Documentación Proyecto 2 Diseño y Análisis de algoritmos

Alejandro Pardo - 202223709

Alejandro Lancheros - 202122797

1. Algoritmo de solución

Para nuestra solución primero implementamos los métodos necesarios para la creación del grafo, estos incluyen el crear los nodos y arcos, luego, implementamos los métodos para poder evaluar las distancias entre cada nodo, nótese que incluimos tanto calcular la distancia manhattan, como calcular la distancia euclidiana, sin embargo, para resolver el problema se decidió hacer uso de la distancia Manhattan (por el bono). Después, para el análisis del flujo implementamos primero el algoritmo de búsqueda en amplitud (BFS), ya que, el algoritmo Edmonds-Karp (El cual calcula el flujo máximo del grafo) hace uso de este.

El algoritmo Edmonds-Karp calcula el flujo máximo en nuestro grafo dirigido desde un nodo fuente (i) hasta un nodo sumidero (f). Primero inicializa el flujo máximo en cero, a medida que encuentra caminos aumentantes irá sumando su capacidad. A partir del diccionario 'parent' se rastrea el nodo desde el cual se llegó a cada vértice en el camino encontrado. Si no se encuentra un camino se detiene el bucle. Luego, se sigue el camino desde el nodo final hacia el inicial usando la información almacenada en el diccionario 'parent', para cada arista en el camino se actualiza al mínimo entre su capacidad y el flujo mínimo actual. Finalmente, se después de varias iteraciones retorna el flujo total.

Por cada nodo dentro del grafo, se obtiene la información necesaria (id, tipo, posicionx, posiciony y péptidos) almacenando dicha información en variables independientes, luego, se itera sobre los posibles nodos adyacentes, en dichos nodos se evalúa bajo condicionales acorde al enunciado la posibilidad de conexión bajo un arco, en caso de que se pueda realizar la conexión se crea el arco indicando el id del nodo adyacente y el número de péptidos que comparten ambas células (nodos).

Inicialmente, para la búsqueda de la célula a bloquear se estableció el flujo total sin nodos eliminados en infinito, también se realiza una copia de la lista de adyacencias original, luego se itera sobre cada célula calculadora potencial del grafo para borrar las conexiones que se generaron en la creación del grafo, con esta nueva lista de adyacencias modificada ejecutamos el algoritmo Edmonds-Karp para conocer el flujo total con esta célula eliminada. Por último, se el flujo anterior con el actual para obtener de esta manera el flujo mínimo posible.

2. Análisis de complejidad espacial y temporal

En el desarrollo de nuestro algoritmo consideramos dos puntos importantes que influyen en la complejidad espacial y temporal, uno de ellos es la creación del grafo la cual después de analizar los ciclos de alta complejidad dentro de esta función entendimos que dicha complejidad temporalmente en notación BIG O es $O(V^2)$, donde V es el número de vértices o células en el grafo. Además, el otro componente costoso que se logró considerar fue el algoritmo que determina que célula calculadora se debe eliminar, donde luego de analizar detenidamente se entendió que la complejidad temporal en notación BIG O $O(K \cdot V \cdot E^2)$, donde K es el número de células calculadoras potenciales a eliminar.

Para la complejidad temporal se entendió que esta se maneja en función de la lista de adyacencias del grafo generado, siendo así la memoria necesaria para almacenar esta lista, en notación BIG O es $O(V + E \quad)$.

Se debe tener en cuenta que ambas complejidades aumentan dependiendo de la cantidad de los casos que se pruebas, es decir, si se hacen N pruebas las complejidades totales serian la multiplicación de N por la complejidad respectiva del algoritmo.

3. Respuesta a los escenarios

Escenario 1:

(i) ¿Qué nuevos retos presupone este nuevo escenario -si aplica-?

Deberíamos tener en cuenta en el momento de crear el grafo este tipo de nuevas conexiones entre células. Después de analizar la situación, logramos entender que el flujo total de mensajes que llegan a las células ejecutoras sería mayor, pues recibirían tanto de las calculadoras como de las iniciadoras.

(ii) Que cambios -si aplica- le tendría que realizar a su solución para que se adapte a este nuevo escenario?

Al momento de crear los arcos del grafo incluiríamos la posibilidad de que una célula de tipo iniciadora lograra conectar con una de tipo ejecutara, tendríamos en cuenta también la cantidad de péptidos que comparten dichas células para así determinar los mensajes que se envían entre sí.

Escenario 2:

(i) ¿Qué nuevos retos presupone este nuevo escenario -si aplica-?

Tendríamos que definir la capacidad de envío de cada célula calculadora como un único mensaje emitido posible, para que de esta manera se cumpla la condición planteada en el escenario. Además, se observa que el tamaño del grafo aumentaría considerablemente lo cual, a su vez, aumentaría la complejidad espacial a la hora de resolver el problema.

(ii) Que cambios -si aplica- le tendría que realizar a su solución para que se adapte a este nuevo escenario?
Al momento de obtener los péptidos comunes entre células, debemos considerar que ahora las células calculadoras únicamente envían un mensaje, por lo que cuando se crean los arcos del grafo para el caso de este tipo de células el peso de salida de todos sus arcos seria 1. Además, entendimos que la respuesta esperada al momento de eliminar una célula calculadora para minimizar los mensajes que se pueden enviar entre sí no va a cambiar dependiendo de la célula escogida a eliminar siempre y cuando estén conectadas a alguna célula ejecutora, pues todas ellas envían un solo mensaje.