La complexité mesurée...

... par la taille des programmes. Selon la théorie de la complexité de Kolmogorov, est complexe ce qu'on ne peut représenter avec concision.
Les objets du monde semblent distribués en fonction de cette complexité.

Jean-Paul Delahaye

Quand vous explorez les données stockées dans la mémoire d'un ordinateur, vous rencontrez beaucoup de séquences régulières : de longues plages sont occupées par des 0, car elles n'ont jamais été utilisées ; d'autres représentent les pixels d'une coloration identique ou très voisine appartenant à une même zone d'une image. Similairement certains groupements de lettres sont fréquents alors que d'autres sont rares, « grzy » par exemple. Les données stockées dans les mémoires d'un ordinateur sous forme de 0 et de 1 ne ressemblent pas aux résultats d'une suite de tirages à pile ou face.

Les objets du monde ne sont pas des mélanges aléatoires homogènes de tous les atomes stables. Un objet, minéral, végétal, animal ou artificiel, a toutes sortes de régularités : symétries, répétitions, groupements homogènes, alignements, filaments, cellules avec membranes protectrices, etc. Les objets du monde réel et les données informatiques ne suivent pas des lois de probabilité uniformes (qui attribuent à chaque possibilité la même probabilité, par exemple 1/6 pour chaque face d'un dé) comme le résultat des tirages du Loto, ou la succession des numéros qui sortent à la roulette d'un casino. L'informatique théorique a récemment fourni une nouvelle compréhension de ces faits, qui sont si naturels qu'on juge rarement utile de les analyser. L'éclaircissement est venu de la théorie algorithmique de l'information ou théorie de la complexité de Kolmogorov, domaine de recherche né de l'étude mathématique des ordinateurs abstraits. Cette nouvelle conception de l'information et de la complexité a été élaborée vers 1970 par les mathématiciens Ray Solomonof, Andreï Kolmogorov, Gregory Chaitin, Per Martin-Löf et Leonid Levin.

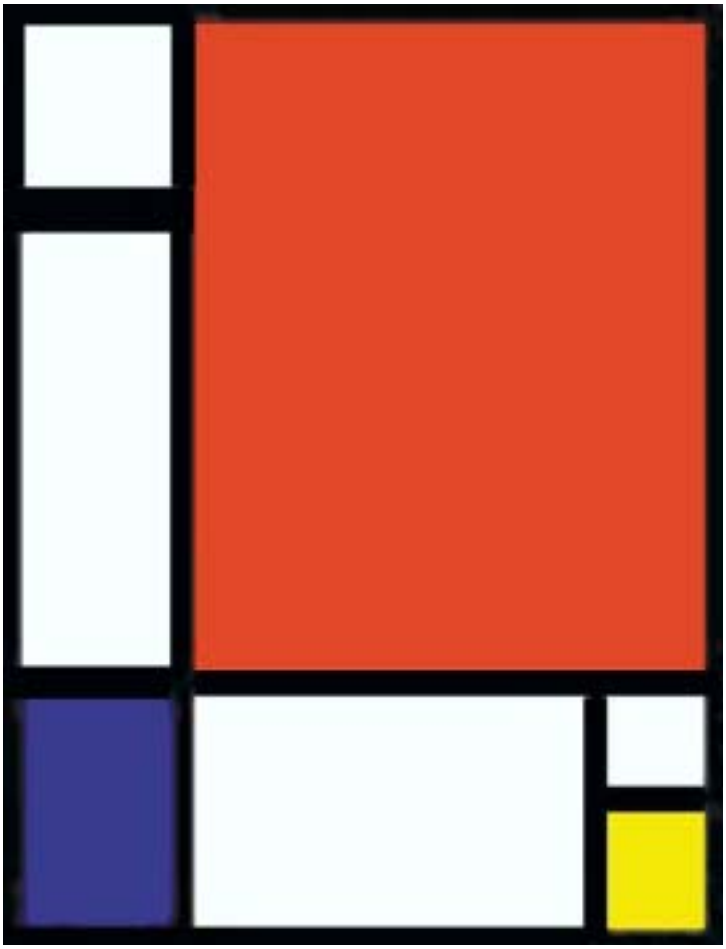
Ce dernier introduisit une nouvelle mesure – dénommée aujourd'hui *mesure de Levin* – qui apparaît aussi importante pour comprendre le monde que les mesures uniformes classiques en théorie des probabilités. Avant de définir la mesure de Levin – qualifiée par Ming Li et Paul Vitanyi de miraculeuse –, nous présenterons la théorie de la complexité en hommage à Kolmogorov dont on a fêté cette année le centenaire de la naissance.

