

R

ENDEZ-VOUS

P.82 Logique & calcul
 P.88 Idées de physique
 P.92 Chroniques de l'évolution
 P.96 Science & gastronomie
 P.98 À picorer

DE L'HYPERCUBE À LA SENSITIVITÉ

La «conjecture de la sensibilité», qui portait sur la complexité des fonctions booléennes, importantes en informatique, résistait depuis plusieurs décennies. Un jeune mathématicien d'origine chinoise, Hao Huang, l'a prouvée en deux pages, en exploitant un détour par les hypercubes.

L'AUTEUR



JEAN-PAUL DELAHAYE
 professeur émérite
 à l'université de Lille
 et chercheur au
 laboratoire Cristal
 (Centre de recherche
 en informatique, signal
 et automatique de Lille)



Jean-Paul Delahaye a notamment publié : **Les Mathématiciens se plient au jeu**, une sélection de ses chroniques parues dans *Pour la Science* (Belin, 2017).

La logique booléenne est celle que vous pratiquez quand vous lisez une carte de restaurant : ((Œuf mayonnaise ou Carottes ou Jambon) et ((Canard et Pommes de terre) ou (Saumon et riz)) et (Tarte ou Glace). Même sans les parenthèses, vous comprenez ! Son nom évoque George Boole (1815-1864), qui proposa de calculer avec les propositions et les connecteurs logiques (NON, ET, OU, IMPLIQUE, etc.) comme on calcule avec les nombres.

Les travaux de l'ingénieur et mathématicien américain Claude Shannon (1916-2001) ont rendu essentiel ce calcul pour la conception des puces électroniques et aujourd'hui pour la cryptographie. Comme toujours en mathématiques, on y rencontre des énoncés que l'on soupçonne vrais sans réussir à les démontrer. C'était le cas de la «conjecture de la sensibilité» devenu en 2019 «théorème de la sensibilité» puisqu'elle a été démontrée. Le plus étonnant est que, après avoir résisté près de trente ans aux meilleurs spécialistes, un jeune mathématicien aux États-Unis, Hao Huang, en est venu à bout dans un article de six pages, l'argument principal de la démonstration occupant moins de deux pages.

LE GRAPHE DE L'HYPERCUBE

Pour analyser ce résultat, examinons d'abord la structure remarquable de l'hypercube de dimension n , ce qui est déjà un défi pour l'imagination que quelques dessins nous aideront à relever.

L'hypercube de dimension n , noté H_n , est un graphe qui généralise le carré et le cube

(voir l'encadré page ci-contre). En dimension 2, le carré, H_2 , possède 4 sommets et 4 arêtes. Sur le plan, on représentera les sommets du carré par les points de coordonnées $A(0, 0)$, $B(0, 1)$, $C(1, 0)$ et $D(1, 1)$. Chaque sommet de ce graphe est relié par deux arêtes à deux voisins ; par exemple, le sommet A est relié à B et à C .

En dimension 3, le cube, H_3 , a 8 sommets et 12 arêtes. Les sommets sont par exemple les points ayant pour coordonnées les 8 triplets de nombres pris parmi 0 et 1 : 000, 001, 010, 011, 100, 101, 110 et 111. Chaque sommet S est relié par trois arêtes à trois voisins. Les voisins de S sont les sommets S' dont les coordonnées ne diffèrent de celles de S que par une coordonnée. Par exemple, les trois voisins de 010 sont 110, 000 et 011. Les 12 arêtes du cube constituent un graphe où, pour se déplacer en n'empruntant que les arêtes, on doit passer d'un sommet à l'autre en ne changeant qu'une coordonnée à la fois. On peut parcourir tous les sommets du cube sans passer deux fois par le même sommet ; cela définit un «chemin hamiltonien», dont un exemple est le parcours :

000 – 001 – 011 – 010 – 110 – 111 – 101 – 100.

Vous l'avez deviné, l'hypercube de dimension 4, H_4 , est défini par ses 16 sommets qui sont tous les quadruplets possibles de 0 et de 1 : 0000, 0001, 0010, ..., 1111. On obtient son graphe en prenant deux copies du graphe du cube de dimension 3 et en ajoutant des 0 devant les coordonnées des sommets du premier et des 1 devant les coordonnées des sommets du second. On ajoute pour finir des arêtes pour joindre les sommets analogues des deux cubes de dimension 3 : ➤

D'UN HYPERCUBE À L'AUTRE

1

Pour dessiner l'hypercube H_n de dimension n , à partir de l'hypercube H_{n-1} de dimension $n-1$, on translate H_{n-1} pour obtenir une copie H_{n-1}' , puis on joint (en bleu) les sommets analogues.

