

Le problème du voyageur de commerce

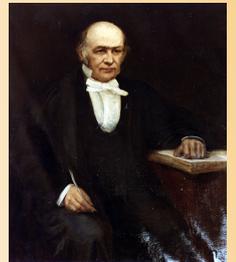
Étant donnée une liste de villes, quel est le plus court circuit passant une seule fois par chacune et revenant au point de départ ?



Problème de graphe

Les villes sont les sommets, les trajets entre 2 villes sont les arêtes, pondérées par la distance ou le temps de parcours. On cherche un circuit qui passe une seule fois par chaque sommet : un **cycle hamiltonien**.

William Rowan Hamilton (1805-1865), mathématicien, physicien et astronome irlandais.



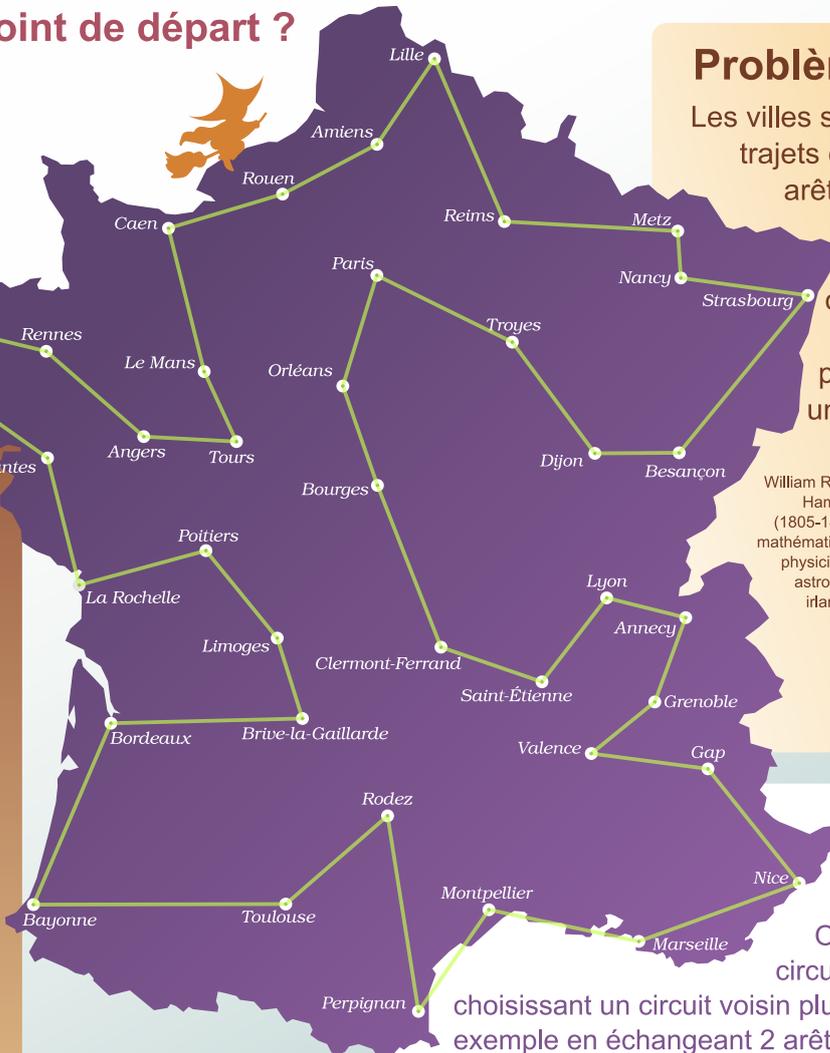
Explosion combinatoire

Le nombre de circuits différents pour parcourir n villes est :

$$\frac{(n-1)!}{2} = \frac{1 \times 2 \times \dots \times (n-1)}{2}$$

Si un ordinateur étudie 1 million de circuits par seconde, il lui faut 0,18 s pour résoudre le problème à 10 villes, 1929 ans pour 20 villes, et des milliards de fois l'âge de la terre pour 30 villes ! Il est donc impossible de passer en revue tous les circuits possibles, et les algorithmes existants ne donnent que des solutions approchées.

Parcours d'un bus de ramassage scolaire, circuit de distribution du courrier..., les applications en logistique sont naturelles. Le problème est aussi utilisé en biologie (séquençage du génome), ou dans l'industrie (construction de circuits imprimés sur des cartes électroniques).



Algorithme de descente

On améliore un circuit de départ en choisissant un circuit voisin plus court, obtenu par exemple en échangeant 2 arêtes dans le parcours, et on recommence tant que c'est possible. L'algorithme reste bloqué en un circuit de longueur « localement » minimale, qui risque d'être bien supérieure à la longueur du circuit optimal.

Recuit simulé

Cet algorithme évite le piège des minima locaux en autorisant l'exploration de solutions moins bonnes avec une certaine probabilité. Celle-ci dépend d'un paramètre, la **température**, que l'on fait décroître progressivement.

Le terme « recuit simulé » est inspiré du processus de réchauffement/refroidissement d'un métal pour améliorer ses propriétés.

