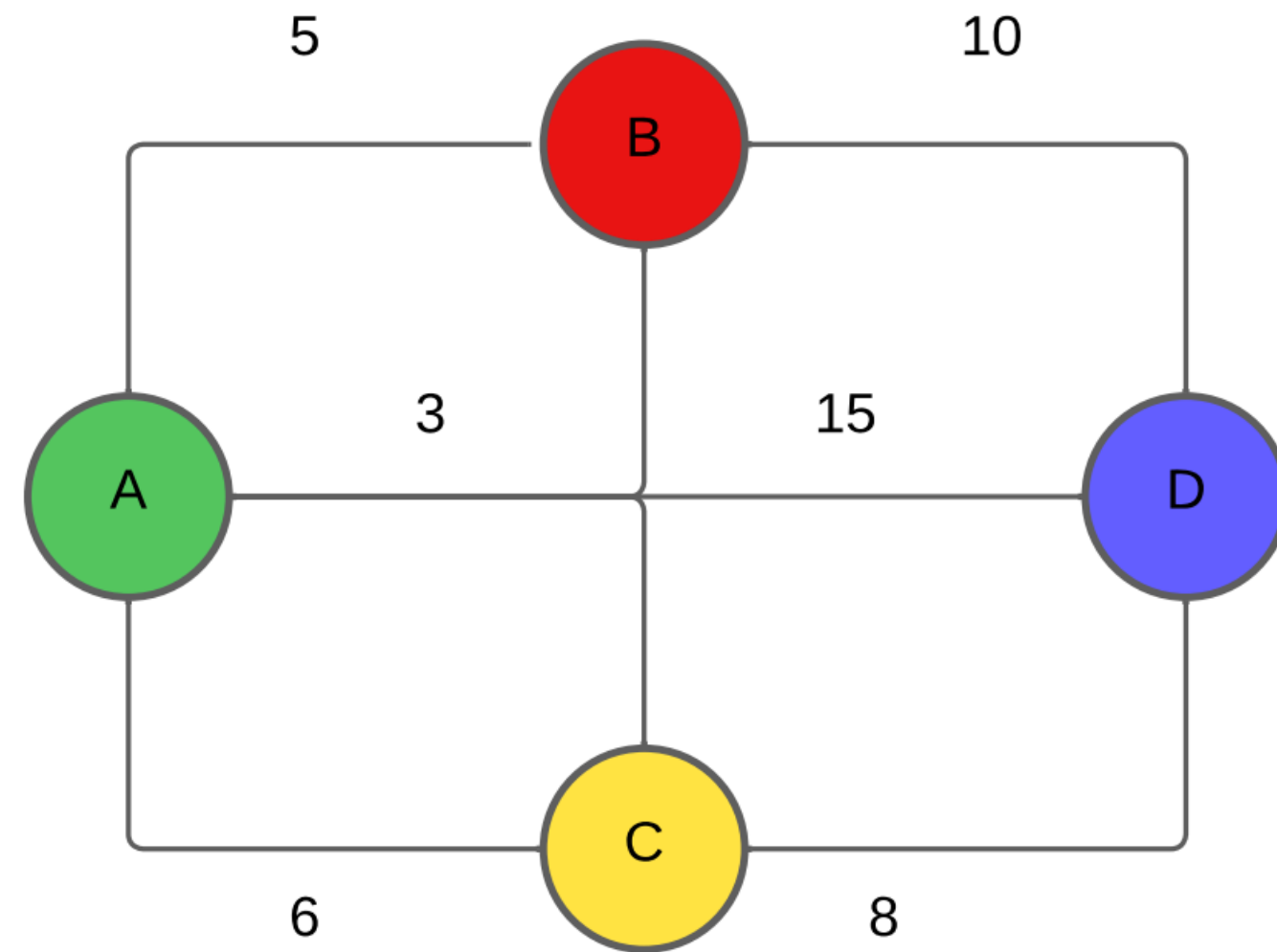
El problema de rutas se resuelve mediante un proceso de optimización evolutivo. Se comienza generando una población inicial de rutas posibles, donde cada ruta representa una solución potencial al problema.



