

Adrián Vázquez Barrera

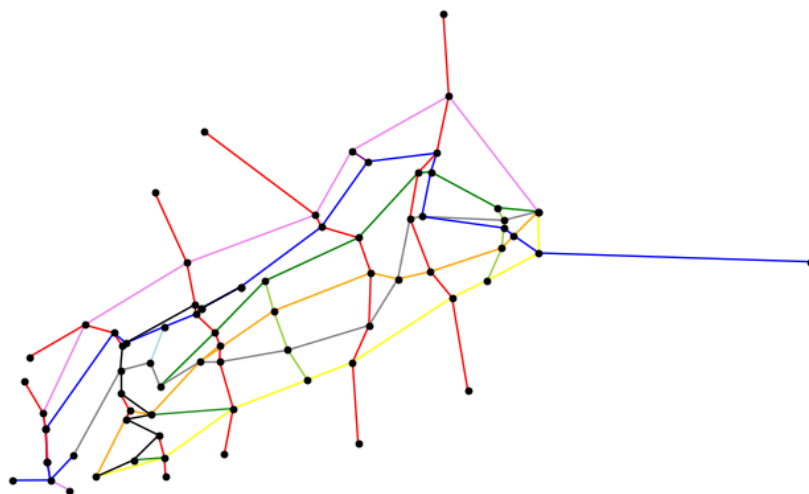
Técnicas de Inteligencia Artificial

**Chinese Postman Problem
2021-2022**



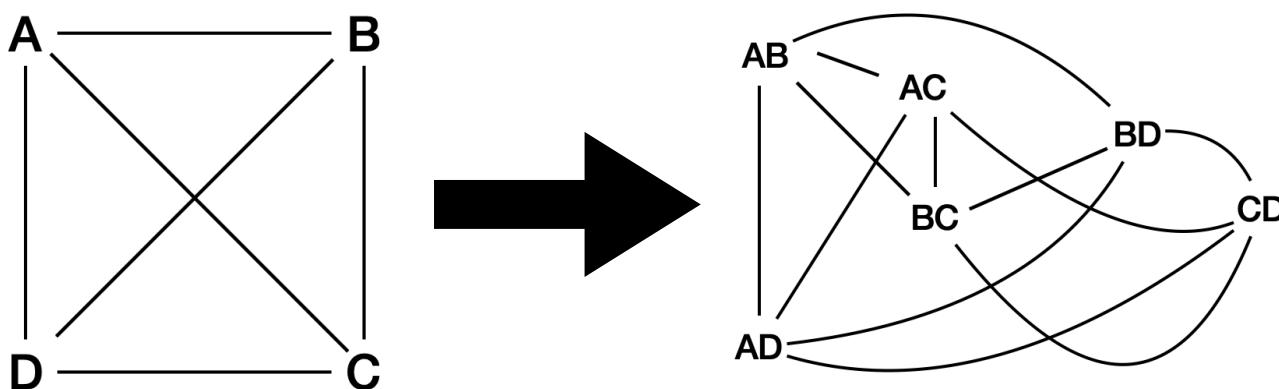
UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

DESCRIPCIÓN DEL PROBLEMA



Un repartidor de correo de origen chino debe recorrer todas las calles de su distrito asignado, repartir la correspondencia y volver al punto de partida. Para llevar a cabo su tarea más eficientemente, se propone una serie de soluciones basadas en técnicas metaheurísticas.

Se deberá, por tanto, recorrer todas y cada una de las aristas del grafo en el menor tiempo posible. Para ello, se propone transformar el espacio de búsqueda en un grafo equivalente donde los vértices y sus conexiones sean equivalentes a las aristas del grafo original.



De este modo, se utilizará el grafo transformado para generar un genotipo de tipo combinatorio, junto a esto, se implementa un algoritmo de *pathfinding* que conecta virtualmente todos los vértices a través del camino más corto entre pares de vértices no conectados físicamente en el grafo original. Este mismo mecanismo se aplica igualmente al grafo transformado.

