

L'histoire de la conjecture

L'étude du problème du sudoku minimal par Gary McGuire et son équipe a commencé en 2005, par la mise au point d'un premier programme recherchant, pour une grille complète fixée, toutes les sous-grilles partielles correctes.

Cependant, c'est seulement en 2009, voyant que la solution du problème était peut-être atteignable, que G. McGuire et ses collègues ont entrepris des travaux systématiques. En effet, Gordon Royle avait à cette époque réuni une collection importante de problèmes corrects différents à 17 données. Il recherchait en même temps, mais en vain, un problème ne demandant que 16 données. Cette situation rendait plausible la conjecture que 16 données ne suffisent jamais. Une étonnante grille ayant 29 sous-grilles correctes à 17 données attira l'attention (*ci-dessous*). On ne connaît aujourd'hui encore aucune grille complète ayant 30 sous-grilles correctes à 17 données. Encouragé par ce grand nombre de sous-grilles correctes à

17 données, on espéra un moment en extraire une sous-grille correcte à 16 données. En vain, ce qui fut considéré comme un indice de plus en faveur de la conjecture.

Avant d'en arriver au résultat que 16 données sont insuffisantes, un calcul mené par une équipe de l'Université de Graz en Autriche montra, en 2008, que 11 ne suffisent pas. Le calcul entrepris par cette équipe pour montrer que 12 ne suffisent pas fut abandonné sans aboutir. On était donc loin du but. Rappelons toutefois que le raisonnement mathématique pur ne réussit pas mieux que 7 (*voir l'encadré page 77*) ; la démonstration par ordinateur était donc déjà en avance.

En 2008, une jeune fille de 17 ans proposa une preuve mathématique que 16 données sont insuffisantes, mais on découvrit

rapidement une erreur dans son raisonnement, que personne n'a su corriger.

En 2010, une équipe de l'Université Chiao Tung à Taiwan entreprit de s'attaquer au problème des 16 données en utilisant une méthode de calcul distribué. Elle évalua alors qu'avec les outils dont elle disposait, le problème serait résolu en menant un calcul de 2 400 années-processeurs, par exemple, en faisant calculer 2 400 ordinateurs de table chacun pendant un an. Le calcul a abouti en septembre 2013 (après celui de G. McGuire). C'est la seule vérification à ce jour du théorème des 16 données.

Le calcul de G. McGuire et de son équipe se déroula sur toute l'année 2011, opérant l'équivalent de 800 années-processeurs de calculs. Pour chaque grille de sudoku complète, le temps de traitement a été enregistré. Il a fallu en moyenne 3,6 secondes par grille. L'article rendant compte du résultat a été publié en 2014.

de savoir si la solution est G ou G', et donc le problème sera incorrect. Il en va de même avec les six cases en orangé : permuer les deux rectangles de trois données {7, 2, 1} et {1, 7, 2} donne une grille G' qui n'est exclue que si l'on spécifie au moins une des six cases.

De tels ensembles se dénomment des *ensembles inévitables*. Si, pour une grille complète G, on en trouve k , et qu'ils sont disjoints, alors on est certain que toute grille partielle conduisant à G comportera au moins une donnée dans chaque ensemble inévitable et donc en comportera au moins k . Lorsque certains ensembles inévitables ont une intersection commune, la situation, un peu plus difficile, impose tout de même un nombre minimum de données à toute grille partielle conduisant à G.

La lutte finale

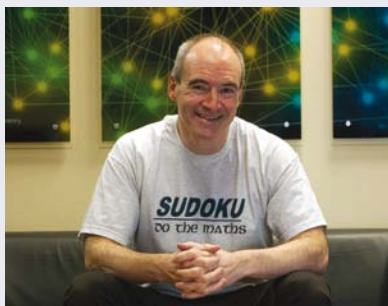
Le travail pour traiter une grille complète G et montrer qu'il faut toujours au moins 17 données se ramène alors à la recherche et à l'exploitation d'ensembles inévitables aussi nombreux que possible. Après divers essais, les chercheurs ont adopté la stratégie suivante, qui a été appliquée à chacune des 5,5 milliards de grilles complètes du catalogue de G. Fowler.

(a) Déterminer (en mettant au point les algorithmes nécessaires) tous les ensembles inévitables de moins de 13 cases. Trouver tous les ensembles inévitables sans limitation de taille, ce qui *a priori* semble meilleur, conduit à plus de calculs au total.

(b) En déduire (toujours en mettant soigneusement au point les algorithmes nécessaires) toutes les sous-grilles de 16 données convenables pour cette famille d'ensembles inévitables, c'est-à-dire ayant au moins un élément dans chaque ensemble inévitable. Cette partie retient considérablement moins de grilles partielles à 16 éléments que les $3,4 \times 10^{16}$ grilles partielles *a priori* envisageables. En moyenne, une grille complète de sudoku a 360 ensembles inévitables de moins de 13 cases.

(c) Pour les grilles partielles retenues, utiliser un programme de test qui indique si oui ou non il y a une solution unique.

6	3	9	2	4	1	7	8	5
2	8	4	7	6	5	1	9	3
5	1	7	9	8	3	6	2	4
1	2	3	8	5	7	9	4	6
7	9	6	4	3	2	8	5	1
4	5	8	6	1	9	2	3	7
3	4	2	1	7	8	5	6	9
8	6	1	5	9	4	3	7	2
9	7	5	3	2	6	4	1	8



LE MATHÉMATICIEN GARY MCGUIRE (*ci-dessus*) et un sudoku ayant 29 sous-grilles correctes à 17 données (*ci-contre*).