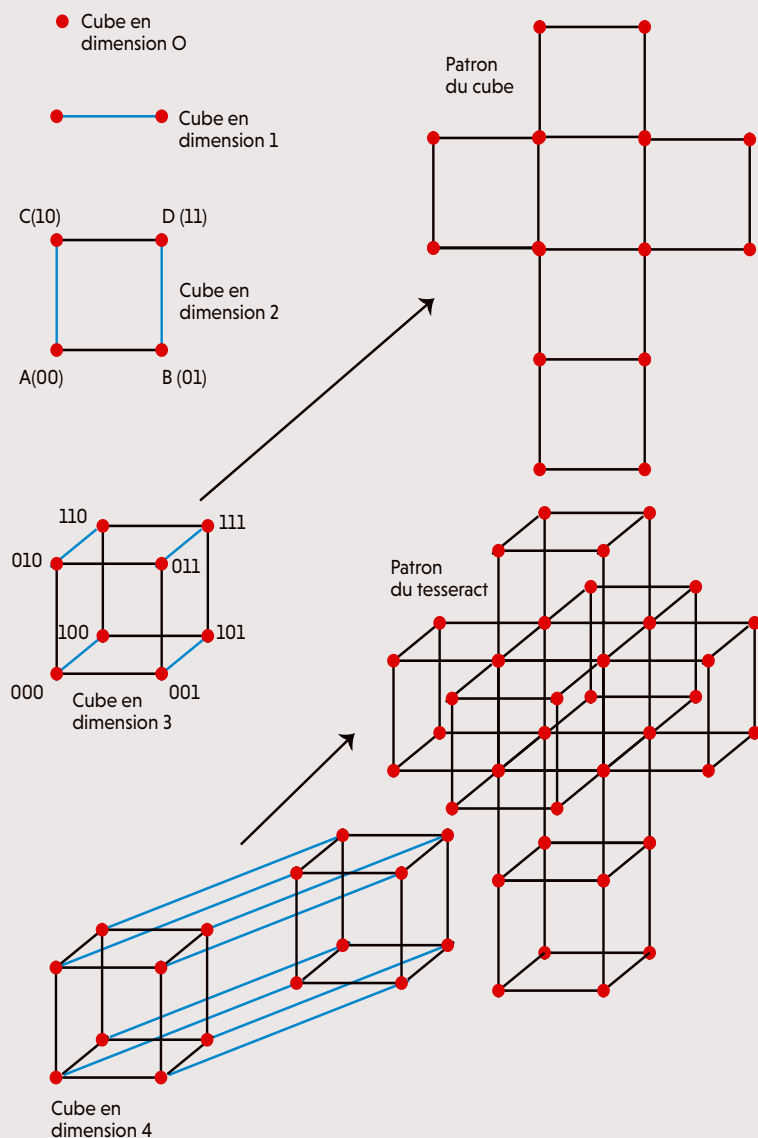
H_n a 2^n sommets qui sont les points dont toutes les coordonnées sont des 0 ou des 1. Il a $n \cdot 2^{n-1}$ arêtes, qui sont tous les segments de droite liant deux sommets ayant exactement $n-1$ coordonnées identiques. Par exemple, les sommets $(1, 0, 0, 1, 0)$ et $(1, 0, 0, 1, 1)$ sont liés par une arête dans le cas de l'hypercube de dimension 5.

Les coordonnées des sommets de H_n sont obtenues en ajoutant un 0 devant chacune des coordonnées des sommets de H_{n-1} et en ajoutant un 1 devant chacune des coordonnées des sommets de H_{n-1}' . Ainsi, les sommets $(0, 0, 1, 0, 0)$ et $(1, 0, 1, 0, 0)$ de H_5 proviennent du sommet $(0, 1, 0, 0)$ de H_4 .

La Grande Arche de la Défense, à Paris, est une représentation en trois dimensions d'un hypercube de dimension 4 parfois dénommé « tesseract » ou « octachore ».

