Notions de NP-Complétude

Il existe des problèmes « faciles » comme : « Trouver la plus petite chaîne simple entre les sommets x et y » et des problèmes « difficiles » comme : « Trouver la plus grande chaîne simple entre les sommets x et y ».

Un problème est dit facile si on connaît un algorithme efficace (complexité polynomiale ou inférieur) pour le résoudre, sinon il est dit difficile (complexité exponentielle ou supérieure).

Pour classer les problèmes on définit deux ensembles :
P : l'ensemble de tous les pb pouvant être résolus par des algo déterministes en temps Polynomial.
NP: ~ ~ ~ ~ ~ ~ ~ ~ Non déterministe en temps Polynomiale.

Un algo non déterministe est un algo qui en présence d'alternatives durant le processus de résolution, peut choisir la bonne et continuer son traitement. Ou d'une autre manière il peut « deviner » la solution puis la vérifie en temps polynomiale. Ce pouvoir de choisir à chaque étape la bonne alternatives est offert par une machine abstraite (théorique) dite machine de Turing Non Déterministe.

Pour montrer qu'un pb est dans NP, il suffit de trouver un algo qui vérifie si une solution donnée est valide en temps polynomiale.

Tout pb dans P est aussi dans NP, mais l'inverse reste à démontrer ou à réfuter.

Parmi les pb classés dans NP et dont on arrive pas à vérifier s'ils sont ou non dans P, on cite :
- le pb du plus long chemin entre un couple de sommet
- le pb du voyageur de commerce (PVC) : trouver une tournée de longueur minimale pour un voyageur de commerce devant visiter un ensemble de villes et revenir au point de départ.
- le pb de satisfiabilité de formules logiques (SAT) : Existe-t-il des valeurs (vrai/faux) que l'on attribue à des variables booléennes (xj) pour qu'une formule logique données soit satisfaite (vraie). Ex de formule : (x1 ou x3 ou x5) et (¬x1 ou x2 ou x4) et (x3 ou ¬x6)
- ...

Si quelqu'un puisse un jour montrer qu'il existe un pb classé dans NP et qui ne peut pas être classé dans P, alors cela voudra dire que P ≠ NP et donc que le non déterminisme est une opération (théorique) extrêmement puissante. Or pour le moment, personne n'a pu trouver un tel problème. C'est à dire personne n'a pu montrer que le non déterminisme serait d'une quelconque aide pour la résolution de problèmes.

Pratiquement personne ne pense que P = NP, cela veut dire que l'on suspecte l'existence de certaines caractéristiques rendant certains pb « difficiles » (donc ne pouvant pas être résolus en temps polynomiale).

Ce qui favorise cette hypothèse est l'existence d'un sous ensemble de NP formé par des pb ayant une caractéristique commune et qui dit que si un de ces pb pourra un jour être résolu en temps polynomiale, alors cela impliquerait que tous les pb de NP pourront aussi être résolus en temps polynomiale (donc P=NP). Les pb appartenant à ce sous ensemble sont dits « NP-Complets »

Beaucoup de pb intéressant sont NP-complets. Le premier à être classé NP-complet est le pb SAT (théorème de Cook – 1971) où il a été démontrer que la machine de Turing Non Déterministe pouvait être entièrement décrite par des formules logiques et que tout programme (non déterministe) s'exécutant sur cette machine est forcément associé à une instance du pb SAT dont la solution donnerait alors celle du programme en question. Donc tout algo déterministe pour résoudre SAT en temps polynomiale pouvait être utilisé pour résoudre (de manière déterministe) n'importe quel pb de NP en temps polynomiale.

Les autres problèmes NP-complets ont été classé en utilisant le mécanisme de réduction polynomiale suivant :

> Pour montrer qu'un nouveau pb A est classé NP-complet, il faut montrer que l'on peut transformer n'importe quelle instance d'un pb B déjà classé NP-complet en une instance de A et que la solution de cette instance de A puisse être transformée en une solution de B. Les transformations doivent pouvoir être faites en temps polynomiale. Donc si jamais il existera un jour un algo en temps polynomiale pour résoudre A cela impliquerait l'existence d'un algo (polynomial) pour résoudre B et de la même manière il existerait alors un algo (polynomial) pour résoudre SAT et qui pourra donc aussi résoudre tout pb de NP en temps polynomial. Le pb A serait donc NP-complet (par définition).

Un exemple de réduction (simple) :
Soit PCH le pb du cycle hamiltonien qui s'énonce comme la recherche d'un cycle simple contenant tous les sommets d'un graphe. Ce pb est déjà classé NP-complet.
On aimerait démontrer (par réduction polynomiale avec PCH) que le pb du PVC est aussi NP-complet. Il faut alors montrer que :

- toute instance de PCH peut être transformée en instance de PVC : les sommets du graphe deviennent les villes a parcourir et les distances entre villes seront positionner à 1 s'il existe une arête entre les sommets correspondants dans le graphe et à 2 sinon.
- La solution du PVC peut se transformer en solution du PCH : si la plus petite tournée du voyageur de commerce à une longueur égale à n (le nombre de sommets du graphe) alors il existe un cycle hamiltonien dans le graphe; sinon si la plus petite tournée du voyageur de commerce > n alors il n'existe pas de cycle hamiltonien dans le graphe.

Donc si un jour quelqu'un trouve un algo efficace (polynomial) pour résoudre PVC, ce même algo et les 2 transformations ci-dessus pourront être utilisée pour résoudre PCH en temps polynomial.

Il existe des réductions plus complexes entre pb très différents comme par exemple un réduction entre SAT et PCH et beaucoup d'autres.