RENDEZ-VOUS

IA JEU, SET ET MATH

La recherche autour d'un jeu de cartes profite des modèles massifs de langage de l'IA générative.

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)

On s'est parfois moqué de ChatGPT et autres modèles massifs de langage (*large language models*, LLM, en Anglais) à cause des réponses qu'on obtient quand on soumet à leurs agents conversationnels des questions mathématiques. Il est vrai que l'intelligence artificielle (IA) générative s'est révélée fort décevante pour le raisonnement logique. De nouvelles expériences viennent cependant montrer que, sans en attendre des capacités générales spectaculaires, en s'y prenant bien et en s'intéressant à des problèmes précis, les modèles massifs de langage produisent des résultats mathématiques intéressants. Ils peuvent même faire avancer certains sujets de recherche au-delà des lignes où les mathématiciennes et mathématiciens professionnels sont bloqués.

Nous allons décrire un problème combinatoire difficile lié au jeu de cartes SET, qui, depuis plus de cinquante ans, intéresse particulièrement les chercheurs et où, récemment, des méthodes d'IA générative ont permis des progrès. La technique utilisée s'appliquera sans doute à d'autres sujets et conjectures difficiles, ce qui fera peut-être de ce type d'IA de précieux assistants mathématiques d'un nouveau genre.

SET est un jeu de cartes où chaque joueur doit identifier le plus vite possible des familles de trois cartes «compatibles». Il est lié à des questions combinatoires qui se sont avérées d'une difficulté insoupçonnée. Des dizaines d'articles mathématiques ont été consacrés au jeu, et l'IA générative vient d'apporter sa contribution.

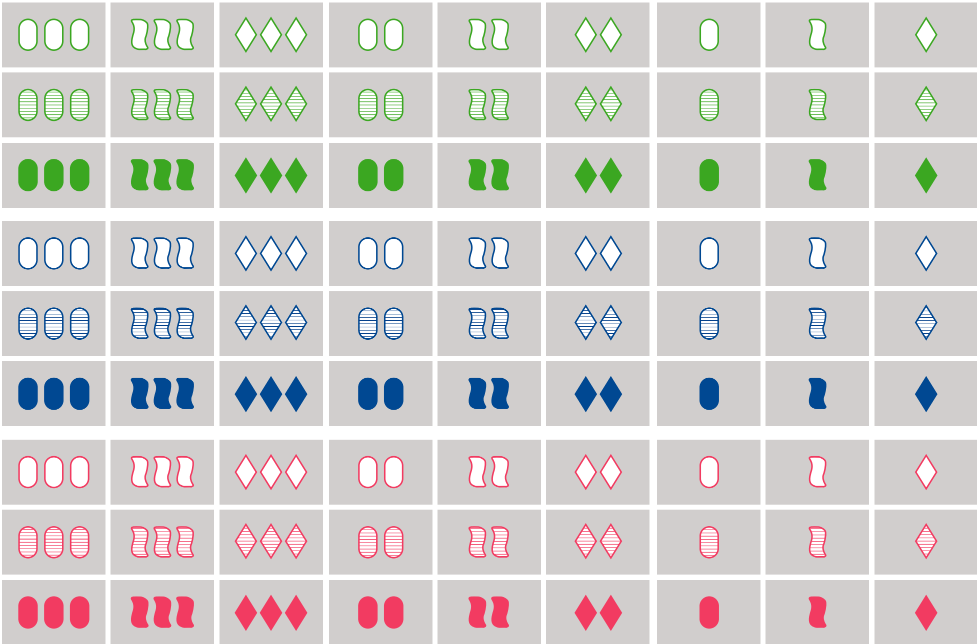
Commençons par décrire ce jeu tel qu'il est proposé par divers éditeurs. Il utilise des cartes très particulières, toutes différentes. Sur chacune on trouve une, deux ou trois formes identiques. Il y a trois formes possibles: l'ovale, la vague et le losange. Il y a également trois couleurs possibles pour les formes: le rouge, le vert et le bleu. Le remplissage du dessin de chaque forme peut être vide, hachuré ou plein. Il y a donc trois possibilités pour chacun des quatre paramètres d'une carte – nombre, forme, couleur, remplissage – ce qui donne $3^4 = 81$ cartes différentes, qui constituent le paquet du jeu.

