Échauffement : deux algorithmes probabilistes.

Exemple 1 (Vérifier la multiplication de matrices). Soient A, B, C trois matrices carrées à coefficients dans $\mathbb{F}_2 = \{0, 1\}$. On cherche à décider $AB = C$.

Idée 1. On calcule AB et on vérifie l'égalité à C. L'algorithme pour calculer AB avec $(AB)_{i,j} = \sum_{k=1}^{n} A_{i,k}B_{k,j}$ se fait avec une complexité en $\mathrm{O}(n^3)$.

On peut améliorer la complexité en $\mathrm{O}(n^\alpha)$ avec $2 < \alpha < 3$ (actuellement, on peut le faire avec $\alpha \approx 2{,}37$) à l'aide de la méthode de Strassen.

Idée 2. On calcule ABx et Cx pour un vecteur $x \in \mathbb{F}_2^n$. On a des multiplications matrices $\times$ vecteurs, en complexité en $\mathrm{O}(n^2)$. Pour trouver un « bon » vecteur x, on le choisit au hasard.

Lemme 1. Si $D \in \mathscr{M}_n(\mathbb{F}_2)$ est non-nulle et $x \in \mathbb{F}_2^n$ est choisi uniformément au hasard, alors on a $\mathrm{P}(Dx \neq 0) \geq \frac{1}{2}$.

Preuve. Au moins un coefficient de D est non-nul et, sans perte de généralité, on peut supposer que $D_{1,n} \neq 0$. Alors,

$$(Dx)_1 = \sum_{i=1}^{n} D_{1,i}x_i = \sum_{i=1}^{n-1} D_{1,i}x_i + x_1.$$

Quels que soient $x_1, \ldots, x_{n-1}$, il y a une probabilité de $\frac{1}{2}$ que

$(Dx)_1 \neq 0$. On en conclut que

$$\mathrm{P}(Dx \neq 0) \geq \mathrm{P}((Dx)_1 \neq 0) = \frac{1}{2}.$$

□

Exemple 2 (suite de 1)**.** Ainsi, si $AB \neq C$, on a donc

$$\mathrm{P}(ABx \neq Cx) \geq \frac{1}{2}.$$

On choisit $x_1, \ldots, x_{100}$ des vecteurs uniformément dans $\mathbb{F}_2^n$. Si on a $AB \neq C$, alors

$$\mathrm{P}(\forall i \in [\![1, 100]\!], ABx_i = Cx_i) \leq \left(\frac{1}{2}\right)^{100}.$$

On a donc un algorithme ayant une complexité $\mathrm{O}(n^2)$ pour détecter, avec grande probabilité, si $AB = C$.

Exemple 3 (Coupe minimale dans un graphe)**.** On considère G un graphe non-orienté sans boucle (éventuellement avec des arêtes multiples). Une *coupe* du graphe est un sous-ensemble $C \subseteq E$ tel que $(V, E \setminus C)$ n'est pas connexe. On cherche une coupe de taille minimale :

$$\mathsf{mincut}(G) = \min\{|C| \mid C \text{ est une coupe}\}.$$

De manière équivalente, on cherche une partition $V = V_1 \sqcup V_2$ (avec $V_1, V_2 \neq \emptyset$) qui minimise le nombre d'arêtes reliant V_1 et V_2.

Étant donné un graphe $G = (V, E)$, et une arête $e = \{x, y\} \in E$, la *contraction de G selon e*, notée G/e, est le graphe où les sommets x et y sont fusionnés en un sommet xy, et les arêtes $\{x, z\}$ ou $\{y, z\}$ sont remplacées en $\{xy, z\}$ si $z \notin \{x, y\}$.

