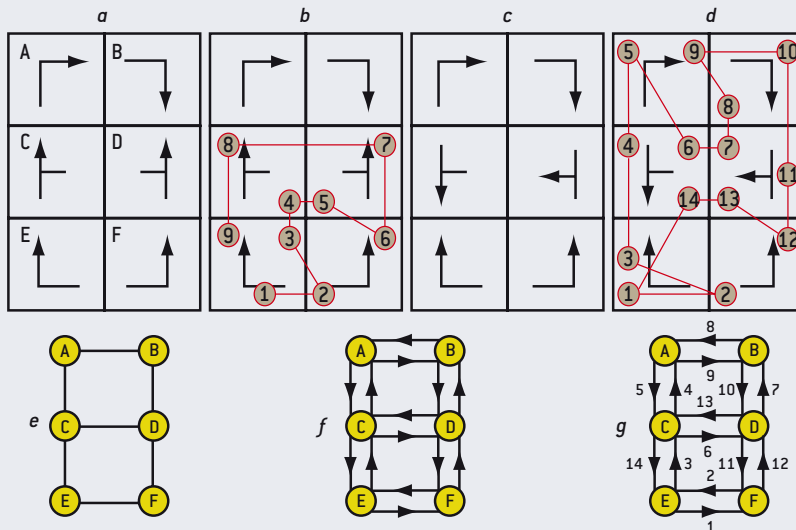


## Le miracle des circuits eulériens



**U**n circuit eulérien sur un graphe orienté (où chaque arête est dotée d'un sens de parcours) est un chemin qui revient à son point de départ en ayant parcouru chaque arête du graphe une fois exactement. Un chemin eulérien passe donc aussi par chaque nœud du graphe.

De façon presque magique, le rotor-router découvre automatiquement et rapidement de tels circuits eulériens. Considérons par exemple six cases d'un quadrillage formant un rectangle  $2 \times 3$  (a). Chaque case est reliée à ses voisins des quatre points cardinaux quand elles existent. Cela détermine un graphe non orienté (e) auquel on associe un graphe orienté (f) en dédoublant chaque arête. D'après le théorème d'Euler, un tel graphe possède des circuits eulériens. Nous allons utiliser un rotor-router pour en découvrir un.

Posons un rotor dans chaque case. La case A est par exemple munie d'un rotor à deux positions,

puisque'elle est reliée à deux autres cases (B et C). Orientons les rotors en prenant la première direction disponible à partir de la position nord en tournant dans le sens horaire. Les six rotors et leurs directions initiales sont dessinés en (a). On pose un agent dans la case E, et on le laisse faire selon la règle de fonctionnement des rotor-routers : l'agent fait tourner le rotor de la case dans lequel il se trouve, et suit la direction qu'il indique alors, et recommence dans la nouvelle case. Le parcours de l'agent suit le chemin indiqué en (b) : EFECDFDCE.

Il revient donc dans la case E sans être passé partout. C'est la phase d'ini-

tialisation. Les rotors ont changé de direction et sont maintenant dans les orientations indiquées par le dessin (c). L'agent reprend son cheminement et, cette fois en 14 étapes, revient en E après avoir parcouru un chemin eulérien. À l'issue de ce parcours, il a remis tous les rotors comme ils l'étaient en (c) : il reprendra donc indéfiniment le chemin eulérien découvert (représenté en (d) et en (g)).

Ce qui se produit ici est général. Quel que soit le graphe non orienté initial, un agent sans mémoire, grâce au système des rotors, finit toujours par découvrir un circuit eulérien du graphe orienté associé, qu'il parcourt alors indéfiniment. Cette méthode permet à un amnésique de sortir d'un labyrinthe quelle que soit sa complexité, pourvu qu'il dispose d'une craie pour dessiner l'équivalent des rotors à chaque carrefour.

plus classique de diffusion centrale nommé IDLA (*Internal Diffusion-Limited Aggregation*, ou agrégation interne limitée par la diffusion) introduit en 1992. Les chemins aléatoires peuvent être conçus de plusieurs façons, mais la plus simple consiste pour chaque pas fait par l'agent à lancer un dé à quatre faces et, selon le résultat, à faire se déplacer l'agent d'une case vers le nord, le sud, l'est ou l'ouest.

James Propp présente le rotor-router comme une « dérandomisation » du modèle classique de diffusion centrale, c'est-à-dire comme un modèle aussi proche que possible du modèle IDLA, mais ne dépendant plus du hasard. Bien que déterministe, le modèle du rotor-router a de nombreuses propriétés communes avec le modèle aléatoire, et sous une forme plus nette. Par exemple, dans le modèle aléatoire IDLA, on observe temporairement des trous dans la forme circulaire construite par les agents arrêtés. Ce n'est jamais le cas avec le rotor-router. Autre exemple : la frontière de l'ensemble des cases occupées, dont on démontre aussi qu'elle devient de plus en plus ronde dans le modèle aléatoire, est bien plus lisse avec le rotor-router.

## Patrouilles sur un graphe

Le mécanisme du rotor-router a une capacité étonnante à parcourir efficacement un graphe quelconque. On se donne un graphe  $G$  non orienté (on ne précise aucun sens de parcours pour les arêtes) à nombre fini de nœuds et d'arêtes. Chaque nœud est muni d'un rotor. On suppose que le graphe est connexe : quels que soient les nœuds A et B du graphe, il existe un chemin allant de A à B. En un nœud quelconque, on pose un agent unique qui se met à suivre les indications des rotors sans jamais s'arrêter, en les faisant tourner à chaque fois qu'il passe. Ce qui se produit est là encore tout à fait remarquable.