QUATRE PARAMÈTRES, TROIS CARTES

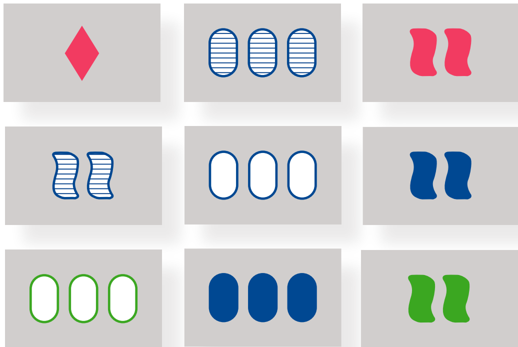
Un SET est un ensemble de trois cartes telles que, pour chacun des quatre paramètres, soit les trois cartes ont la même valeur du paramètre, soit les cartes possèdent toutes une valeur différente du paramètre (*voir l'image page ci-contre*).

Pour jouer, on dispose sur la table douze des quatre-vingt-une cartes du jeu, choisies au hasard. Elles sont rendues bien visibles aux joueurs, qui peuvent être deux ou plus. Le premier joueur qui repère un SET l'indique en disant «SET!». S'il s'est trompé, il ne pourra rejouer qu'après qu'un autre joueur a réussi. S'il a vraiment repéré un SET, il retire les trois cartes correspondantes et les garde. On prend alors trois nouvelles cartes dans la pioche, pour remplacer celles qui ont été retirées, et la partie reprend selon la même règle. Il peut parfois y

Jean-Paul Delahaye a également publié: **Au-delà du Bitcoin** (Dunod, 2022).



Les quatre-vingt-une cartes du jeu *(ci-contre)*, et trois exemples de SET *(ci-dessous)*. Un set est constitué de trois cartes telles que chacun de leurs quatre paramètres (forme, nombre, couleur, remplissage) est soit identique pour les trois cartes, soit différent sur chaque carte.



avoir un blocage, si tous les joueurs s'accordent à dire qu'il n'y a pas de SET à retirer parmi les douze cartes posées. Dans ce cas, on ajoute trois nouvelles cartes aux douze déjà posées, pour en avoir quinze sur la table. Si le jeu est encore bloqué avec quinze cartes, on en ajoute trois nouvelles, etc. Quand le nombre de cartes posées dépasse douze et qu'un SET est repéré et retiré, on ne remplace pas les cartes retirées, de manière à ramener le plus vite possible le nombre de cartes exposées à douze. La partie est terminée quand le paquet de quatre-vingt-une cartes a été vidé et que plus aucun SET n'est possible avec les cartes qui restent sur la table. Le gagnant est celui qui a retiré le plus grand nombre de SET pendant la partie.

Il faut remarquer que, si l'on considère deux cartes du jeu, il existe toujours une unique troisième carte, qui, ajoutée aux deux premières, permettra de composer un SET. En effet, pour chaque paramètre *P*, si les deux cartes ont la même valeur pour *P*, cela impose à celle recherchée d'avoir encore cette même valeur pour *P*, et si les deux cartes ont des valeurs différentes pour *P*, cela impose à celle qu'on recherche d'avoir la troisième valeur possible pour *P*. La carte manquante est donc unique et entièrement déterminée par ce raisonnement.

Une façon de jouer consiste à envisager le plus vite possible toutes les paires qu'on peut faire avec les cartes posées, et à s'interroger sur la présence ou non sur la table de celle qui complète la paire. C'est un exercice d'observation et de vivacité mentale, et bien sûr il est facile d'écrire un programme informatique basé sur cette méthode, qui battra n'importe quel humain, même très entraîné ! Ce n'est pas cette capacité combinatoire des ordinateurs qui est l'objet des récentes découvertes – on sait depuis longtemps qu'elle surpasse la nôtre –, mais bien l'implication des LLM dans la résolution d'une autre énigme combinatoire posée par le jeu de SET et ses généralisations, et que nous allons maintenant décrire.