On peut déplier les faces d'un cube pour les aplatir, ce qui donne six carrés collés en forme

de croix (l'une des branches étant plus longue) C'est ce que l'on désigne par un « patron ». De même, on peut déplier les « faces » de l'hypercube de dimension 4 qui sont des cubes de dimension 3 ; cela donne une sorte de croix composée de huit cubes (voir le schéma). Cette croix a servi de base au tableau *La Crucifixion (Corpus hypercubus)*, de Salvador Dalí, peint en 1954 ; la géométrie lui avait été enseignée par le peintre mathématicien Marcel Duchamp. Sur le carrelage représenté, on distingue le patron du cube à trois dimensions.



2

> on joint 0000 à 1000, puis 0001 à 1001, etc. Au total, on a les 12 arêtes du premier cube, les 12 du second et 8 pour les relier, ce qui fait $2 \times 12 + 8 = 32$ arêtes. Une arête correspond à un segment reliant un sommet, par exemple 0100, à un sommet dont les coordonnées ne diffèrent que par une coordonnée: les quatre voisins de 0100 sont 1100, 0000, 0110 et 0101.

On parcourt les sommets du graphe H_4 sans repasser deux fois par le même et en n'utilisant que les arêtes de H_4 en empruntant le chemin: 0000-0001-0011-0010-0110-0111-0101-0100-1100-1101-1111-1110-1010-1011-1001-1000.

On a repris le parcours hamiltonien de H_3 en ajoutant un 0 devant chaque triplet. Puis on a repris à nouveau le même parcours à l'envers en ajoutant un 1 à chaque triplet. Par construction, à chaque fois que l'on passe d'un sommet au suivant, une seule coordonnée change.

Ce que nous venons d'expliquer se généralise de façon évidente pour l'hypercube de dimension n , H_n , qui a 2^n sommets: il se construit en prenant deux copies de l'hypercube de dimension $n-1$ et en y ajoutant des arêtes pour joindre les sommets analogues.

Le nombre d'arêtes de H_n est donc le double du nombre d'arêtes du graphe H_{n-1} , auquel on ajoute le nombre de sommets du graphe H_{n-1} , ce qui donne $n2^{n-1}$. Par la même méthode que pour H_4 , on dispose d'un parcours hamiltonien de H_n : on ajoute des 0 en tête des sommets du

parcours hamiltonien de H_{n-1} , puis, en reprenant ce parcours à l'envers, on ajoute des 1 en tête de chaque sommet d'une copie de H_{n-1} .

Voici maintenant trois énigmes, de la plus facile à la plus difficile, concernant H_3 , H_4 et H_5 qui éclairent la conjecture devenue théorème grâce à Hao Huang.

SOMMETS NON LIÉS

Problème 1. Quel est le nombre maximum de sommets que l'on peut choisir dans le cube H_3 tels qu'aucun ne soit relié à un autre par une arête?

La réponse est 4 (voir l'encadré 3). Il faut quelques secondes de réflexion pour se persuader qu'il est impossible de faire mieux. Si l'on veut être parfaitement rigoureux, on recherchera un raisonnement (ce n'est pas difficile), ou bien on énumérera les 56 façons de choisir 5 sommets parmi les 8 du cube et l'on constatera que pour chacune, il y a deux sommets reliés par une arête... et même qu'il existe toujours un sommet relié à deux autres.

Problème 2. Quel est le nombre maximum de sommets que l'on peut choisir dans l'hypercube H_4 de dimension 4, tels qu'aucun ne soit relié à un autre par une arête? La réponse est 8. L'encadré 3 indique comment choisir ces 8 sommets non reliés par une arête.

Montrer qu'il est impossible de faire mieux que 8 est plus compliqué que dans le problème 1. On se persuadera peut-être en essayant

UN JEUNE CHERCHEUR OBSTINÉ

Hao Huang, jeune mathématicien actuellement en poste à l'université Emory, aux États-Unis, a résolu en 2019 la « conjecture de la sensibilité ». Il avait entendu mentionner cette conjecture en 2012 par Michael Saks, du MIT (Massachusetts Institute of Technology). Il raconte : « À partir de ce moment, séduit par la simplicité et l'élégance de l'énoncé, je suis devenu obsédé par cette conjecture. À chaque fois que j'avais terminé un travail et qu'il avait été accepté pour publication, je revenais à ce problème. Bien sûr, j'abandonnais au bout de quelque temps et retournais vers des questions plus abordables. » Pendant six ans, Hao Huang a donc repensé au problème et y a épisodiquement consacré du temps.

