

d'un entier et le problème du voyageur de commerce, qui consiste à déterminer le plus court chemin reliant plusieurs villes données, sont des problèmes de classe non- P (bien que cela n'ait pas encore été démontré).

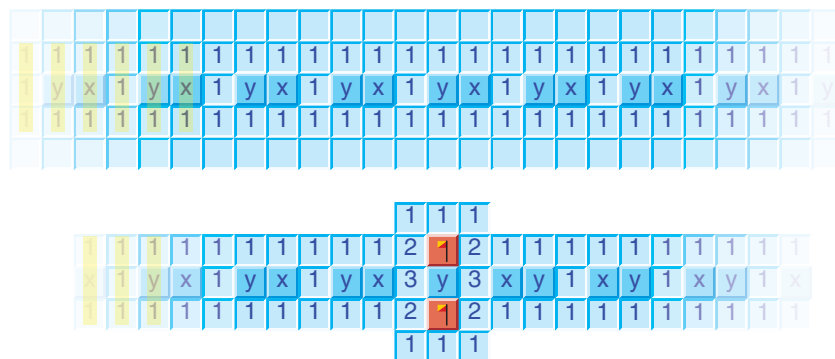
Pourquoi prouver qu'un problème est de classe non- P est-il si difficile ? Parce qu'il faut traiter tous les algorithmes possibles – et non pas un seul – et montrer qu'aucun ne résout le problème en un temps polynomial. Aujourd'hui, on sait qu'une vaste catégorie de problèmes candidats à la classe non- P sont sur un pied d'égalité : si l'un d'eux peut être résolu en un temps polynomial, alors tous peuvent l'être. Ces problèmes ont un comportement polynomial non déterministe, c'est-à-dire qu'avec de la chance – l'algorithme inclut une étape aléatoire –, on peut les résoudre en un temps polynomial ; ils sont de la classe NP .

NON- P , NP ET NP -COMPLET

Les classes non- P et NP sont différentes : un problème est NP dès qu'une solution proposée est bien en temps polynomial. Il est plus facile de vérifier qu'une solution proposée est bien en temps polynomial que de trouver la solution elle-même. Mon exemple favori qui illustre cette différence est le puzzle : en venir à bout peut être difficile, mais vérifier une solution proposée ne requiert souvent que quelques secondes ; le temps d'examiner que chaque pièce s'ajuste aux pièces voisines. Ce temps, environ proportionnel au nombre de pièces, est polynomial. En revanche, on ne peut effectuer un puzzle en un temps polynomial : le nombre de solutions croît plus vite que toute puissance donnée du nombre de pièces.

Beaucoup de problèmes NP ont des temps de traitement équivalents. Un problème NP est dit NP -complet quand l'existence d'une solution en un temps polynomial déterministe, c'est-à-dire sans processus aléatoire, entraîne que tout problème NP a une solution en un temps polynomial. Ainsi, si on peut résoudre un problème NP -complet en un temps polynomial, on peut résoudre tous les problèmes NP en un temps polynomial. Et la question « $P = NP$?» ? Bien que les apparences soient contraires, on peut s'interroger sur l'égalité des classes P et NP . La réponse attendue est négative, cependant, au cas où tous les problèmes NP -complets ont une solution en temps polynomial, alors les deux classes P et NP sont égales.

Nombre de problèmes sont NP -complets : un des plus simples est le problème SAT, qui inclut des circuits booléens.



2. L'électronique du démineur convertit des circuits booléens en position de jeu. Un seul type de carrés (x ou y) cache des mines. Des connexions propagent un signal (y suivi de x) le long de la grille (en haut). Un opérateur *non* inverse l'ordre des x et des y (en bas).

Ces circuits sont construits avec des opérateurs logiques, tel «*et*», «*ou*» et «*non*», qui s'appliquent à des valeurs soit vraies (V), soit fausses (F). Chaque opérateur fournit à partir de valeurs d'entrée une valeur de sortie : par exemple, l'opérateur *non* transforme la valeur V en la valeur F . Le problème SAT s'intéresse à l'existence pour un circuit booléen donné de valeurs d'entrée tel que la valeur de sortie soit V . Pour des circuits simples, le problème est un jeu d'enfants ; il devient ardu lorsque le circuit a un grand nombre d'opérateurs et d'entrées.

LA CONSISTANCE DU DÉMINEUR

Le lien avec le démineur apparaît quand on s'intéresse à la consistance du jeu, c'est-à-dire à la vraisemblance d'une position du jeu. Par exemple, la figure 1 montre une position inconsistante : aucune répartition des mines ne peut mener aux nombres affichés dans les carrés de la grille.

R. Kaye a montré que le problème SAT pour un circuit booléen donné peut-être transformé en un problème de consistance pour une certaine grille du jeu du démineur. En outre, il montre que la conversion des circuits booléens en positions du démineur s'exécute en un temps polynomial. Dans cette conversion, les entrées et les sorties du circuit deviennent une répartition des mines sur la grille : à un carré miné correspond une valeur V , et à un carré sans mine, une valeur F . Les connexions des opérateurs du circuit sont représentées sur la figure 2 : les carrés marqués y ont des valeurs opposées de celles des carrés marqués x . Notons que les chiffres des autres carrés sont corrects que x soit V ou F . Tels des «câbles électroniques», les connexions propagent chaque valeur (V ou F) sur la longueur de la grille.

On représente les opérateurs logiques par le même type de procédé. Par

exemple, la position du démineur de la figure 2 illustre l'opérateur *non* : le bloc du milieu échange les positions de x et de y ; x suit y avant le bloc et précède y après le bloc. Les valeurs des carrés sont inversées, que x soit V ou F , comme après un opérateur *non* dans un circuit booléen.

L'électronique du démineur est plus complexe : elle requiert par exemple de plier ou de scinder des «fils». R. Kaye franchit tous ces obstacles et conclut que quiconque prouve la consistance d'une position donnée du démineur en un temps polynomial, résout aussi le problème SAT du circuit équivalent en un temps polynomial. En d'autres termes, le démineur est un problème NP -complet : une solution en un temps polynomial à la consistance du démineur entraîne que tous les problèmes NP -complets ont des solutions en un temps polynomial : les classes P et NP sont égales. À l'inverse, si l'on montre qu'il n'existe pas de telle solution au problème du démineur, alors les classes P et NP sont différentes. Dans les deux cas, la question serait résolue.

Avant de vous lancer sur votre ordinateur, rappelez-vous que le problème de la consistance du démineur n'est pas une mince affaire : déterminer si une position de jeu est consistante devient vite difficile quand la taille de la grille augmente et la plupart des mathématiciens et des informaticiens pensent qu'il n'y a pas de solution générale traitable en un temps polynomial. De surcroît, l'Institut Clay a imposé des règles strictes au concours : avant d'accepter une solution, elle doit être publiée dans un journal important et être «généralement acceptée» par la communauté mathématique dans les deux années de sa publication.

Richard Kaye, *Minesweeper is NP-Complete*, in *Mathematical Intelligencer*, vol. 22, n° 2, 2000.