Informatique et sciences numériques

M. Bernard CHAZELLE,
professeur à l'université de Princeton (États-Unis),
professeur invité sur chaire annuelle

L'ALGORITHMIQUE ET LES SCIENCES

L'objectif de ces leçons [a] fut de présenter les aspects les plus en pointe de l'algorithmique et de ses relations avec les sciences. Le cours fut divisé en deux grandes parties, l'une tenant aux questions proprement épistémologiques de la discipline, souvent issues de la cryptographie (qu'est-ce qu'une identité, qu'est-ce qu'une preuve ?) et l'autre aux questions scientifiques extérieures à l'informatique (économie, recherche opérationnelle, biologie, etc.). Les séminaires ont étés tenus par Jacques Stern (cryptographie), Claire Mathieu (approximations), Christos Papadimitriou (évolution), Tim Roughgarden (théorie des jeux), Alexandre d'Aspremont (apprentissage), Herbert Edelsbrunner (topologie algorithmique), Kurt Mehlhorn (calcul biologique), et François Baccelli (réseaux).

La complexité de l'aléa

Historiquement, les trois ressources (théoriques) principales du calcul sont *le temps, la mémoire*, et *l'aléa*. C'est sur cette dernière que tous les grands succès des dernières décennies convergent. La question principale est la suivante : est-ce que l'aléa est indispensable pour vaincre l'*intraitabilité* ? En d'autres termes, les bits aléatoires sont utiles en pratique, mais sont-ils nécessaires en théorie ? Le consensus est qu'ils ne le sont pas. Ceci semble paradoxal. Après tout, dans bien des domaines, l'aléa est indispensable : sondages d'opinions, stratégies mixtes en théorie des jeux, problèmes classiques en calcul distribué, méthode de Monte Carlo pour le calcul de la fonction

a. La leçon inaugurale, L'algorithmique et les sciences, a été publiée sous forme de livre imprimé (Paris, Collège de France/Fayard, coll. « Leçons inaugurales du Collège de France », 2013) et de livre numérique (Collège de France, 2013) ; texte intégral disponible en ligne : http://books.openedition.org/cdf/1296. Les vidéos de cette leçon inaugurale, des cours et des séminaires sont disponibles sur le site Internet du Collège de France : http://www.college-de-france.fr/site/bernard-chazelle/ [Ndlr].

de partition en physique statistique, etc. Dans d'autres domaines, l'intuition porte à croire que l'aléa est incontournable : le test d'identité polynomiale par exemple.

La question se pose donc de savoir si tout algorithme probabiliste polynomial peut être dérandomisé. En géométrie algorithmique, la plupart des plus grands résultats des années 90 ont été obtenus précisément à coup de dérandomisation : enveloppes convexes, diagramme de Voronoï, etc. En algorithmique des nombres, la découverte la plus importante a été la dérandomisation de tests de primalité (Rabin-Miller, Solovay-Strassen) due à Agrawal, Kayal, et Saxena. Il reste bien sûr des problèmes où la dérandomisation nous échappe complètement : par exemple, factoriser un polynôme à plusieurs variables, approximer le volume d'un polyèdre convexe, générer un nombre premier entre n et $2n$, ou encore produire un arbre recouvrant minimum en temps linéaire.

La classe BPP regroupe les langages dont l'appartenance peut être décidée en temps polynomial (probabiliste) avec une probabilité de succès supérieure à 2/3. La conjecture prévalente est que P = BPP, ce qui est une façon formelle de dire que l'aléa n'est pas indispensable pour NP. La raison qui se trouve derrière cette croyance est l'existence supposée de fonctions difficiles ou de fonctions à sens unique. Le générateur de Nisan-Wigderson convertit la « dureté » d'une fonction en séquences dont le pseudo-aléa ne peut être contredit sans vaincre la dureté de ladite fonction. La quasi-équivalence entre la complexité d'une fonction et l'(in)existence du pseudo-aléa est un développement récent majeur en théorie informatique.