1. Guillaume d'Occam (1280-1349 ?) prônait de ne garder que les concepts strictement nécessaires aux raisonnements théologiques. En suivant ce principe, pour choisir entre les théories compatibles avec les faits observés, on retient la plus simple (le rasoir d'Occam tranche en sa faveur). La notion de simplicité a souvent été considérée comme subjective, mais grâce à la complexité de Kolmogorov elle prend un sens déterminé : la théorie la plus simple est celle correspondant au programme le plus court reproduisant les données.

Est complexe ce qui ne peut se décrire brièvement

Kolmogorov a fondé sa théorie sur une idée naturelle : un objet est complexe quand il n'en existe pas de description courte. Par objets, nous considérerons d'abord des suites de 0 et de 1 auxquelles on ramène tout en informatique, une photographie numérisée ou un programme d'ordinateur. La complexité de Kolmogorov, $K(s)$, d'une suite s est la taille du plus petit programme, noté s^*, qui engendre s, cette taille étant mesurée en bits, l'unité élémentaire d'information.

Haags Gemeentemuseum, Netherlands / Bridgeman Art Library



Piet mondrian, "the winkel mill in sunlight," 1908. Haags Gemeentemuseum. © 2002 mondrian/holtzman trust.

2. Évolution de la complexité des tableaux de Mondrian. À gauche, un tableau de jeunesse et à droite un exemple d'art dépouillé par lequel il a connu la célébrité. Dans un format comprimé, la seconde illustration nécessite six fois moins de bits que la première. La complexité de Kolmogorov des tableaux de Mondrian a baissé au fur et à mesure que son œuvre évoluait.

Dans la théorie de Kolmogorov, on tente d'évaluer la longueur du plus petit programme s^* qui produit la suite s des bits de cet objet. Cette complexité varie peu quand on change de langage, et cette invariance fait de la complexité de Kolmogorov un outil universel de mesure de complexité. Il est difficile à mettre en œuvre pratiquement, mais il est le concept le plus profond pour mesurer la complexité d'objets individuels. Donnons quelques exemples.

La complexité de Kolmogorov d'une suite de un million de 0 est faible (quelques dizaines de bits) : le court programme *pour i allant de 1 à 1 000 000 imprimer 0* produit une suite de un million de 0. De même, la suite du premier million de chiffres binaires de π a une faible complexité de Kolmogorov, car de nombreux programmes courts (fondés sur des séries convergentes vers π) calculent ce million de chiffres.

En revanche, un tirage au hasard d'un million de chiffres binaires, par exemple à pile ou face, donne une suite s qui a toutes les chances d'avoir une complexité de Kolmogorov proche du million. En effet, *(voir la figure 3)*, moins d'une suite sur 1 000 de longueur 1 000 000 a une complexité inférieure à 999 990, et moins d'une sur un million, une complexité inférieure à 999 980. Toute régularité d'un objet permet de raccourcir les programmes qui l'engendrent : ainsi, dans le cas d'un objet symétrique, il suffit de décrire la moitié de cet objet. Il est donc équivalent de dire qu'un objet a une faible complexité de Kolmogorov ou qu'il est régulier et fortement structuré.

