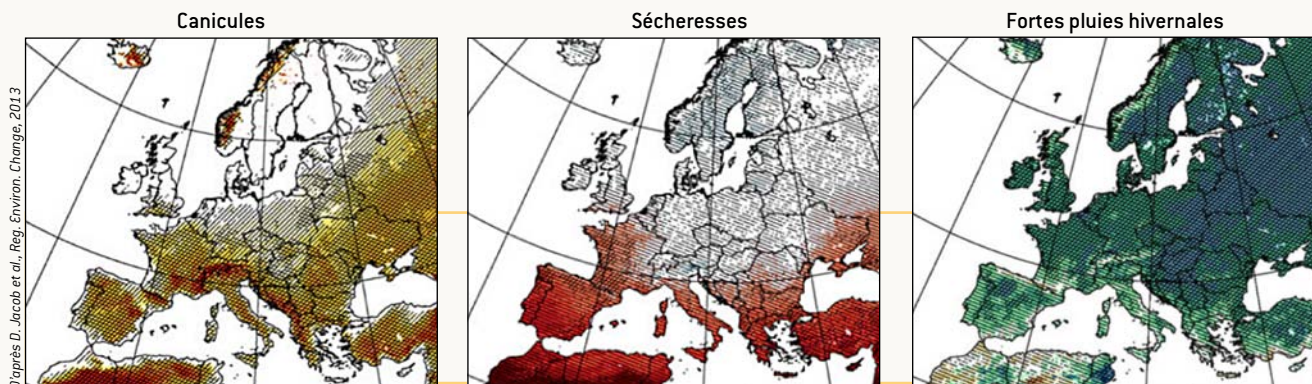




Événements climatiques extrêmes en Europe à la fin du siècle [années 2071-2100 comparées aux années 1971-2000] pour un scénario de fortes émissions de gaz à effet de serre (les zones de forte augmentation sont en rouge pour les canicules et les sécheresses, en bleu pour les pluies).

Dans un scénario à +4 °C, on verrait apparaître des risques systémiques du changement climatique à la fin du XXI^e siècle : réduction des ressources en eau souterraines et superficielles exacerbant les concurrences entre l'agriculture et les autres secteurs, forte réduction du potentiel de production des forêts européennes et de la valeur du foncier, baisse des rendements agricoles et réduction de l'ensemble des services des écosystèmes en Europe du Sud. Les conséquences économiques d'un réchauffement de cette ampleur ont été estimées à une perte annuelle moyenne du produit intérieur brut de un pour cent par an pour l'ensemble de l'Europe, alors que la croissance moyenne historique du produit intérieur brut européen a été de deux pour cent par an.

À l'échelle mondiale, la hausse de la demande alimentaire liée à la croissance démographique et à l'accroissement des richesses pourrait dépasser dans ce scénario l'offre alimentaire étant donné les impacts négatifs du changement climatique. Il en résulterait une baisse de la sécurité alimentaire mondiale et une augmentation de la mortalité infantile.

Vers la transition écologique

Pour éviter ce scénario du pire, il faut trouver des solutions rendant compatibles la production agricole, la réduction des gaz à effet de serre et l'adaptation au changement climatique. L'agriculture intelligente face au climat a été définie comme une agriculture qui augmente durablement la productivité et la résilience (adaptation),

réduit les émissions de gaz à effet de serre (atténuation) et améliore la sécurité alimentaire et le développement. Des systèmes plus résilients peuvent avoir des effets secondaires bénéfiques comme la séquestration du carbone et des réductions de gaz à effet de serre par unité de produit. Une intensification agricole durable permettrait de combler les déficits de rendement et d'augmenter l'efficacité d'utilisation des ressources naturelles par l'agriculture, en particulier dans les pays en développement. Cette stratégie pourrait améliorer la sécurité alimentaire et contribuer à atténuer les changements climatiques en mettant un terme à la déforestation et à l'expansion de l'agriculture sur des écosystèmes sensibles.