VISION GÉOMÉTRIQUE

Commençons par interpréter le jeu en termes mathématiques – et même, plus précisément, géométriques. Plutôt que de parler de forme, de couleur, etc. il est plus simple de parler de coordonnées d'un 4-uplet. À chaque carte, on fait correspondre un 4-uplet dont chaque coordonnée correspond à un paramètre (forme, remplissage…), et peut donc prendre une valeur parmi 1, 2 ou 3. On obtient donc, pour chaque carte, un 4-uplet correspondant, comme par exemple (1, 2, 1, 3). Un SET correspond à un ensemble de trois 4-uplets où, quand on prend les composantes en position k de chaque vecteur, soit on observe trois fois le même nombre soit on observe trois nombres différents. Par exemple, les trois 4-uplets $(1, 2, 1, 3)$, $(1, 1, 1, 2)$ et $(1, 3, 1, 1)$ constituent un SET.

En termes mathématiques, on considère l'ensemble $\{1, 2, 3\}$, noté $\mathbb{Z}_3$ (ou parfois $\mathbb{Z}/3\mathbb{Z}$), muni de l'opération d'addition qu'on définit par $1+1=2$; $1+2=3$; $1+3=1$; $2+1=3$; $2+2=1$; $2+3=2$; $3+1=1$; $3+2=2$; $3+3=3$. On prend cet ensemble à la puissance 4, $\mathbb{Z}_3^4$, et on montre alors qu'un SET est, au sens de la géométrie de $\mathbb{Z}_3^4$, un ensemble de trois points alignés de l'espace. Bien sûr, une fois le problème envisagé de cette façon, la généralisation est facile: elle consiste à envisager l'espace $\mathbb{Z}_3^n$ des n-uplets de nombres à coordonnées dans l'ensemble $\{1, 2, 3\}$, et à appeler SET les ensembles de trois points alignés au sens de la géométrie de $\mathbb{Z}_3^n$ – ce qui revient, comme précédemment, à imposer aux paramètres de même coordonnée d'être soit égaux pour les trois points, soit tous différents. Cela revient à considérer un jeu de cartes ayant chacune n paramètres (on pourrait, par exemple, rajouter l'épaisseur des traits, la taille des symboles…), et à rechercher des SET constitués de trois cartes ayant, pour chacun de ces paramètres, soit la même valeur, soit des valeurs toutes différentes.

Cette façon de traduire le jeu en un problème de géométrie en fait un outil remarquable pour découvrir et explorer divers concepts mathématiques. D'ailleurs, un livre entier, publié en 2017 par les presses de l'université Princeton, explore cette richesse du jeu SET.

Quand on joue avec des vraies cartes à quatre paramètres, une question vient rapidement à l'esprit: quel est nombre minimal k de cartes qui garantit qu'il y a au moins un SET quand on prend k cartes parmi les quatre-vingt-une? Formulé de manière équivalente: quel est le nombre maximum de cartes parmi les quatre-vingt-une du jeu constituant un ensemble qui ne contient aucun SET?

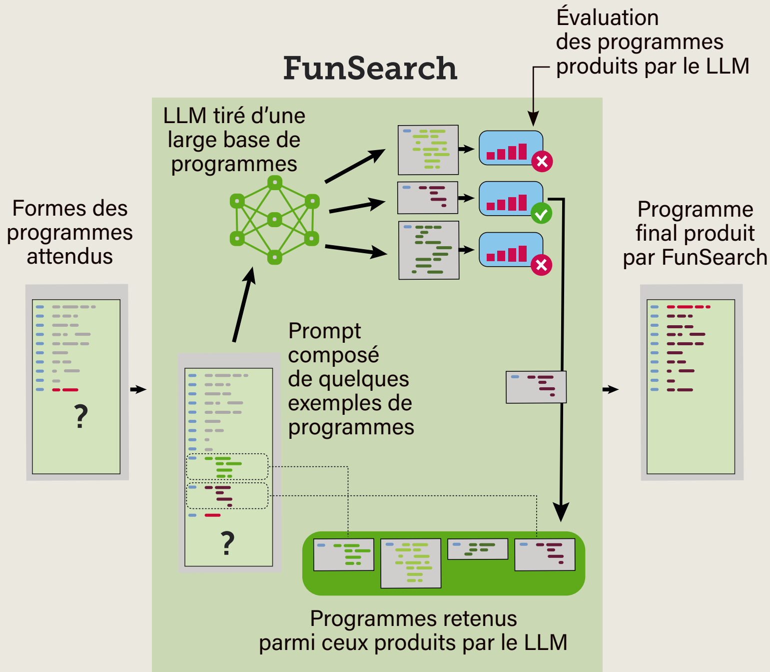
UNE MÉTHODE GÉNÉRALE

Généralisé à $\mathbb{Z}_3^n$, c'est un problème qui occupe les mathématiciens depuis les années 1970, lorsqu'ils se sont aperçus que la question n'était pas du tout facile. Il s'agit de trouver la plus grande taille d'un ensemble de points de $\mathbb{Z}_3^n$ sans aucun alignement de trois points. Un tel ensemble se nomme un Cap SET. La question combinatoire est donc: pour chaque entier n, quelle est la taille du plus grand Cap SET de l'espace $\mathbb{Z}_3^n$?