Les preuves interactives et l'épistémologie algorithmique

La notion de preuve interactive provient de la cryptographie et de la vérification des programmes. L'idée est de formaliser le concept intuitif qu'un enseignement interactif est plus efficace qu'un cours magistral. Le grand théorème IP = PSPACE de Shamir nous en donne la preuve. Rappelons que la classe NP recouvre tous les problèmes dont une solution peut être vérifiée en temps polynomial. Prenons le cas du coloriage à trois couleurs. Étant donné un graphe, je peux vous convaincre facilement de sa tricolorité en vous indiquant comment colorier chaque nœud du graphe. C'est un problème dans NP : sa vérification est déterministe et en forme de monologue. Élargissons le modèle au dialogue en acceptant des questions probabilistes. L'asymétrie intrinsèque de NP se retrouve maintenant sous la forme d'une dichotomie prouveur / vérifieur. Je vais pouvoir vous convaincre que le graphe est tricoloriable sans dévoiler la moindre information sur la façon de le faire : ce sont les preuves interactives *à divulgation nulle*. On peut procéder de même pour toute question dans NP, comme la primalité d'un entier, et même l'élargir à des problèmes en dehors de NP : est-ce que ce jeu a une stratégie gagnante ? Est-ce que ces meubles peuvent être sortis de la salle où ils sont sans être démontés (un problème dur pour PSPACE). Un des grands résultats de la théorie des preuves interactives à divulgation nulle est que l'existence de fonctions à sens unique (*grosso modo*) entraîne la possibilité de divulgation nulle pour tous les problèmes d'intérêt.

De ces travaux ont émergé la classe PCP et le plus grand théorème en informatique des vingt dernières années : NP = PCP (log n, 1). Un langage L appartient à PCP (log n, 1) si $x \in L$ entraîne l'existence d'une preuve de ce fait qui peut être vérifiée en ne consultant que $O(1)$ bits et n'utilisant que $O(\log n)$ bits aléatoires (où n est

la taille de x) ; d'un autre coté, si x n'est pas dans L, alors aucune preuve n'accepte x avec une probabilité supérieure à 1/2. Bien sûr, on peut diminuer cette probabilité d'erreur en répétant le test. La notion de preuve polynomialement vérifiable est fondamentalement différente de celle de divulgation nulle. Dans le premier cas, le prouveur veut convaincre le vérifieur mais ce dernier n'a que très peu de temps à offrir ; dans le deuxième cas, c'est le prouveur qui résiste à dévoiler quoi que ce soit sur la preuve elle-même. Le modèle PCP a permis des avances considérables sur la complexité de l'approximation. Par exemple, on sait que trouver une clique maximale dans un graphe est NP-complet. Grace à PCP, on sait que ça le reste même si on se contente d'une clique polynomialement plus petite.

La complexité de la communication

Des travaux de Chomsky jusqu'au moteur de recherche de Google, l'informatique a longtemps privilégié la syntaxe à la sémantique. Il y a deux raisons simples à cela : l'une est que la syntaxe se formalise naturellement ; la deuxième est que ce choix a été fructueux en pratique. Andy Yao fut un des premiers à systématiser l'observation que beaucoup d'arguments de bornes inférieures en complexité ne mettent en jeu que des arguments de communication. Il en vint à définir un modèle de calcul entre deux participants, Bob et Alice, où le seul coût est le nombre de bits échangés. Par exemple, il est facile à ces deux individus de s'assurer que leurs copies respectives de *Hamlet* sont identiques en n'échangeant qu'un nombre constant de bits (la méthode naïve consiste à échanger le texte dans son intégralité). Ces algorithmes ne peuvent qu'être probabilistes donc il existe un risque d'erreur, qu'on peut faire tendre vers zéro.

Un problème classique de « streaming » est d'estimer le second moment des fréquences d'un texte qui défile devant nous, c'est-à-dire la somme des carrés de m_i, de 1 à n, où n mesure la taille de l'alphabet et m_i le nombre d'apparitions de la i-ème lettre. Si on dénote par m la somme des m_i, il est alors évident de calculer F_2 en utilisant une quantité de mémoire $O(n \log m)$. Un joli résultat de Alon, Matias, et Szegedy (1999) nous dit qu'en fait on peut le faire avec une mémoire de seulement $O(\log n + \log m)$. Le curieux de la chose est que le second moment est un cas isolé : pour tous les autres moments, il est impossible d'obtenir une estimation correcte avec un coût de communication logarithmique. Ceci confirme ce que les géomètres savent depuis les Anciens : la norme euclidienne est très spéciale ! L'argument d'optimalité suit un raisonnement emprunté à la complexité de la communication. Souvent les problèmes de ce genre se réduisent à des estimations de rangs de matrices (qui peuvent être très délicates) ou à des arguments en théorie de la discrépance.