Pour mettre tout cela en place et le programmer aussi efficacement que possible, il a fallu développer quelques résultats mathématiques sur les ensembles inévitables, et surtout être soigneux dans la programmation des algorithmes.

Cette qualité des programmes de l'équipe de G. McGuire a permis de gagner la compétition engagée avec une équipe chinoise (parvenue au même résultat, mais plus d'un an après). Le calcul des Chinois n'a cependant pas été vain, puisqu'en aboutissant au même résultat, il appuie l'affirmation que 17 est bien le nombre minimum de données d'une grille partielle correcte de sudoku.

La partie finale de la méthode exige l'utilisation d'un programme pour résoudre les sudokus, programme qui doit être capable d'indiquer qu'une grille partielle est incorrecte car elle autorise plusieurs solutions.

Là encore, le partage gratuit de programmes entre chercheurs a été utile. Plutôt que de développer leur propre programme de résolution de sudokus, l'équipe a préféré rechercher le meilleur de ceux déjà développés. Leur choix s'est porté sur un programme *open-source* (donc libre d'utilisation, contrôlable et modifiable) dû à Brian Turner, un ingénieur de Google qui développe des logiciels de jeux pour son plaisir. Ce programme rapide a cependant été encore accéléré et peut tester en une seconde 50 000 problèmes à 16 données.

Une fois tous les éléments en place et vérifiés, l'exploration du catalogue à la recherche d'une grille partielle à 16 données a été engagée. Après un an de calcul, la réponse est arrivée : il n'en existe pas !

Définitivement résolu ?

Ce formidable travail collectif et informatique pour réussir à démontrer « Le théorème du sudoku » pose cependant un problème : peut-on faire confiance à ce calcul fondé sur une série de programmes délicats fonctionnant une année entière sur des centaines de processeurs différents, et qui n'a été vérifié qu'une fois depuis par une méthode assez proche ?

Le point qui rend difficile une certitude absolue est dû à la nature du calcul, qui n'est

que la recherche infructueuse d'une grille correcte à 16 données. Lorsqu'un ordinateur trouve une solution à un problème et que cette solution est contrôlable par une autre méthode plus rapide que la phase de recherche (comme dans la chasse aux grands nombres premiers), on a peu de raisons de douter de l'exactitude de la solution trouvée. Le premier programme peut comporter des erreurs ou dysfonctionner à cause d'un bug interne ou dans le système d'exploitation, il se peut aussi que le programme oublie une partie du calcul que les programmeurs prévoyaient, etc. Rien de tout cela n'est grave si, à l'issue du calcul, on propose quelque chose de rapidement et indépendamment contrôlable par d'autres programmes ; on se moquera de la phase préliminaire éventuellement très longue et qui n'a plus d'importance pour l'affirmation mathématique.

Ici, la situation est opposée : le programme a recherché quelque chose pendant un an et n'a rien trouvé, et il faut être certain que rien n'a été oublié. Il faut donc être certain que les idées mathématiques sont chacune parfaitement maîtrisées et mises en place, que les programmes ont été écrits sans la moindre erreur, qu'il n'y a pas de bug dans le système d'exploitation de la machine (qui aurait pu faire sauter l'exécution d'une partie des instructions) et qu'à chaque instant le calcul a bien été mené comme il devait l'être sans être perturbé par une panne de disque dur ou par un rayon cosmique. Les craintes sont compréhensibles.

Seule la poursuite du travail sur le sujet, la mise au point de nouvelles méthodes, mathématiques ou techniques, conduisant à d'autres confirmations (si possible plus rapides et avec des programmes indépendants) fournira la certitude attendue. Un assistant de preuve (voir l'encadré page 78) ne serait guère utile, car il est à parier qu'il serait beaucoup trop lent (valider les calculs oblige ici à les refaire, même pour l'assistant de preuve). L'ordinateur est utile, mais restons conscients des erreurs possibles dans les preuves informatiques de non-existence, qui ne doivent pas être considérées avec le même œil que les preuves d'existence. ■

L'AUTEUR



J.-P. DELAHAYE est professeur émérite à l'Université de Lille et chercheur

au Laboratoire d'informatique fondamentale de Lille (LIFL).

BIBLIOGRAPHIE

G. McGuire *et al.*, There is no 16-clue sudoku : Solving the sudoku minimum number of clues problem via hitting set enumeration, *Experimental Mathematics*, vol. 23(2), pp. 190-217, 2014.

J. Cooper et A. Kirkpatrick, Critical sets for sudoku and general graph colorings, *Discrete Mathematics*, vol. 315, pp. 112-119, 2014.

J. Rosenhouse et L. Taalman, Taking Sudoku Seriously : The Math Behind the World's Most Popular Pencil Puzzle, Oxford University Press, 2012.

H.-H. Lin et I.-C. Wu, An efficient approach to solving the minimum sudoku problem, *ICGA Journal*, vol. 3, pp. 191-208, 2011.

Références supplémentaires sur le site www.pourlascience.fr



Retrouvez la rubrique
Logique & calcul sur
www.pourlascience.fr