Pour le cas $n=4$, c'est-à-dire pour le jeu à quatre-vingt-une cartes, il existe une méthode simple pour construire un Cap SET de seize points (donc un ensemble de seize cartes sans SET). Pour chaque coordonnée des points, on choisit deux nombres parmi les trois possibles.

1 FONCTIONNEMENT DE FUNSEARCH

Le programme FunSearch, de la société DeepMind, a battu le record humain dans la recherche de grands Cap SET. Expliquons comment. Un modèle massif de langage (LLM) a été entraîné sur un très gros ensemble de programmes. Cela lui a permis, en quelque sorte, de comprendre la syntaxe du langage de programmation et le type d'utilisations qu'en font les programmeurs humains. Ce LLM a ensuite été utilisé pour produire des programmes ayant une fonction particulière, déterminée par la « forme des programmes attendus » *(voir le schéma ci-dessous)*. Dans le cas qui nous intéresse, ces programmes servent à la recherche de Cap SET.
Plus précisément, après sa phase d'entraînement, on fournit au LLM un *prompt* initial, composé de quelques programmes simples respectant la forme attendue. Puis, d'un cycle à l'autre, ce *prompt* est enrichi de nouveaux programmes produits par le LLM et sélectionnés en raison de la qualité des résultats qu'ils fournissent. Comme les meilleurs programmes sont ajoutés au fur et à mesure au *prompt*, le LLM s'appuie dessus pour en produire de nouvelles variantes, dont on espère qu'elles seront encore meilleures. Le LLM profite ainsi à la fois de tout ce qu'il « connaît » en programmation (le résultat de sa phase d'entraînement initial) et des programmes présents dans le *prompt*, qui le guident dans ses tentatives d'en créer de meilleurs. Après quelques cycles de fonctionnement, on extrait un nouveau bout de programme qui, si tout a bien fonctionné, est meilleur que ceux du *prompt* de départ.
Pour le problème des Cap SET, cela a conduit à un meilleur résultat que ceux qu'ont obtenus préalablement les mathématiciennes et mathématiciens. L'intérêt d'un tel fonctionnement est de bénéficier de la puissance d'un LLM en évitant, grâce à la phase d'évaluation du cycle, de générer des résultats faux ou délirants, ce qu'on a pu reprocher à d'autres LLM.



Par exemple, 1 et 2 pour la première coordonnée, 1 et 3 pour la deuxième, 1 et 2 pour la troisième, 2 et 3 pour la quatrième. On considère tous les points qui utilisent un des nombres choisis pour chacune des coordonnées. Comme il y a deux possibilités pour la première coordonnée, deux pour la seconde, la troisième et la quatrième, il y a en tout 2^4 façons de faire ces choix, ce qui donne 16 points (16 cartes). Avec notre exemple: (1, 1, 1, 2); (1, 1, 1, 3); (1, 1, 2, 2); (1, 1, 2, 3); (1, 3, 1, 2); (1, 3, 1, 3); (1, 3, 2, 2); (1, 3, 2, 3); (2, 1, 1, 2); (2, 1, 1, 3); (2, 1, 2, 2); (2, 1, 2, 3); (2, 3, 1, 2); (2, 3, 1, 3); (2, 3, 2, 2); (2, 3, 2, 3).

Par construction, ces 16 points ne contiennent aucun SET et constituent donc un Cap SET. En effet, par construction, la seule façon pour une coordonnée d'un ensemble de trois cartes prises parmi ces seize de satisfaire la contrainte «constante» ou «toutes différentes» est: «constante». Or si les trois cartes ont la première coordonnée constante, mais aussi la seconde, la troisième et la quatrième, alors il s'agit de trois fois la même carte, ce qui n'est pas possible! Cette méthode fonctionne pour toutes les dimensions et donne donc, pour chaque entier n, un Cap SET de 2^n cartes.

