Universidad Nacional Autónoma de México

FACULTAD DE CIENCIAS

Análisis y Diseño de Algoritmos

TAREA2

Martínez Avila Santiago
Reynoso Sánchez Arturo Yitzack

29 de octubre de 2021

1

Peso: 10 puntos. Considera el problema de la *mochila fraccional.* Entrada: una capacidad de mochila C y una secuencia de n parejas $item_i = (v_i, w_i)$, de tal forma que el elemento i-ésimo tiene valor total v_i, y su peso total es w_i. Salida: un vector $q_1, ..., q_n$ con entradas en $[0, 1]$, que indican las fracciones del total de los respectivos elementos, de tal forma que:

- La suma de los pesos de las fracciones de los elementos es compatible con la capacidad de la mochila ($\sum_{i=1}^{n} q_i \cdot w_i \leq C$), y
- La suma de las fracciones de los valores de los elementos ($\sum_{i=1}^{n} q_i \cdot v_i \leq V$) donde V sea el mayor valor total posible.

Por simplicidad, y para que la solución sea única, supón que el valor unitario (v_i/w_i) de cada elemento es único.
Para este problema:

1. Enuncia y demuestra un lema de intercambio, que te permita transformar toda solución no óptima A en una solución de mejor valor A'.

   Lema de intercambio.

   Supongamos que $\sum_{i=1}^{n} w_i > C$ (el peso total de los elementos excede la capacidad C de la mochila) y además:

   - Sea A un arreglo de n elementos, con entradas $A[1]$, $A[2]$, ..., $A[n]$, donde $A[k]$ representa la fracción del total del k-ésimo elemento ($A_k = q_k$) y A es compatible con la capacidad de la mochila.
   - Sean $i, j \in \{1, 2, ..., n\}$ donde $\frac{v_i}{w_i} > \frac{v_j}{w_j}$ (el valor unitario de i es mayor que el de j)
   - $q_j > 0$ y $q_i < 1$.

   Sea $y > 0$ tal que $0 < y \leq q_j \cdot w_j \quad \text{y} \quad 0 < y \leq (1 - q_i) \cdot w_i$. Definimos un arreglo B como:

   - $B[k] = q_k$ para $k \neq i$ y $k \neq j$.
   - $B[i] = q_i + \frac{y}{w_i}$
   - $B[j] = q_j - \frac{y}{w_j}$

   es decir, en B quitas una cantidad y del elemento j y lo aumentas al elemento i. Entonces la suma de pesos de los elementos de B es compatible con la capacidad de la mochila y la suma de valores de los elementos de B es mayor que la suma de valores de los elementos de A.

   Demostración.

   Sean A y B con las condiciones dadas en el lema, V el valor de los elementos de A y V' el valor de los elementos de B. Tenemos entonces:

el peso de los elementos en B es el mismo que el de A:

$$\begin{aligned}
\text{Capacidad de B} &= \sum_{\substack{k=1\\k\neq i\\k\neq j}}^{n} q_k \cdot w_k + \left(q_i + \frac{y}{w_i}\right) \cdot w_i + \left(q_j - \frac{y}{w_j}\right) \cdot w_j \\
&= \sum_{\substack{k=0\\k\neq i\\k\neq j}}^{n} q_k \cdot w_k + q_i \cdot w_i + y + q_j \cdot w_j - y \\
&= \sum_{k=1}^{n} q_k \cdot w_k = \text{ Capacidad de A}
\end{aligned}$$

La suma de valores de elementos en B es mayor que en A:

$$V = \sum_{\substack{k=1\\k\neq i\\k\neq j}}^{n} q_k \cdot v_k + q_i \cdot v_i + q_j \cdot v_j$$

$$V' = \sum_{\substack{k=1\\k\neq i\\k\neq j}}^{n} q_k \cdot v_k + \left(q_i + \frac{y}{w_i}\right) \cdot v_i + \left(q_j - \frac{y}{w_j}\right) \cdot v_j$$

Entonces:

$$\begin{aligned}
V' - V &= \frac{y}{w_i} \cdot v_i - \frac{y}{w_j} \cdot v_j \\
&= y\left(\frac{v_i}{w_i} - \frac{v_j}{w_j}\right) > 0
\end{aligned}$$

2. Demuestra con el lema anterior, que la solución es única (usando la hipótesis de que los valores unitarios no se repiten).