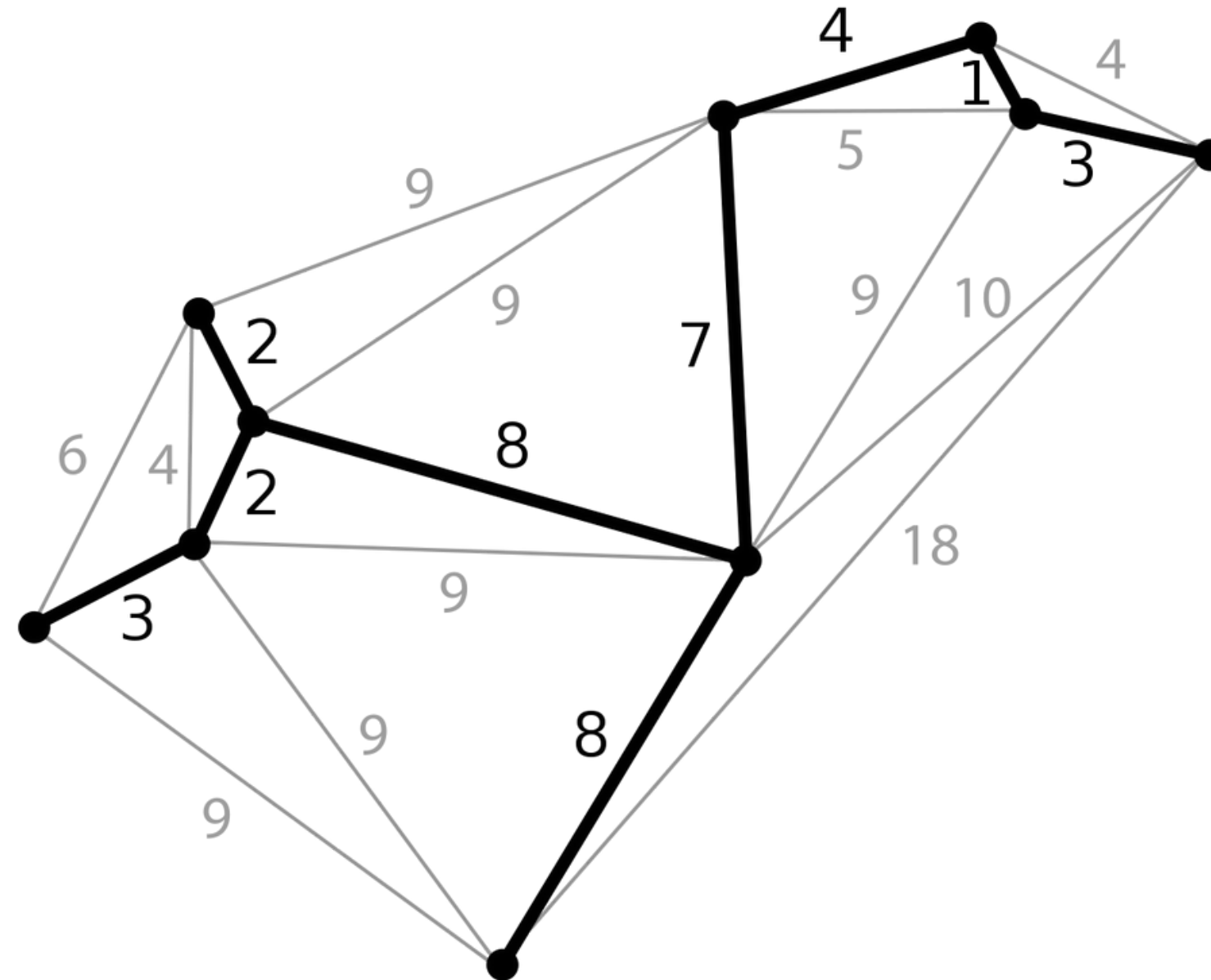
Enfoque de fuerza bruta



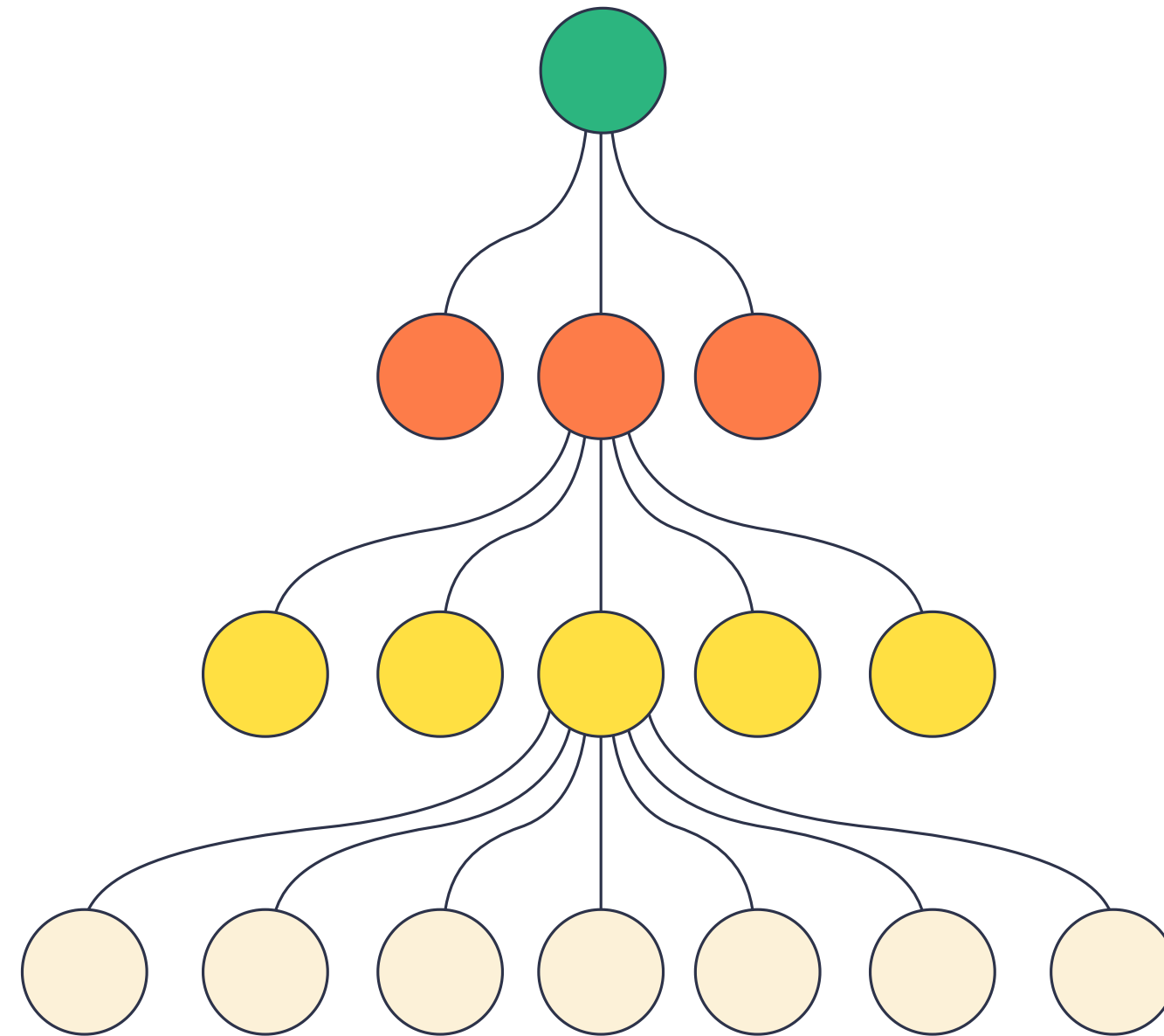