Gracias a estas transformaciones, conseguimos reducir las restricciones del problema, lo que redundará en un mejor desempeño de las metaheurísticas al añadir un suavizado que orienta a los algoritmos a moverse por el espacio de búsqueda sin muchos exabruptos, fruto de dichas restricciones.

ALGORITMO GENÉTICO

Para almacenar las distancias entre vértices, se propone la siguiente representación matricial.

$$A = \begin{bmatrix} 0 & d_{12}^2 & d_{13}^2 & \dots & d_{1n}^2 \\ d_{21}^2 & 0 & d_{23}^2 & \dots & d_{2n}^2 \\ d_{31}^2 & d_{32}^2 & 0 & \dots & d_{3n}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{n1}^2 & d_{n2}^2 & d_{n3}^2 & \dots & 0 \end{bmatrix}$$

Para conectar todos los vértices originalmente no conexos, se ha implementado el algoritmo A*.

Las aristas que componen el genotipo de un individuo son de la forma:

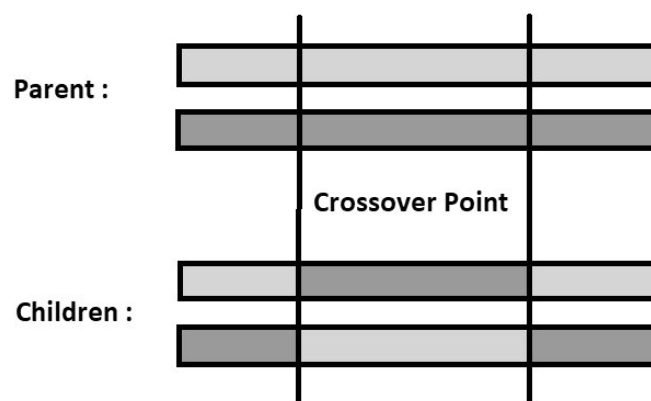
$$G = [x_0, x_1, x_2, \dots, x_{n-1}]$$

Y por tanto su función de coste:

$$f(x) = \text{dist}(y_{n-1}, y_0) + \sum_{i=1}^n \text{dist}(x_i, x_{i-1})$$

Siendo **X** los nodos del grafo transformado de aristas y el par de valores **Y** la distancia entre el vértice final de la solución y de partida. En cualquier caso, suponemos que la primera arista del genotipo contiene el nodo de partida del cartero.

La selección de padres se realiza por ruleta sin discriminar entre individuos, el 30% de los mejores pasan a la siguiente solución, cada 5 iteraciones se realiza una compactación de la población y el cruce entre individuos se realiza en bloques intermitentes de longitud variable, como se muestra en la figura:



ENFRIAMIENTO SIMULADO

Con todo lo descrito anteriormente, el enfriamiento simulado difiere en el método de generación de vecinos. Al tratarse de un problema combinatorio, los vecinos se generan a partir de permutaciones sobre una solución inicial, generalmente proveniente de un algoritmo voraz. Enfriamiento simulado requiere de una temperatura inicial y ratio de enfriamiento como parámetros.

$$G = [x_0, x_{n-1}, x_2, \dots, x_1]$$

Neighbourhood generated by swapping!

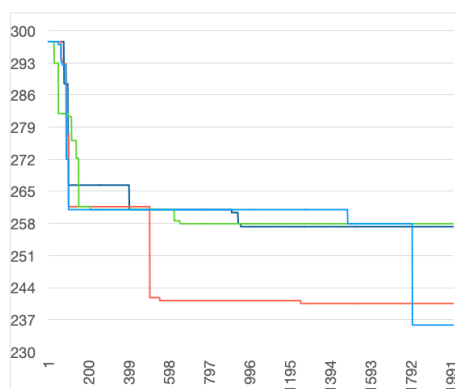
ALGORITMO MEMÉTICO

En esencia es el mismo algoritmo genético descrito anteriormente, sin embargo, cada 50 iteraciones utiliza enfriamiento simulado para mejorar el mejor individuo de la población.

