par Jean-Luc Stehlé

L'algorithme de Dijkstra

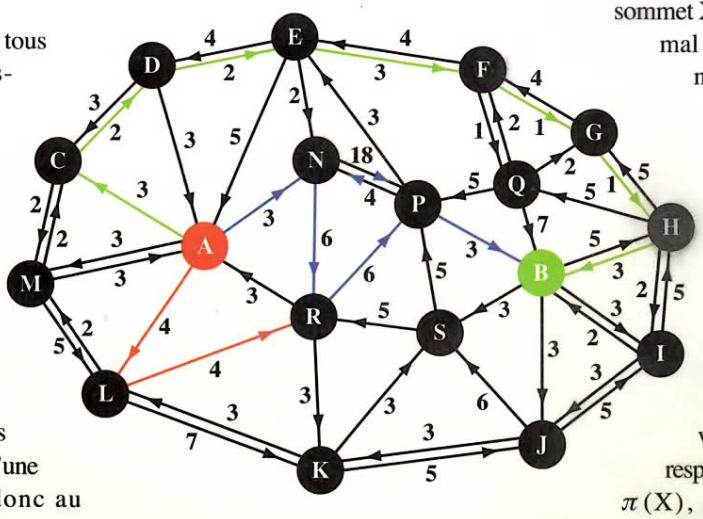
Savoir quelles sont les routes embouteillées, c'est bien, mais encore faut-il déterminer rapidement le chemin à suivre pour aller le plus vite possible d'un point à un autre. L'algorithme de Dijkstra est celui qui, à l'heure actuelle, est le plus performant pour résoudre ce problème.

Un problème important est de savoir comment rejoindre le plus vite possible un point B partant d'un point A lorsque le système de navigation d'un conducteur automobile lui fournit les temps moyens de parcours de chacun des axes possibles en fonction de la densité du trafic. En voici un exemple :

Si on veut essayer tous les chemins possibles de A vers B (en excluant les cycles), on arrive vite à un nombre colossal.

En moyenne, trois flèches partent de chaque point : en partant de A, les trajets composés d'une arête seraient donc au nombre de 3, de deux arêtes, au nombre de 3^2, et ainsi de suite. On trouve donc 3^n trajets de n arêtes. En se limitant à sept arêtes, on trouve déjà plus de deux mille trajets.

Pour des trajets sur des graphes comportant des centaines d'arêtes, avec un ordinateur examinant un milliard de trajets à la seconde, il faudrait plus de 10^{30} années pour venir à bout du problème. Le monde est loin d'être aussi vieux !



L'algorithme de Dijkstra permet d'éviter l'explosion combinatoire.

Trouver sa route rapidement

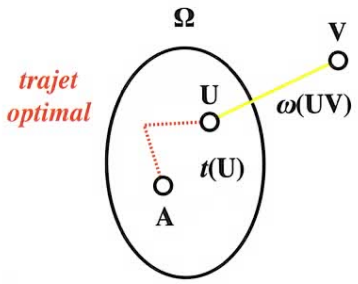
Comment faire ? Une solution est apportée pa l'algorithme de Dijkstra. L'observation clé est l suivante : si un chemin optimal de A vers B pass par un sommet X, alors la partie de ce chemi allant de A à X est un itinéraire optimal de A ver X. En effet, s'il existait un chemin plus rapide d A vers X, il permettrait de réduire le trajet de A ver B. On associe alors, à chaque sommet X (autr que A), son *prédécesseur* $p(X)$, c'est-à-dire le som met immédiatement précédent dans le chemin opt mal menant de A à X. Dans l'exemple ci-dessu on a $p(B) = H$, $p(H) = G$, et ainsi de suite jusqu' $p(C) = A$. La fonction p permet de reconstruir de proche en proche, pour chaqu sommet X, un chemin opt mal de A vers X. O note $t(X)$ le temp de parcours co respondant.

Au départ, o ne dispos pas de valeurs déf nitives d $t(X)$ et d $p(X)$: on s contente donc d valeurs provisoire respectivement $\tau(X)$ e $\pi(X)$, données par l meilleur itinéraire de A vers X actuellement connu. Ces valeurs vont évo luer, par améliorations successives, vers les valeur définitives.

Appelons Ω l'ensemble des sommets du graph pour lesquels on sait que les fonctions t et p ont atteir leurs valeurs définitives.

L'algorithme de Dijkstra consiste à faire grossi de proche en proche Ω en lui adjoignant, à chaqu itération, un nouveau sommet. Voici comment. A démarrage, on pose $\Omega = \{A\}$ avec $\tau(A) = t(A) = 0$ (le trajet optimal de A vers A pren un temps nul, par un chemin formé de zéro arête)

Pour les sommets X différents de A, on ne connaît pas de chemin et on initialise donc $\tau(X)$ à $+\infty$ en convenant que, tant que $\tau(X)$ n'aura pas de valeur finie, $\pi(X)$ ne sera pas défini.



