GROSPELLIER Antoine

928 : Problèmes NP-Complets : exemples de réductions

I Introduction

But : Pour un problème, déterminer si il est "Résoluble en temps raisonnable" ou non / "Vérifiable en temps raisonnable" ou non.
"Temps raisonnable" : Complexité temporelle polynomiale.

Ex 1 : ☆ (k-)SAT :
Entrée : Une liste de listes d'entiers qui représente une formule booléenne CNF (avec des clauses de taille k).
Sortie : La formule est-elle satisfiable ?

- 2-SAT est résoluble en temps linéaire } Temps raisonnables
- k-SAT est vérifiable en temps linéaire }

☆ k-couleur (-planaire) :
Entrée : Listes d'adjacences d'un graphe non orienté (planaire)
Sortie : Le graphe est-il k-colorable ?

- 3-couleur (-planaire) est vérifiable en temps linéaire
- 4-couleur-planaire est résoluble en temps constant (toujours vrai).

Def 2 : • $P = \bigcup_{k \in \mathbb{N}} DTIME(n^k)$ (=Résoluble en temps raisonnable)
• $Exp = \bigcup_{k \in \mathbb{N}} DTIME(2^{n^k})$ (=Temps non raisonnable).

Rq 3 : • On s'interesse aux problèmes de décision
• Problème d'optimisation → Problème de décision

Ex 4 : ☆ Clique-optim :
Entrée : Listes d'adjacences d'un graphe non orienté
Sortie : Une clique de taille maximale

☆ (k-) Clique :
Entrée : Listes d'adjacences d'un graphe non orienté ($+ k \in \mathbb{N}$ si ✗ Clique).
Sortie : Existe-t-il une clique de taille supérieure ou égale à k ? ($\Leftrightarrow$ Existe-t-il une clique de taille k ?).

Ex 5 : Temps nécessaire à l'exécution d'un algorithme : 10^9 op/s

Temps	$n=5$	$n=10$	$n=30$	$n=55$	$n=100$
$O(n)$	5 ns	10 ns	30 ns	55 ns	100 ns
$O(n^2)$	25 ns	100 ns	900 ns	3 µs	10 µs
$O(2^n)$	32 ns	1 µs	1 s	1 an + 52 j	4.10^{13} ans
$O(3^n)$	243 ns	59 µs	2 j + 9 h	$5,5.10^9$ ans	$1,6.10^{31}$ ans

âge de l'univers $\simeq 14.10^9$ ans
• Taille résoluble si on augmente la puissance de calcul en 1 s :

Multiplication	$O(n)$	$O(n^2)$	$O(2^n)$	$O(3^n)$
×1	$N_1 = 10^9$	$N_2 = 3.10^5$	$N_3 = 30$	$N_4 = 19$
×10	$10 N_1$	$3 N_2$	$N_3 + 3$	$N_4 + 2$
×1000	$1000 N_1$	$32 N_2$	$N_3 + 10$	$N_4 + 6$
×10^6	$10^6 N_1$	$1000 N_2$	$N_3 + 20$	$N_4 + 13$

[2]

Def 5 : $NP = \bigcup_{k \in \mathbb{N}} NTIME(n^k)$

Prop 6 : $L \in NP$ ssi il existe $\begin{cases} c \in \mathbb{N} \\ A \text{ une machine à deux entrées} \end{cases}$ telle que $L = \{ x \in \{0;1\}^* \mid \exists y \in \{0;1\}^{|x|^c} \text{ tel que } A(x,y) = 1 \}$

Rq 7 : • Si $L \in NP$ alors :
- $x \in L \Rightarrow \exists y,\ A(x,y) = 1$
- $x \notin L \Rightarrow \forall y,\ A(x,y) \neq 1$

• y s'appelle une certificat pour x
• $P \subseteq NP \subseteq Exp$.