Enfoque de la ruta más corta



Heuristica MST con DPF

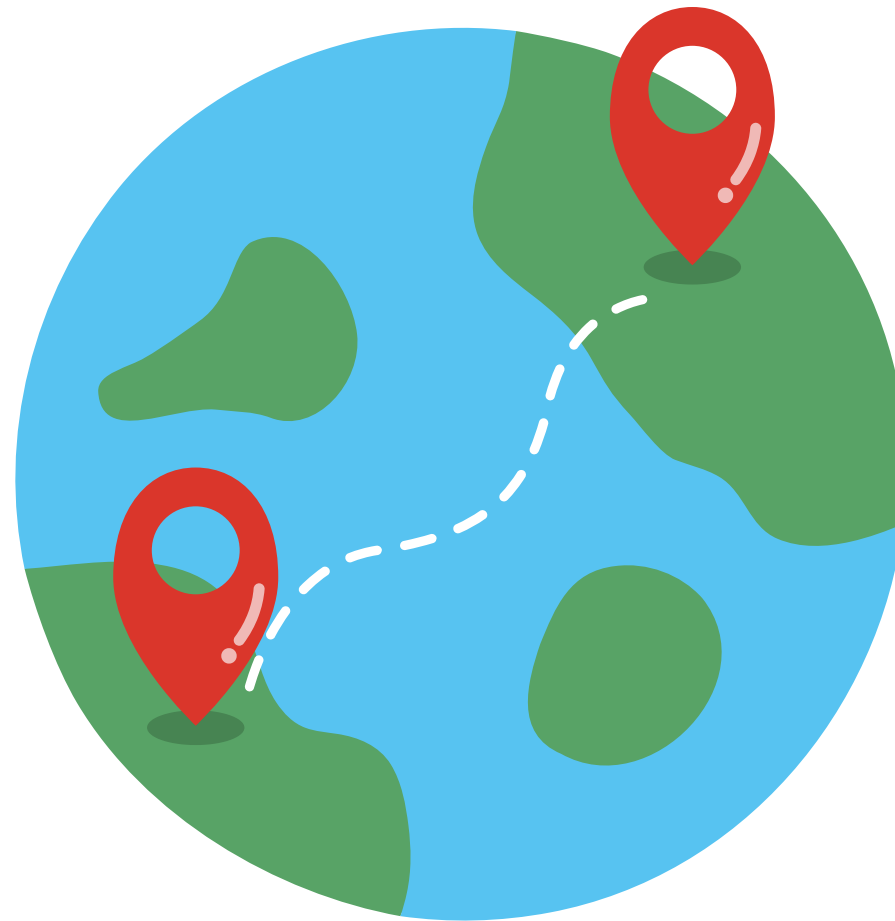


Enfoque de algoritmos genéticos

