Ros Roger, Alexandre

Abril 2023

5. Planificació. Tenim un conjunt de n tasques. La tasca i es descriu per un parell $T_i = (s_i, d_i)$ on s_i és el seu temps de disponibilitat i d_i la seva duració. L'objectiu és planificar totes les tasques en un processador de manera que cap tasca s'executa abans del seu temps de disponibilitat, les tasques s'executen sense interrumpció, i el temps de finalització del procesament de totes les tasques sigui mínim.

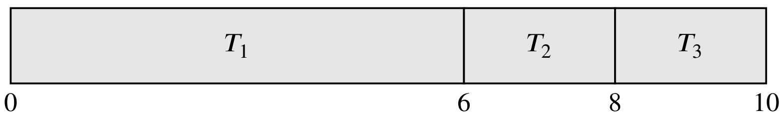
Se sap que, quan es possible interrompre qualsevol tasca en execuci o i reiniciar-la més tard des del punt d'aturada, l'algorisme voraç que executa en cada punt de temps la tasca (disponible) més propera al seu temps de finalització, aconsegueix el desitjat mínim. Anomenarem a aquest algorisme `Voraç1`. Considereu el procediment següent:

1. Executa `Voraç1`.
2. Ordena les tasques en l'ordre ascendent del seu temps de finalització d'acord amb la solució proporcionada per `Voraç1`.
3. Les tasues s'executeu en aquest ordre i sense interrupció, afegint el temps d'espera necessari fins que la tasca estigui disponible.

Demostreu que aquest algorisme es una 2-aproximació pel problema plantejat

Solució:

Primerament, podem observar que `opt(x)` $\geq \sum_i d_i + K$, on K és el temps perdut entre execucions del algorisme `Voraç1` (definim K com $T_{\text{voraç}} - \sum_i d_i$). En el cas òptim, totes les tasques podran ser executades seqüencialment, sense esperes entremig i sense interrupcions. El següent diagrama en mostra un exemple, on tenim les tasques $T_1 = (0, 6)$, $T_2 = (6, 2)$, $T_3 = (8, 2)$:



Seguidament, no és difícil veure que l'ordre d'execució de tasques que retorna el nostre algorisme ordena les tasques per $d_i + s_i$ ascendentment. Llavors, la primera tasca en el nostre algorisme acabarà en la marca $d_0 + s_0$ sempre. Seguidament, la segona acabarà a la marca de temps $\mathtt{max}(s_0 + d_0, s_1) + d_1$.

Recursivament, la tasca i-èsima acabarà a la marca $\mathtt{max}(t_{i-1}, s_i) + d_i$ on t_{i-1} és l'instant de finalització de la tasca anterior.

Veiem, però, que $\mathtt{max}\ s_i$ és com a màxim $\sum_i d_i + K$. Per tant, tenim que la última marca serà de com a màxim $2 * \sum_i d_i + K$ (per les sumes en l'expressió de la tasca i-èsima, hi haurà algun màxim on saltarà el $\mathtt{max}\ s_i$ i la resta són sumes de d_i).

Per tant:

$\sum_i d_i + K \leq \mathtt{opt(x)} \leq A(x) \leq 2 * \sum_i d_i + K \leq 2 * \left(\sum_i d_i + K\right)$.