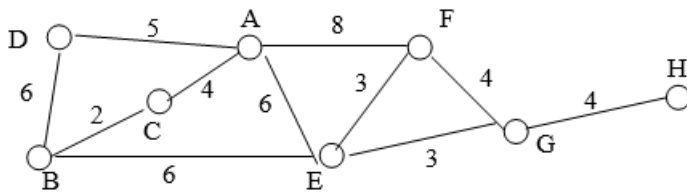


Exercices d'I.A.

Exercice corrigé 1 :

Appliquez l'algorithme de Dijkstra au graphe ci-dessous pour déterminer le plus court chemin entre **D** et **E** (peu importe les autres chemins). Montrez l'évolution des ensembles \mathcal{F} et \mathcal{O} qui désignent respectivement la liste des sommets "fermés" et la liste des sommets "ouverts", le tableau des distances et le tableau des prédécesseurs. Vous arrêterez la simulation de l'algorithme aussitôt que le meilleur chemin entre D et E est trouvé, en précisant quelle est la condition d'arrêt requise.



Réponse :

- | | |
|---------------------------------------|--------------------------------|
| (1) $\mathcal{F} = \{\}$ | $\mathcal{O} = \{D\}$ |
| (2) $\mathcal{F} = \{D\}$ | $\mathcal{O} = \{A, B\}$ |
| (3) $\mathcal{F} = \{D, A\}$ | $\mathcal{O} = \{B, C, E, F\}$ |
| (4) $\mathcal{F} = \{D, A, B\}$ | $\mathcal{O} = \{C, E, F\}$ |
| (5) $\mathcal{F} = \{D, A, B, C\}$ | $\mathcal{O} = \{E, F\}$ |
| (6) $\mathcal{F} = \{D, A, B, C, E\}$ | |

E passe dans les fermés, c'est la condition d'arrêt de l'algorithme, le plus court chemin entre D et E est trouvé, c'est D, A, E.

Evolution du tableau des distances :

	A	B	C	D	E	F	G	H
(1)	∞	∞	∞	0	∞	∞	∞	∞
(2)	5	6	∞	0	∞	∞	∞	∞
(3)	5	6	9	0	11	13	∞	∞
(4)	5	6	8 (!!!)	0	11	13	∞	∞
(5)	5	6	8	0	11	13	∞	∞
(6)	5	6	8	0	11	13	∞	∞

Evolution du tableau des prédécesseurs :

	A	B	C	D	E	F	G	H
Préd. :	D	D	A	-	-	-	-	-
(1) à (3)								
(4)	D	D	B (!)	-	A	A	-	-
(5) et (6)	D	D	B	-	A	A	-	-

Exercice corrigé 2 :

L'algorithme de Dijkstra permet-il de trouver le plus court chemin si on l'applique à un graphe dans lequel on a associé à certains arcs une valeur négative ? Etayez votre propos par une démonstration ou un contre-exemple.

Réponse :

Contrexemple : il suffit de modifier le coût de l'arête FE et de lui donner la valeur négative -10. Dans ce cas, le plus court chemin serait D,A,F,E. Or, l'algorithme de Dijkstra commencerait comme ci-dessus et après que E soit passé dans les fermés, il n'y aurait plus de mise à jour possible du chemin menant à E. Donc Dijkstra ne trouverait pas le chemin D, A, F, E. CQFD, Dijkstra ne fonctionne pas si les coûts sont négatifs.

Exercice corrigé 3 :

Pour déplacer un robot de sa position actuelle à une position finale souhaitée, on construit une grille imaginaire sur laquelle le robot peut se déplacer horizontalement, verticalement ou diagonalement d'une case à la fois et on tente de trouver un chemin sur cette grille. Lorsqu'un obstacle est présent dans la case considérée, même partiellement, la case est considérée comme inaccessible. Pour un déplacement horizontal ou vertical d'une case, on associe un coût de 1 et pour un déplacement en diagonal, on associe un coût de 1,414. Pour déterminer le plus court chemin de la position courante du robot à la position recherchée, on utilise l'A*.