   Demostración

   Sean $A = [q_{A1}, ..., q_{An}]$ y $B = [q_{B1}, ..., q_{Bn}]$ dos soluciones distintas al problema. Primero notemos que la suma de los pesos de las fracciones de ambos arreglos es C, ya que si no fuera así, podrías aumentar la fracción de algún elemento y te generará un mayor valor. Además como A y B son soluciones al problema, alcanzan el máximo valor posible, digamos V.

   Notamos que existen índices i y j distintos tales que

   $$B[i] > A[i] \text{ y } B[j] < A[j]$$

   ya que A y B tienen la misma capacidad C y el mismo valor V pero son distintos. Como los valores unitarios no se repiten, entonces tenemos que el elemento i o j tiene mayor valor unitario que el otro. Digamos que el elemento i tiene mayor valor unitario que j.

Entonces tenemos $q_{Aj} > 0$ y $q_{Ai} < 1$. Por el **lema de intercambio**, podemos construir otro arreglo D cuya suma de valores de sus elementos es mayor al de A. Pero esto es una contradicción al hecho de que A y B tenían el valor máximo posible. Por lo tanto, la hipótesis de que A y B son dos soluciones diferentes es incorrecta. Por lo tanto, la solución del problema es única.

3. Enuncia y demuestra un lema que caracterice la estructura de la solución única (que es la única a la que ya no se le puede aplicar un intercambio que la mejore).

   Lema Sea $i_1, i_2, ..., i_n$ un reordenamiento de los elementos de manera descendente en valor unitario. La estructura de la solución única es la siguiente:

$$(*) \ldots q_{i_1} = \begin{cases} 1, & \text{si } w_{i_1} \leq C \\ \frac{C}{w_{i_1}}, & \text{si } w_{i_1} > C \end{cases}$$

$$(**) \ldots q_{i_j} = \begin{cases} 1, & \text{si } w_{i_j} \leq C - \sum\limits_{k=1}^{j-1} q_{i_k} \cdot w_{i_k} \\ \frac{C - \sum\limits_{k=1}^{j-1} q_{i_k} \cdot w_{i_k}}{w_{i_j}}, & \text{si } w_{i_j} > C - \sum\limits_{k=1}^{j-1} q_{i_k} \cdot w_{i_k} \end{cases} \qquad \text{para } j = 2, ..., n$$

   Demostración

   - **n=1**
     - Caso 1 $w_{i_1} \leq C$
       $\implies q_{i_1} = 1$
       Sea $0 \leq q < 1 \implies q \cdot v_{i_1} < v_{i_1} = q_{i_1} \cdot v_{i_1}$
       $\therefore q_{i_1}$ es solución y es mayor que cualquier otra solución.
     - Caso 2 $w_{i_1} > C$
       $\implies q_{i_1} = \frac{C}{w_{i_1}}$
       Si $q > \frac{C}{w_{i_1}} \implies q \cdot w_{i_1} > C \implies q$ no es solución
       Si $q < \frac{C}{w_{i_1}} \implies q \cdot v_{i_1} < C\frac{v_{i_1}}{w_{i_1}} = q_{i_1} \cdot v_{i_1}$
       $\therefore q_{i_1}$ es solución y es mayor que cualquier otra solución.