Compression et classification

La complexité de Kolmogorov est associée à la compression de données. En effet, si une suite codant un texte peut être comprimée cela entraîne que sa complexité de Kolmogorov est inférieure à la taille de cette version comprimée : la version comprimée d'un texte est une sorte de programme engendrant le texte, et la complexité de Kolmogorov d'une suite est la taille de sa meilleure version comprimée. Dans le cas de la compression d'images et de films, la remarque n'est valable que pour des compressions sans perte, c'est-à-dire permettant la restitution parfaite des objets comprimés. Ainsi, toute méthode de compression sans perte constitue un outil pour le calcul approché de la complexité de Kolmogorov, et c'est pourquoi $K(s)$ est parfois dénommé contenu incompressible d'information de s.

Les applications pratiques de la théorie de la complexité de Kolmogorov sont nombreuses : citons la classification automatique des langues, des morceaux de musique, la reconsti-

tution des arbres phylogénétiques. Toutes utilisent des techniques de compression pour le calcul approché de K(s). La thermodynamique, comme l'a montré Charles Bennet à travers les concepts d'ordre et d'entropie, est aussi associée à la théorie algorithmique de l'information.

Le langage de programmation fixé, on espérerait calculer exactement K(s) pour chaque s donné. Hélas, aucun procédé général ne calcule exactement K(s) en temps fini pour toute suite binaire s et le recours à des calculs approchés de K(s) dans les applications pratiques est inévitable.

Condamné aux approximations par l'indécidabilité

L'impossibilité du calcul exact de K(s) ne résulte pas d'un manque de maturité du domaine de recherche, mais d'une situation mathématique mise en évidence par Kurt Gödel en 1931 : l'indécidabilité logique. Certains énoncés, exprimables dans le langage d'une théorie T et dont on est certain qu'ils sont vrais (parce qu'on les démontre à l'aide d'autres théories), ne sont pas démontrables avec les moyens de T : ce sont les indécidables de la théorie T.

Une théorie mathématique ne connaît la fonction K que pour quelques suites s, et au-delà d'une certaine taille pour s, ne connaît plus aucune valeur de K(s). Il en résulte en particulier qu'aucune méthode algorithmique générale ne détermine K(s) pour toute suite finie s. Cette indétermination a une raison profonde : des structures, invisibles au premier abord, peuvent se trouver cachées dans les objets qui apparaissent alors, à tort, aléatoires.

Nul doute, la complexité est difficile à appréhender et à mesurer. Le fait que la théorie de la calculabilité et les résultats d'indécidabilité fassent de nos intuitions des vérités précises et démontrables est un grand succès de la logique mathématique. Il est étonnant, mais merveilleux, que la théorie de la complexité de Kolmogorov ait si bien traduit formellement l'idée que le complexe est difficile et incalculable. Mais ce n'est pas son seul succès, et la mesure de Levin en est un autre d'une portée philosophique surprenante.

La mesure de Leonid Levin

Parmi toutes les suites binaires, peu d'entre elles sont compressibles. En revanche, nous avons remarqué que, dans la nature, la redondance (et donc la compressibilité) résultant de symétries et de structures prévaut. La mesure $m(s)$ (ou probabilité) de Levin d'une suite s de chiffres binaires prend en compte cet état de fait : cette mesure $m(s)$, liée à la complexité de Kolmogorov K(s), est définie par $m(s) = 1/2^{K(s)}$.

La définition de $m(s)$ exprime qu'un objet résultant d'un programme court a une probabilité forte (m grand). Plus une suite binaire s de longueur donnée est structurée (et donc compressible), plus K(s) est petit et plus la mesure de Levin $m(s)$ est grande : un objet structuré est plus probable qu'un objet aléatoire. En informatique, on a remarqué depuis longtemps qu'une part importante des données se comprime, ce qui explique le succès des méthodes de compression de données sans perte qui n'auraient aucune utilité si les données informatiques suivaient la loi uniforme.

En examinant 40 bits consécutifs provenant d'un support informatique ou résultant de la numérisation d'une expérimentation sur le monde réel, nous rencontrerons plus souvent la suite 0000000000000000000000000000000000000000 que la suite 1101100010111011101000011111101110100100.