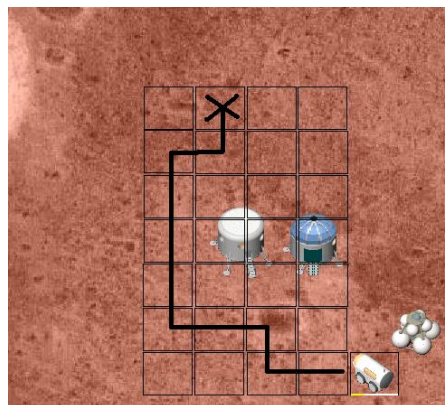


Figure 1 : Illustration du déplacement d'un robot (sans les diagonales pour cet exemple)

Question 1 : Que proposez-vous comme fonction heuristique pour minimiser les temps de calculs ? Pour rappel, une fonction heuristique d'un A* est une fonction qui permet d'estimer, de façon approximative, le chemin restant à parcourir entre une position envisagée et la position terminale recherchée. Plus l'estimation est proche de la véritable distance à parcourir tout en y étant inférieure ou égale, plus la recherche est efficace. La qualité de l'heuristique proposée sera prise en compte dans la notation. Expliquez votre choix.

Question 2 : Sur la figure 1, on montre le déplacement d'un robot sur une grille imaginaire en interdisant les déplacements en diagonal. De plus, la taille des cases est relativement importante, ce qui conduit à des trajectoires non optimales. Pourquoi de tels choix ? A quelle condition peut-on construire un graphe avec des arcs correspondant à des déplacements en diagonal ?

Réponse 1 :

On note (x_f, y_f) la position de la case à atteindre et (x, y) la position de la case qu'on est en train d'examiner, c'est-à-dire celle associée au nœud du graphe qu'on va passer dans les fermés dans l'algorithme de Dijkstra/A*. Appelons ce nœud N et h la fonction heuristique.

* Première possibilité simple, prendre en compte la distance à vol d'oiseau.

Dans ce cas : $h(N) = \text{racine carrée} ((x_f - x)^2 + (y_f - y)^2)$.

* Deuxième possibilité, essayons d'affiner notre estimation, tout en préservant la contrainte d'être toujours inférieur à la véritable distance qu'il reste à parcourir. Dans le cas le plus favorable, il n'y a aucun obstacle à contourner et la distance à parcourir est égale à un déplacement en diagonal + un déplacement horizontal ou vertical, selon le contexte. Distinguons les 2 cas :

On note "abs" la valeur absolue.

si $(\text{abs}(x_f - x) > \text{abs}(y_f - y))$, $h(N) = 1,414 * \text{abs}(y_f - y) + (\text{abs}(x_f - x) - \text{abs}(y_f - y))$

sinon $h(N) = 1,414 * \text{abs}(x_f - x) + (\text{abs}(y_f - y) - \text{abs}(x_f - x))$

Réponse 2 :

La taille des cases correspond à la taille de l'objet à déplacer. Si on prend des cases plus petites, le risque est de trouver un chemin qui conduira l'objet à heurter des obstacles au niveau de ses parties qui dépassent de la case (de même, un humain ne choisit pas de faire passer son centre de gravité à 1 cm d'un mur car il risque de toucher le mur au niveau des jambes, des bras ou des hanches !).

En ce qui concerne les déplacements diagonaux, ce qui peut poser problème, c'est d'avoir la case de départ libre, la case située en diagonale également libre, mais une des 2 cases adjacentes occupée par un obstacle. Lors du passage diagonal, il y a donc un risque de toucher un obstacle. Pour assurer un déplacement diagonal entre une case A et une case B, la contrainte est d'avoir les 2 cases simultanément adjacentes à A et B également libres.

Exercice 4 :

Problème du voyageur de commerce

Un commerçant souhaite visiter toutes les villes d'une région en effectuant le plus court chemin dans un réseau routier dont on connaît les distances entre villes. On suppose qu'une ville est présente à chaque intersection et que le voyageur part obligatoirement d'une ville donnée et finit dans cette même ville.

a) Modélisez la résolution de ce problème en définissant toutes les modalités permettant d'effectuer une recherche efficace dans l'espace d'états, notamment comment est défini un état de l'espace d'états, comment est caractérisé l'état initial, comment est définie la fonction successeur, etc.

