

LOGIQUE & CALCUL

Tous les chemins mènent au rond

Le rotor-router est un mécanisme déterministe de cheminement fondé sur des règles simples. Il forme cependant des structures aux propriétés étonnantes et, à certains égards, plus performantes que celles des parcours aléatoires.

Jean-Paul DELAHAYE et Philippe MATHIEU

Laissée à elle-même, la matière dans l'espace se met en boule. La forme ronde apparaît spontanément même si au départ rien n'est rond. En mathématiques, il existe aussi des situations où se forment des ronds alors qu'on ne les attend pas. La situation de ce type la plus extraordinaire est le « rotor-router ». Il a déjà été évoqué dans cette rubrique [voir Mathématiques expérimentales, *Pour la Science* n° 331, mai 2005], mais il est intéressant d'y revenir, car un énorme travail de recherche a été mené à son sujet depuis dix ans et a dévoilé les propriétés de ce mécanisme de déplacement sur un graphe.

Girouettes déterministes

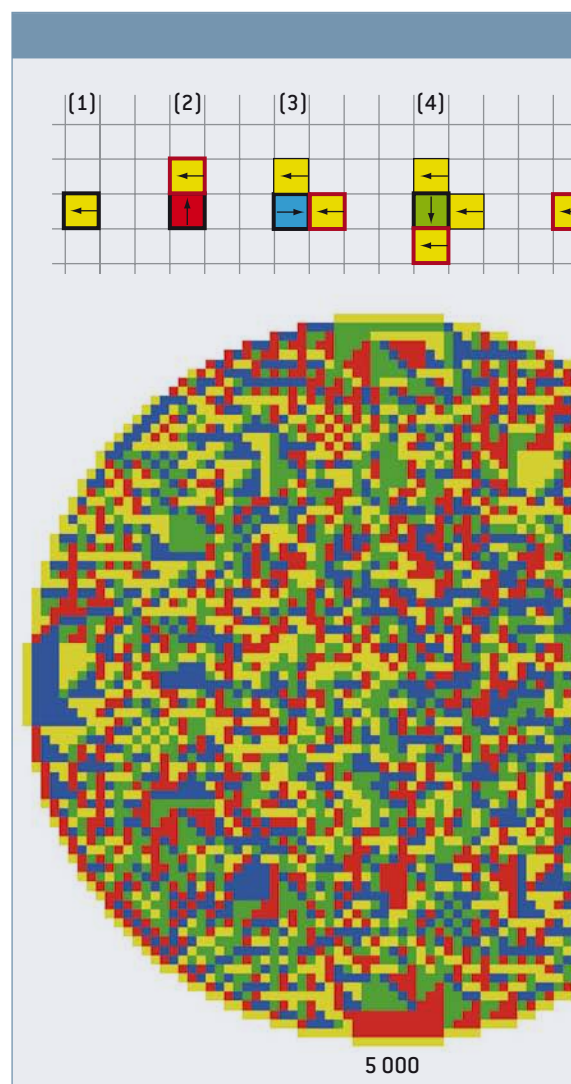
Imaginons un terrain semé de poteaux munis chacun d'une flèche à leur sommet et pointant vers un autre poteau du même type. Un agent mobile, par exemple un personnage, se place en un point *O* où est planté un poteau, et procède ainsi : il fait tourner la flèche du poteau où il se trouve, puis se déplace vers le poteau qu'elle indique alors ; arrivé à ce deuxième poteau, il en fait tourner la flèche et la suit, etc. L'agent continue alors sa promenade qui peut-être le conduira à revenir en *O*, une ou plusieurs fois.