Supposons que ces suites de longueur 40 résultent d'un tirage à pile ou face, 1 pour pile et 0 pour face. Ces deux suites ont pourtant la même probabilité : $1/2^{40}$ (mesure uniforme). La mesure uniforme n'est pas un bon indicateur de la probabilité de rencontrer une séquence s.

La première séquence se définit très simplement : 40 « 0 ». L'autre ne peut guère être définie autrement qu'en énumérant ses chiffres : 1, puis 1, puis 0, puis 1, puis 1, etc., ce qui bien sûr est nettement plus long. La probabilité donnée par m est donc bien supérieure pour la suite de 40 « 0 » que pour l'autre suite et cela est conforme à l'idée qu'on rencontrera plus fréquemment la suite 40 « 0 ».

L'idée derrière la définition de Levin est celle d'un monde où les objets sont produits par des programmes ou des mécanismes assimilables à des programmes. Les programmes de ce monde sont d'autant plus probables qu'ils sont courts. Ce que nous rencontrons, ce ne sont donc pas des résultats de tirages directs et uniformes, mais les résultats de tirages indirects : le monde serait équivalent à un ensemble de programmes produits avec une probabilité dépendant seulement de leur taille (les plus longs,

Kolmogorov

3. Suites complexes

Parmi les suites de longueur n, moins d'une sur 2^m a une complexité inférieure à $n - m$.

(a) Parmi toutes les suites binaires de longueur 1 000, une suite sur 2^{10} (= 1024) au plus possède une complexité de Kolmogorov plus petite que 990.

(b) Au plus une suite sur 2^{20} (= 1 048 576) a une complexité inférieure à 980. En choisissant au hasard une suite de longueur 1000, il y a donc très peu de chances de tomber sur une suite compressible.

L'affirmation précédente provient du raisonnement suivant :

(a') Le nombre de programmes binaires de longueur k est inférieur ou égal à 2^k, le nombre de suites de 0 et de 1 de longueur k.

(b') Le nombre de programmes de longueur strictement inférieure à k est inférieur ou égal à $1 + 2 + 2^2 + 2^3 + ... + 2^{k-1} = 2^k - 1 < 2^k$.

(c') En conséquence, parmi les 2^n suites de 0 et de 1 de longueur n, au plus 2^{n-m} ont un programme de longueur strictement inférieure à $k = n - m$, autrement dit encore la proportion des suites de longueur n ayant une complexité de Kolmogorov strictement inférieure à $n - m$ est inférieure ou égale à : $2^{n-m}/2^n = 1/2^m$.

On conclut de ce résultat que si les fichiers informatiques étaient tirés au hasard, ils ne seraient pas compressibles. Or ils le sont le plus souvent. Cela prouve que les fichiers informatiques ne sont pas distribués selon la loi uniforme : ils le sont selon la loi de Levin.

parce que les plus nombreux, étant moins probables), puis ces programmes produiraient les objets que nous rencontrons réellement et qui suivent donc la mesure de Levin : les objets provenant de programmes courts étant plus fréquents que ceux provenant de programmes longs.

Le monde fonctionnerait-il selon une dynamique calculatoire et ne serait-il qu'un grand calcul ? Une telle conception de l'Univers est sujette à débat, mais séduit certains physiciens. Il ne fait aucun doute cependant que même si elle n'est que grossièrement vraie, elle dépeint mieux le monde qu'un désordre total.

Le Soleil se lèvera demain

De cette vision du monde, on tire une conception épistémologique intéressante selon laquelle le travail scientifique est une version algorithmique du principe du rasoir d'Occam. Celui-ci affirme qu'on doit toujours préférer les explications simples aux explications complexes, sa version algorithmique précise que le simple est par définition ce qui est engendré par un programme court.

