TP 1 - Algo III

Nicolas Celie

April 2023

1 Problema a resolver

Dado un conjunto de actividades $A = \{A_1, ..., A_n\}$, el problema de selección de actividades consiste en encontrar un subconjunto de actividades S de cardinalidad máxima, tal que ningún par de actividades de S se solapen en el tiempo. Cada actividad A_i se realiza en algún intervalo de tiempo (s_i, t_i), siendo $s_i \in N$ su momento inicial y $t_i \in N$ su momento final. Suponemos que $1 \leq s_i < t_i \leq 2n$ para todo $1 \leq i \leq n$.
Considerar la siguiente estrategia golosa para resolver el problema de selección de actividades: elegir la actividad cuyo momento nal sea lo más temprano posible, de entre todas las actividades que no se solapen con las actividades ya elegidas. Demostrar que un algoritmo goloso que implementa la estrategia anterior es correcto.

2 Demostración

En primera instancia, nuestro algoritmo greedy para resolver el problema se basará en ordenar los intervalos de cada actividad tomando como parámetro de ordenamiento al límite superior de cada intervalo (t_i). Luego tomaremos el primer intervalo y a partir de él recorreremos (en el orden establecido) los siguientes, agregando a la solución siempre el primer intervalo que tenga límite inferior mayor o igual al límite superior del ultimo intervalo elegido. Es decir, suponiendo $t_1 \leq t_2 \leq ... \leq t_k \leq ... \leq t_i$, si el ultimo intervalo elegido fue $[s_k, t_k]$ entonces el siguiente intervalo $[s_{k+1}, t_{k+1}]$ será tomado si y solo si $s_{k+1} \geq t_k$.

¿Por qué nuestro algoritmo greedy es correcto?

Inicialmente vamos a demostrar que todo intervalo con el mínimo límite superior siempre va a pertenecer a alguna solución óptima.
Supongamos $S = \{S_1, S_2, ..., S_k\}$ el conjunto de soluciones óptimas posibles y $A_O = \{[s_1, t_1], [s_2, t_2], ..., [s_i, t_i]\}$ el conjunto de actividades ordenadas de la forma ya descripta.

Quiero ver que:
Como $t_1 = min(\{t/t \in A_{O1}(Conjunto\ de\ limites\ superiores\ de\ A_O)\}) \Longrightarrow t_1 \in S_O$ para algún $S_O \in S$

Supongamos que $\exists B \subset A$ tal que $B \in S$ y $t_1 \notin B$.

Entonces:
Si $\#B = 1 \Longrightarrow t_1$ puede tomar el lugar de aquel $t \in B$

Si $\#B > 1 \Longrightarrow \exists t_a, t_b \in B$ tales que $\#([s_a, t_a] \cap [s_b, t_b]) \leq 1 \Longrightarrow t_a \leq s_b$
Como $t_1 = min(\{t/t \in A_{O1}\}) \Longrightarrow t_1 < t_a \Longrightarrow t_1 \leq s_b \Longrightarrow (B - \{[s_a, t_a]\}) \cup \{[s_1, t_1]\} \in S$

Basicamente en esta ultima línea estamos diciendo que si B tiene más de un intervalo, entonces existen al menos dos elementos que no se intersecan o solo lo hacen en sus extremos (cardinal de su intersección menor o igual a 1). Como esto ocurre y sabemos que t_1 es el mínimo límite superior del conjunto A, entonces t_1 puede ocupar el lugar del intervalo con menor límite superior de estos dos que estamos contemplando y, por construcción, tendríamos un B' tal que $B' \in S$.
Con esto demostramos que siempre podremos llegar a una solución óptima tomando algun intervalo con el

menor limite superior.

Ahora veamos que, siguiendo la lógica de siempre tomar el primer intervalo que no se solape con el anterior, obtenemos una solución óptima.
Lo demostraremos utilizando el siguiente invariante:
Q(i): "S_O es un conjunto de intervalos que no se solapan entre sí y está contenido en una solución óptima".

Probamos por inducción (Consideramos S una solución óptima existente):

Caso Base: Q(1): $[s_1, t_1] \in S_O$ y $S_O \in S$
Trivialmente porque ya probamos que siempre va a poder estar en una solución optima.

Paso inductivo: Suponiendo que se cumple Q(i), quiero ver que vale Q(i+1)

$\Longrightarrow$ Q(i): $S_O = \{[s_1, t_1], [s_2, t_2], ..., [s_i, t_i]\}$ y $S_O \in S$
Quiero ver que: Q(i+1): $S_O \cup \{[s_{i+1}, t_{i+1}]\}$ y $S_O \cup \{[s_{i+1}, t_{i+1}]\} \subset S$

Tenemos 4 casos:

Caso 1: No agregamos $[s_{i+1}, t_{i+1}]$ a S_O y $S_O \cup \{[s_{i+1}, t_{i+1}]\} \not\subset S$
Esto es verdadero trivialmente porque sigue valiendo la implicación. Es decir, sigue habiendo al menos una extensión del conjunto S_O que es igual a S.

Caso 2 y 3: Agregamos $[s_{i+1}, t_{i+1}]$ a S_O y $S_O \cup \{[s_{i+1}, t_{i+1}]\} \not\subset S$ o
No agregamos $[s_{i+1}, t_{i+1}]$ a S_O y $S_O \cup \{[s_{i+1}, t_{i+1}]\} \subset S$

Sabemos que $S_O \subset S \Longrightarrow \{[s_1, t_1], ..., [s_i, t_i]\} \subset S$

- Si $s_{i+1} < t_i \Longrightarrow$ Los intervalos se solapan $\Longrightarrow$ Absurdo, pues nuestro algoritmo se encargaba se asegurarse que $s_{k+1} \geq t_k$ para incluirlo en el conjunto óptimo.
- Si $s_{i+1} \geq t_i$, llamemos t_{optimo} al límite superior de un intervalo óptimo de la posición i+1.
  1. Veamos el caso en el que $t_{i+1} \leq t_{optimo}$.
     Análogamente a la demostración de por qué todo intervalo con mínimo límite superior siempre puede pertenecer a una solución óptima. En este caso, tomar el intervalo de t_{i+1} o tomar el intervalo de t_{optimo} sería igual de eficiente. Veamos que el siguiente intervalo óptimo tendrá un límite inferior, llamémoslo $s_{optimo2}$, tal que $s_{optimo2} \geq t_{optimo} \geq t_{i+1} \Longrightarrow s_{optimo2} \geq t_{i+1}$, luego, también sería óptima la elección.
  2. Por otro lado, veamos $t_{i+1} > t_{optimo}$
     Absurdo, pues como los t_i estaban ordenados de menor a mayor, t_{i+1} nunca será mayor que t_{optimo}

Caso 4: Agregamos $[s_{i+1}, t_{i+1}]$ a S_O y $S_O \cup \{[s_{i+1}, t_{i+1}]\} \subset S$
Trivialmente, al igual que el primer caso, se cumpliría que sigue valiendo la implicación anterior porque la extensión óptima de la sigue siendo la misma.