Ce mécanisme élémentaire est celui du rotor-router : chaque flèche tournante, nous dirons rotor, indique la route à l'agent

mobile qui n'a aucune mémoire propre et se fie pour son chemin aux indications des rotors qui pivotent à chaque passage. Dans certains cas, plusieurs agents se déplaceront simultanément et interagiront donc par la rotation des rotors. Ce type de communication sans mémoire propre des agents porte le nom de stigmergie. Dans la nature, il est utilisé par certains insectes, fourmis ou termites, qui, en déposant des phéromones là où ils passent, organisent ainsi des comportements collectifs d'où émergent des activités cohérentes, telle la construction d'un nid.

Si l'on se donne un graphe quelconque – des nœuds reliés par des arêtes (dont on supposera qu'elles n'ont pas d'orientation) –, il est possible de l'équiper de rotors et d'envisager de le parcourir selon le mécanisme décrit. Pour qu'aucun nœud du graphe ne soit oublié, on imposera que le rotor de chaque nœud pointe successivement, de manière cyclique, vers tous les nœuds voisins. Si, par exemple, le nœud *N* a trois nœuds voisins *A*, *B* et *C*, le rotor placé en *N* pointerait vers *A*, puis *B*, puis *C* au cours des passages successifs, puis recommencerait : *A*, *B*, *C*, *A*, *B*, *C*, ...

Le modèle du mécanisme général du rotor-router a été proposé la première fois en 1996 par le Russe Viatcheslav Priezzhev et les Indiens Deplap Dhar, Abhishek Dhar, et Supriya Krishnamurthy sous le nom de « promeneur eulérien » (nous verrons plus loin pourquoi). Le modèle a été redécouvert



5 000

par James Propp, du MIT, qui l'a nommé « rotor-router » en 2005. Plusieurs dizaines d'articles scientifiques et plusieurs thèses lui ont été consacrées.

Commençons par le miracle des cercles parfaits avec la version du rotor-router sur un plan quadrillé. Chaque case du quadrillage est reliée à ses voisines nord, sud, est et ouest, ce qui définit un graphe infini où chaque nœud a quatre voisins.

Initialement, il n'y a aucun rotor, et on dépose les uns après les autres des agents sur une case fixe du plan, dénommée case centrale. Dès qu'un agent trouve une case vide, il y dépose un rotor pointant vers l'ouest, lequel occupe la case. Quand la case

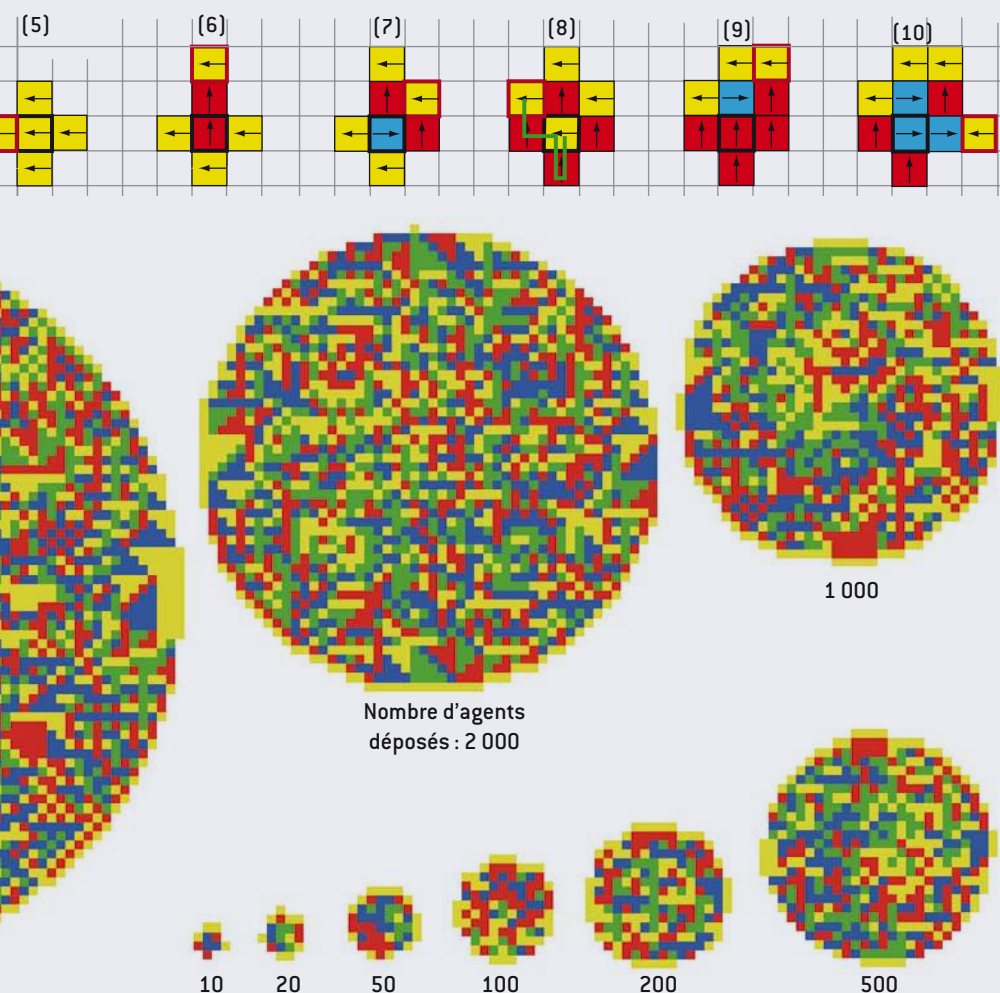
où arrive un agent est occupée (par un agent et le rotor qu'il a déposé), l'agent fait tourner le rotor et suit la direction alors indiquée. Le premier agent déposé s'arrête donc sur la case centrale, le second sur la case au nord de la case centrale, etc. (voir l'encadré ci-dessous). Petit à petit, des cases du plan sont occupées.