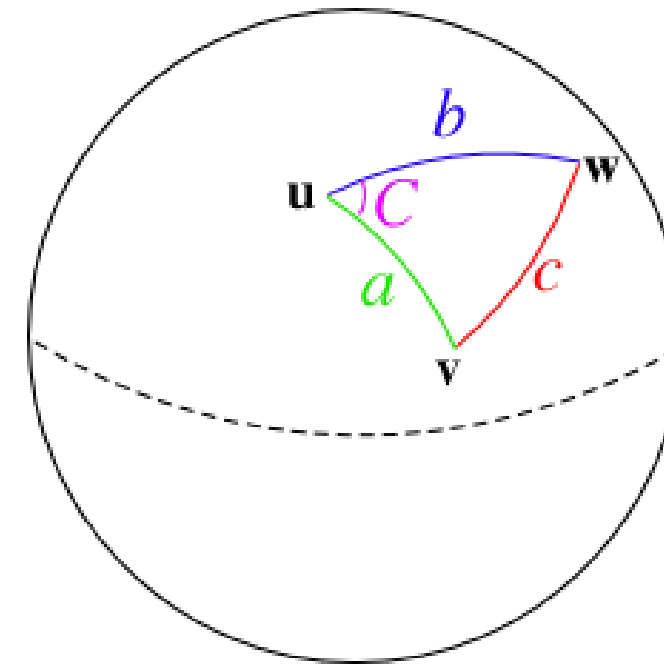
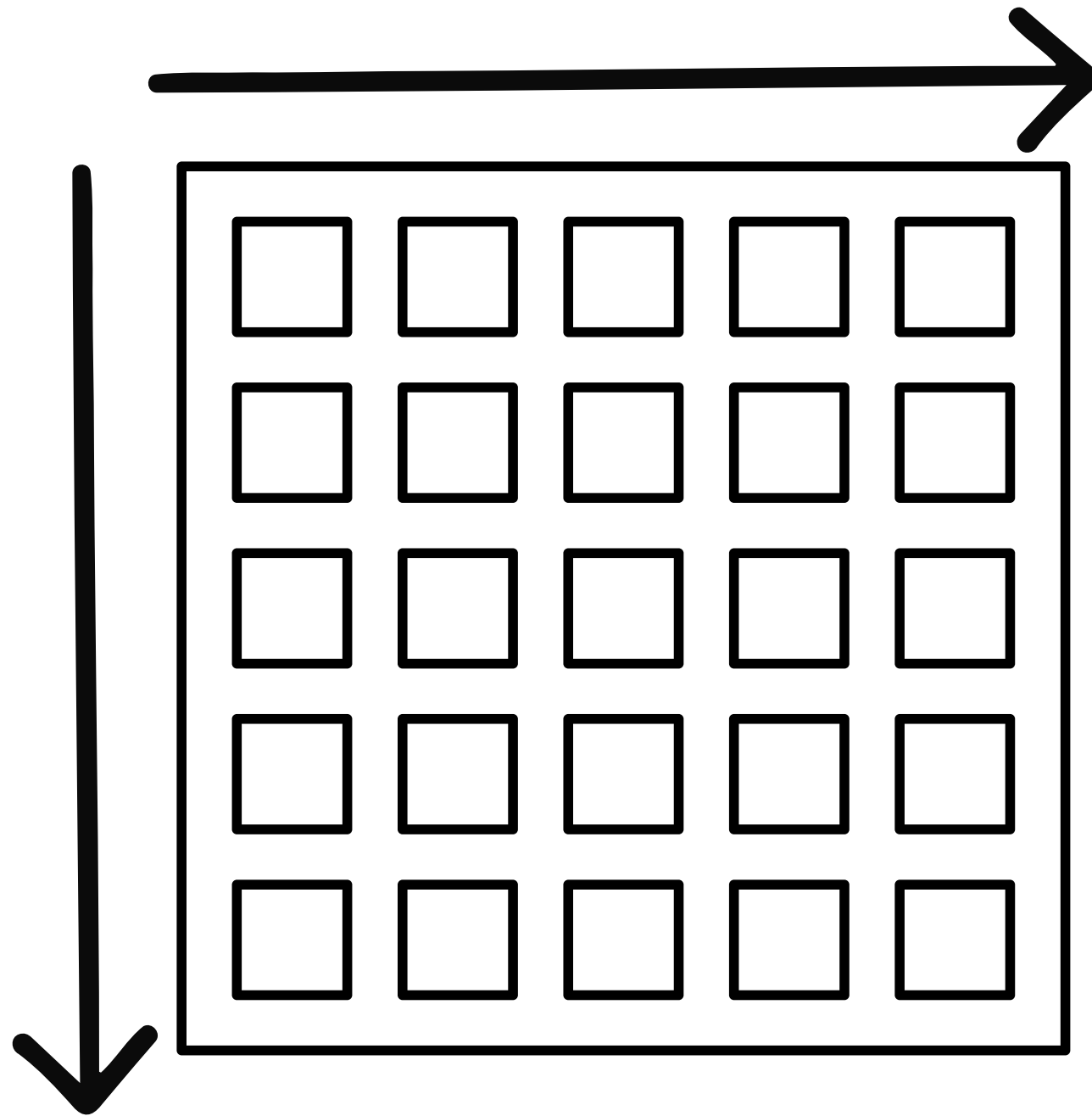


Distancia Haversine

$$d = 2 * r * \arcsin(\sqrt{\sin^2(\frac{\Delta lat}{2}) + \cos(lat_1) * \cos(lat_2) * \sin^2(\frac{\Delta lon}{2})})$$

