TP 2 info: Programmation dynamique

Références: [1] chapitre 15.

Exercice 1. *Multiplication de matrices*. Le but de cet exercice est d'optimiser le calcul d'un produit matriciel $P = M_1 \dots M_n$. Plus précisément, il s'agit d'exploiter l'associativité de la multiplication matricielle pour trouver un parenthésage de l'expression $M_1 \dots M_n$ qui minimise le temps de calcul.

1. Donner la complexité (en nombre de multiplications de scalaires) de l'algorithme naïf pour la multiplication de deux matrices de tailles $p \times q$ et $q \times r$ respectivement.

2. Construire une famille de matrices P_n, Q_n, R_n tel que le quotient du nombre de multiplications scalaires dans le calcul de $(P_nQ_n)R_n$ par le nombre de mulitiplications scalaires dans le calcul de $P_n(Q_nR_n)$ tende vers l'infini.

3. Démontrer que le nombre P_n de parenthésages possible d'un produit de n termes vérifie la récurrence suivante :

$$\begin{cases} P_1 = 1 \\ P_n = \sum_{k=1}^{n-1} P_k P_{n-k} \text{ si } n \geq 2 \end{cases} \tag{1}$$

En déduire que $P_n = \Omega(2^n)$. Ainsi, on ne peut espérer trouver le parenthésage optimal par énumération des parenthésages possibles.

4. La suite de matrice $M_1 ... M_n$ *est maintenant fixée.* Soit $p_0, p_1, \dots, p_n, p_n$ la suite des dimensions des espaces vectoriels en jeu, c'est-à-dire que pour tout $1 \leq i \leq n$, M_i est une matrice de dimension $p_{i-1} \times p_i$. On note $m[i,j]$ le parenthésage optimal du produit matriciel $M_i ... M_j$. Montrer la formule suivante :

$$m[i,j] = \min_{i \leq k \leq j} m[i,k] + m[k+1,j] + p_{i-1}p_k p_j \tag{2}$$

5. En déduire un algorithme pour calculer un parenthésage qui minimise de multiplications scalaires dans le produit matriciel $M_1 \dots M_n$. Quelle est sa complexité ?

6. Montrer qu'on peut calculer le tableau $m[i,j]$ $(j \geq i)$ en temps $\mathrm{O}(n^3)$ et en espace $\mathrm{O}(n^2)$. (Indication : on commence par initialiser $m[i,i] = 0$ pour tout i, puis on calcule $m[i,i+1]$ pour tout i en utilisant (2), puis $m[i,i+2]$ etc.)

7. Modifier l'algorithme de la question précédente, qui ne fait que calculer le coût optimal, afin de pouvoir également calculer le parenthésage optimal.

Références

[1] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to algorithms.* MIT press, 2001.