Algoritmos voraces. Ejercicio: El fontanero diligente[1]

- Un fontanero tiene que hacer n reparaciones urgentes, y sabe por adelantado el tiempo que necesitará para cada una de ellas: en la tarea $i-ésima$ tardará t_i minutos. Como en su empresa le pagan de acuerdo con la satisfacción del cliente, necesita decidir el orden en el cual atenderá los avisos para minimizar el tiempo medio de espera de los clientes.

- En otras palabras, si denominamos T_i el tiempo que espera el cliente $i-ésimo$ hasta que sea reparada su avería por completo, el fontanero necesita minimizar la expresión:

$$E_n = \sum_{i=1}^{n} T_i$$

Sea $m_j \in \{1, 2, \ldots, n\}$ $\forall j \in \{1, 2, \ldots, n\}$, la tarea que se realizará en $j-ésimo$ lugar. Claramente,

$$T_i = \sum_{j=1}^{i} t_{m_j}$$

Es decir, el tiempo que espera el cliente $i-ésimo$ es la suma del tiempo en realizar su tarea, que se realiza en $i-ésimo$ lugar, y el de todas las tareas que se han realizado con anterioridad. Por lo tanto,

$$E_n = \sum_{i=1}^{n}\sum_{j=1}^{i} t_{m_j} = \sum_{i=1}^{n}(n+1-i)t_{m_i},$$

el tiempo de espera sumado para los k primeros clientes contiene k veces el tiempo que tardó la reparación del primer cliente m_1, $k-1$ veces lo del cliente m_2, etc.

La solución óptima, de manera intuitiva, es dejar para el final los clientes con trabajos de mayor duración, puesto que aparecerán multiplicados por un factor $n+1-i$ más pequeño. Por lo tanto hay que seleccionar los índices $m_1, m_2, \ldots, m_n$ de forma que se cumpla una ordenación ascendente de tiempo. Es decir, si $i < j$ entonces $t_{m_i} \le t_{m_j}$ $\forall i, j \in [1, n]$.

- Demostración:
Si asumimos que esta distribución de las visitas es la óptima, cualquier permutación de dos visitas (por ejemplo, la visita al cliente $j \in [1, n-1]$ y la visita a un cliente más tardío $j+k$ con $k \in [1, n-j]$) tendría que aumentar el valor de E_n para cualesquier valores de j y k. Veámoslo:

Con la intención de clarificar el proceso, comenzaremos extrayendo del sumatorio que expresa el valor de E_n, las visitas j y $j+k$:

$$E_n = \sum_{\substack{i=1,\\ i\neq j,\\ i\neq j+k}}^{n} (n+1-i)t_{m_i} + (n+1-j)t_{m_j} + (n+1-(j+k))t_{m_{j+k}}$$

Llamando E'_n al nuevo tiempo de espera con la permutación mencionada y restando:

$$E'_n = \sum_{\substack{i=1,\\ i\neq j,\\ i\neq j+k}}^{n} (n+1-i)t_{m_i} + (n+1-(j+k))t_{m_j} + (n+1-j)t_{m_{j+k}}$$

$$\begin{aligned} E'_n - E_n &= (n+1-j)(t_{m_{j+k}} - t_{m_j}) + (n+1-(j+k))(t_{m_j} - t_{m_{j+k}}) \\ &= k(t_{m_{j+k}} - t_{m_j}) > 0 \end{aligned}$$

Es decir, el tiempo aumenta para cualquier permutación de dos clientes; por lo tanto, la distribución propuesta era la óptima. El problema permite una solución voraz: el fontanero atiende los clientes en orden ascendente del tiempo que dedicará a cada uno de ellos para minimizar el tiempo de espera total de su clientela.

[1]Este problema también se denomina problema del almacenamiento óptimo en cintas (“optimal storage on tapes”), Horowitz y Sahni (1978, p. 153–155).