On trouve des applications de la complexité de la communication dans l'algorithmique des enchères, la profondeur des circuits booléens, les bornes inférieures pour structures de données (*cell probe model*), et les jeux de Karchmer-Wigderson.

La théorie algorithmique des jeux

La question philosophique sous-jacente est celle-ci : comment concevoir des mécanismes d'interaction économique tels qu'un comportement individuel égoïste se traduise automatiquement par un comportement collectif (quasi) optimal.

L'exemple classique en théorie des jeux est le fameux dilemme du prisonnier. Bob et Alice sont accusés d'un crime : s'ils coopèrent en ne disant rien, ils écopent de 2 ans ; s'ils se trahissent, ils reçoivent une peine de 5 ans ; si l'un trahit et l'autre coopère, ils écopent respectivement de 0 et 9 ans. Le seul équilibre de Nash (et stratégie dominante) est la trahison mutuelle, ce qui de toute évidence n'est pas aussi favorable que la coopération mutuelle. Même dans une version répétitive de ce jeu, la trahison mutuelle est la seule solution rationnelle sauf si on change les conditions de terminaison, où des variantes comme *tit-for-tat* (donnant-donnant) sont effectives. La théorie des jeux proposée par von Neumann et Morgenstern a eu une grande influence en économie et en théorie de la décision, ainsi qu'en philosophie politique (*cf.* la *Théorie de la Justice* de John Rawls), en théorie de l'évolution et en psychologie (*cf.* la théorie des perspectives [*Prospect Theory*] de Kahneman et Tversky).

La *tragédie des biens communs* trouve des illustrations dans l'infrastructure même d'internet ; par exemple, en créant des situations où le comportement égoïste optimal se retourne contre les participants. La solution est de créer des mécanismes où l'altruisme est préférable pour des raisons d'intérêt purement personnel. Haldane et Wilson ont étudié cette idée en biologie évolutionnaire. Un théorème célèbre de Nash garantit l'existence de stratégies (mixtes) optimales pour n'importe quel jeu fini. Ceci reflète une structure topologique reliée aux points fixes (théorème de Brouwer). La question algorithmique qui se pose est d'évaluer la complexité de trouver un équilibre de Nash. Il a été démontré que le problème, bien que sans doute plus facile que NP, est probablement intraitable : il est complet pour la classe PPAD. La différence essentielle avec NP est l'existence garantie de solutions : une situation qui ne prévaut pas pour les solutions d'équations booléennes.

Des travaux récents sur les mécanismes d'incitation ont produit toute une liste d'avancées sur le « prix de l'anarchie » et, en particulier, sur les protocoles de ventes aux enchères et de routage, notamment pour circonvenir aux paradoxes tels que celui de Braess : rajouter une bretelle à un circuit autoroutier peut conduire à ralentir la circulation. Parmi les critiques valides des modèles de décision reliés aux mécanismes d'incitation, on trouve en premier lieu l'intraitabilité (conjecturée) des équilibres de Nash. Il y a également la critique des hypothèses de rationalité : des recherches actuelles en économie « behavioriste » remettent fortement en question ce fondement des théories néoclassiques.

L'apprentissage automatique et les algorithmes auto-améliorants

Comment prendre des décisions dans un environnement incertain ou, plus généralement, en l'absence d'information ? Imaginons que nous ayons accès à n experts qui, chaque jour, profèrent des conseils s'avérant le lendemain incorrects ou judicieux. Il est surprenant qu'il existe une stratégie nous permettant d'atteindre asymptotiquement la performance du meilleur expert, sans connaître son identité au départ. Intuitivement, la *méthode des poids multiplicatifs* maintient une distribution de probabilités parmi les experts en fonction de leurs performances passées et amplifie exponentiellement le poids des meilleurs. On arrive ainsi à une performance égale à celle du meilleur expert à un coût additif proportionnel à la racine carrée du nombre d'étapes.

La technique des poids multiplicatifs est d'une utilité quasiment universelle. On la retrouve en théorie des jeux, en apprentissage (*boosting*), en calcul de flot maximum dans des graphes pondérés, dans le lemme XOR de Yao, en théorie de l'évolution et en géométrie algorithmique. Cette méthode permet de démontrer le joli théorème suivant : un nombre fini de points dans le plan peuvent être joints pour former un polygone simple tel qu'aucune droite ne peut couper plus d'arêtes qu'un nombre proportionnel à la racine carrée du nombre de points. Ce résultat, qui se généralise facilement en dimension arbitraire, est très utile pour faire de la recherche multidimensionnelle.