En 2018, il remarqua qu'un théorème vieux de deux cents ans démontré par le mathématicien français Augustin Louis Cauchy pouvait peut-être l'aider. Ce théorème établit des relations

entre les « valeurs propres » d'une matrice et celles de ses sous-matrices, valeurs propres qui sont en rapport avec le nombre de voisins des sommets d'un graphe... ce qui justement était la question à laquelle avait été ramenée la conjecture de la sensibilité.

Alors qu'il était en train de rédiger une demande de subvention pour approfondir son idée, il comprit soudain qu'en changeant les signes d'une matrice, tout se mettait en place pour faire tomber la conjecture. En fait, tout était assez simple, il fallait juste lier plusieurs résultats et les manipuler pour voir surgir la solution dont la démonstration n'occupe finalement que deux pages dans l'article que Hao Huang a publié. Depuis, Donald Knuth, célèbre informaticien auteur d'une bible sur la programmation (*The Art of Computer Programming*), a d'ailleurs proposé une démonstration de la conjecture qui tient sur une seule page (voir la bibliographie).



LA CONJECTURE DE LA SENSITIVITÉ

3

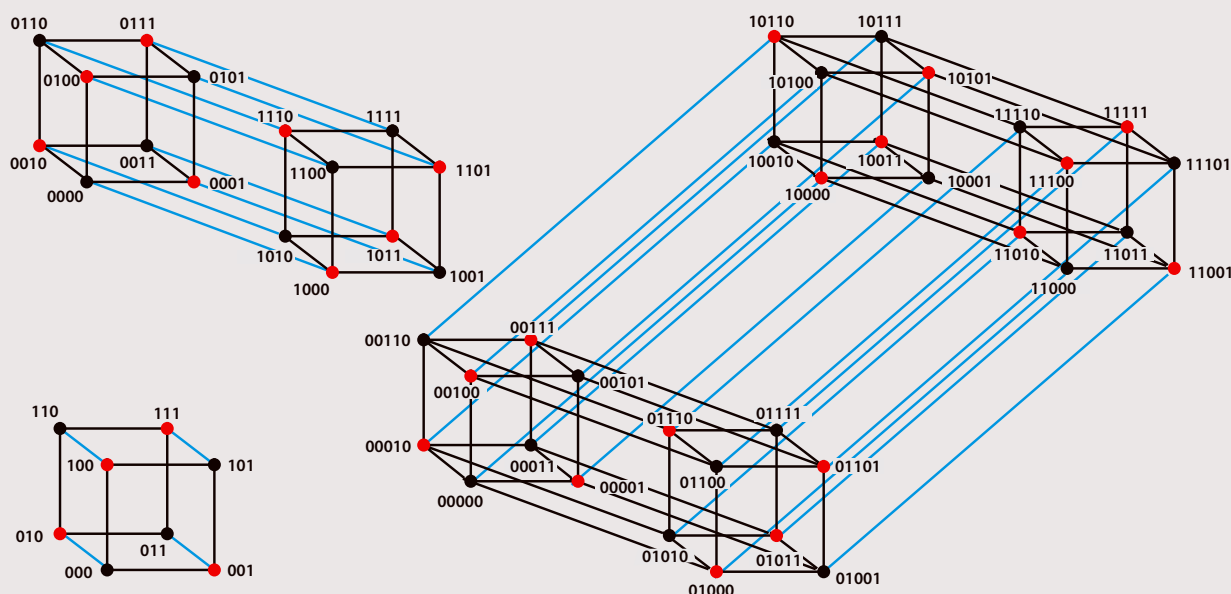
On avait montré que la conjecture de la sensibilité, qui porte sur les mesures de complexité des fonctions booléennes, est équivalente à une conjecture sur les sommets de l'hypercube H_n et c'est cette dernière formulation que Hao Huang a démontrée. Il a montré que si l'on prend $2^{n-1} + 1$ sommets de H_n , alors l'un au moins de ces sommets est relié à au moins \sqrt{n} autres sommets parmi ceux retenus.

