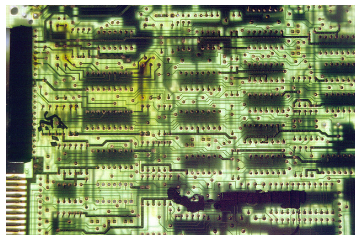


Le plus court chemin

Voici un problème à plusieurs millions ! C'est très sérieux : résolvez ce problème, et vous serez riches et célèbres. Mais malheureusement, c'est un moyen plutôt compliqué de devenir riche.

Imaginez que vous fabriquiez des circuits électroniques, comme des téléviseurs. Pour souder les composants sur le circuit, il faut percer des trous dans votre plaque de PCB. Pour cela, un robot doté d'une perceuse va d'un emplacement à l'autre pour faire les trous.



On sent bien que si on s'y prend mal, la tête fait trop de déplacements et perd du temps, donc de l'argent. En revanche, on voit beaucoup moins bien comment trouver le meilleur chemin possible. . .

Matériel

- Une planche plantée de clous, disposés au hasard,
- une ficelle suffisamment longue (et pas trop élastique),
- un marqueur (pour marquer sur la ficelle la longueur du chemin)

Le but du jeu est de trouver le plus court chemin passant une fois et une seule par chaque clou et terminant sur le clou de départ (la perceuse doit être prête pour la prochaine plaque). On attache une ficelle à un clou et on fait passer la ficelle de clou en clou, en bouclant.

Parviendrez-vous à trouver le plus court chemin ?

Recherche de la solution optimale

Ce problème a de nombreuses applications dans la vie de tous les jours : minimiser la tournée du facteur, la longueur des pistes d'un circuit imprimé, les déplacements d'un bras robotique... C'est un problème très étudié, plus connu sous le nom de « **problème du voyageur de commerce** ».

Le problème du voyageur de commerce appartient à une catégorie de problèmes très difficiles à résoudre. Pour ces problèmes, on ne dispose pas d'algorithmes assez performants pour le résoudre quand n est élevé. Le problème du voyageur de commerce ayant fait l'objet de nombreuses études, beaucoup d'algorithmes ont été proposés pour le résoudre le plus efficacement possible.

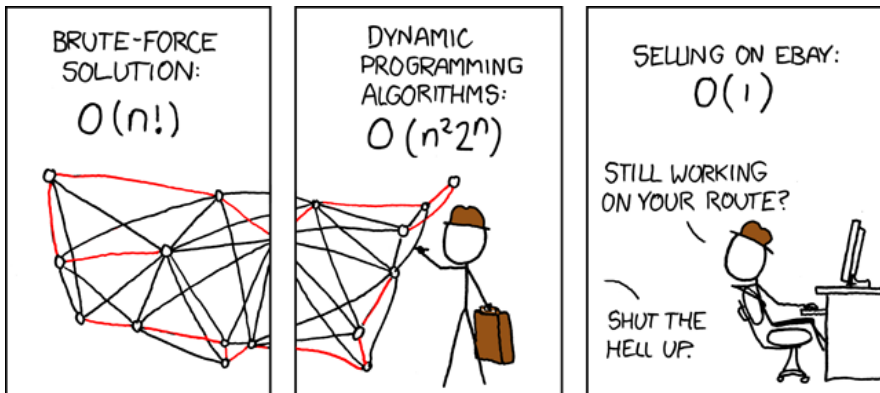
Comparaison de deux algorithmes

L'approche naïve consiste à calculer la longueur de tous les chemins possibles, et comparer les résultats pour ne retenir que le plus court. Pour n villes, le nombre de chemins possible est $n!$. La performance de l'algorithme est donc $O(n!)$.