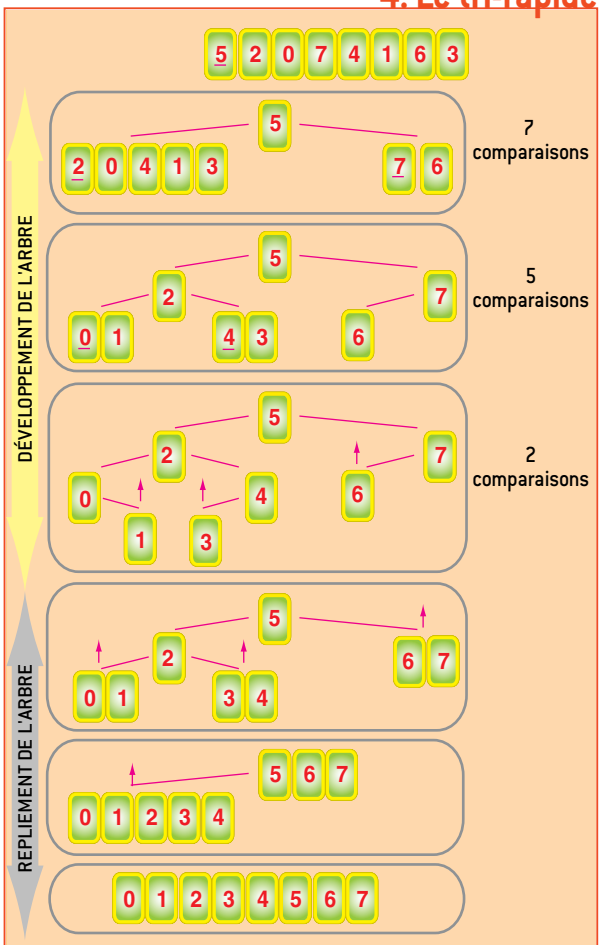
Pour présenter cette conception de la science, défendue par Walter Kirchherr, Ming Li et Paul Vitanyi, imaginons que nous observions des résultats successifs d'une expérience opérée devant nous. Les 10 premiers résultats ont donné : 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0.

On nous demande alors de parier sur le résultat suivant (qui viendra en onzième place). Même sans disposer d'explications précises du phénomène et malgré l'absence complète d'informations sur l'origine des 10 premières données, la grande majorité des gens considère que l'alternance de 0 et de 1 est suffisamment nette et qu'il va de soi qu'on doit parier que la onzième donnée sera un 1. Une telle attitude pourrait être critiquée, car sans informations particulières et puisque les deux séquences de onze données : 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1 et 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0 possèdent *a priori* autant de chances de se présenter l'une que l'autre ($1/2^{11}$) lorsqu'on dispose de 10 résultats consécutifs, il n'y a aucune raison de préférer l'hypothèse que le 1 va venir comme onzième résultat à l'hypothèse que le 0 va apparaître.

Ce raisonnement serait justifié si l'on savait que les données consignent les lancers successifs d'une pièce non truquée, mais ici nous ne faisons aucune hypothèse sur l'origine des données. La critique présuppose, sans justification, que les données suivent une loi uniforme. Elle s'appliquerait même si l'alternance de 0 et de 1 s'était produite mille fois ou un million de fois, elle conduirait aussi à penser que même si le Soleil se lève tous les matins depuis des milliards d'années, on ne doit pas favoriser l'hypothèse qu'il se lèvera demain par rapport à l'hypothèse qu'il ne se lèvera pas.

C'est absurde et nos actions quotidiennes contredisent cette conception uniformaliste de l'Univers. Il n'est pas vrai que les deux séquences de 12 éléments 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1 et 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0 soient également probables. La première est plus vraisemblable, car plus simple, c'est-à-dire définissable brièvement. La première possède une complexité de Kolmogorov plus

4. Le tri-rapide



Le tri-rapide est un algorithme permettant de classer par ordre croissant des nombres ou des données. Sa description est ici expliquée avec un jeu de cartes.

