Schémas algorithmiques
Annexe

Guillaume Rousseau
MP2I Lycée Pierre de Fermat
guillaume.rousseau@ens-lyon.fr

5 mai 2024

A Glouton des intervalles : preuve de correction

On considère une instance du problème de programmation d'évènements constituée des intervalles $(d_1, f_1), \ldots, (d_n, f_n)$ avec $n \in \mathbb{N}$. Montrons que l'algorithme glouton prenant les évènements par date de fin croissante renvoie une solution optimale. On suppose dans la suite que les évènements sont triés selon ce critère, i.e. que $f_1 \leq f_2 \leq \cdots \leq f_n$.

On considère $I^* = \{i_1 < \cdots < i_k\}$ une solution optimale, et $I = \{j_1 < \cdots < j_l\}$ la solution renvoyée par l'algorithme glouton. Notons qu'alors, par optimalité, $k \geq l$.
Montrons par récurrence que $\forall t \in [\![0, l]\!], I_t^* = \{j_1, \ldots, j_t, i_{t+1}, \ldots i_k\}$ est une solution admissible (et optimale, car de même cardinal que I^*).

- Initialisation : pour $t = 0$, on a $I_0^* = I^*$, c'est donc une solution admissible.
- Soit $t \in [\![0, l-1]\!]$. On suppose que I_t^* est une solution admissible. Montrons qu'alors I_{t+1}^* est aussi une solution admissible. On a :

$$\begin{array}{rcl} I_t^* & = & \{j_1, \ldots j_t, i_{t+1}, i_{t+2}, \ldots i_k\} \\ I_{t+1}^* & = & \{j_1, \ldots j_t, j_{t+1}, i_{t+2}, \ldots i_k\} \end{array}$$

  On sait déjà par HR que les évènements $j_1, \ldots j_t, i_{t+2}, \ldots i_k$ sont compatibles, i.e. deux à deux disjoints. Il reste à montrer que j_{t+1} est compatible avec $j_1, \ldots j_t, i_{t+2}, \ldots i_k$.

  — $j_1, \ldots j_t, j_{t+1}$ font partie de la solution admissible I, et sont donc compatibles.
  — Soit $x \in [\![t+2, k]\!]$. Montrons que j_{t+1} et i_x sont compatibles. Pour cela, nous allons montrer que $f_{j_{t+1}} \leq d_{i_x}$. On sait que $i_{t+1} < i_x$ donc $f_{i_{t+1}} \leq f_{i_x}$, et comme les deux évènements sont compatibles (car faisant partie de la solution I^*), on a nécessairement que l'évènement i_{t+1} est intégralement avant l'évènement i_x, i.e. que $f_{i_{t+1}} \leq d_{i_x}$.
    Il suffit donc de montrer que $f_{j_{t+1}} < f_{i_{t+1}}$. Pour cela, on remarque que j_{t+1} est l'évènement choisi par le glouton après $j_1, \ldots, j_t$. Or, i_{t+1} est compatible avec les évènements $j_1, \ldots j_t$ car la solution I_t^*, admissible par HR, les contient. Ainsi, l'algorithme glouton a déterminé que j_{t+1} était meilleur que i_{t+1}, i.e. que $f_{j_{t+1}} < f_{i_{t+1}}$. Donc, $f_{j_{t+1}} < f_{i_{t+1}} \leq d_{i_x}$, et donc les évènements j_{t+1} et i_x sont compatibles.

  Nous avons bien montré que I_{t+1}^* est une solution admissible.

Par principe de récurrence, $\forall t \in [\![0, l]\!], I_t^* = \{j_1, \ldots, j_t, i_{t+1}, \ldots i_k\}$ est une solution admissible.

Supposons que $l < k$. Alors, $l+1 \leq k$, et donc l'évènement i_{l+1} existe bien. Il est compatible avec $j_1, \ldots j_l$, mais pourtant n'a pas été choisi par l'algorithme glouton, qui s'est arrêté après avoir choisi $j_1, \ldots j_t$. C'est absurde, et donc $j = k$.

B Complexité de l'algorithme DPR pour la multplication de polynômes

Notons $C(n)$ le coût de multiplication de deux polynômes de taille n par l'algorithme DPR naïf. $C(n)$ vérifie :

$$C(n) = 4C(\frac{n}{2}) + \Theta(n)$$

Autrement dit, il existe deux constantes $A, B \in \mathbb{R}^*$ et un rang $n_0 \in \mathbb{N}$ tel que pour $n \geq n_0$:

$$4C(\frac{n}{2}) + An \leq C(n) \leq 4C(\frac{n}{2}) + Bn$$

Quitte à prendre un rang n_0 plus grand, on peut supposer sans perte de généralité que n_0 est une puissance de 2. Commençons par dérouler l'inégalité de droite :

$$\begin{array}{rcl} C(n) & \leq & 4C(\frac{n}{2}) + Bn \\ & \leq & 4(4C(\frac{n}{4}) + B\frac{n}{2}) + Bn \\ & \leq & 16C(\frac{n}{4}) + Bn(1+2) \\ & \leq & 16(4C(\frac{n}{16}) + B\frac{n}{4}) + Bn(1+2) \\ & \leq & 64C(\frac{n}{16}) + Bn(1+2+4) \end{array}$$

On peut montrer par récurrence que pour $k \in \mathbb{N}$ tel que $\frac{n}{2^k} \geq n_0$, on a :

$$C(n) \leq 4^k C(\frac{n}{2^k}) + Bn(2^k - 1)$$

En particulier, pour $k = \log_2 n - \log_2 n_0$, on a $\frac{n}{2^k} = n_0$, et :

$$\begin{array}{rcl} C(n) & \leq & 4^{\log_2 n - \log_2 n_0} C(n_0) + Bn(2^{\log_2 n - \log_2 n_0} - 1) \\ & \leq & \frac{n^2}{n_0^2} + Bn(\frac{n}{n_0} - 1) \\ & \leq & \mathcal{O}(n^2) \end{array}$$

Donc $C(n) = \mathcal{O}(n^2)$. En déroulant l'inégalité de gauche, on montre de manière analogue que $C(n) = \Omega(n^2)$, et donc que $C(n) = \Theta(n^2)$.