Les changements alimentaires et les politiques bioénergétiques peuvent également contribuer. Par exemple, passer d'une consommation de viande d'animaux (bovins, ovins) nourris au grain à celle d'animaux nourris à l'herbe et ne pas utiliser les cultures alimentaires comme source de biocarburants pourrait améliorer la disponibilité mondiale de calories et réduire les impacts environnementaux de l'agriculture. Il s'agit aussi de réduire les pertes après récolte, par l'amélioration du stockage et du transport des aliments dans les pays en

développement. Dans les pays industrialisés, ce sont principalement les gaspillages d'aliments dans la distribution et la consommation qu'il faudra limiter.

À l'échelle d'un pays comme la France, des scénarios illustrent le potentiel d'une transition écologique qui permettrait de réduire d'ici 2050 de 30 à 50 pour cent les émissions de gaz à effet de serre en agriculture (voir l'encadré page 83). L'augmentation de la résilience des systèmes agricoles n'a pas encore été suffisamment explorée dans ces scénarios, même si plusieurs options ont été proposées.

Les efforts de recherche en cours devraient contribuer à combler cette lacune et à dégager les bases d'une transition écologique de l'agriculture, de la forêt et de la gestion de la biodiversité (voir l'encadré page ci-contre). À brève échéance, ces recherches pourraient aussi contribuer à évaluer les potentiels des politiques à mener dans la perspective d'un nouvel accord de l'ensemble des pays, en développement, émergents et industrialisés. Cet accord sera discuté lors de la négociation internationale qui aura lieu à Paris fin 2015 dans le cadre de la XXI^e conférence des parties à la Convention cadre des Nations unies sur le changement climatique.

Bibliographie

- J.-F. Soussana (sous la direction de), *S'adapter au changement climatique, Agriculture, écosystèmes et territoires*, Quae, 2013.
- S. Pellerin et al., *Quelle contribution de l'agriculture française à la réduction des émissions de gaz à effet de serre ?*, INRA, 2013.
- D. Jacob D. et al., *EURO-CORDEX : New high-resolution climate change projections for European impact*, Regional Environmental Change, 2013.
- J. Ciscar et al., *Physical and economic consequences of climate change in Europe*, PNAS, 2010.

LOGIQUE & CALCUL

Les preuves de travail

Donner à des ordinateurs des problèmes à résoudre permet de les freiner. Cette procédure est utile pour lutter contre les attaques sur le réseau Internet, pour limiter l'envoi de spams ou pour organiser des courses entre machines.

Jean-Paul DELAHAYE

Aujourd'hui, les ordinateurs qui sont à notre disposition sont très puissants et nous nous en réjouissons, car cela les rend plus utiles : ils manipulent de longs textes, des photos ou même de la vidéo, ils explorent rapidement des pages Internet à l'autre bout du monde, ils exécutent des programmes de jeux aux effets graphiques époustouflants, etc.

Cependant, cette puissance est parfois mal utilisée et crée des dysfonctionnements. Ainsi, n'importe qui peut envoyer des dizaines de milliers de messages électroniques qui encombrer les boîtes aux lettres au point de les rendre inutilisables : c'est le problème des courriels non sollicités, ou spams. Par ailleurs, quelques ordinateurs coordonnés sont en mesure de soumettre, plusieurs milliers de fois par seconde, des requêtes à un même service en ligne et ainsi d'en empêcher le fonctionnement : ce sont les attaques par « déni de service » ou DoS (pour *Denial of Service*).

Une idée étonnante a été proposée pour contrôler les effets de cet excès de puissance de nos machines. Il s'agit de soumettre aux ordinateurs des problèmes à résoudre, afin qu'ils calculent longuement pour trouver la solution ; alors seulement, ces ordinateurs ont le droit d'accéder à une boîte à lettres ou à un service informatique en ligne.

Ces « preuves de travail » jouent un rôle de plus en plus important. Elles sont combinées aux outils de base de la cryptographie

moderne (chiffrement, signature numérique, authentification, etc.) pour concevoir des protocoles complexes réalisant des opérations considérées comme impossibles il y a peu. Les monnaies numériques décentralisées, dont le célèbre et controversé *Bitcoin*, fonctionnent toutes selon un protocole où les « preuves de travail » jouent un rôle essentiel. Nous y reviendrons.