Autrement dit : dès qu'on prend 5 sommets du cube H_3 , l'un des sommets est lié à 2 autres ; dès qu'on prend 9 sommets de H_4 , l'un des sommets est lié à 2 autres ; dès qu'on prend 17 sommets de H_5 , l'un des sommets est lié à 3 autres ; et ainsi de suite.

Cette affirmation est d'autant plus étonnante que l'on peut trouver 4 sommets du cube H_3 sans aucun lien, 8 sommets de H_4 sans

aucun lien, 16 sommets de H_5 sans aucun lien, etc.

Les dessins ci-dessous représentent ces façons de choisir $2^{n-1} + 1$ sommets de H_n sans liens entre eux. La méthode utilisée pour ces choix est assez simple : prendre les sommets dont les coordonnées comportent un nombre pair de 1 (en noir sur le dessin).



de trouver 9 sommets et en constatant que l'on n'y arrive jamais. Cependant, pour l'établir en toute rigueur, il faut soit trouver un raisonnement mathématique, soit envisager tous les sous-ensembles de 9 sommets pris dans les 16 sommets de H_4 et constater pour chacun que l'on a deux sommets reliés par une arête. Cela exige d'envisager $16!/(9!7!) = 11440$ cas (nombre de sous-ensembles de 9 éléments pris parmi un ensemble de 16 éléments).

Ce travail d'exploration de 11440 cas est difficilement envisageable à la main. J'ai écrit un programme qui réalise cette énumération et, effectivement, il m'a indiqué que pour chacune des 11440 façons de prendre 9 sommets de H_4 , il y a au moins un sommet relié à un autre. Mon programme m'a aussi signalé que dans chaque cas, il y a au moins un sommet relié à deux autres !

Problème 3. Quel est le plus grand nombre de sommets que l'on peut choisir dans H_5 de telle façon qu'aucun ne soit relié à un autre par une arête ?

La réponse est 16. D'une part, en s'inspirant de ce qui s'est passé pour la dimension 4, il est assez facile de choisir 16 sommets sans

que deux soient reliés par une arête. On prend la solution à 8 éléments pour le premier hypercube H_4 qui engendre H_5 . On prend la solution complémentaire pour le second et tout est bon (voir l'encadré 3). On remarque alors que les sommets retenus sont ceux dont les coordonnées comportent un nombre pair de 1. Il est clair qu'un sommet ayant un nombre pair de 1 dans ses coordonnées n'est relié qu'à des sommets ayant un nombre impair de 1 dans ses coordonnées, ce qui confirme que la solution proposée est satisfaisante. Elle se généralise et montre que l'on peut toujours choisir 2^{n-1} sommets du graphe H_n sans que deux soient reliés.

Bien sûr, pour démontrer que 17 est impossible, le problème combinatoire est devenu colossal, car il y a $32!/(17!15!) = 565722720$ cas à envisager.

Ce qui s'est passé pour le problème 2 se reproduit en plus étonnant encore. Dès que l'on prend 17 sommets, on ne peut éviter qu'un sommet soit lié à trois autres. Le nombre maximum de voisins d'un sommet d'un graphe se nomme le « degré du graphe ». Le plus petit degré possible pour un sous-graphe de 16 sommets de H_4 , qui est 0, passe donc à 3 avec un

> sous-graphe de 17 sommets. De 16 sommets à 17 sommets, le degré minimum passe brusquement de 0 à 3.

Cette fois, ce n'est pas mon ordinateur qui m'a donné cette information, car mon programme n'était pas assez rapide: c'est le théorème de Hao Huang, qui résout la conjecture de la sensibilité.

Le résultat démontré par Hao Huang en 2019, qui a surpris tous les spécialistes, est maintenant facile à comprendre. Il s'énonce ainsi: «Si l'on choisit $2^{n-1}+1$ sommets dans l'hypercube H_n , alors l'un au moins des sommets est relié à au moins \sqrt{n} autres sommets par des arêtes.»

Lorsque $n=3$, on retrouve qu'avec H_3 , dès que l'on prend 5 sommets, l'un est relié à deux autres ($\sqrt{3}=1,7320\dots$), qu'avec H_4 , dès que l'on prend 9 sommets, l'un est relié à deux autres ($\sqrt{4}=2$) et qu'avec H_5 , dès que l'on prend 17 sommets, l'un est relié à trois autres ($\sqrt{5}=2,2360\dots$).