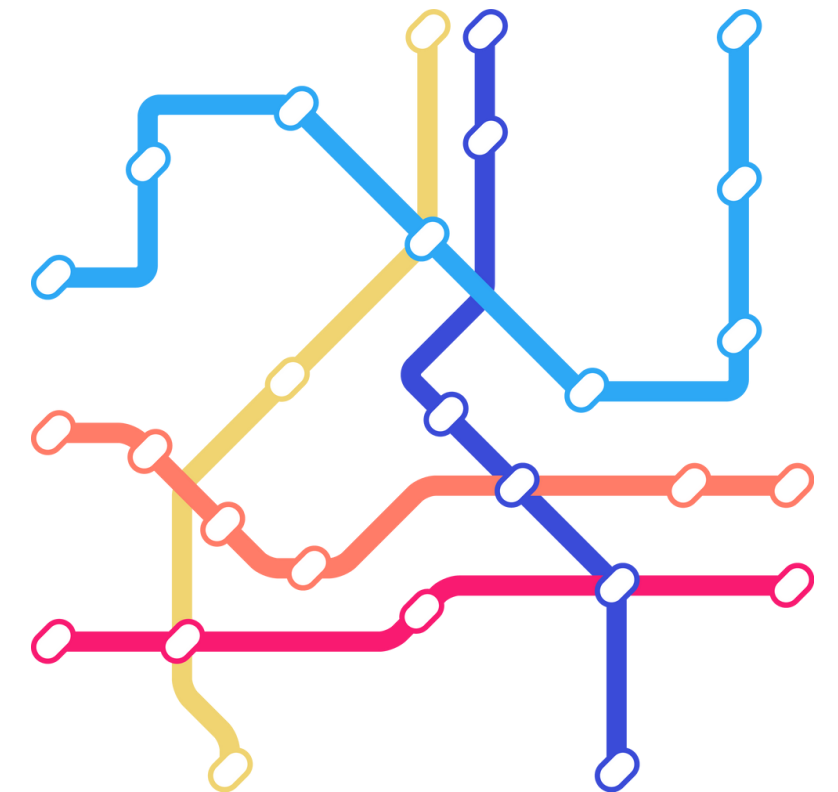


Cálculo de matriz de distancia



Inicialización

- Generación de una población de rutas
- El primer individuo se genera con usando Greedy
- Los demás individuos son rutas aleatorias



Fitness Function

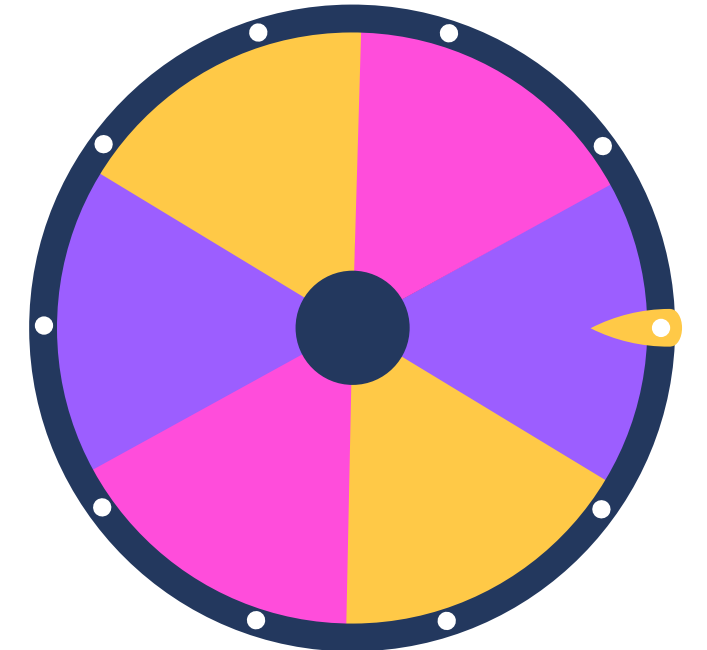
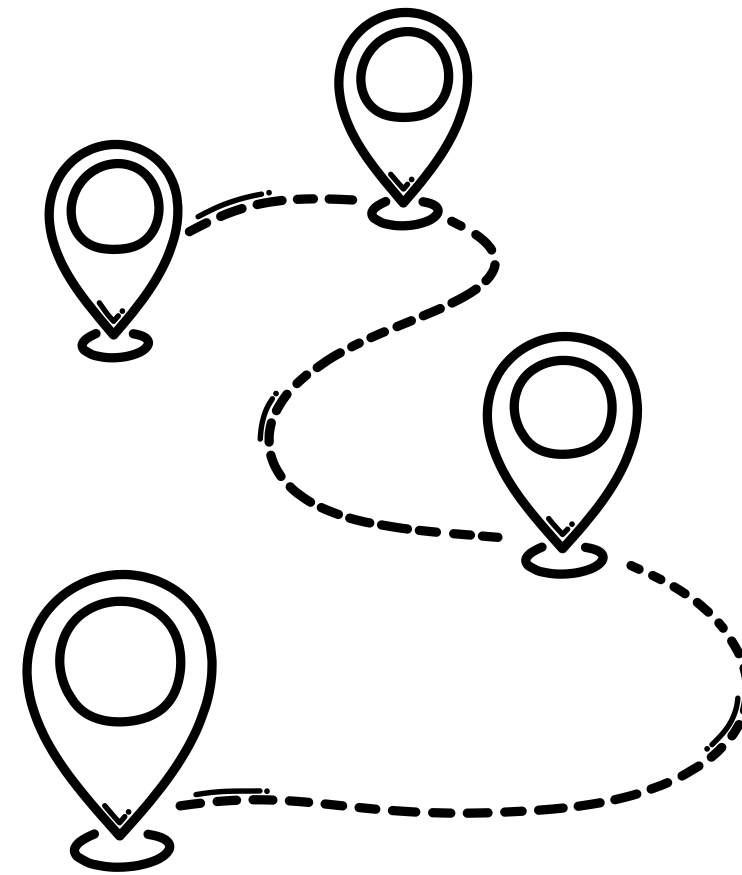
- Se define al individuo más óptimo de forma inversa a la distancia total
- Rutas más cortas obtienen mayor valor fitness
- Prioriza soluciones de mayor calidad