- NP = Non-déterministe ≠ Non-polynomial.
- La question $NP \subseteq P$ (c.à.d $P = NP$) reste ouverte et vaut 1 Million de dollars.

II) Les problèmes NP-Complets

Les problèmes NP-Complets sont les problèmes les plus durs de NP. Si $L \in$ NP-Complet et $L \in P$ alors $P = NP$.

Déf 8 : • Soit L_1 et L_2 deux problèmes de décision. L_1 se réduit polynomialement en L_2 (on note $L_1 \leq_p L_2$) ssi Il existe un algorithme polynomial $R : \{0;1\}^* \to \{0;1\}^*$ tel que $i \in L_1$ équivaut à $R(i) \in L_2$ pour tout $i \in \{0;1\}^*$.

- NP-difficile est l'ensemble des problèmes L_0 tels que $L_0 \geq_p L$ pour tout $L \in NP$.
- NP-Complet = (NP-difficile) $\cap$ NP.

Prop 9 : $P = NP$ ssi il existe $L_0 \in$ (NP-Complet) tel que $L_0 \in P$.

Prop 10 : Si $L_1 \in$ (NP-Difficile) et $L_1 \leq_p L_2$ alors $L_2 \in$ (NP-difficile)

Rq 11 : • Si on a un problème NP-Complet on peut en construire un autre avec une seule réduction polynomiale.
- Il faut un problème NP-Complet de base. La difficulté est de réduire tout problème NP à celui-ci.

Théo 12 de Cook : SAT $\in$ NP-Complet

III) Exemples de problèmes (Voir ordre des réductions en annexe (i))

Prop 13 (Formules booléennes) : • (2-SAT) $\in P$ et même (2-SAT) $\in$ DTIME(n).
- (3-SAT) $\in$ NP-Complet
- (k-SAT) $\in$ NP-Complet pour $k \geq 3$.

Prop 14 (coloration de graphes) : • (2-couleur) $\in P$, (2-couleur) $\in$ DTIME(n) [1]
- (3-couleur) $\in$ NP-Complet (utilisation du gadget en annexe (ii))
- (k-couleur) $\in$ NP-Complet si $k \geq 3$.
- (3-couleur-planaire) $\in$ NP-Complet (utilisation du gadget en annexe (iii))

Développement 1

Théo 15 des quatre couleurs : (4-Couleur-planaire) $\in$ DTIME(1).
Car tout graphe planaire est colorable avec 4 couleurs (admis)

☆ Couverture de Sommets :
Entrée : Listes d'adjacences d'un graphe non orienté $G = (S, A)$ + $k \in \mathbb{N}$
Sortie : Existe-t-il $S' \subseteq S$; $|S'| = k$ tel que pour tout $\{u,v\} \in A$, $u \in S'$ ou $v \in S'$.

☆ Cycle Hamiltonien :
Entrée : Listes d'adjacences d'un graphe non orienté
Sortie : Existe-t-il un cycle simple qui contient tous les sommets ?

☆ Voyageur de commerce :
Entrée : $m \in \mathbb{N}^*$ le nombre de sommet d'un graphe complet $G = (S, A)$, $c : A \to \mathbb{N}$ une fonction de coût et $b \in \mathbb{N}$ un entier objectif
Sortie : Existe-t-il un cycle Hamiltonien dans G de coût inférieur à b qui visite tous les sommets ?

Prop 16 : Clique, Couverture de Sommets, Cycle Hamiltonien, Voyageur de commerce $\in$ NP-Complet. [1]

☆ Somme-Sous-ensemble :
Entrée : S un ensemble fini d'entiers positifs et un entier objectif t.
Sortie : Existe-t-il un sous-ensemble $S' \subseteq S$ tel que $\sum_{x \in S'} x = t$?