L'intérêt du résultat de Hao Huang est qu'il est vrai pour tout entier positif n , ce qu'aucun ordinateur procédant par énumération ne pourra jamais établir. En soi, le résultat est surprenant puisque, associé à celui indiquant que

2^{n-1} sommets de H_n peuvent n'avoir aucun lien entre eux, il montre que, par exemple pour $n=100$, dès qu'on veut prendre un sommet de plus que 2^{99} (soit $6,338\dots \times 10^{29}$) sommets dans H_{100} , alors on crée toujours un sommet lié à dix autres au moins. De 2^{99} à $2^{99}+1$, c'est-à-dire en ajoutant un seul sommet, on passe du degré minimal 0 au degré minimal 10...

COMPLEXITÉ DES FONCTIONS BOOLÉENNES

Cependant, l'intérêt principal du résultat de Hao Huang n'est pas dans ce petit problème combinatoire relatif à l'hypercube, mais dans ses conséquences sur la mesure de la complexité des fonctions booléennes.

Les fonctions booléennes sont utiles pour la conception des circuits électroniques et en particulier des microprocesseurs. Elles servent aussi en cryptographie pour concevoir des fonctions pseudoaléatoires ou des fonctions de chiffrement. Ce sont simplement des fonctions dont l'ensemble de départ est H_n , l'hypercube de dimension n , et qui prennent une valeur qui peut être 0 ou 1. Le 0 et le 1 sont assimilés à *Faux* et *Vrai* quand on envisage les variables des fonctions booléennes.

Un exemple de fonction booléenne est:

$$f(A, B, C) = ((A \text{ ET } B) \text{ OU } (\text{NON}(A) \text{ ET } \text{NON}(B))) \text{ IMPLIQUE } C.$$

Les variables A , B et C peuvent chacune prendre deux valeurs 0 ou 1, ce qui donne 8 valeurs possibles pour le triplet (A, B, C) .

Les connecteurs **OU**, **ET** et **IMPLIQUE** sont définis par leur sens logique habituel:

$$\begin{aligned} 0 \text{ OU } 0 &= 0; 0 \text{ OU } 1 = 1; 1 \text{ OU } 0 = 1; 1 \text{ OU } 1 = 1; \\ 0 \text{ ET } 0 &= 0; 0 \text{ ET } 1 = 0; 1 \text{ ET } 0 = 0; 1 \text{ ET } 1 = 1; \\ 0 \text{ IMPLIQUE } 0 &= 1; 0 \text{ IMPLIQUE } 1 = 1; \\ 1 \text{ IMPLIQUE } 0 &= 0; 1 \text{ IMPLIQUE } 1 = 1, \end{aligned}$$

$$\begin{aligned} \text{On en tire par exemple que } f(1, 1, 1) &= ((1 \text{ ET } 1) \text{ OU } (\text{NON}(1) \text{ ET } \text{NON}(1))) \text{ IMPLIQUE } 1 \\ &= (1 \text{ OU } (0 \text{ ET } 0)) \text{ IMPLIQUE } 1 \\ &= (1 \text{ OU } 0) \text{ IMPLIQUE } 1 = 1 \text{ IMPLIQUE } 1 = 1. \end{aligned}$$

Les fonctions booléennes utilisées en informatique peuvent comporter des milliers de variables et avoir des expressions comportant des milliers de connecteurs logiques. Chaque fonction s'écrit d'une multitude de façons et, bien sûr, on recherche les écritures les plus simples, car elles conduisent aux circuits et aux programmes les plus petits et les plus efficaces. Il est important de mesurer la complexité des fonctions booléennes; en cryptographie, c'est essentiel, car des fonctions booléennes trop simples seraient faciles à attaquer.