D'un étonnement à l'autre

Ce qu'on voit alors se dérouler suscite de nombreuses surprises (on trouve des programmes pour observer ce processus sur www.cs.uml.edu/~ffjpropp/rotor-router-model/#01 ou <http://rotor-router.mpi-inf.mpg.de/applet/>).

Premier étonnement : jamais un agent ne tourne en rond indéfiniment, tous finissent par trouver une case vide où s'arrêter. La propriété se démontre en effectuant un raisonnement par l'absurde. Supposons qu'un agent se mette à repasser toujours par le même chemin et parcoure donc indéfiniment et périodiquement le même ensemble de cases. L'ensemble des cases de ce circuit est fini et a donc un bord (des cases du circuit adjacentes à des cases n'appartenant pas au circuit). En passant plusieurs fois sur une telle case, l'agent finit par faire pointer le rotor de la

Le rotor-router sur le plan



On pose successivement des agents mobiles sur la case centrale (entourée d'un cadre noir) d'un quadrillage infini du plan. Quand un agent arrive sur une case vide, il y dépose un rotor (une flèche

tournante) en l'orientant vers l'ouest et reste sur la case. Ainsi, le premier agent déposé occupe la case centrale (1). Quand la case sur laquelle arrive un agent est occupée, il fait tourner le rotor de 90° dans le sens horaire et suit la direction alors indiquée en se déplaçant d'une case, où le processus recommence (la dernière cellule créée est encadrée de rouge). Le jaune indique que le rotor pointe vers l'ouest, le rouge vers le nord, le bleu vers l'est, le vert vers le sud. Ainsi, l'agent 2 trouve une case centrale occupée, fait tourner son rotor de 90° qui pointe alors vers le nord et y dépose un rotor jaune (2). L'agent 3 trouve la case centrale occupée par un rotor rouge, le fait tourner pour qu'il devienne bleu et se déplace vers l'est, où il dépose un rotor jaune (3). Etc.

Avec ces règles, l'agent revient parfois sur ses pas et fait tourner un rotor plusieurs fois (trajet vert de l'étape (8)). La zone occupée par des agents arrêtés s'étend, mais est de plus en plus arrondie.