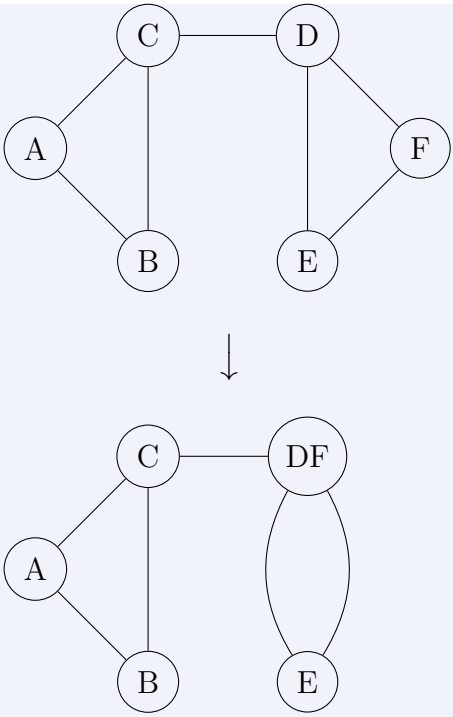

Figure 1 | *Contraction de l'arête* $\{\mathrm{D}, \mathrm{F}\}$.

On a que $\mathsf{mincut}(G/e) \geq \mathsf{mincut}(G)$.

On utilise l'*algorithme de Krager* (1993). On contracte successivement selon des arêtes choisies uniformément au hasard, jusqu'à n'obtenir que 2 sommets, ce qui donne une coupe du graphe initial.

Lemme 2. La coupe C produite par l'algorithme de vérifie

$$\mathrm{P}(|C| = \mathsf{mincut}(G)) \geq \frac{2}{n^2},$$

où $n = |V|$. □

Preuve. Soit $k = \mathsf{mincut}(G)$ et C une coupe de taille k. Montrons que $\mathrm{P}(\text{l'algorithme renvoie la coupe } C) \geq 2/n^2$. Notons A_i (pour $i \in [\![1, n-2]\!]$) l'événement « l'arête contractée à la i-ème étape est dans C », et B_i l'événement complémentaire. L'algorithme renvoie la coupe C si et seulement si tous les événements $B_1, \dots, B_{n-2}$ sont vérifiés. On a $\mathrm{P}(A_1) = k/|E| \leq 2/n$. Or, tout sommet a un degré $\geq k$, et donc $|E| \geq nk/2$. Conditionnellement à B_{11}, le graphe obtenu après contraction de la première arête vérifie $\mathsf{mincut}(G/e) = k$ donc $\mathrm{P}(A_2 \mid B_1) \leq 2/(n-1)$. De même, $\mathrm{P}(A_j \mid B_1 \cap \cdots \cap B_{j-1}) \leq 2/(n+1-j)$, pour tout $j \in [\![1, n-2]\!]$. On a donc $\mathrm{P}(A_{n-2} \mid B_1 \cap B_{n-2}) \leq \frac{2}{3}$, et donc

$$\begin{aligned}
\mathrm{P}(B_1 \cap \cdots \cap B_{n-2}) &= \mathrm{P}(B_1)\mathrm{P}(B_2 \mid B_1) \cdots \mathrm{P}(B_{n-2} \mid B_1 \cap \cdots \cap B_{n-1}) \\
&\geq \left(1 - \frac{2}{n}\right)\left(1 - \frac{2}{n-1}\right) \cdots \left(1 - \frac{2}{3}\right) \\
&\geq \frac{n-2}{n}\frac{n-3}{n-1} \times \cdots \times \frac{2}{3} \\
&\geq \frac{2}{n(n-1)} \geq \frac{2}{n^2}.
\end{aligned}$$

□

Exemple 4 (suite de 3). On répète $N = 50n^2$ fois cet algorithme (tous les choix étant indépendant). On note k_i la taille de la coupe obtenue à la i-ème itération, et alors

$$\mathrm{P}(k_i = \mathsf{mincut}(G)) \geq \frac{2}{n^2},$$

d'où $\mathrm{P}(k_i \neq \mathsf{mincut}(G)) \leq 1 - \frac{2}{n^2}$.

On en conclut que

$$\begin{aligned}\mathrm{P}(\forall i, k_i \neq \mathsf{mincut}(G)) &\leq \left(1 - \frac{2}{n^2}\right)^{50n^2} \\ &\leq \exp\left(-\frac{2}{n^2}50n^2\right) \\ &\leq \exp(-100).\end{aligned}$$

Chaque itération prend un temps en $\mathrm{O}(n^2)$, on obtient donc un algorithme en $\mathrm{O}(n^4)$ qui calcule une coupe minimale avec très grande probabilité.