$$\textit{Valor fitness} = \frac{1}{\textit{distancia total}}$$



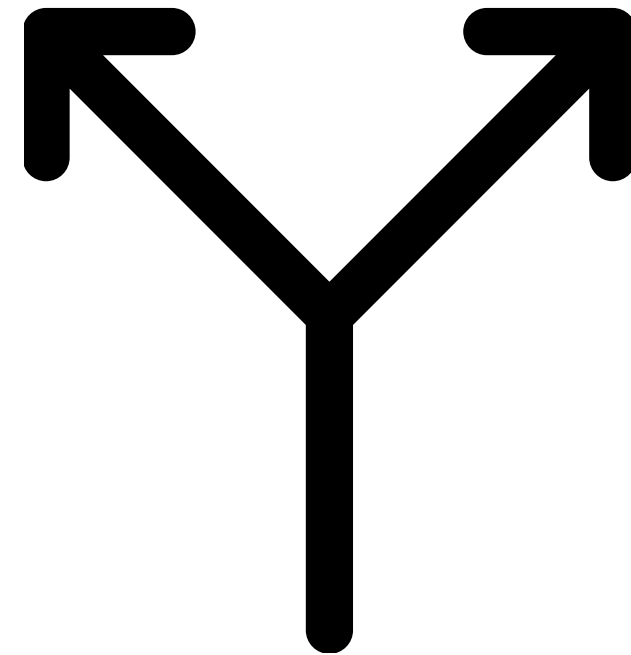
Selection

- Se eligen las rutas padre
- La probabilidad de ser seleccionado es proporcional al valor fitness
- Propaga las mejores soluciones en las poblaciones generadas