En revanche, cette construction ne fournit pas la taille maximum pour un Cap SET en dimension n. Le problème est apparu en 1970 quand le mathématicien italien Giuseppe Pellegrino a trouvé, avant même que le jeu ne devienne populaire, un Cap SET de 20 points dans le cas $n=4$, et démontré que 20 ne pouvait pas être dépassé. Voici ce Cap SET de taille maximale: (1, 1, 1, 1); (1, 1, 3, 1); (1, 1, 1, 3); (1, 1, 3, 3); (2, 1, 2, 2); (3, 1, 2, 1); (3, 1, 1, 2); (3, 1, 3, 2); (3, 1, 2, 3); (1, 2, 2, 2); (3, 2, 2, 2); (1, 3, 2, 1); (1, 3, 1, 2); (1, 3, 3, 2); (1, 3, 2, 3); (2, 3, 2, 2); (3, 3, 1, 1); (3, 3, 3, 1); (3, 3, 1, 3); (3, 3, 3, 3).

Pour les joueurs, le résultat de Giuseppe Pellegrino signifie que, dans le pire cas, douze cartes posées sur la table et ne présentant aucun SET obligeront à en ajouter trois supplémentaires (pour atteindre quinze cartes), puis trois autres (dix-huit cartes), puis trois encore (vingt et une cartes), mais qu'une fois arrivé à vingt et une, les joueurs seront certains d'avoir au moins un SET à découvrir.

Si on note $max(n)$ le nombre maximum de points d'un Cap SET de $\mathbb{Z}_3{}^n$, le résultat de Giuseppe Pellegrino établit que: $max(4)=20$. Mais combien vaut $max(n)$ pour les autres valeurs de n?

On a pu démontrer que $max(2)=4$; $max(3)=9$; $max(4)=20$; $max(5)=45$; $max(6)=112$. Le résultat $max(6)=112$ n'a été obtenu qu'en 2008 par Aaron Potechin, un chercheur de l'université Princeton, aux États-Unis. Le raisonnement prouvant qu'on ne peut pas faire mieux que 112 occupe un article de seize pages.

Le célèbre mathématicien Terence Tao, lauréat de la médaille Fields en 2006 et professeur à l'université de Californie à San Francisco, a indiqué que c'était son problème favori. C'est lié à de la simplicité de l'énoncé, qui n'interdit pas au problème d'être d'une formidable difficulté. Pour Terence Tao, la situation est un modèle pour la théorie de Ramsey, qui cherche à découvrir de l'ordre dans les grandes structures mathématiques, et le mathématicien pense que des progrès sur ce problème permettraient probablement de faire avancer l'ensemble de la théorie.

DES ENCADREMENTS

Le nombre de cas à considérer pour résoudre le problème pour des valeurs au-delà de $n=6$ est considérable, tant et si bien qu'aucune approche systématique, même avec les meilleurs programmes et ordinateurs, n'est aujourd'hui en mesure de produire un résultat certain. Les mathématiciens s'intéressent donc à des encadrements de $max(n)$, du type: $236 \leq max(7) \leq 292$.

Il en résulte que les chercheurs travaillent à la fois à trouver des Cap SET aussi grands que possible et à démontrer qu'un Cap SET ne peut pas avoir plus d'un certain nombre d'éléments. Pour la dimension 7, on a trouvé un Cap SET avec 236 points, ce qui fournit la borne inférieure dans l'inégalité ci-dessus. On a par ailleurs su démontrer qu'un Cap SET de $\mathbb{Z}_3{}^7$ a au maximum 292 éléments, ce qui fournit la borne supérieure. Le tableau des encadrements obtenus à partir de $n=6$ est donné ci-dessous.

La construction expliquée plus haut montre que $max(n) \geq 2^n$, et bien sûr on a $max(n) \leq 3^n$ puisque 3^n est la taille de $\mathbb{Z}_3{}^n$.

Bornes inférieures et supérieures pour l'encadrement de *max*(*n*), pour différentes valeurs de *n*.

n	6	7	8	9	10	11	12	13	14	15
inf.	112	236	496	1064	2240	5040	12544	26432	52864	112896
sup.	112	292	771	2070	5619	15425	42659	118237	330222	926687

Notez que si l'on connaît un Cap SET de dimension n et un Cap SET de dimension m, on peut construire un Cap SET de dimension $n+m$ en construisant chaque vecteur de dimension $n+m$ de ce nouveau Cap SET en «accolant» un des vecteurs du Cap SET en dimension m et un des vecteurs du Cap SET de dimension n. Cela démontre que $max(n+m) \geq max(n) \times max(m)$.