★ 2-Partition :
Entrée : Un (multi-)ensemble fini S d'entiers positifs
Sortie : Existe-t-il $I \subseteq S$ tel que $\sum_{x \in I} x = \sum_{x \notin I} x$

[1] Prop 17 (Problèmes de sous-ensembles) : Somme-Sous-ensemble, 2-Partition $\in$ NP-Complet

Algo 18 : reduction-Somme-Sous-ensemble-vers-2-Partition (S, t) : =
Soit $A :=$ la somme des entiers de S ;
Soit $S_0 := S \cup \{2A - t ; A + t\}$;
Retourner (S_0).

★ Racine-polynome :
Entrée : P un polynome donné sous forme de liste de couples degré / coefficient + $M \in \mathbb{N}$
Sortie : Existe-t-il $\omega \in \mathbb{C}$ tel que $\omega^M = 0$ et $P(\omega) = 0$?

★ PGCD-Polynomes :
Entrée : P et Q deux polynomes donnés sous forme de listes de couples degré/coefficient
Sortie : Est-ce que $PGCD(P, Q) = 1$.

[3] Prop 19 (Problèmes polynomes) : Racine-polynome, PGCD-Polynomes $\in$ NP-difficile.

Rq : • On ne sait pas si ils sont dans NP.
• Le caractère polynomial de la rédaction $(3\text{-SAT}) \leq_p$ (Racine-Polynome) n'est pas immédiat (il faut utiliser que le $n^{ième}$ nombre premier est $p_n = O(n \log n)$ ...).
• Exemple de problème pour lequel l'encodage modifie la complexité.

IV Que faire d'un problème NP-Complet :

Solutions : • Résolution de petites instances : SAT-Solveur ...
• Regarder la complexité en moyenne
• Utiliser des algorithmes d'approximation.

$\neq (S, A)$

Prop 20 : Si on choisit un graphe G au hasard (chaque arrête apparait avec probabilité p) alors si on essaie de 3-coloré le graphe par brute force, l'algorithme s'arrête en moyenne en 197 pas pour tout $|S|$ (voir Algo 21). [4]

Algo 21 : Colorer $(u \in S)$: =
Soit (L : liste) := Couleurs-possibles-en-fonction-voisins (u).
Si $L = []$ alors $\perp$
Sinon Tant que $L \neq []$: Essayer (Couleur$(u) \leftarrow$ Tête(L) ;
lancer Colorer(v) pour tout v voisin de u non coloré)
$\perp \Rightarrow L \leftarrow$ Queue(L)

Def 22 : $p(n)$-approximation.

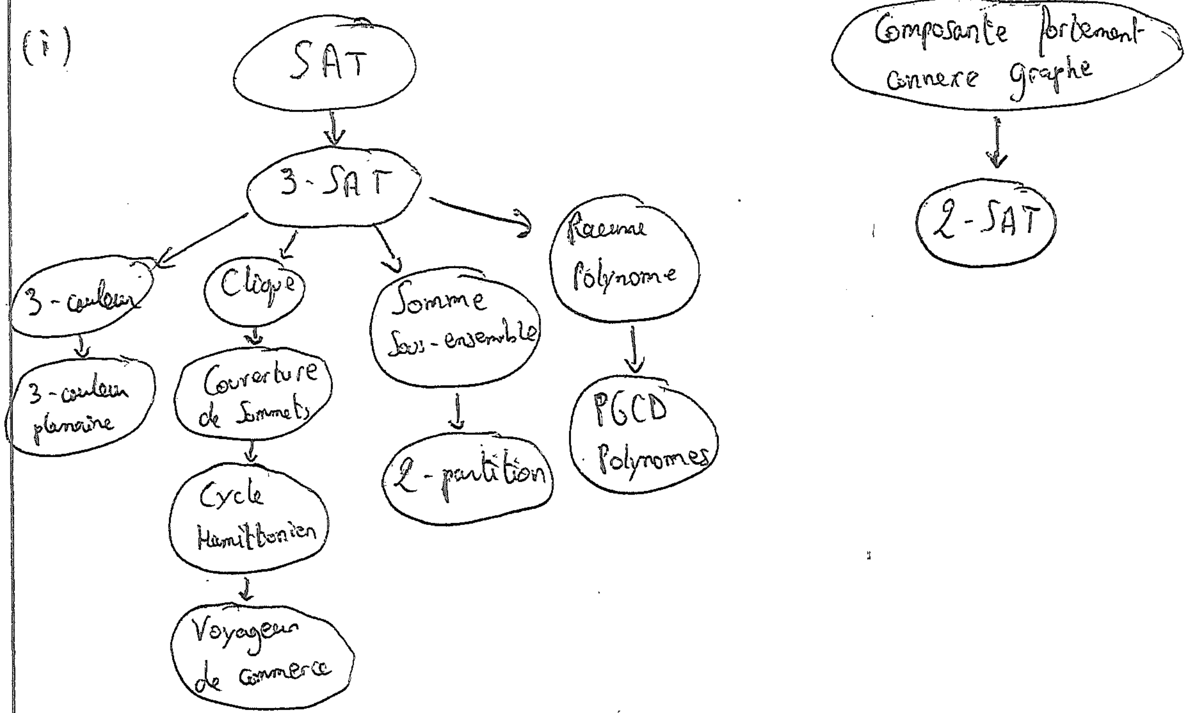
Rq 23 : Cette notion ne se transmet pas avec les reductions.