On a donc introduit des mesures de complexité des fonctions booléennes. Plusieurs se présentent naturellement. Quand on veut disposer d'une fonction booléenne avec la certitude qu'elle est complexe, il faut que toutes les mesures de complexité de cette fonction soient élevées, sinon cela signifie sans doute

LA SENSIBILITÉ, MESURE DE COMPLEXITÉ

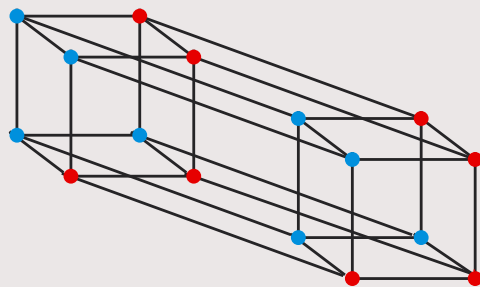
4

Une fonction booléenne de n variables est une fonction qui associe un 0 ou un 1 à chaque sommet de l'hypercube H_n . On peut représenter une fonction booléenne de n variables en dessinant H_n et en coloriant en bleu les sommets où la fonction prend la valeur 1, les autres étant en rouge.

La sensibilité d'une fonction booléenne en un point est le nombre de sommets liés à ce point où la fonction change (de rouge à bleu ou de bleu à rouge). La sensibilité d'une fonction f est le maximum

de la sensibilité de f en x , quand on fait varier x . C'est une sorte de dérivée, liée aux mouvements plus ou moins nombreux de la fonction quand on se déplace sur l'hypercube en suivant les arêtes.

La fonction représentée ci-dessous a une sensibilité égale à 2, car les sommets rouges ne sont jamais liés à plus de 2 sommets bleus, et les sommets bleus ne sont jamais reliés à plus de 2 sommets rouges. La sensibilité est une mesure de complexité pour les fonctions booléennes.



qu'on peut la casser. Pour en être certain, le plus simple est de disposer de résultats mathématiques généraux reliant les diverses mesures de complexité les unes aux autres et qui assurent que, dès qu'une mesure pour f est élevée, c'est aussi le cas pour les autres.

De tels résultats servent à concevoir aisément des fonctions booléennes résistantes aux attaques. Mais un autre motif pour rechercher des liens entre les diverses mesures de complexité des fonctions booléennes est simplement que les mathématiciens aiment savoir, quand ils introduisent des notions portant sur les mêmes objets, quelles relations les lient.

Le résultat de Hao Huang sur les hypercubes a justement comme conséquence qu'une mesure naturelle de complexité, la «sensitivité» (définie plus loin) est liée aux autres. Les travaux antérieurs avaient relié toutes les mesures naturelles de complexité des fonctions booléennes, sauf la sensibilité, qui faisait bande à part.

LA SENSITIVITÉ BOOLÉENNE

Pour saisir l'éclaircissement que Hao Huang a apporté, considérons trois des mesures de complexité les plus courantes et indiquons où le résultat de Huang intervient. Commençons par la mesure de complexité récalcitrante, la «sensitivité».

La sensibilité d'une fonction booléenne f au point $x = (b_1, b_2, \dots, b_n)$ est le nombre de x' liés à x sur l'hypercube H_n tels que $f(x') \neq f(x)$; autrement dit, c'est le nombre de coordonnées de x qui changent la valeur $f(x)$ quand on les modifie. Cette mesure est une sorte de dérivée de f en x : si f bouge beaucoup quand on envisage des points proches de x sur H_n , alors la sensibilité de f est élevée en x .

Par définition, la sensibilité d'une fonction booléenne f est le maximum de la sensibilité de f en x quand on fait varier le point x dans H_n . Cette mesure est une évaluation globale de l'agitation de f : on la note $s(f)$.

Par exemple, la fonction booléenne $f(A, B) = (A \text{ ou } B)$ a une sensibilité 0 en $x = (1, 1)$; en effet, $(1 \text{ ou } 1) = 1$ et quand on change la première coordonnée ou la seconde, le résultat ne change pas, puisque $(0 \text{ ou } 1) = (1 \text{ ou } 0) = 1$. Elle a en revanche une sensibilité égale à 2 en $x = (0, 0)$, car $(0 \text{ ou } 0) = 0$, valeur qui change en modifiant la première ou la seconde coordonnée: $(1 \text{ ou } 0) = 1$, $(0 \text{ ou } 1) = 1$. La sensibilité de la fonction booléenne $f(A, B) = (A \text{ ou } B)$ est donc 2: $s(f) = 2$.

