Complexité de problèmes

du facile au difficile

Groupe algorithmique de l'IREM de Grenoble
{**Anne.Rasse**,Jean-Marc.Vincent,Benjamin.Wack}@imag.fr
{Maryline.Bruel}@ac-grenoble.fr

Annecy Février 2016

COMPLEXITÉ D'ALGORITHMES

La **complexité** ($\sim$ le coût) d'un algorithme est l'ordre de grandeur du **nombre d'opérations** élémentaires qu'il exécute en fonction de la **taille des données**.

Le temps d'exécution d'un programme est lié à la complexité de l'algorithme qu'il met en œuvre.

Exemples

n : taille des données

Complexité	Temps d'exécution	Pour $n = 10^3$	Pour $n = 10^6$
$\mathcal{O}(n)$	$k \times n$	$k \times 10^{-6}$ s	$k \times 10^{-3}$ s
$\mathcal{O}(n^3)$	$k \times n^3$	k s	$k \times 30$ ans
$\mathcal{O}(2^n)$	$k \times 2^n$	$> 10^{300}$ ans	...

(sur un processeur à 1 GHz)

Complexité de problème

La complexité d'un **problème** est celle du **meilleur** algorithme qui le résout.

Par exemple :

- la complexité du tri de n valeurs dans un tableau est $\mathcal{O}(n \times \log n)$
- la complexité de l'algorithme "classique" de multiplication de matrices carrées $n \times n$ est $\mathcal{O}(n^3)$ mais la complexité de ce problème, actuellement inconnue, est de la forme $\mathcal{O}(n^{2+\varepsilon})$ algorithme de Strassen en $\mathcal{O}(n^{2,807})$.

Maison pour la science AU SERVICE DES PROFESSEURS AIX MARSEILLE UNIVERSITÉ ALPES DAUPHINE

Complexité de problème (2)

Et pour certains problèmes (parmi ceux proposés au cours de ce stage) :

- trouver un circuit dans un graphe qui passe par tous les sommets une fois et une seule (le “Tour du Monde”)
- garantir le remplissage d’un camion/sac à dos . . . (“Alice déménage”)

on ne connaît pas d’algorithme **efficace** qui permette de les résoudre.

algorithme “efficace” = algorithme de complexité polynômiale en fonction de la taille des données $\mathcal{O}(n^k)$

COMPLEXITÉ DE PROBLÈME (3)

Pour ces problèmes, on connaît par contre un algorithme “inefficace” qui consiste à **énumérer tous** les chemins du graphe, **énumérer toutes** les façons de placer les cartons dans le camion . . . .

Par contre, on peut **vérifier facilement** (efficacement, i.e. avec un coût polynômial) si une solution répond au problème :

- vérifier si un circuit est hamiltonien
- vérifier que le remplissage du camion répond aux contraintes

CLASSIFICATION DES PROBLÈMES

Soit l'algorithme suivant :

- exhiber une solution (au hasard ?) $\rightarrow$ polynômial
- vérifier qu'elle est correcte $\rightarrow$ polynômial

si la réponse est oui : le problème est résolu
si la réponse est non : on n'a pas eu de chance

On a donc un algorithme **non déterministe** polynômial qui résout notre problème. (si on a de la chance . . .)

Classification des problèmes

Classes de problèmes :

- Classe $\mathcal{P}$: problèmes pour lesquels il existe un algorithme polynômial
- Classe $\mathcal{NP}$: problèmes pour lesquels il existe un algorithme non déterministe polynômial

Évidemment : $\mathcal{P} \subseteq \mathcal{NP}$

Question à 1 million de dollars :

$$\mathcal{P} = \mathcal{NP} \ ?$$

ET POUR UN PROBLÈME DIFFICILE ...

Aider la chance ?

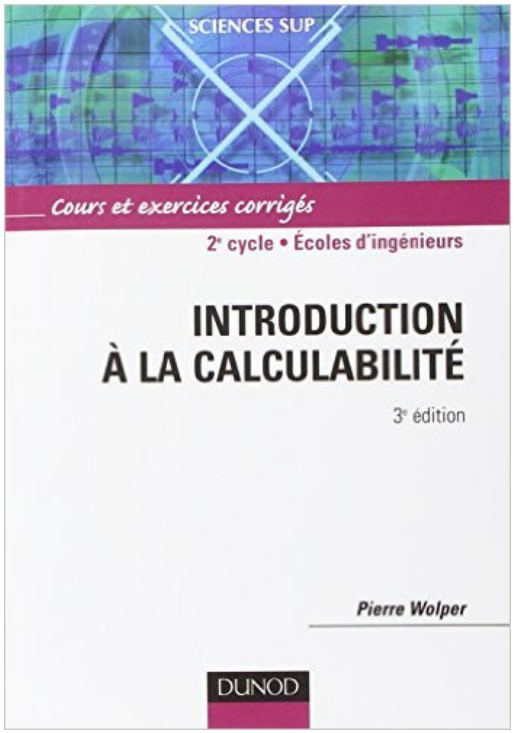
Il y a des stratégies (heuristiques) qui semblent prometteuses :

- placer les plus hauts cartons d'abord
- . . .

La solution obtenue est "souvent" correcte, ou sinon approche une solution correcte.

RÉFÉRENCES

Pour aller plus loin :
https://interstices.info/jcms/c_21832/p-np-un-probleme-a-un-million-de-dollars



Introduction à la calculabilité, Pierre Wolper, Dunod 2006