La théorie des algorithmes auto-améliorants repose sur le principe suivant : peut-on concevoir des algorithmes qui, ne faisant aucune hypothèse de départ sur la distribution des données qui leur sont offertes, s'améliorent d'eux-mêmes progressivement ? C'est la différence entre l'apprentissage descriptif et prescriptif : plus prosaïquement, c'est la différence entre apprendre pour son propre plaisir et apprendre pour réussir un examen. On peut montrer que la version prescriptive de l'apprentissage est, conformément à notre intuition, bien plus facile. Par exemple, il est possible de définir un algorithme de tri optimal dont la phase d'apprentissage est tellement courte qu'elle ne permet même pas d'estimer l'entropie de la distribution des données. L'idée de base de ces algorithmes auto-améliorants est de construire un classificateur permettant de distinguer entre données typiques et atypiques. Dans le premier cas, on stocke des échantillonnages de solutions, qui, plus tard, nous donnent un accès rapide à des « pré-solutions ». Sinon, on utilise un algorithme ordinaire. Ceci fonctionne à merveille pour des problèmes d'optimisation où des solutions approximatives peuvent être facilement transformées en solutions optimales : ceci est souvent le cas pour les algorithmes de graphes ou d'optimisation linéaire.

Les algorithmes naturels

Le mot *complexité* est à la mode. Le terme, certes, tend à flatter les disciplines qui s'en réclament et ceci explique peut-être cela. Toujours est-il que le mot recouvre plusieurs sens qu'il convient de discerner. Pour la majorité des gens, la signification est *sémantique* : est « complexe » ce qui est difficile à comprendre. Il y a aussi un sens *épistémologique* d'usage courant dans la théorie des systèmes dynamiques, qui concerne la difficulté à prédire (par exemple le temps qu'il fera dans un mois). En informatique théorique, le mot *complexe* a un sens *instrumental* se référant à la difficulté de faire des calculs (classes de langage telles que P, NP, PSPACE, etc.) Il y a un quatrième sens du mot *complexe*, qui est *descriptif* et concerne ce qui est difficile à modéliser. La biologie diffère de la physique par l'existence et l'importance de l'histoire (évolutionnaire) des organismes et des espèces. Pour cette raison, il est maintenant traditionnel de dire que le monde du vivant parle le langage des *algorithmes naturels*. Ce n'est pas qu'un point sémantique : cette approche conduit à un traitement différent de la complexité biologique. Quelle est la pertinence d'une démarche algorithmique dans les sciences de la vie ? En présence d'une masse de données hétérogènes, l'analyse statistique ne suffit-elle pas ? Il y a fort à parier que la réponse est non. Aussi indispensables soient-telles, les statistiques de vol ne vous éclaireront pas plus sur les nuées d'oiseaux qu'une analyse de fréquence de mots dans *Madame Bovary* ne nous dévoilera la psychologie d'Emma. Une approche algorithmique est indispensable.

De même qu'on étudie les équations avec des équations, on doit étudier les algorithmes naturels avec des algorithmes.

On définit comme « agent » toute entité capable de communiquer, calculer et agir par le biais d'un algorithme : cela peut être une personne, un oiseau, un insecte, un microbe ou une molécule. On s'intéresse de préférence à des comportements collectifs, un réseau social par exemple, une volée de grues, une termitière, etc. Un des grands enjeux est de construire des techniques algorithmiques nous permettant d'analyser des algorithmes naturels à des échelles différentes. C'est ce que la *renormalisation* nous permet de faire en physique statistique et l'*abstraction* en informatique. La spécificité des algorithmes naturels est que l'influence entre échelles va dans les deux sens : non seulement le petit *crée* le grand, mais des effets macroscopiques peuvent en retour avoir des effets microscopiques.