Quels que soient le graphe, la position initiale des rotors et l'endroit d'où part l'agent, il visitera chaque nœud une infinité de fois sans en oublier aucun. Pour le démontrer, on note qu'il existe nécessairement un nœud  $N$  du graphe qui est

visité une infinité de fois. On raisonne alors comme pour notre « premier étonnement » à propos du rotor-router sur le plan : les nœuds voisins de  $N$  voient aussi passer l'agent une infinité de fois, de même que les voisins des voisins de  $N$ , etc.

Plus étonnant et même franchement miraculeux : après une période d'errance, l'agent entre dans un parcours cyclique qui est un parcours eulérien du graphe orienté  $G'$  associé à  $G$ . Précisons ici ce que sont les parcours eulériens de  $G'$ . Le graphe que nous avons considéré a des arêtes non orientées. Si l'on remplace chacune des arêtes de  $G$  par deux arêtes orientées, une pour chaque sens de parcours, cela donne un graphe  $G'$  orienté associé à  $G$ . Un parcours eulérien du graphe  $G'$  est alors un chemin qui emprunte chaque arête orientée une fois exactement. Ce chemin empruntera donc chaque arête du graphe  $G$  deux fois exactement.

## Un parcours eulérien pour sortir de tout labyrinthe

Le théorème d'Euler indique que si un graphe orienté connexe est tel que chaque nœud a autant d'arcs arrivant sur lui que d'arcs qui en sortent, alors il existe nécessairement un parcours eulérien du graphe. Ici, le graphe  $G'$  a la propriété demandée (car chaque arc non orienté de  $G$  a donné deux arcs orientés opposés dans  $G'$ ). Il n'y a donc pas de surprise à ce qu'un tel parcours eulérien de  $G'$  existe. En revanche, que l'agent posé n'importe où en trouve un tout seul, simplement parce qu'il utilise les rotors du graphe, est inattendu. L'agent n'a aucune mémoire propre, et pourtant il réalise une tâche qui semble exiger de se souvenir de ce que l'on fait et d'avoir une vision globale du graphe (*voir l'encadré page précédente*) !

Ce résultat fournit à un agent sans aucune mémoire une méthode sûre pour parcourir entièrement un graphe. Concrètement, cela vous indique un procédé pour sortir d'un labyrinthe, aussi compliqué soit-il... si vous disposez d'une craie. En effet :

— À chaque fois que vous passez dans un carrefour du labyrinthe, vous dessinez au centre du carrefour une flèche que vous faites tourner à chaque passage (vous effacez la précédente, et vous dessinez une nouvelle flèche pointant vers la branche suivante du carrefour dont, lors du premier passage, vous avez numéroté les branches). Cette flèche joue le rôle d'un rotor. Même si vous êtes totalement amnésique, vous serez assuré de parcourir tout le labyrinthe et donc d'en trouver la sortie... ou le trésor.

Terminons en mentionnant quelques jolis résultats obtenus sur le problème de la patrouille sur un graphe infini connexe.

Supposons un graphe (non orienté) infini et connexe (par exemple celui associé au quadrillage infini du plan). On a fixé des rotors en chaque nœud (ou case) et déposé un agent en un nœud  $N$ . L'agent se déplace indéfiniment sur le graphe en faisant tourner les rotors. Deux cas seulement sont possibles : soit chaque nœud est visité une infinité de fois (cas dit récurrent), soit aucun nœud n'est visité une infinité de fois (cas transitoire).

La démonstration s'appuie à nouveau sur l'idée que si un nœud est visité une infinité de fois, il en est de même de ses voisins, donc des voisins de ses voisins, etc. Plus inattendus sont les résultats suivants :

— Si un graphe infini connexe est donné ainsi qu'une configuration des rotors, et qu'en déposant un agent en un nœud  $N$ , on se trouve dans le cas récurrent, alors en déposant l'agent n'importe où, on se trouvera aussi dans le cas récurrent.

— Quand on considère le graphe du quadrillage (chaque case est reliée à ses quatre voisines nord, sud, est et ouest) ou sa généralisation en dimension  $k$  ( $k > 2$ ), il existe des configurations initiales des rotors qui produisent le cas récurrent.

— Pour tout graphe infini connexe, il existe une façon de fixer les positions initiales des rotors de manière à avoir une configuration récurrente.

Le simplissime mécanisme du rotor-router n'a pas encore livré tous ses secrets. Il occupera encore un bon moment mathématiciens et informaticiens. ■

## LES AUTEURS



Jean-Paul DELAHAYE et Philippe MATHIEU sont professeurs à l'université de Lille et chercheurs au laboratoire CRISTAL (Centre de recherche en informatique, signal et automatique de Lille).

## BIBLIOGRAPHIE

A. Kosowski et D. Pajak, *Does adding more agents make a difference ? A case study of cover time for the rotor-router*, *Automata, Languages and Programming*, LNCS 8573, Springer, pp. 544-555, 2014.

D. Dereniowski *et al.*, *Bounds on the cover time of parallel rotor walks*, *STACS 2014*, pp. 263-275, 2014.

L. Florescu *et al.*, *The range of a rotor walk*, prépublication arXiv:1408.5533, 2014.

O. Angel et A. Holroyd, *Rotor walks on general trees*, *SIAM Journal on Discrete Mathematics*, vol. 25(1), pp. 423-446, 2011.

Références supplémentaires sur [www.pourlascience.fr](http://www.pourlascience.fr)



Retrouvez la rubrique Logique & calcul sur [www.pourlascience.fr](http://www.pourlascience.fr)