     De ambos casos, q_{i_1} como en (*) es la estructura de la solución única.
   - **H.I.**
     Supongamos que para toda n, q_{i_1} como en (*) y q_{i_j} como en (**) es la estructura de la solución única.
   - **P.I.**
     Sea $i_1, i_2, ..., i_n, i_{n+1}$ un reordenamiento de los elementos de manera descendente en valor unitario. Consideremos únicamente los primeros n elementos del reordenamiento.

Por H.I., q_{i_1} como en (*) y q_{i_j} como en (**) es la estructura de la solución única. Ahora veamos qué pasa con $q_{i_{n+1}}$

- Caso 1 $w_{i_{n+1}} \leq C - \sum_{k=1}^{n} q_{i_k} \cdot w_{i_k}$

  $\implies q_{i_{n+1}} = 1 \implies q \cdot w_{i_{n+1}} \leq C - \sum_{k=1}^{n} q_{i_k} \cdot w_{i_k} \implies \sum_{k=1}^{n+1} q_{i_k} \cdot w_{i_k} \leq C$

  $\therefore q_{i_j}$ como en (*) y (**) es solución.

  Sea $0 \leq q < 1$

  $\implies q \cdot v_{i_{n+1}} < v_{i_{n+1}} = q_{i_{n+1}} \cdot v_{i_{n+1}} \implies \sum_{k=1}^{n} q_{i_k} \cdot w_{i_k} + q \cdot v_{i_{n+1}} < \sum_{k=1}^{n+1} q_{i_k} \cdot w_{i_k}$

  Por lo que $q_{i_{n+1}}$ es solución parcial y es mayor que cualquier otra solución parcial. Y como el resto de q_{i_j} ya eran óptimos para n elementos, $q_{i_1}, ..., q_{i_n}, q_{i_{n+1}}$ es la mejor solución.

- Caso 2 $w_{i_{n+1}} > C - \sum_{k=1}^{n} q_{i_k} \cdot w_{i_k}$

  $\implies q_{i_{n+1}} = \frac{1}{w_{i_{n+1}}}(C - \sum_{k=1}^{n} q_{i_k} \cdot w_{i_k}) \implies q_{i_{n+1}} \cdot w_{i_{n+1}} = C - \sum_{k=1}^{n} q_{i_k} \cdot w_{i_k} \implies \sum_{k=1}^{n+1} q_{i_k} \cdot w_{i_k} = C$

  $\therefore q_{i_j}$ como en (*) y (**) es solución.

  Si $q > \frac{1}{w_{i_{n+1}}}(C - \sum_{k=1}^{n} q_{i_k} \cdot w_{i_k}) \implies q_{i_{n+1}} \cdot w_{i_{n+1}} > C - \sum_{k=1}^{n} q_{i_k} \cdot w_{i_k} \implies \sum_{k=1}^{n} q_{i_k} \cdot w_{i_k} + q_{i_{n+1}} \cdot w_{i_{n+1}} > C \implies q$ no es solución.

  Si $q < \frac{1}{w_{i_{n+1}}}(C - \sum_{k=1}^{n} q_{i_k} \cdot w_{i_k}) \implies q < q_{i_{n+1}} \implies q \cdot v_{i_{n+1}} < q_{i_{n+1}} \cdot v_{i_{n+1}} \implies \sum_{k=1}^{n} q_{i_k} \cdot w_{i_k} + q \cdot v_{i_{n+1}} < \sum_{k=1}^{n+1} q_{i_k} \cdot w_{i_k}$

  Por lo que $q_{i_{n+1}}$ es solución parcial y es mayor que cualquier otra solución parcial. Y como el resto de q_{i_j} ya eran óptimos para n elementos, $q_{i_1}, ..., q_{i_n}, q_{i_{n+1}}$ es la mejor solución.

De ambos casos, q_{i_j} como en (*) y (**) es la estructura de la solución única.

4. Propón un algoritmo que, apoyándose en el lema anterior, dada una instancia del problema, construya la solución óptima, en tiempo $O(n \log n)$.