Cela conduit à prouver qu'il existe une constante c telle que $max(n)$ tend vers l'infini de la même manière que c^n, ce qui veut dire que la limite de $max(n)^{1/n}$ tend vers la constante c quand n tend vers l'infini. Les deux inégalités au-dessus indiquent que: $2 \leq c \leq 3$. Mais les chercheurs aimeraient en savoir plus sur la valeur précise de c!

Pour améliorer cet encadrement, ils cherchent donc la plus grande constante c telle que, pour tout entier n, $max(n) \geq c^n$. Le Cap SET de vingt cartes pour $n = 4$ indique que $c > 20^{1/4}$, c'est-à-dire $c > 2{,}114742...$ Trois autres résultats ont lentement fait progresser les choses. En 1994, Robert Calderbank et Peter Fishburn prouvent que $c > 2{,}210147...$ En 2002, Yves Edel prouve que $c > 2{,}217389...$ Et en 2022, Fred Tyrrell prouve que $c > 2{,}218021...$

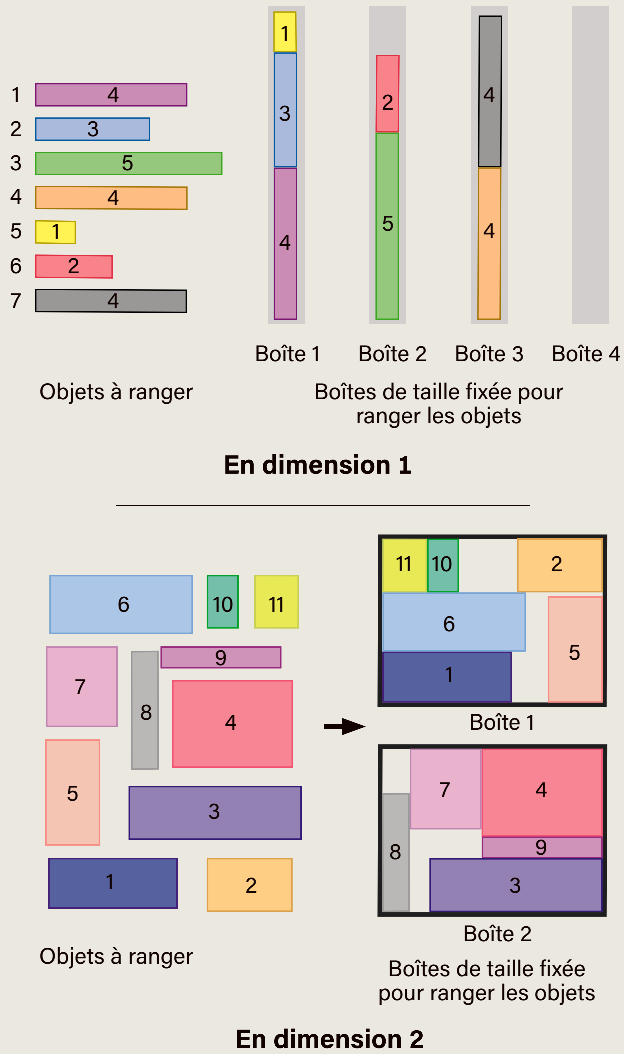
TROUVER LE BON PROGRAMME

Cependant, au grand étonnement des spécialistes, un progrès d'origine informatique vient juste d'être publié. Grâce à une prometteuse méthode d'IA utilisant un LLM, on sait à présent que $c > 2{,}2202...$, ce qui constitue le nouveau record de précision.

