

TP3 – Programmation dynamique (1/2)

Problème du maître-nageur

Le problème du maître-nageur, qui consiste à chercher le plus court chemin entre deux points dans un milieu non homogène, a été décrit en cours. Ce problème d'optimisation compliqué peut être résolu par *programmation dynamique*.

Tout algorithme de résolution d'un problème d'optimisation par programmation dynamique peut se ramener à la recherche du plus court chemin dans un graphe. Pour trouver le plus court chemin dans un graphe pondéré (de poids positifs), l'algorithme le plus connu est l'*algorithme de Dijkstra* (1959).

Exercice 1 : familiarisation avec la fonction `dijkstra` de Matlab

Commencez par lire la documentation de la fonction `dijkstra` de Matlab, en tapant `doc dijkstra`, puis écrivez un script de nom `exercice_1` visant à résoudre le problème de plus court chemin vu en cours (graphe à six sommets A, B, C, D, E, F) par l'algorithme de Dijkstra. L'appel à la fonction s'écrit, par exemple :

```
[couts,chemins] = dijkstra(M,P,4)
```

où M est une matrice de booléens de taille 6×6 décrivant les adjacences du graphe, P est une deuxième matrice de taille 6×6 contenant les poids des arêtes du graphe, et 5 est l'indice du sommet E , si les six sommets sont ordonnés par ordre alphabétique. Attention : les matrices M et P doivent être symétriques.

Le script `exercice_1` doit retourner les coûts optimaux menant de E aux six sommets du graphe, ainsi que les six chemins optimaux correspondants. Par exemple, le chemin optimal menant de E à A s'obtient en tapant la commande :

```
>> chemins{1}
```

Exercice 2 : résolution du problème du maître-nageur

Pour résoudre le problème du maître-nageur par l'algorithme de Dijkstra, il est nécessaire de construire un graphe tel que :

- Les sommets sont les pixels.
- Les arêtes correspondent au système de voisinage des « 8 plus proches voisins ».
- Le poids d'une arête est égal à la longueur de cette arête (cette longueur ne peut valoir que 1 ou $\sqrt{2}$) divisée par la vitesse de déplacement du maître-nageur, qui est proportionnelle à la moyenne des niveaux de gris des deux extrémités de l'arête.

Complétez le script `exercice_2` de manière à mener à bien cette résolution sous forme interactive : l'utilisateur est invité à choisir la position du maître-nageur, puis celle du noyé. Le chemin optimal doit alors s'afficher quasi-instantanément.