Une deuxième mesure de complexité pour les fonctions booléennes est la «complexité en arbre de décision». Une fonction booléenne étant fixée, on considère les algorithmes qui posent des questions de type: «Quelle est la valeur de la variable X ?» puis, prenant en compte les réponses précédentes, «Quelle est la

valeur de la variable Y ?», etc., jusqu'à avoir assez d'information pour déterminer $f(x)$. Le meilleur algorithme est celui qui sait déterminer $f(x)$ pour tout x en un minimum de questions; ce minimum est la complexité en arbre de décision de f , notée $ad(f)$.

Prenons l'exemple $f(A, B, C, D, E) = (A \text{ ET } B \text{ ET } E) \text{ OU } (C \text{ ET } D \text{ ET NON}(E))$.

Bien qu'il y ait cinq variables, trois questions suffiront toujours pour connaître la valeur de f en un point, car selon que E est vrai (1) ou faux (0), il faut s'intéresser à la première partie de l'expression définissant f ou à la seconde. Le meilleur algorithme d'interrogation est donc:

- 1) Demander la valeur de E ;
- 2) Si E est vrai, demander A , puis, si A est vrai, demander B ;
- 3) Si E est faux, demander C , puis, si C est vrai, demander D .

En trois questions et parfois deux, on connaît ainsi la valeur de f . On ne peut pas faire mieux que 3, donc $ad(f) = 3$.

Une troisième mesure de complexité d'une fonction booléenne est le «degré» défini de la façon suivante. Toute fonction booléenne peut s'écrire de façon unique comme une somme de conjonctions du type:

$$f(A, B, C) = (A \text{ ET } B) + (B \text{ ET } C) + (A \text{ ET } C)$$

où le + désigne le «ou exclusif»: $P + Q$ est vrai si « P est vrai et Q faux», ou si « P est faux et Q vrai». Le degré de f , noté $deg(f)$, est le plus grand nombre de variables qu'on trouve dans un terme de cette expression de f . Ici, c'est 2: $deg(f) = 2$.

Quels sont les liens entre ces trois mesures de complexité et d'autres encore? Pour les deux dernières et les autres, la réponse était connue: chacune est majorée par un polynôme en fonction de l'autre, puisque l'on a toujours:

$$deg(f) < ad(f) < 16 [deg(f)]^8.$$

Pour avoir des relations liant la sensibilité $s(f)$ aux autres mesures de complexité, il était établi que la question se ramenait à celle concernant les sous-graphes de H_n que Hao Huang a résolue et que nous avons expliquée. Ce travail a permis de savoir que $[s(f)]^2 \geq deg(f)$, ce qui, avec les autres résultats connus, permet de conclure à l'équivalence, à un polynôme près, des mesures de complexité des fonctions booléennes.

Le résultat de Hao Huang est théorique, mais il est aussi pratique puisque la vitesse optimale de calcul de certains circuits électroniques est directement liée à ces mesures de complexité. La preuve en deux pages de Hao Huang est assez simple. En une heure de lecture, un mathématicien connaissant le sujet peut la vérifier.

Scott Aaronson, de l'université du Texas à Austin, remarque qu'une recherche pendant trente ans par des dizaines de mathématiciens, certains très brillants, a été nécessaire pour aboutir à une preuve qui se vérifie en une heure: prouver est bien plus difficile que vérifier! ■

BIBLIOGRAPHIE

R. Karthikeyan et al., **On the resolution of the sensitivity conjecture**, *Bull. of the Amer. Math. Soc.*, vol. 57(4), pp. 615-638, 2020.

H. Huang, **Induced subgraphs of hyper-cubes and a proof of the sensitivity conjecture**, *Ann. of Math.*, vol. 190(3), pp. 949-955, 2019.

D. Knuth, **A computational proof of Huang's degree theorem** (démonstration en une page du résultat de Hao Huang), 2019 : <https://www.cs.stanford.edu/~knuth/papers/huang.pdf>

S. Jukna, **Boolean Function Complexity**, Springer, 2012.

H. Buhrman et R. de Wolf, **Complexity measures and decision tree complexity: a survey**, *Theoret. Comput. Sci.*, vol. 288(1), pp. 21-43, 2002.

C. Gotsman et N. Linial, **The equivalence of two problems on the cube**, *J. Combin. Theory Ser. A*, vol. 61(1), pp. 142-146, 1992.

N. Nisan et M. Szegedy, **On the degree of Boolean functions as real polynomials**, *Computational Complexity*, vol. 4, pp. 301-313, 1994.