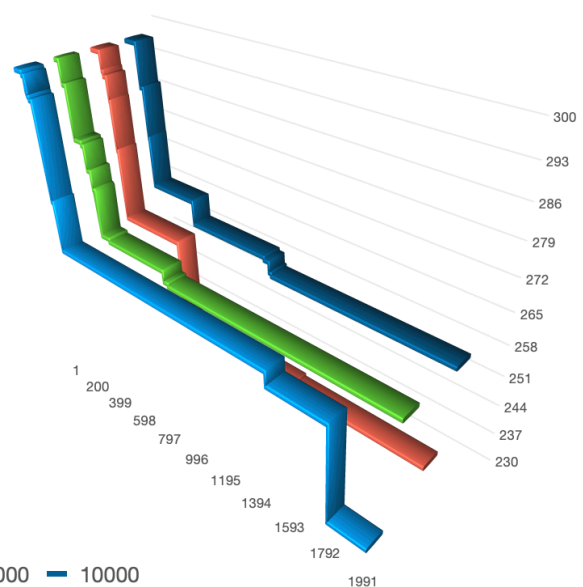
MÉTRICAS

2,000 iter. 77 edges	Greedy	Simulated Annealing	Genetic Algorithm	Memetic Algorithm
Best fitness	350.47	287.60	236.48	235.20
Fitness Calls	1	250,800	9,127,701	29,224,069
Computation time	10 ms	2.3 s	4 min.	6 min.

EVOLUCIÓN DE LA POBLACIÓN SEGÚN MEMÉTICO



— 10 — 100 — 1000 — 10000



EVOLUCIÓN DE LOS RESULTADOS EN FUNCIÓN DE LOS PARÁMETROS

ALGORITMO GENÉTICO

- **Número de iteraciones:** Indica el número máximo de generaciones, a mayor número de iteraciones, mayor probabilidad de encontrar el óptimo global.
- **Tamaño de la población:** Indica el número de individuos de la población, a mayor número, explora más pero hace que se ejecute y converja más lento. Para este problema el número óptimo está en 10 individuos.
- **Vértice inicial:** Punto de partida del cartero.
- **Probabilidad de mutación:** Probabilidad de que un individuo mute tras el cruce. A menor valor, el algoritmo puede estancarse, sin embargo, un valor demasiado alto genera demasiada dispersión.
- **Ratio de reemplazo:** Porcentaje de individuos de la anterior generación que pasan a la siguiente. A mayor valor, la población puede estancarse más fácilmente y si se toman valores menos conservadores pueden perderse soluciones prometedoras.
- **Ratio de compactación:** Número de iteraciones tras la que se eliminan individuos repetidos que se sustituyen por otros de generación aleatoria. Una excesiva compactación puede provocar ralentizaciones en la ejecución.

ALGORITMO MEMÉTICO

- **Memetización:** Número de iteraciones tras las que se mejora el mejor individuo de la población via enfriamiento simulado.

ENFRIAMIENTO SIMULADO

- **Temperatura:** Temperatura inicial.
- **Ratio de enfriamiento:** Velocidad de enfriamiento, afecta al tiempo de ejecución de enfriamiento simulado. Es un valor real positivo mayor que cero y menor que uno.

TECNOLOGÍAS UTILIZADAS

Para el desarrollo de la aplicación se ha utilizado Java 8 como lenguaje principal y Maven para la gestión de dependencias como JFreeChart y OpenCSV.

CONCLUSIONES

Como se ha podido observar, el algoritmo memético es el que aporta la mejor solución a igualdad de condiciones. Sin embargo, el coste computacional y el número de llamadas a la función fitness que realiza, lo hace demasiado lento en comparación.

Para disminuir el impacto en el rendimiento que producen una serie de llamadas repetidas a la función de fitness (véase al ordenar la población), se ha optado por almacenar el valor de este en un atributo privado, por lo que solo se calculará una vez y solo se volverá a computar si el genotipo se modifica.

Cabe resaltar que, tanto los algoritmos genético, memético y enfriamiento simulado parten de la misma solución generada por un algoritmo voraz. Esto es así pues el espacio de búsqueda es demasiado grande (77 nodos y 123 aristas) como para que puedan converger en un tiempo razonable.