

- 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 $6,338... \times 10^{29}$) 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:

$0 \text{ OU } 0 = 0; 0 \text{ OU } 1 = 1; 1 \text{ OU } 0 = 1; 1 \text{ OU } 1 = 1;$
 $0 \text{ ET } 0 = 0; 0 \text{ ET } 1 = 0; 1 \text{ ET } 0 = 0; 1 \text{ ET } 1 = 1;$
 $0 \text{ IMPLIQUE } 0 = 1; 0 \text{ IMPLIQUE } 1 = 1$
 $1 \text{ IMPLIQUE } 0 = 0; 1 \text{ IMPLIQUE } 1 = 1,$

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

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

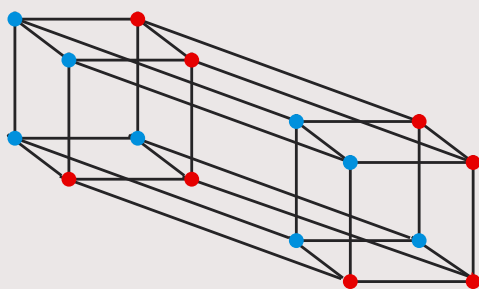
LA SENSITIVITÉ, MESURE DE COMPLEXITÉ

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 assureront 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.