Ces preuves de travail sont toujours des problèmes dont la solution est difficile à trouver, mais facile à vérifier. L'ordinateur qui doit résoudre le problème soumis ne peut le faire rapidement, mais quand il propose une solution pour accéder au service qui l'exige, la vérification de la solution est immédiate. Il existe de nombreuses méthodes pour fabriquer les problèmes des preuves de travail. On préfère bien sûr celles dont il est facile d'ajuster la difficulté à la situation, pour que le temps de résolution ne soit ni trop court ni trop long.

Des preuves avec ou sans échanges de messages

Deux types de preuves de travail sont utilisés : les systèmes fonctionnant par échanges multiples de messages et les systèmes à un seul envoi.

Dans le cas des « preuves de travail par échanges multiples », voici les étapes du protocole. La machine D (pour *Demandeur*) qui souhaite accéder au service S le contacte ; le service S élabore un problème

et l'envoie à D ; la difficulté du problème est ajustée en prenant par exemple en compte la surcharge du service S à l'instant de la demande, ou ce que S sait déjà de D ; la machine D résout le problème, ce qui exige d'elle un certain travail, donc du temps ; quand D a résolu le problème, il envoie la solution au service S ; le service S vérifie que la solution soumise est bonne, ce qui est facile pour lui, et, si c'est le cas, donne accès au service, c'est-à-dire accepte de placer un message dans la boîte à lettres, ou ouvre la page d'accueil du service en ligne que demande D.

Dans le cas du protocole des « preuves de travail sans échange », tout se passe au cours d'un seul envoi. Selon des paramètres que le demandeur D connaît sans avoir à contacter le service S, un problème à résoudre est défini. La façon de le définir est fixée une fois pour toutes et connue d'avance. Le contenu du message que D veut envoyer détermine par exemple le problème (et donc le problème à résoudre change à chaque demande, ce qui est évidemment essentiel). Le demandeur D connaît donc, sans contacter le service S, le travail qu'il doit réaliser. Quand D a résolu le problème, ce qui lui prend un peu de temps, il envoie la solution trouvée au service S qui ne donne suite à sa demande que si la solution proposée est correcte.

La seconde méthode est souvent préférable : elle est plus simple pour le service qui n'a pas à s'occuper de définir un pro-

blème particulier à chaque demande. Mais la première méthode permet une prise en compte plus fine de la surcharge du service à l'instant de la demande ou d'informations connues de S concernant D .

Le délai nécessaire pour que le demandeur résolve le problème posé est le plus souvent fondé sur un calcul à faire par le demandeur. Le temps qu'il met à résoudre le problème dépend donc de sa puissance de calcul et les machines puissantes sont avantagées. Aussi a-t-on envisagé des preuves de travail exigeant beaucoup de mémoires (plutôt que beaucoup de calculs) ou nécessitant la recherche d'informations sur le réseau. Dans ce dernier cas, les machines puissantes ne sont pas avantagées, puisque ce qui nécessite du temps est lié à la vitesse de circulation des informations sur le réseau, qui est la même pour tout le monde.

L'idée des preuves de travail est due à Cynthia Dwork et Moni Naor qui, dès 1993, suggérèrent cette méthode pour lutter contre les spams. Le système *Hashcash* conçu par Adam Back en 1997 (indépendamment de C. Dwork et M. Naor) et servant aujourd'hui de base dans la plupart des mises en œuvre pratiques des preuves de travail, utilise des « fonctions à sens

unique » pour définir les problèmes soumis et en contrôler finement la difficulté (voir l'encadré 2 pour des exemples).

Une fonction f à sens unique se calcule facilement, mais s'inverse difficilement. Autrement dit, il est rapide de passer de x à $f(x) = y$, mais difficile de trouver, à partir d'un y donné, un x tel que $y = f(x)$. On exigera souvent aussi que les valeurs de $f(x)$ quand x varie se présentent comme si elles étaient tirées au hasard.