Étape 1. On prend la première carte (la carte « pivot »), on la place sur la table au milieu en laissant un espace en dessous d'elle (on imagine cet espace divisé en une succession de lignes). La carte pivot constitue la ligne 1. On forme alors deux paquets en comparant les cartes restantes avec la carte pivot : les cartes plus faibles sont mises à gauche sur la ligne 2, les plus fortes sont mises à droite sur la ligne 2.

Étape 2. On prend les cartes du paquet de gauche de la ligne 2, sauf une qui sert comme précédemment de pivot et qu'on laisse sur la ligne 2. On sépare ce paquet en deux nouveaux paquets plus petits qu'on place sur la troisième ligne comme précédemment. Le paquet de droite de la ligne 2 est traité de la même façon. Il y a donc maintenant une carte sur la ligne 1, deux sur la ligne 2, et quatre petits paquets (au plus) sur la ligne 3.

Étape 3. On retire ensuite une carte de chaque paquet de la ligne 3 qui y reste et on sépare le reste de chaque paquet comme précédemment en deux, ce qui donne huit paquets, au plus, sur la ligne 4.

On continue jusqu'à n'avoir plus que des cartes seules, chaque paquet ayant été totalement fractionné. Les cartes forment alors un arbre à l'envers. On ramasse maintenant les cartes en faisant un repliement de l'arbre, comme l'indique la figure. La complexité de cet algorithme de tri est très délicate à évaluer et seule l'utilisation de la mesure de Levin conduit à des conclusions satisfaisantes.

faible, ou ce qui revient au même une plus grande probabilité au sens de la mesure de Levin. Parier que le 1 sera le onzième résultat est justifié.

Ce qui fonde nos capacités inductives et prédictives est que nous attribuons une plus grande probabilité *a priori* aux hypothèses les plus simples. Sans le savoir, nous utilisons la mesure de Levin ou quelque chose qui y ressemble. On peut défendre que la science fonctionne de la même manière : face à des données provenant du monde réel, nous retenons les hypothèses compatibles avec ces données, et entre ces hypothèses concurrentes nous choisissons en donnant préférence aux plus simples, c'est-à-dire à celles considérées plus probables par la mesure de Levin.

Que la théorie de complexité de Kolmogorov nous aide à mieux comprendre la nature de l'activité scientifique s'ajoute au fait qu'elle est utile en biologie et en physique. Il existe un autre domaine encore où elle produit des résultats remarquables et profonds : l'analyse d'algorithmes.

La complexité moyenne

Lorsque l'on compare des algorithmes, deux notions doivent être considérées et soigneusement distinguées : la complexité dans le pire des cas, et la complexité en moyenne.

Par exemple, le fameux algorithme de tri de données inventé par C. Hoare en 1962, nommé tri-rapide *(voir la figure 4)*, classe par ordre croissant toute suite de nombres en opérant au maximum n^2 comparaisons entre nombres : la complexité dans le pire cas du tri-rapide d'une suite de n nombres est n^2.

De plus, l'étude de cet algorithme, quand on l'applique systématiquement pour classer toutes les suites de n nombres, montre qu'il faut opérer en moyenne environ $n\log(n)$ comparaisons entre entiers par tri. Le nombre de comparaisons $n\log(n)$ nécessaires en moyenne est bien inférieur au nombre de comparaisons nécessaires dans le pire cas. La raison en est simplement que certaines distributions des nombres à classer piègent l'algorithme et le font travailler stupidement.

Se pose une question : en pratique doit-on considérer que le tri-rapide possède une complexité en $n\log(n)$ – c'est alors un algorithme recommandable – , ou que sa complexité est en n^2 – c'est alors un algorithme médiocre ?