Il existe cependant des algorithmes plus efficaces : par exemple, la performance de l'algorithme Held-Karp est $O(n^2 2^n)$. Pour illustrer la différence, comparons l'augmentation des calculs nécessaires à mesure que n augmente :

Nombre de sommets	5	10	15	20
Méthode naïve $O(n!)$	120	3628800	1.3×10^{12}	2.4×10^{18}
Held-Karp $O(n^2 2^n)$	800	102400	7.3×10^6	4.1×10^8

Ce tableau indique qu'en testant un milliard de chemins par seconde, il faudrait à la méthode naïve plus de **77 ans** pour trouver le chemin le plus court entre 20 sommets ! Dans les mêmes conditions, l'algorithme Held-Karp met moins d'**une demi-seconde** pour trouver le même résultat.

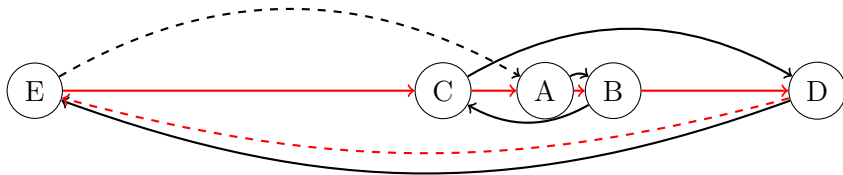


Recherche d'une solution approchée

Pour ces problèmes complexes, on préfère souvent trouver une solution raisonnablement bonne (solution approchée) très rapidement, plutôt que de chercher très longtemps la solution optimale. Une **heuristique** est une méthode pour fouiller intelligemment l'espace des solutions possibles à la recherche des bonnes solutions. La recherche d'heuristiques efficaces fait partie du travail des chercheurs en informatique.

Heuristique : le plus proche voisin

Partez d'une ville, et allez vers la ville la plus proche par laquelle vous n'êtes pas encore passé. Recommencez, jusqu'à passer par toutes les villes, et revenez au point de départ. Cette heuristique aura tendance à prendre de préférence les distances les plus courtes... Mais elle n'aboutira pas forcément à la meilleure solution. Par exemple :



Si on démarre de \textcircled{A} , aller toujours vers le voisin le plus proche nous fait faire un trajet (\longrightarrow) beaucoup plus long que le trajet optimal (\longrightarrow).

Heuristiques et méta-heuristiques

Une heuristique est spécifique au problème qu'elle traite : elle exploite certaines propriétés du problème pour orienter la recherche vers des « régions » susceptibles de contenir des bonnes solutions. Cependant, certaines heuristiques reposent sur des concepts assez génériques pour s'appliquer à de nombreux problèmes, moyennant quelques adaptations. On parle alors de **méta-heuristiques**. Les méta-heuristiques sont souvent inspirées de la nature. En voici quelques exemples.

Le recuit-simulé s'inspire d'un processus utilisé en métallurgie pour minimiser l'énergie d'un matériau.

Les colonies de fourmis s'inspirent du comportement des insectes sociaux : avez-vous remarqué que les fourmis finissent toujours par trouver le chemin le plus court entre la fourmilière et la source de nourriture ?

Les algorithmes génétiques reproduisent les mécanismes de l'évolution dans le vivant :

- une population de solutions aléatoires est créée ;
- on soumet cette population à une sélection naturelle (les meilleures solutions sont les plus adaptées) ;
- on crée de nouvelles solutions à partir des solutions existantes par croisements et mutations ;

Génération après génération, on observe une amélioration des solutions.

Le coin de l'animateur

L'objectif de cette activité est d'introduire la notion de complexité des problèmes.

- Commencez par donner un exemple de chemin, en faisant volontairement des détours. Laissez ensuite les participants trouver de meilleures solutions.
- À mesure qu'ils avancent, il sera de plus en plus difficile d'améliorer les résultats. Insistez sur le fait que lorsqu'ils trouvent une meilleure solution, ils ne peuvent pas être sûrs qu'il n'en existe pas une meilleure.
- Pour qu'ils se fassent une idée de la complexité du problème, vous pouvez demander aux participants de calculer le nombre de chemins possibles ($n!$), et les amener à refaire l'estimation du temps de calcul nécessaire à raison d'un milliard de chemins testés par seconde.