   - Demuestra que el algoritmo es correcto.
     Demostración
     El algoritmo empieza creando dos listas (o vectores) U y Q; las entradas de U corresponden al valor unitario $\frac{v_i}{w_i}$ y el índice i, y las entradas de Q son 0 pues sólo se necesitaba crear un vector de n entradas.

Algorithm 1 Mochila Fraccional

```
procedure Mochila(item_1, item_2, ..., item_n, C)
    C' = 0
    U=[ ]
    Q=[ ]
    for i in range(n) do
        Q.append(0)
        U.append(item_i[0] / item_i[1], i)
    end for

    MergeSort(U)        ▷ Ordenar U en orden descendente con respecto de item_i[0] / item_i[1]

    i1=U[0][1]
    if item_i1[1] ≤ C then
        Q[i1]=1
        C' = item_i1[1]
    else
        Q[i1] = C / item_i1[1]
        C'=C
    end if
    U.pop(0)

    for u in U do
        i=u[1]
        if item_i[1] ≤ C then
            Q[i]=1
            C' = C' + item_i[1]
        else
            Q[i] = (1 / item_i[1]) (C − C')
            C'=C
        end if
    end for
    return Q
end procedure
```

Lo siguiente es ordenar U en orden descendente con respecto a los valores unitarios, es decir el primer elemento de cada entrada. Aquí se usa una función auxiliar (MergeSort). Ahora es donde se usa el lema que caracteriza la estructura de la solución única. De acuerdo a este lema q_{i_1} es el único de los q_{i_j} que se calcula distinto por lo que primero calculamos este usando la definición dada en el lema.
C' se utiliza para hacer la suma de los $q_{i_j} \cdot w_{i_j}$
Si $q_{i_1} = 1$ entonces $q_{i_1} \cdot w_{i_1} = C$ por lo que $C' = C$. Y si $q_{i_1} = \frac{C}{w_{i_1}}$ entonces $q_{i_1} \cdot w_{i_1} = w_{i_1}$ por lo que $C' = w_{i_1}$. Una vez realizado esto, se quita el primer elemento de U pues ya no se necesita volver a usar.
Usando la definición del lema, calculamos el resto de los q_{i_j}. Aquí es donde se empieza a usar *C'* ya que del Lema, es necesario conocer la suma de lo $q_{i_1} \cdot w_{i_1}, \ldots, q_{i_{j-1}} \cdot w_{i_{j-1}}$ para calcular $q_{i_j} \cdot w_{i_j}$. Y notemos que si $q_{i_j} = 1$, $\sum\limits_{k=1}^{j} q_{i_k} \cdot w_{i_k} = \sum\limits_{k=1}^{j-1} q_{i_k} \cdot w_{i_k} + w_{i_1}$ por lo que $C' = C' + item_i[1]$. Y si $q_{i_j} = \frac{1}{w_{i_j}}(C - \sum\limits_{k=1}^{j} q_{i_k} \cdot w_{i_k})$, $q_{i_j} \cdot w_{i_j} = C - \sum\limits_{k=1}^{j} q_{i_k} \cdot w_{i_k}$ y así $\sum\limits_{k=1}^{j} q_{i_k} \cdot w_{i_k} + q_{i_j} \cdot w_{i_j} = C$ por lo que $C' = C$.
Finalmente notemos que como lo que hace el algoritmo es encontrar el valor de q_{i_j} de acuerdo al Lema, y el Lema da la estructura de la solución única, el algoritmo nos da la solución única. Por lo tanto, el algoritmo es correcto.

- Demuestra que su complejidad es la requerida.
  Demostración

  Como los pasos con complejidad *O(1)* no afectan, solo nos interesan los ciclos *for* y *MergeSort*. Como los pasos dentro de los *for* tiene complejidad *O(1)*, la complejidad de los ciclos es *O(n)*. Y la complejidad de *MergeSort* es *O(n logn)*. Por lo tanto la complejidad del algoritmo es *O(n logn)*.