TD-TP : QUELQUES VARIANTES SUR LES PLUS COURTS CHEMINS

Dominique Michelucci, Université de Dijon

1 TP : PLUS COURTS CHEMINS AVEC DES REPRESENTATIONS MATRICIELLES

Pour générer le graphe, vous pouvez tirer au hasard des coordonnées de points (entre 10 et 500, pour simplifier l'affichage). Vous pouvez refuser de générer des points dans certaines zones, pour simuler des obstacles. La longueur des arcs sera la distance (arrondie sur l'entier le plus proche si vous le souhaitez) entre les 2 sommets, s'ils sont suffisamment proches.

Programmer tous les algorithmes de plus courts chemins vus en cours : Ford, Floyd, Dantzig, Dijkstra, pseudo-multiplication de matrice, en représentant les graphes par des matrices carrées. Pour Dijkstra, vous utiliserez un tableau de booléens pour savoir si un sommet a été "atteint" (la distance à la source d'un sommet atteint ne peut plus être modifiée par l'algorithme) ou non.

Ces algorithmes sont décrits sur :

`http://ufrsciencestech.u-bourgogne.fr/master1/mi1-tc5/COURS_ALGO_HTML/cours_algo017.html`

Gérez aussi un tableau (ou une matrice) de prédécesseurs. Ecrivez une procédure pour reconstituer le plus court chemin jusqu'à un sommet donnè. Vous pouvez stocker le chemin dans un tableau.

Dessinez les arêtes du graphe en noir, puis l'arbre des plus courts chemins en rouge (autrement dit les arcs joignant un sommet à son prédécesseur).

2 TP : DIJKSTRA AVEC UNE REPRESENTATION PAR LISTE DE VOISINS

Le même graphe est ce coup ci représenté par un tableau de liste des arcs : $T[s]$ est une liste d'arcs sortant de s ; un arc contient les champs : t un sommet

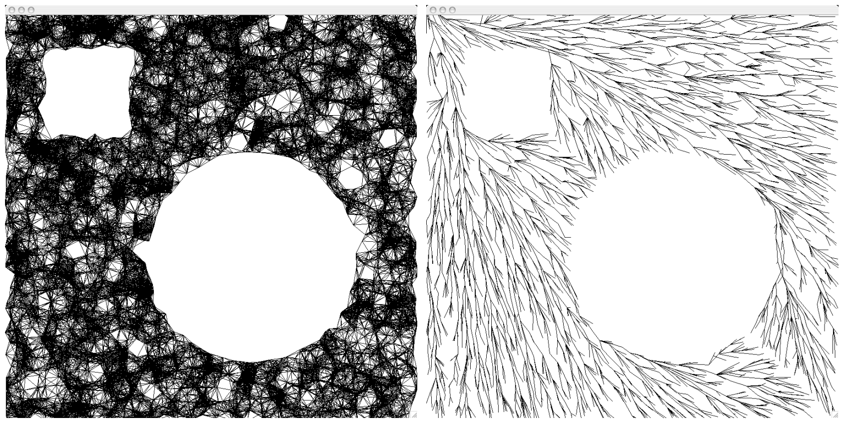

FIG. 1 – A gauche, le graphe. A droite, l'arbre des plus courts chemins, issus du point en haut à gauche (masqué par la bannière de la fenêtre...).

voisin de s, et d la longueur de l'arc $s \to t$ (vous pouvez aussi recalculer à chaque fois la distance entre les 2 sommets ; ainsi un arc se réduit au numéro du sommet t et vous pouvez réutiliser les listes d'entiers...).

Le programme Dijkstra est fourni ; mais il utilise une liste et non un tas pour gérer les sommets non atteints. Le programme Heapsort est fourni ; il gère un tas, représenté par un tableau. Il vous faut modifier le programme Heapsort pour qu'un sommet (donné par son indice s) "connaisse" sa position dans le tas (indication : utilisez un tableau position[s] qui donne l'indice de la case du tableau/tas qui contient le sommet numéro s ; si s n'est plus dans le tas, utiliser l'indice -1. La procédure swap, ou échange, doit mettre à jour ce tableau position.)

3 ECARTEMENT et METHODE PAC

Voici une méthode probabiliste ("probablement approximativement correcte", "Probably Approximately Correct", PAC) pour trouver une approximation du chemin le plus court entre 2 points A et B quand le graphe est symétrique ($M = M^t$). Echantillonner l'espace libre, comme dans l'exercice précédent ; calculer l'arbre des plus courts chemins issus de A, et celui des plus courts chemins issus de B. L'écartement d'un sommet S est, par définition, $AS + SB - AB$ (où AS, SB, AB sont les longueurs des plus courts chemins de A à S, de S à B, de A à B). Si S est sur le plus court chemin de A vers B, alors son écartement est nul. Pour préciser le plus court chemin AB, il faut re-échantillonner les zones peu écartées du plus court chemin AB, où, par exemple, l'écartement est 10% de la longueur du chemin AB. Cette méthode est utilisée en robotique pour la planification des trajectoires.