Fonctions à sens unique, avec ou sans paramètre

Si l'on dispose d'une telle fonction f , on en déduit facilement une preuve de travail : le service S choisit un y et exige du demandeur D qu'il trouve un x tel que $f(x) = y$. Le demandeur ne peut guère faire mieux qu'essayer des valeurs de x , jusqu'à trouver la bonne, ce qui est impossible à faire rapidement. Il s'agit ici d'une impossibilité pratique : trouver le x est possible en essayant de manière ordonnée toutes les x envisageables, mais cela nécessite du temps.

Plus subtil, car cela ne demande pas l'intervention de S pour la formulation du problème (donc utilisable pour les preuves de travail sans échange), on utilise une

fonction à sens unique dépendant d'un paramètre p , $f_p(x)$, et dont le résultat, y , peut être interprété comme un nombre réel positif. Une telle interprétation est toujours possible : on écrit y en binaire, par exemple 01001110, et on assimile cette suite de 0 et de 1 au nombre réel dont l'écriture en base 2 est 0,01001110. Le service S attend que D lui propose un x tel que $f_p(x) < y$, où p est par exemple le message que D veut envoyer et y un nombre réel, appelé seuil, fixé à l'avance par S . La valeur de y détermine très précisément la difficulté de la preuve de travail, car plus y est petit, plus il est difficile de trouver un x convenable.

Les preuves de travail fournissent aussi un moyen simple d'organiser des courses entre machines sur un réseau. Or ces courses sont au cœur des protocoles de fonctionnement des cryptomonnaies.

Ces monnaies, dont le *bitcoin* est l'exemple le plus connu, fonctionnent sans autorité centrale, la gestion des comptes étant opérée par les machines des volontaires qui se contrôlent mutuellement. Ces contrôles empêchent toute manipulation du cahier de compte de la monnaie (la « blockchain »), lequel indique qui détient des devises (voir *Pour la Science*, décembre 2013).

1. L'idée des preuves de travail

Pour limiter le pouvoir de nuisance que chacun possède du fait de la puissance de son ordinateur, on peut soumettre des énigmes obligeant les machines à de longs calculs et dont seul l'aboutissement leur donne le droit de mener une action, comme envoyer un message électronique ou accéder à un service en ligne. Une machine soumise à ces énigmes ne pourra pas envoyer des milliers de messages non sollicités (spams), ou participer à une tentative de mise en panne d'un service informatique en le submergeant de requêtes (attaque par déni de service).

Le temps de calcul imposé est un prix à payer pour que l'ordinateur ac-

quière le droit de mener une action. Il doit travailler et le prouver par un résultat. La solution produite est la preuve qu'il a fourni ce travail. C'est l'idée des « preuves de travail ».

Les problèmes qu'on demande de résoudre pour ces défis doivent être asymétriques. Il faut que le calcul de la solution prenne du temps, mais que la vérification soit rapide ; sinon, celui qui se protégerait par les preuves de travail serait autant ennuyé que ceux dont il tenterait de se protéger.

On connaît une multitude de problèmes asymétriques. En voici deux exemples :

– L'entier b étant fixé, et p étant un grand nombre premier, trouver un



nombre a tel que $a^2 = b \pmod{p}$. Ce problème de la « racine carrée modulo un grand nombre premier » a été proposé comme preuve de travail dès 1993 par Cynthia Dwork et Moni Naor.

– Trouver les deux nombres premiers p et q dont le produit est le

nombre n qu'on vous a donné sans vous avoir dévoilé p ni q . Il est facile de formuler de telles énigmes – on multiplie deux grands nombres premiers entre eux, ce qui produit n – mais aucun algorithme de factorisation rapide n'est connu, et il faut donc du temps pour retrouver p et q à celui qui ne connaît que n . Il ne faut pas prendre p et q trop grands, car le problème de la factorisation de n deviendrait impossible à résoudre en un temps raisonnable.

Un troisième exemple – le plus important, car le plus utilisé – est expliqué en détail dans les encadrés 2 et 3 et se fonde sur les fonctions « à sens unique ».