Prenons un exemple classique de calcul distribué « naturel » : les *systèmes de Hegselmann-Krause*, qui s'occupent de l'évolution des opinions au sein d'un groupe d'individus. Chaque *agent i* détient, pour prendre un exemple concret, une sensibilité politique exprimée par deux nombres réels x_i et y_i, compris entre 0 et 1. La valeur de x_i signifie la position de l'agent i sur l'échiquier politique (0 pour l'extrême gauche, 1 pour l'extrême-droite, et toutes les valeurs entre les deux). La valeur de y_i indique le libéralisme de l'agent : anarchique si $y_i = 0$ et autoritaire si $y_i = 1$. Les individus communiquent entre eux suivant un principe « d'influence par affinité » selon lequel un agent n'est réceptif qu'aux arguments de ceux dont les opinions sont similaires aux siens. On fixe donc un seuil $r > 0$ et, à chaque étape $t = 1, 2, \ldots$, l'opinion (x_i, y_i) de l'agent i, interprétée comme un point dans le plan, se déplace vers le centre de gravité de tous les points (x_j, y_j) à une distance de au plus r. Les opinions de tous les agents sont mises à jour de cette manière et le processus se répète ainsi indéfiniment. Quand la communication va dans les deux sens, on peut montrer que le système converge toujours. L'absence de réciprocité, par contre, change tout.

Les *systèmes d'influence* étendent ce modèle. On définit l'état d'un agent par un point $\mathrm{x}_i = (x_{1,1}, \ldots, x_{i,d})$. Par exemple, on peut imaginer que l'on modélise ainsi la position de l'agent, son âge, sa température, ses préférences alimentaires, ses opinions politiques, etc. Tous ces systèmes étant markoviens, est stockée dans x_i toute information sur le passé de l'agent i susceptible d'être pertinente pour l'avenir : en d'autres termes, x_i représente la mémoire de l'agent i. Chaque agent dispose de son propre *algorithme de communication* lui permettant de déterminer à tout moment qui est une source d'influence en fonction de l'état actuel de tout le monde. Il a également accès à son *algorithme d'action*, qui lui dit comment mettre à jour son vecteur d'état x_i en fonction de l'état actuel de ceux qui l'influencent. Chaque agent est à même de choisir ses propres algorithmes, tous différents des autres. Notons la haute complexité descriptive potentielle d'un tel système, où chaque agent est muni de son propre algorithme. Bien sûr, certains d'entre eux peuvent être inertes, comme des phéromones ou des signaux chimiotactiques.

Formellement, un système d'influence se définit comme un système dynamique à temps discret muni de deux fonctions f et $\mathcal{G}$. D'une part, $x \in (\mathbb{R}^d)^n \to f(x) \in (\mathbb{R}^d)^n$, où n représente le nombre d'agents, d est la dimension de l'espace ambiant et chaque x_i de $\mathrm{x} = (\mathrm{x}_1, \ldots, \mathrm{x}_n) \in (\mathbb{R}^d)^n$ code la position de l'agent i dans $\mathbb{R}^d$. D'autre part, à tout x correspond un graphe orienté $\mathcal{G}(x)$, muni d'un sommet par agent et donc plongé dans $\mathbb{R}^d$. La fonction-coordonnée f_i de $f = (f_1, \ldots, f_n)$ prend comme entrées les voisins émanant de l'agent i dans $\mathcal{G}(x)$ ainsi que leurs positions, et calcule la nouvelle position $f_i(\mathrm{x})$ de l'agent i dans $\mathcal{R}^d$.

On distingue entre algorithmes d'action f et de communication $\mathcal{G}$ par désir de séparer le syntaxique (qui parle à qui) du sémantique (qui fait quoi). La philosophie du modèle est de s'intéresser à l'émergence de phénomènes macroscopiques dus à des agents cognitivement limités. De fait, il est bien connu que les insectes sociaux (comme les termites) sont souvent à même d'accomplir des tâches d'une sophistication bien supérieure à leurs capacités individuelles. Parmi les applications des systèmes d'influence, on compte les réseaux sociaux dynamiques, les communautés d'insectes, les oscillateurs couplés, les bancs de poissons, la circulation automobile, les mouvements de foule, les circuits cellulaires, les réseaux protéiques, les transmissions de rumeurs, les mouvements de foule, les épidémies, la polarisation politique, la dynamique des populations, le modèle d'Ising, ou encore les réseaux neuronaux ou bayésiens.