Aide : Avoir déjà visité toutes les villes imposées et être dans la ville finale (état terminal) ou n'avoir visité aucune ville imposée et être dans la ville finale sont 2 états différents. Pour les distinguer, la position n'est pas suffisante, il faut caractériser un état par des informations complémentaires.

b) Quelle heuristique proposez-vous pour accélérer la recherche ?

Exercice corrigé 5 :

Voici ci-dessous une énigme classique trouvée dans une revue, dans laquelle il faut déterminer les 5 lettres d'un mot. On peut noter que toutes les lettres de ce mot se trouvent dans les autres mots de la grille.



Question 1 : On souhaite résoudre cette énigme à l'aide d'un programme informatique. A quelle catégorie de problèmes appartient cette énigme ? Justifiez votre réponse en donnant d'autres exemples de problème du même style.

Question 2 : Proposez une modélisation informatique du problème en effectuant une recherche dans l'espace d'états selon une technique classique de résolution générale de problèmes. Répondez notamment aux questions suivantes :

- Quelle structure de données proposez-vous pour modéliser un état du problème ?
- Quel est l'état initial ?
- Comment est caractérisé l'état final ?
- Donnez les grandes lignes de la fonction "successeur" permettant de trouver les états suivants d'un état donné.
- Quelles sont les grandes lignes de la fonction permettant de vérifier qu'un état est consistant par rapport aux contraintes du problème ?
- Au niveau de la fonction successeur, quelle stratégie proposez-vous pour déterminer le choix de la lettre à placer ?

Réponse 1 : Dans la catégorie des problèmes de satisfactions de contraintes, comme Sudoku, 8 reines, ...

Réponse 2 : *Quelle structure de données proposez-vous pour modéliser un état du problème ?*

Un état est typiquement déterminé par des lettres placées dans le mot secret, entre 0 et 5 lettres. structure typique, un tableau de 5 caractères :

char[5] t; // "?" pour les cases non remplies

Le tableau des contraintes pourra être stocké en variable globale initialisée avant la recherche dans l'espace d'état. Structures utilisées : 1 tableau de 5x10 cases de string pour les mots et dans un tableau de 10 cases d'entiers pour les scores.

- *Quel est l'état initial ?* un tableau rempli de 5 "?"
- *Comment est caractérisé l'état final ?* par un tableau rempli de 5 lettres différentes de "?"
- *Donnez les grandes lignes de la fonction "successeur" permettant de trouver les états suivants d'un état donné.*

Pour un état N donné, comprenant k cases non encore déterminées, on procède en 2 étapes :

- a) Choix de la nouvelle case à placer (un indice entre 0 et 4 parmi les k indices encore possibles)
- b) On regarde les lettres possibles pour la case choisie : elle doit faire partie de la liste des lettres contenues dans le tableau des contraintes dans la même colonne (même indice). Cette liste fournit autant d'états successeurs potentiels. Cependant, il faut vérifier qu'en ajoutant une lettre, les contraintes sont toujours satisfaites. La liste des états successeurs est donc donnée par la liste des états formés en ajoutant une nouvelle lettre dans la colonne choisie tout en respectant les contraintes.

- *Quelles sont les grandes lignes de la fonction permettant de vérifier qu'un état est consistant par rapport aux contraintes du problème ?*

On prend chaque mot du tableau des contraintes et on le compare au mot associé à l'état dont on veut vérifier la consistance. Il faut compter le nombre de lettres qui sont les mêmes et comparer ce chiffre au chiffre correspondant du tableau des scores. Si c'est inférieur ou égal et qu'il reste suffisamment de cases pour atteindre le chiffre des scores, c'est consistant, sinon ça n'est pas consistant. Un état est consistant si et seulement si la consistance est validée pour tous les mots du tableau des contraintes.

- *Au niveau de la fonction successeur, quelle stratégie proposez-vous pour déterminer le choix de la lettre à placer ?*

La stratégie la plus simple consiste à commencer à l'indice 1 et de continuer de 1 en 1 jusqu'à l'indice 5. Par exemple si on a déjà placé les n premières lettres du mot à trouver, on s'intéressera à la lettre à placer à l'indice n+1. Une stratégie plus pertinente consiste à choisir la colonne (l'indice) la plus contrainte, parmi les colonnes non encore choisies. Pour cette application, on peut déterminer le nombre de lettres différentes de la colonne et choisir comme colonne celle où il y a le moins de choix (moins de choix, donc + de contraintes).