4 QUEL EST LE SOMMET SOURCE ?

Un graphe est donné, par sa matrice des distances (non négatives) entre sommets. Un sommet source, inconnu, de ce graphe émet un signal au temps 0. Le signal se propage à la vitesse 1 dans le graphe, en utilisant toujours les plus courts chemins. En quelques sommets, un capteur mesure la date d'arrivée du signal. Proposez une méthode pour déterminer quel est le sommet source. Par exemple, si un capteur est par chance sur le sommet source, il indique la date d'arrivée 0, et donne donc la source.

Même question, mais la date d'émission du signal est inconnue ; seules les dates de réception sont connues.

Un algorithme similaire a été proposé pour déterminer les sources des rumeurs sur internet...

5 CHAÎNES DE MARKOV

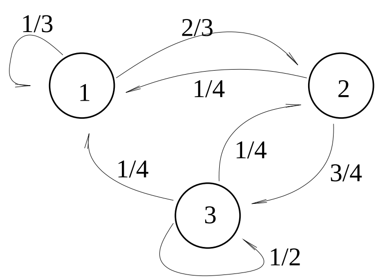

Fig. 2 – Au premier temps d'internet, il y avait 3 pages, avec les hyperliens ci dessus. Soit (p_1, p_2, p_3), avec $p_1 + p_2 + p_3 = 1$ la distribution de probabilité d'un internaute. Après un clic, quelle est la distribution de probabilité ?

5.1 Chemin le plus probable

Les arcs d'un graphe sont étiquetés par des probabilités ; pour chaque sommet, la somme des probabilités des arcs sortants du sommet vaut 1. Ces graphes sont appelés chaîne de Markov et très utilisés en mathématiques et en informatique (par exemple, pour la reconnaissance automatique de la parole ou des textes, ou comme représentation des liens entre pages sur Internet). Quelle est la probabilité d'un chemin ? Comment calculez le chemin les plus probable entre 2 sommets ?

Si vous voulez généraliser la méthode par pseudo produit de matrices, quel pseudo produit devez vous utiliser pour calculer le chemin le plus probable ?

5.2 Distribution stationnaire

Les chaînes de Markov ergodiques (irréductibles, non périodiques) ont une distribution stationnaire. Calculez celle du graphe de la Fig. 2.

Indication : un clic change la distribution de probabilité (p_1, p_2, p_3) en

$$(p'_1, p'_2, p'_3) = (p_1, p_2, p_3)M, \quad M = \begin{pmatrix} 1/3 & 2/3 & 0 \\ 1/4 & 0 & 3/4 \\ 1/4 & 1/4 & 1/2 \end{pmatrix}$$

Pour trouver la distribution stationnaire, résolvez $X = XM$ (indication : $(15, 16, 24) = (15, 16, 24)M$). Informatiquement, on peut calculer M^∞ ou bien M^{2^k} avec k "grand" : M^{2^k} converge assez vite en général vers une matrice dont toutes les lignes sont égales à la distribution stationnaire.

Note : au tout début, Google utilisait la probabilité stationnaire pour mesurer la pertinence des pages internet... Si $P = PM$, P_i est la probabilité qu'un internaute aléatoire se trouve sur la page i : plus P_i est grand (intuitivement : plus la page est référencée, par des pages elles mêmes bien référencée), et plus la page est pertinente. Ce critère est manipulable par les plaisantins.

Exemple linguistique : la probabilité de "mushroom soup" est plus grande que celle de "much rooms hope". En français, considérez : (sert, sers, cerf, serre) (toi, ta, toit...) (un, hun, hein) (verre, vert, vers...)

Loi de Benford : en base 10, les nombres commençant par le chiffre 1 sont plus probables : empiriquement, ils sont rencontrés plus souvent. Voir : `http://ufrsciencestech.u-bourgogne.fr/master1/mi1-tc5/COURS_ALGO_HTML/cours_algo030.html#toc94` Par exemple dans la suite 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096... il semble bien que cela soit vrai. Les chaînes de Markov peuvent être utilisées pour calculer les probabilités du premier chiffre dans cette suite.

6 CHEMIN LE PLUS SÛR

Les arcs d'un graphe sont étiquetés par une probabilité de mourir quand cet arc est utilisé. Quelle est la probabilité de mourir pour un chemin donné ? (Indication : considérez plutôt la probabilité de survie). Comment calculer le chemin le plus sûr entre 2 sommets ? Si vous voulez généraliser la méthode par pseudo produit de matrices, quel pseudo produit devez vous utiliser pour calculer le chemin le plus sûr ?