Nuées d'oiseaux

Des techniques issues de l'algorithmique ont eu un succès considérable sur des questions théoriques concernant les nuées d'oiseaux. Les modèles sont tous des variantes plus ou moins complexes de celui-ci : le vecteur d'état de l'oiseau i contient six coordonnées indiquant sa position et sa vélocité, ce qui rend la dimension de l'espace de phase égale à $6n$ pour un système de n oiseaux. On définit une relation de voisinage entre les oiseaux, établissant ainsi un réseau dynamique (c'est-à-dire qui change au cours du temps). L'algorithme d'action conduit le vecteur de vélocité à se transformer à chaque étape en la moyenne de la vélocité de ses voisins. L'analyse montre qu'après une certaine période de « pré-mixage », les oiseaux finissent par former des groupes cohésifs qui ne peuvent plus jamais se fragmenter. Le graphe de communication finit par se figer et la vélocité moyenne de chaque composante connexe devient constante. Nos recherches ont déterminé les valeurs asymptotiques maximales de ces intervalles de temps.

Pour prouver ce résultat, il est utile de comprendre pourquoi les méthodes traditionnelles (basées sur les fonctions de Lyapunov) ne peuvent pas réussir. Une approche classique serait de prétendre que le graphe de communication est fixe, de résoudre le système, puis d'étendre le raisonnement au cas dynamique où les arêtes changent au cours du temps. Ceci pose de grandes difficultés. Fixer les relations de voisinage produit un oscillateur couplé amorti. Utiliser les équations du mouvement pour suivre directement la position de chaque oiseau dans l'espace se fait donc en estimant leurs *modes*, ce que la théorie des matrices stochastiques nous permet de faire. Malheureusement, dès qu'on redonne aux oiseaux leur faculté de changer de voisinage en fonction de leurs positions, cette approche échoue : en pratique, il n'y a plus de modes !

On oublie donc l'algèbre linéaire et on met en chantier une approche algorithmique. On introduit la *s-énergie*, une série de Dirichlet qui est un paramètre global nous informant de la cohésion cinétique des oiseaux, c'est-à-dire de leur comportement à petite échelle. Pour borner cette *s-énergie*, on suit une approche d'analyse amortie héritée de l'analyse de structures de données. On distribue à chaque oiseau un certain « capital d'argent ». L'idée est de mettre en place une économie avec des règles de capital qui stipulent combien d'argent chaque oiseau doit avoir à tout moment, ainsi que des règles d'échange qui spécifient combien d'argent un oiseau doit passer, quand et à qui. La morale de l'histoire est qu'un algorithme naturel de nuées s'analyse par le biais d'un autre algorithme, très différent, d'échanges

économiques. En mathématiques, le moyen classique d'étudier une équation est d'en produire une autre. Similairement, comprendre les algorithmes naturels nécessite la conception et l'utilisation (purement analytique) d'autres algorithmes.

Pour une vision macroscopique qui nous renseigne, par exemple, sur la possibilité que deux nuées éloignées puissent plus tard fusionner, on fait appel au *filet de vol*, une structure géométrique en dimension 4. Savoir si deux nuées sont amenées à fusionner par la suite se réduit à détecter la présence de chemins presque droits dans le filet de vol. On retrouve là un écho des intégrales de chemin en physique quantique : il s'agit d'affirmer (ou d'infirmer) une propriété parmi un nombre exponentiel de chemins possibles. Pour ce faire, on répète la démarche antérieure et on invente un nouvel algorithme. On place une « brindille » dans le bec d'un oiseau et on lui donne la liberté de la passer à n'importe lequel de ses voisins quand il le désire. Question : y a-t-il un algorithme de passage de brindille qui la garde plus ou moins le long d'une ligne droite ? Malgré la possibilité de vols en spirale, la réponse est affirmative. Une fois de plus, on étudie un algorithme par le biais d'un autre. Les preuves ne sont plus algébriques mais algorithmiques.

Les systèmes d'influence diffusifs

Les systèmes de Hegselmann-Krause sont *diffusifs*, ce qui indique une transmission d'information par prise de moyennes pondérées avec ses voisins. Une propriété essentielle de ces systèmes est qu'ils se figent dès qu'ils atteignent un consensus : parler à des gens qui pensent comme vous n'est pas de nature à vous faire changer d'avis ; quand toute la cuillère atteint la même température, elle y reste ; etc. Ces systèmes généralisent les processus de diffusion en chimie et biologie. Les algorithmes de communication, eux, restent arbitraires. Pour fixer les idées : on leur permet de calculer toute formule exprimable dans la *théorie du premier ordre des réels*, une théorie décidable et d'une telle généralité qu'elle inclut à peu près tous les systèmes concevables en pratique. Contrairement aux modèles de Hegselmann-Krause, la communication peut être soit réciproque soit à sens unique (graphe orienté).