Exercice corrigé 6 :

On considère le problème de l'affectation d'un chiffre entre 0 et 9 aux variables D,E,U,X,S,P,T,N et F, pour que le résultat de l'opération suivante soit exact : DEUX + SEPT = NEUF, avec comme contrainte que chaque variable ait une valeur différente des autres.

Donnez les grandes lignes d'une méthode de résolution de ce problème par un programme informatique en vous inspirant de techniques connues pour résoudre des problèmes similaires.

Réponse :

Il s'agit d'un problème classique de satisfaction de contraintes, comme le problème des 8 reines ou du sudoku (voir cours).

Ce type de problème est caractérisé par un ensemble de variables dont il faut déterminer la valeur et un ensemble de contraintes à respecter.

Les variables sont les 9 entiers D,E,U,X,S,P,T,N et F.

La contrainte principale peut s'écrire de la manière suivante :

$$(1000D + 100E + 10U + X) + (1000S + 100E + 10P + T) = (1000N + 100E + 10U + F)$$

A laquelle s'ajoute les contraintes de non égalité des variables et la contrainte d'intervalle de valeurs entre 0 et 9 pour chacune d'elles.

On peut également ajouter des contraintes qui découlent de l'analyse du problème.

Par exemple, on ne peut avoir $T=0$ (sinon $X=F$) ni $X=0$ (sinon $T=F$) ni $F=0$ (sinon $X=T=0$ ou $X=T=5$).

De plus, E vaut 0 ou 9 pour pouvoir avoir $E + E +$ retenue possible 0 ou 1 = E

La résolution du problème (voir cours) consiste à suivre les étapes algorithmiques suivantes, qui s'apparentent à une recherche d'un plus court chemin dans l'espace d'états selon un parcours en profondeur d'abord.

(0) Aucune variable n'est affectée

(1) Choisir la variable la plus contrainte parmi celles qui ne sont pas encore affectées

(2) Affecter une valeur à cette variable parmi celles qui n'ont pas encore été essayées (chemins ouverts de l'espace d'états) et qui permettent de respecter les contraintes. S'il ne reste plus de valeur possible, aucune solution, fin.

(3) Si toutes les variables sont affectées, fin, on a trouvé

(4) Aller en (1)

NB : la variable la plus contrainte est celle qui comporte le moins de chiffres possibles. Par exemple, après l'analyse proposée, c'est E puisqu'elle ne peut prendre comme valeur que 0 ou 9, ensuite T, X et F.

Exercice 7 :

Sudoku

Proposez une modélisation de la recherche de solution pour le Sudoku. Spécifiez en particulier l'espace d'états, l'état initial et la fonction successeur. Argumentez vos choix.

Exercice 8 :

Mots croisés indexés

Dans certains mots-croisés, la liste des mots utilisés pour la grille est fournie et l'objectif consiste à retrouver la place de chacun. Proposez une modélisation de la recherche de solution pour ce jeu. Spécifiez en particulier l'espace d'états, l'état initial et la fonction successeur.

Exercice 9 :

Proposez le schéma d'un bras manipulateur comportant uniquement des liaisons pivot et dont la seule fonction est de prendre avec une pince des crayons posés à plat sur une table, sachant qu'il n'y a pas d'autres objets. La prise doit se faire avec une pince placée verticalement au-dessus du crayon et avec chaque mâchoire placée symétriquement de part et d'autre du crayon. Quel est le nombre minimal de servomoteurs ? Justifiez votre réponse.

Exercice 10 :

Proposez plusieurs méthodes et capteurs utilisés en robotique pour déterminer la distance des obstacles proches du robot, en précisant brièvement le principe de fonctionnement.

Exercice 11 :

En robotique, on est amené à gérer de nombreuses « incertitudes » liées à la position et orientation du robot. D'où proviennent ces incertitudes ? Développez.

Exercice 12 :

Le problème des 8 reines consiste à placer 8 reines sur un échiquier de 8x8 cases sans qu'aucune ne soit en prise avec une autre (pas la même ligne, pas la même colonne et pas la même diagonale). Pour résoudre le problème des 8 reines, peut-on exploiter un algorithme génétique ? Si oui, préciser la mise en œuvre et sinon pourquoi.