Integrantes
Juan David Obando Novoa 202123148
Alejandro Mariscal

1. Algoritmo de solución

Nuestro algoritmo se divide en dos etapas fundamentales. La primera de encontrar un camino y la segunda calcular la mínima cantidad de LTP.

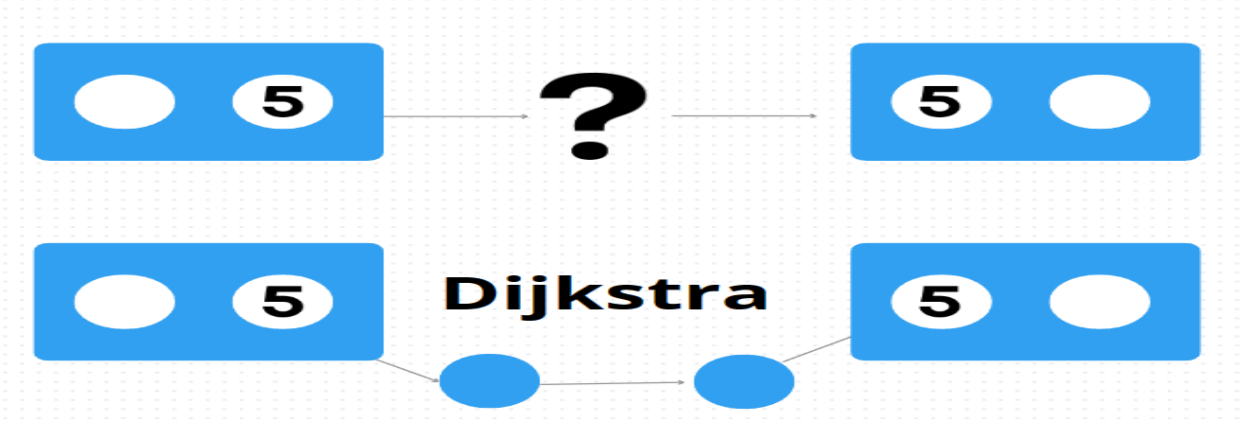
Primera etapa: Encontrar un camino

El propósito de esta etapa es encontrar un camino en el que todos los elementos fundamentales se puedan unir entre ellos por medio de un enlace bolt. Con todos los elementos vamos a encontrar un camino euleriano que pase pase por todos los vértices pero que no repita aristas. La lógica del algoritmo es la siguiente: Vamos a tener una lista, si la lista de aristas está vacía, entonces hemos encontrado un camino que visita todas las aristas, por lo que devolvemos el camino euleriano. Si la solución parcial no está vacía, intentamos extenderla seleccionando una arista que comienza o termina en el último nodo visitado (representado por "inicio"). Si la solución parcial está vacía, intentamos comenzar un camino en un nodo que aparece un número impar de veces en las aristas, si existe tal nodo. Si no, simplemente comenzamos en el primer nodo de la primera arista.De esta manera logramos obtener el camino que debe seguir el compuesto.

Segunda etapa: Calcular la mínima cantidad de LTP

El propósito de esta etapa consiste en que dado un camino se va a calcular la mínima cantidad de ltp que se necesita. Esto es importante porque nos dirá cuales son los átomos libres que necesitamos para unir los compuestos de camino.

Comenzamos con la lista que contiene el camino que hallamos en la primera etapa. Con este camino creamos una lista de adyacencia que tendrá como llaves los valores positivos y negativos de cada elemento dentro de los elementos fundamentales . Ya con eso, calcularemos la energía requerida con la ecuación de los ltps desde cada llave al resto de ellas. Una vez calculadas las energías haremos dijkstra por cada conexión, hallando así, el camino mínimo entre elementos fundamentales (cabe recalcar que este proceso se puede optimizar utilizando dp). Una vez hallamos la energía mínima de cada unión de elementos, la sumamos y conseguimos el total requerido. De esta manera hallamos el mínimo costo de LTP requerido por el compuesto.



2. Análisis de complejidades espacial y temporal Cálculo de complejidades y explicación de estas.

Temporal: En el caso de nuestro algoritmo, Dijkstra es el que toma más tiempo en realizarse ya que tiene una complejidad de O(v^2) donde cada nodo (V) representa un átomo dentro del grafo al cual le estamos realizando el proceso para identificar cual es el camino con el menor número de LTPs, sin embargo, este proceso se realiza un total de N veces, donde N es el número de compuestos fundamentales que se nos da en el problema, de modo que la complejidad total de este algoritmo viene dada por O(N*V^2)

Espacial: Para la complejidad espacial se analiza que inicialmente en memoria se utilizan los espacios para N nodos que viene siendo el mismo N que se tuvo en el numeral anterior. Así al crear las cadenas entre nodos se utilizan x átomos libres donde 0<x<N y x es un número entero, así a lo sumo se puede tener una complejidad espacial de O(N+xN) pues contamos con los N nodos iniciales más N veces (realmente sería N-1 veces pero eso produce una constante que se desprecia para la complejidad) el número de x átomos entre las conexiones, es decir, O(N)

3. Respuestas a los escenarios de comprensión de problemas algorítmicos. Respuesta a las preguntas establecidas en cada escenario. No tiene que implementar la solución a estos escenarios, el propósito es meramente analítico.

3.1 ESCENARIO 1: Se admiten compuestos en los que un elemento fundamental aparezca más de una vez

a. Nuevos retos

La presencia de elementos fundamentales repetidos aumenta la complejidad del problema, ya que se debe considerar la gestión de múltiples instancias del mismo elemento en el compuesto final. Esto podría afectar la complejidad temporal del algoritmo, especialmente en la función

buscarFundamentales, que realiza una búsqueda exhaustiva de todas las permutaciones posibles de la lista de entrada.

b. Nuevos cambios

Para abordar este escenario, se debe eliminar la restricción que exige la unicidad de los elementos fundamentales. Esto se puede lograr eliminando la declaración set que se utilizaba para almacenar los elementos y reemplazándola por una estructura de datos que permita almacenar elementos repetidos.

3.2 ESCENARIO 2: Se admiten compuestos que incluyan directamente enlaces toll

a. Nuevos retos

La inclusión de enlaces toll introduce una nueva dimensión de complejidad, ya que se deben evaluar las posibles combinaciones de enlaces bolt y toll para formar el compuesto final. Esto puede aumentar la cantidad de soluciones válidas y complicar la búsqueda de la solución óptima.

b. Nuevos cambios

Para incorporar enlaces toll, se deben realizar modificaciones en la función buscarFundamentales. La lógica de búsqueda de un nodo para extender la solución parcial debe considerar no solo los enlaces bolt, sino también los enlaces toll con el mismo peso pero diferente carga.

Además, se debe implementar un mecanismo para distinguir entre enlaces bolt y toll en el cálculo de los LTP, evitando que se calculen los LTP de los enlaces toll. Esto se puede lograr mediante un indicador que diferencie los dos tipos de enlaces.