Quand on recherche un Cap SET par programme, explorer l'ensemble des possibilités est inconcevable à cause de la taille de l'espace des solutions possibles. En revanche, on peut utiliser ce qu'on appelle des heuristiques pour n'explorer que certaines parties de l'espace du problème, en tentant de donner la priorité à celles où il paraît plus vraisemblable de trouver une bonne solution. La façon la plus naturelle de créer des Cap SET de grande dimension – ou des objets de même nature appelés les «ensembles admissibles», qui sont un moyen de déduire des Cap SET – est de partir d'un point de l'espace $\mathbb{Z}_3^n$, de lui ajouter un second point en le choisissant «au mieux», etc. Autrement dit: partant d'une liste des points déjà obtenus et constituant un début de solution, on tente à chaque étape de la prolonger d'un nouveau point, qui en fera une solution meilleure. Pour cela, on utilise un petit programme – appelé «fonction de priorité» – qui exprime l'heuristique de la méthode en indiquant les candidats qui semblent les meilleurs. Pour se donner des chances de battre les records précédents, on peut donc chercher de bonnes

2 « BIN PACKING »

Le système FunSearch, qui a permis une meilleure approximation de la constante *c* recherchée par les mathématiciens dans le problème des Cap SET, est utilisable pour d'autres problèmes dont celui qu'on appelle *bin packing* (« remplissage de paquets »).
C'est un problème d'optimisation pour des rangements. En dimension 1, le problème consiste à utiliser un nombre aussi petit que possible de longues boîtes d'une taille fixée, pour y ranger des objets de tailles plus petites *(voir le dessin ci-contre)*. Le problème existe aussi en dimensions 2 et 3, et a une importance pratique pour le découpage des tissus ou des planches (dimension 2) et pour le rangement des containers et le remplissage des soutes de camions et d'avions (dimension 3).
Les objets à ranger peuvent être donnés tous ensemble au départ (problème statique), ou arriver les uns après les autres (problème dynamique).
Plusieurs méthodes sont utilisables pour remplir les boîtes au mieux. On peut prendre en premier les gros objets, ou au contraire les petits ; on peut remplir les boîtes au mieux les unes après les autres, ou toutes en même temps ; etc. Ces méthodes sont ce qu'on appelle des « heuristiques ». Elles sont parfois complexes, et s'expriment sous la forme de programmes. FunSearch a mené une recherche des meilleurs programmes pour un problème dynamique de *bin packing*. Là aussi, l'IA a généré d'intéressantes heuristiques, battant les plus classiques.
Les problèmes où FunSearch est susceptible de donner des résultats sont ceux où, d'une part, l'espace des solutions possibles est un espace de programmes, et où d'autre part on dispose d'une méthode de comparaison des programmes proposés. Cette seconde hypothèse est indispensable à la phase d'évaluation du cycle de fonctionnement de FunSearch.
Les sujets scientifiques pouvant bénéficier des programmes d'intelligence artificielle générative à base de LLM sont divers, et depuis deux ans des chercheurs de différentes disciplines s'en préoccupent. Certains défendent même l'idée d'une méthodologie générale de recherche de ce type : une méthode fonctionnant en boucles et s'appuyant sur l'intelligence artificielle générative pour dépasser ce que les chercheurs humains réussissent à faire seuls.



fonctions de priorité. C'est la qualité de la fonction de priorité – donc de l'heuristique qu'elle représente – qui fera qu'on ira loin ou pas, et qui conduira peut-être à un nouveau record de taille pour le Cap SET déduit.

Les modèles massifs de langage, sur lesquels reposent ChatGPT, Bard, Gemini, Llama, etc., ont beaucoup fait parler d'eux en 2022 et 2023. Ce sont des IA qui utilisent, pour leur apprentissage, de grandes quantités d'exemples de textes dont ils extraient les régularités statistiques pour ensuite en déduire une sorte de logique de fonctionnement général. Quand il s'agit de textes en langages naturels, cela permet aux IA de proposer par la suite de nouveaux textes cohérents, utilisant une orthographe et une grammaire correctes. Ces modèles massifs de langage fonctionnent aussi en partant de textes de programmes informatiques, ce qui leur donne la capacité d'engendrer eux-mêmes des programmes. Il faut vérifier ce qu'ils proposent, mais cela aide réellement certains programmeurs et améliore parfois leur productivité – en atteste notamment le succès de GitHub Copilot, une IA d'OpenAI pour la programmation basée sur un modèle massif de langage.

Exploiter un LLM pour produire des programmes est l'idée reprise et mise en œuvre par Bernardino Romera-Paredes, de la firme DeepMind, à Londres, et son équipe. C'est ce qui a permis de battre les records des mathématiciennes et mathématiciens pour le calcul de la constante *c* des Cap SET. L'IA de cette équipe se nomme «FunSearch» pour «searching in the function space». Elle s'attache à trouver le meilleur programme fixant la fonction de priorité exprimant une heuristique de recherche. Pour arriver au résultat de cette équipe, plus de deux millions de programmes ont été générés et comparés pour nourrir le LLM, en plus des programmes utilisés dans une première étape d'apprentissage général (*voir l'encadré 1*).