Est-il possible de classifier *tous les comportements possibles* des systèmes d'influence diffusifs en quelques catégories simples et universelles ? La réponse, surprenante, est oui, et la preuve, bien entendu, est algorithmique. Il y a trois types de comportements, dont l'un est extrêmement rare. Il peut y avoir convergence : le système finit par se fixer sur une configuration de laquelle il se rapproche progressivement. On peut y trouver une périodicité asymptotique : après une phase initiale, le système se met à répéter la même suite de configurations dans le même ordre ou à s'en rapprocher inexorablement. La période du cycle limite est souvent courte mais elle peut être exponentielle dans le nombre d'agents. Enfin, cas plus rare, on peut avoir la complétude de Turing : le système peut simuler un ordinateur et il est donc impossible, par un théorème classique de Rice, de prévoir toute propriété non triviale. Le système peut être chaotique (avec une entropie topologique positive), ce qui signifie que prédire les k prochaines étapes nécessite une précision exponentielle en k. Le chaos et l'universalité de Turing requièrent tous les deux un ajustement fin des paramètres.

Comme dans les systèmes de spins en mécanique statistique, la frontière qui sépare l'ordre du désordre, ici, le chaos du comportement attracteur (point fixe ou cycle limite), résulte d'un équilibre extrêmement délicat entre l'entropie et l'énergie : une sorte de match nul entre deux combattants. La frontière critique apparaît sous

forme d'ensemble de Cantor dans l'espace des paramètres. Les discontinuités des algorithmes de communication créent un quasi-aléa parmi les changements de voisinage – n'oublions pas que la dynamique reste entièrement déterministe. En passant sous silence quelques conditions techniques, les systèmes d'influence de type diffusif sont presque toujours attracteurs. De plus, les communications réciproques (téléphone, Facebook, courriel) stabilisent les opinions qui finissent toujours par se figer alors que celles à sens unique (journaux, radio, télévision) conduisent à la périodicité. Ce sont des résultats théoriques qui vont devoir être validés expérimentalement.

Il est surprenant qu'il soit possible de classifier des systèmes d'une telle généralité. L'analyse nécessite un équivalent algorithmique du groupe de renormalisation, un outil essentiel en physique quantique et statistique. L'idée est de traiter récursivement tout sous-système qui s'isole du reste pendant un certain temps. Par exemple, un groupe d'agents n'ayant aucun contact avec le reste de la communauté peut être étudié séparément jusqu'à ce que la communication reprenne. Cela revient à définir des « super-agents » consistant en des sous-graphes induits du réseau de communication. La propagation d'information est alors traitée comme un flot dans des canalisations. Là encore, le traitement est entièrement algorithmique.

Bien sûr, avant d'analyser un algorithme naturel, il faut le spécifier. À l'inverse des algorithmes humains, ceux de la nature ne sont pas lisibles à l'œil nu. L'approche ascendante (*bottom-up*) qui consiste à décrire la fonctionnalité de chaque agent à la main, comme on le fait couramment dans les modèles essaimage, repose sur des arguments de plausibilité parfois ténus. Une approche descendante (*top-down*) à la fois fonctionnaliste, phénoménologique, et bien sûr statistique est sans doute incontournable. C'est un sujet ouvert.

Séminaire

Chaque leçon fut suivie d'un séminaire sur le même thème, apportant un éclairage différent mais complémentaire.

Jacques Stern (ENS Ulm) a parlé d'alea et de complexité en cryptographie. Claire Mathieu (ENS Ulm, CNRS) a discuté des développements récents sur les algorithmes d'approximations. Christos Papadimitriou (UC Berkeley) a presenté son projet de plongement de l'évolution néo-darwinienne dans la théorie du calcul. Tim Roughgarden (Stanford) a parlé de la difficulté de quantifier l'inefficacité des équilibres en théorie des jeux. Alexandre d'Aspremont (École polytechnique) a discuté des liens étroits reliant l'apprentissage a l'optimisation. Herbert Edelsbrunner (IST Autriche) a parlé des relations entre la biologie et la topologie, notamment par le biais du concept d'homologie persistante. Kurt Mehlhorn (Institut Max Planck) a aborde le sujet du calcul biologique chez les myxomycètes (une classe de protistes). Pour conclure cette série, François Baccelli (UT Austin et ENS Ulm) a discuté des problèmes d'analyse probabilistes des réseaux.