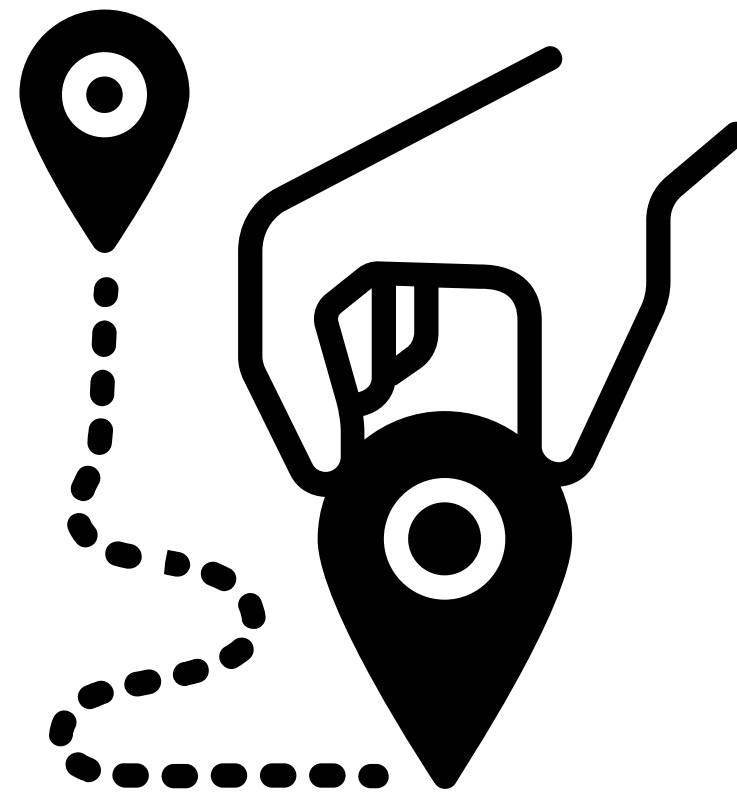
Crossover

- Se implementa el método de cruce Order CrossOver (OX)
- Se combinan y generan nuevas rutas que heredan las mejores cualidades
- Se respeta la naturaleza de permutación del problema



Mutation

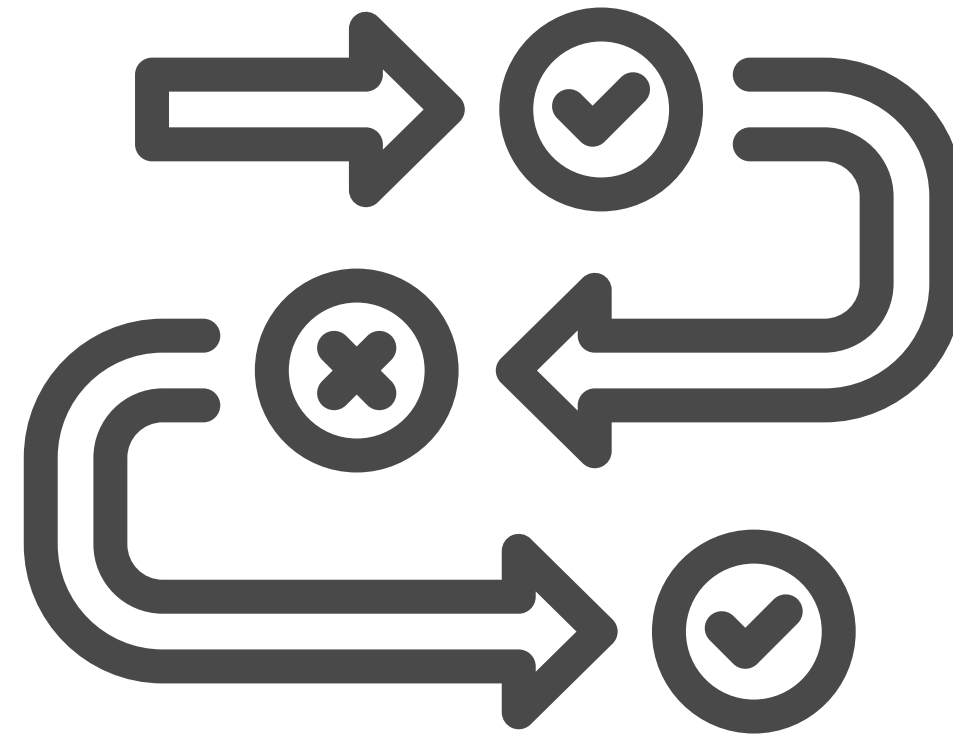
- Se implementa mutación por intercambio
- Se alteran aleatoriamente las rutas
- Evitar estancamiento y variar la población



Aceptación en la simulación

- Si el hijo tiene una menor distancia se acepta
- Si el hijo se considera peor, se acepta si tiene una probabilidad esperada que depende de la relación entre temperatura y diferencias de distancia

$$probabilidad = e^{\frac{\text{diferencia total}}{\text{temperatura}}}$$



Gracias