Une itération se fait en deux étapes. D'abord, on sélectionne toutes les arêtes UV dont l'origine U est dans Ω et dont l'extrémité V est extérieure à Ω. Pour chacune d'elles, on connaît un trajet optimal de A vers U (car U est dans Ω). En prolongeant ce trajet jusqu'à V, on a un temps total de parcours égal à $t(U) + \omega(UV)$, où $\omega(UV)$ est le « poids » de l'arête UV (proportionnel au temps de parcours de U à V).
Si cette somme est inférieure à $\tau(V)$, ce trajet pourrait être un chemin optimal vers V, et on le retient en posant $\tau(V) = t(U) + \omega(UV)$ et $\pi(V) = U$.
Ces valeurs restent toujours relatives au meilleur itinéraire actuellement connu (notons que V peut être touché par plusieurs arêtes).

Bourgeonnements successifs

Dans une seconde étape, on choisit, parmi les sommets X extérieurs à Ω et tels que $\tau(X) < +\infty$ (et où $\pi(X)$ est donc défini), un X qui minimise $\tau(X)$. Le trajet optimal de A vers $\pi(X)$ (qui est dans Ω) suivi de l'arête joignant $\pi(X)$ à X est alors un trajet optimal vers X. En effet, s'il existait un trajet plus rapide, en notant Y le prédécesseur de X dans ce dernier trajet, Y devrait être dans Ω (car Y a dû, lors d'une étape précédente, être inclus dans Ω), ce qui contredit la définition de X.
Le sommet X ainsi trouvé est ensuite rajouté à l'ensemble Ω, et on pose $p(X) = \pi(X)$ et $t(X) = \tau(X)$, ce qui termine cette itération : on recommence alors avec le nouvel ensemble Ω « grossi » qui, de proche en proche, finira ainsi par contenir le point B qui nous intéresse (sauf si le schéma de circulation rend inaccessible le point B partant de A), marquant la fin de l'exécution de l'algorithme.

L'algorithme de Dijkstra fournit l'itinéraire optimal de A vers B ainsi que les itinéraires optimaux pour tous les points qui, partant de A, sont plus rapides à atteindre que B. Le temps de calcul, proportionnel au carré du nombre d'éléments de Ω, reste raisonnable, même pour des réseaux routiers à l'échelle d'un pays.

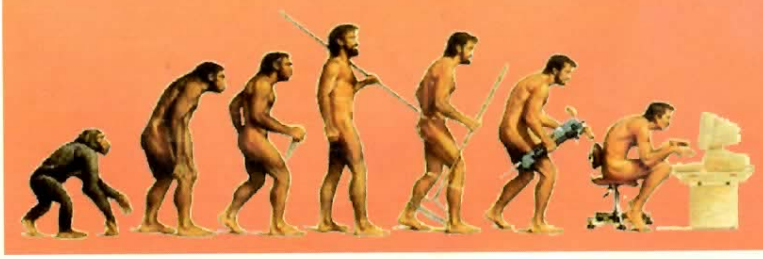
Exemple d'exécution de l'algorithme de Dijkstra

Dans l'exemple de la page précédente, on initialise l'algorithme avec $\Omega = \{A\}$, $t(A) = 0$ et $\tau(X) = +\infty$ pour tout sommet X autre que A.
Première itération
On considère les arêtes ayant leur origine en A : AC, AL, AM et AN, ce qui nous amène à poser :
$\tau(C) = 3, \tau(L) = 4, \tau(M) = 3, \tau(N) = 3,$
$\pi(C) = \pi(L) = \pi(M) = \pi(N) = A.$
Parmi ces quatre sommets (C, L, M et N), il faut en trouver un minimisant le temps de parcours : C par exemple.
On pose donc $t(C) = 3$ et $p(C) = A$.
Continuons l'algorithme avec $\Omega = \{A, C\}$, $t(A) = 0, t(C) = 3,$
$p(C) = A, \tau(L) = 4, \tau(M) = 3, \tau(N) = 3$ et les autres à $+\infty$,
$\pi(L) = \pi(M) = \pi(N) = A.$

Seconde itération
On considère les arêtes ayant leur origine en A ou C, c'est-à-dire AL, AM, AN, CD et CM.
La première étape conduit à poser $\tau(D) = 6, \pi(D) = C$.
Choisissons M. On pose donc $t(M) = 3, p(M) = A$.
Poursuivons donc l'algorithme avec $\Omega = \{A, C, M\}$,
$t(A) = 0, t(C) = 3, t(M) = 3,$
$\tau(D) = 6, \tau(L) = 4, \tau(N) = 3$ et les autres à $+\infty$,
$\pi(L) = \pi(N) = A$ et $\pi(D) = C$.

Troisième itération
On considère les arêtes AL, AN, CD et ML. Les deux étapes conduisent à $\Omega = \{A, C, M, N\}$, $t(N) = 3$ et $p(N) = A$.

Quatrième itération
Les arêtes à considérer sont AL, CD, ML, NP et NR. Le point suivant à être englobé dans Ω est L avec $t(L) = 4$ et $p(L) = A$.



Pour un réseau routier réel, avec des temps de trajets à peu près proportionnels aux distances, la zone Ω analysée ressemble très grossièrement à un disque centré en A et de rayon la distance AB. Elle regroupe un nombre de sommets proportionnel au carré de cette distance et, donc, les temps de calculs croissent donc à peu près comme la puissance quatrième de AB.

J.-L S.