LA MÉTHODE FUNSEARCH

Le calcul a très concrètement produit un nouveau record, avec un Cap SET de 518 vecteurs en dimension 8, dont on a déduit l'inégalité $c>2{,}2202...$, qui est plus fine que toutes celles obtenues par les chercheurs humains.

Une explication de l'efficacité de FunSearch est qu'il opère dans l'espace des programmes. Plutôt que de rechercher directement des solutions au problème combinatoire dans l'espace énorme et peu structuré associé naturellement au problème, FunSearch essaie et met en compétition des programmes générant des solutions. Sa connaissance statistique des programmes écrits par des humains lui permet de naviguer «intelligemment», à la recherche des meilleures heuristiques.

Un autre intérêt de la méthode est qu'en examinant les programmes trouvés qui ne sont pas très longs et qui déterminent les heuristiques gagnantes, les chercheurs en extraient des idées mathématiques utiles pour aller plus loin. «La procédure que nous avons suivie est un exemple concret de la manière dont les approches basées sur les LLM peuvent être utilisées dans les sciences mathématiques, expliquent les auteurs de ces travaux. FunSearch propose une solution, que les chercheurs examinent, les conduisant à en comprendre certaines caractéristiques intéressantes. Ces caractéristiques sont alors utilisées pour affiner la recherche, ce qui permet d'obtenir d'encore meilleures solutions. Ce processus peut être itéré, associant en boucle l'homme et la machine.» Cette méthode n'engendre donc pas seulement un résultat brut qui fonctionne sans qu'on comprenne pourquoi: elle donne aussi accès à des idées nouvelles utiles aux chercheurs.

Cette utilisation des LLM dans la recherche en combinatoire est un exemple de synthèse de programmes, c'est-à-dire une méthode qui permet d'engendrer des programmes. D'autres méthodes existent depuis longtemps en IA pour mener des telles synthèses. La question se pose donc de savoir si cette nouvelle méthode à base de LLM dépasse vraiment les plus anciennes. Diverses comparaisons ont été menées, en particulier avec la technique des algorithmes génétiques, où l'espace des programmes est assimilé à un espace de séquences génétiques et dans lesquels le calcul opère des mutations et des mélanges (*crossing over*) dans le but d'obtenir de meilleurs programmes. Aucune des autres méthodes testées n'a produit le résultat de FunSearch concernant la constante *c*. Il semble donc bien qu'on dispose, avec les LLM, d'un outil nouveau pour la synthèse de programmes et la recherche mathématique. ■

Sa connaissance statistique des programmes écrits par des humains lui permet de naviguer intelligemment à la recherche des meilleures heuristiques

BIBLIOGRAPHIE

B. Romera-Paredes *et al.*, Mathematical discoveries from program search with large language models, *Nature*, 2024.

H. Wang *et al.*, Scientific discovery in the age of artificial intelligence, *Nature*, 2023.

H. Zenil *et al.*, The future of fundamental science led by generative closed-loop artificial intelligence, *arXiv preprint*, 2023.

D. Gijswijt, The Beautiful mathematics of the card game SET, *STAtOR*, 2019.

J. Ellenberg et D. Gijswijt, On large subsets of $\mathbb{F}_n^q$ with no 3-term arithmetic progression, *Annals of Mathematics*, 2017.

L. McMahon *et al.*, *The Joy of SET. The Many Mathematical Dimensions of a Seemingly Simple Card Game*, Princeton University Press, 2017.

A. Potechin, Maximal caps in AG(6,3), *Designs, Codes and Cryptography*, 2008.

B. Davis et D. Maclagan, The Card Game SET, *The Mathematical Intelligencer*, 2003.

Y. Edel *et al.*, The classification of the largest caps in AG(5,3), *Journal of Combinatorial Theory*, 2002.

R. Calderbank et P. Fishburn, Maximal three-independent subsets of $\{0, 1, 2\}^n$, *Designs, Codes and Cryptography*, 1994.

G. Pellegrino, Sul massimo ordine delle calotte in S4,3, *Matematiche* (Catania), 1970.