Une réponse tentante consiste à dire que, dans la réalité, les pires cas ont une faible probabilité de se produire – sinon la moyenne ne serait pas ce qu'elle est ! –, qu'on doit les négliger et qu'on doit considérer que le tri-rapide a une complexité pratique en $n\log(n)$. Il faudrait donc recommander son utilisation. Il se trouve que les programmeurs savent que ce n'est pas vrai : le tri-rapide est dangereux, car l'expérience montre qu'on a souvent à trier des suites presque triées et qu'alors justement il se comporte mal et opère un nombre de comparaisons proche de n^2. L'expérience semble nous dire que le pire cas est probable, alors que le bon sens dicte qu'il ne devrait pas l'être. Comment sortir de ce qui apparaît comme un paradoxe ?

La clef de l'énigme se trouve dans la mesure de Levin. En effet, le résultat de complexité en moyenne (qui indique une complexité de $n\log(n)$) est démontré en attribuant le même poids à toutes les suites de n nombres qu'on doit trier. Cette façon de calculer la moyenne est naturelle et c'est celle usuellement admise lorsqu'on parle de complexité en moyenne d'un algorithme. Mais cette façon de calculer la moyenne correspond justement à ce que nous avons dénommé l'uniformalisme. Elle n'est pas satisfaisante pour traiter les données réelles qu'on trouve dans les mémoires de nos ordinateurs. Pour obtenir une évaluation réaliste de l'efficacité en moyenne du tri-rapide, il faudrait calculer sa complexité en moyenne en pondérant les divers cas par leur probabilité de se présenter, c'est-à-dire par la mesure de Levin.

Un tel calcul de complexité en moyenne pondérée par la mesure de Levin est possible et a été mené par M. Li et P. Vitanyi. Ils ont trouvé que le tri-rapide possède une complexité en moyenne pondérée par m essentiellement équivalente à la complexité dans le pire cas, c'est-à-dire n^2. Ce résultat se généralise et s'applique à tout algorithme : la complexité dans le pire cas est essentiellement égale à la complexité en moyenne pondérée par m. En deux mots, dans le domaine de l'analyse d'algorithme, le pire est (presque) certain.

Dans le cas du tri-rapide, la complexité en moyenne véritable (pondérée par m) du tri-rapide est mauvaise et il ne faut donc pas l'utiliser. La raison de ces résultats est que les pires cas ont une complexité de Kolmogorov faible (car on peut les caractériser à partir des algorithmes) et donc une mesure de Levin importante qui les fait peser lourd lorsqu'on calcule la moyenne pondérée par la mesure de Levin.

Ici, un résultat connu intuitivement des programmeurs a été expliqué par les théoriciens qui de leur côté peuvent y voir la confirmation d'une conception philosophique improuvable, mais intéressante : les données dans le monde informatique se présentent à nous non pas uniformément, mais selon la mesure de Levin, qui sans doute aussi est le meilleur modèle qu'on puisse proposer (en l'absence d'informations précises) pour des données quelconques tirées du monde réel.

Mathématiques, informatique, physique et philosophie se rejoignent dans notre tentative de maîtriser la complexité. La mesure de Levin nous fait progresser en enrichissant notre compréhension du comportement des algorithmes, de l'induction scientifique et du rôle des mécanismes de calculs qui semblent partout présents dans l'Univers.

J.-P. DELAHAYE est professeur d'informatique théorique à l'Univ. de Lille.

Auteur & Bibliographie

L'héritage de Kolmogorov en physique, sous la direction de Roberto Livi et Angelo Vulpiani, Éditions Belin, 2003.

J.P. DELAHAYE, *Information, complexité et hasard*, Éditions Hermès, Paris, 1999, (contient une introduction à la théorie de la complexité de Kolmogorov).

W. KIRCHHERR, M. Li et P. VITANYI, *The Miraculous Distribution* in *The Mathematical Intelligencer*, vol. 19,4, pp.7-14, 1997.

M. Li, P. VITANYI. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer-Verlag, New York, Second édition, 1997 (il s'agit du traité de référence dans le domaine).

De nombreux articles sur l'utilisation de la complexité de Kolmogorov et la mesure de Levin sont téléchargeables à partir des pages internet http://www.cs.ucsb.edu/~mli/ et http://www.cwi.nl/~paulv/kolmcompl.html