Prop 24 : Il existe une 2-approximation polynomiale pour Couverture de Sommets. [1] Dévelop 2

Prop 25 : Si $P \neq NP$ alors pour tout $p \in \mathbb{N}^*$, il n'existe pas de p-approximation polynomiale pour le problème du voyageur de commerce avec p constant.

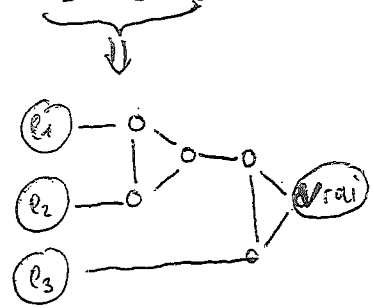
Prop 26 : Pour tout $\varepsilon > 0$, il existe une $(1+\varepsilon)$-approximation polynomiale pour Somme-Sous-Ensemble.

(i)

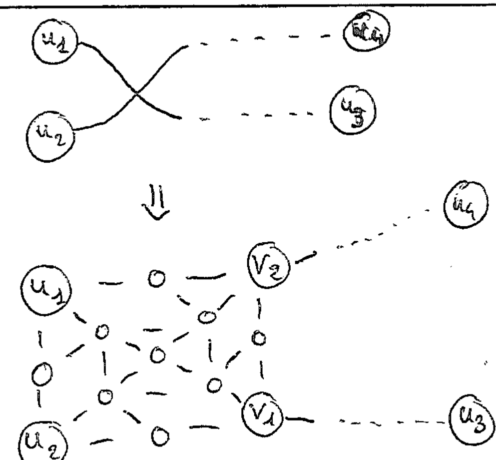


(ii) 3-SAT : $\ell_1 \vee \ell_2 \vee \ell_3$

3-couleur :



(iii) 3-couleur :



3-couleur-planaire

$\begin{cases} \text{Couleur}(u_1) = c_1 \\ \text{Couleur}(v_1) = c_2 \end{cases} \Rightarrow \begin{cases} \text{Couleur}(v_1) = c_1 \\ \text{Couleur}(v_2) = c_2 \end{cases}$

Réf : [1] Cormen
[2] Garey / Johnson : "Computers and intractability : A guide to the theory of NP-completness"
(3) Plaisted : "New NP-hard and NP-complete polynomial and integer divisibility problems"
[4] Herbert S. Wilf : "Backtrack : an $O(1)$ expected time algorithm for the graph coloring problem".

Liste de problemes NP-Durs originaux dans [2].