Intelligence artificielle (IF06M100) • TD 3

Algorithmes et recherches heuristiques

Une stratégie de recherche permet de choisir l'ordre dans lequel les états sont développés

Stratégies d'exploration informées (heuristiques)

une stratégie d'**exploration informée** (une stratégie qui utilise des connaissances du problème en plus de sa définition) peut trouver des solutions plus efficacement qu'une stratégie non informée.

L'approche générale que nous examinons s'appelle **exploration par le meilleur d'abord**.

Cette exploration par le meilleur d'abord est un cas particulier des algorithmes généraux EXPLORATION-ARBRE et EXPLORATION-GRAPHE, dans lesquels on choisit un nœud à développer en se fondant sur une **fonction d'évaluation** $f(n)$.

La fonction d'évaluation est vue comme une estimation des coûts ;

le nœud qui a la valeur de l'évaluation *la plus faible* est développé d'abord.

La plupart des algorithmes par le meilleur d'abord incluent une **fonction heuristique**, notée $h(n)$ comme composante de f :

$$h(n) = \text{coût estimé du chemin le moins coûteux de l'état au nœud } n \text{ à un état but.}$$

Par exemple, en Roumanie, on pourrait estimer le coût du chemin le moins coûteux d'Arad à Bucarest par la distance à vol d'oiseau d'Arad vers Bucarest.

Les fonctions heuristiques sont la manière la plus habituelle de fournir des connaissances supplémentaires sur le problème à l'algorithme d'exploration.

Pour l'instant, nous les considérons comme des fonctions arbitraires, non négatives, spécifiques aux problèmes concernés, avec une contrainte : si n est un nœud but, $h(n) = 0$.

deux manières d'utiliser l'information heuristique pour guider l'exploration.

Exploration gloutonne par le meilleur d'abord

Exploration A* : minimisation du coût total estimé de la solution

Source : Artificial Intelligence: A Modern Approach by S. Russell and P. Norvig (French Version)

Exercice 1

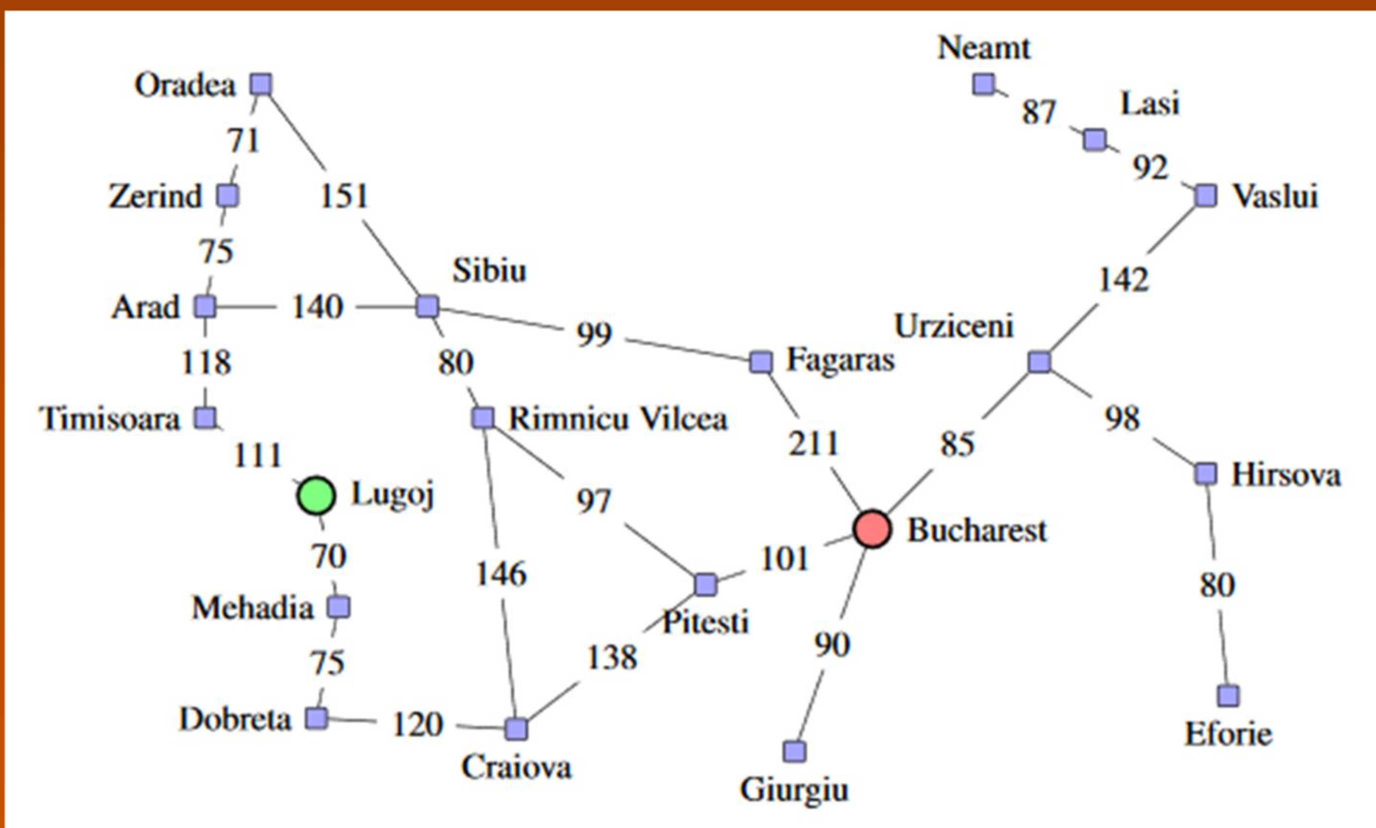
Appliquez l'algorithme A^* au problème du voyage en Roumanie en appliquant l'heuristique de la distance à vol d'oiseau.

Vous supposerez que vous voulez voyager de Lugoj à Bucharest.

Pour chaque nœud, vous donnerez les valeurs de f, g et h.

Si un même état apparaît dans deux nœuds différents, avec deux valeurs de f différentes, on conserve seulement celui avec la meilleure (la plus petite) valeur de f.

On supposera aussi que l'on ne passera pas deux fois par la même ville sur le même chemin (la même branche de l'arbre de recherche).



Ligne droite jusqu'à Bucharest					
Arad	366	Hirsova	151	Rimnicu Vilcea	193
Bucharest	0	Lasi	226	Sibiu	253
Craiova	160	Lugoj	244	Timisoara	329
Dobreta	242	Mehadia	241	Urziceni	80
Eforie	161	Neamt	234	Vaslui	199
Fagaras	176	Oradea	380	Zerind	374
Giurgiu	77	Pitesti	100		

3.5.2 Exploration A* : minimisation du coût total estimé de la solution

La forme la plus connue d'exploration par le meilleur d'abord s'appelle l'**exploration A*** (prononcez « exploration A étoile »). Elle évalue les nœuds en combinant $g(n)$, le coût pour atteindre le nœud, et $h(n)$, le coût pour aller du nœud au but :

$$f(n) = g(n) + h(n).$$

Puisque $g(n)$ indique le coût de chemin du nœud initial au nœud n, et que $h(n)$ est le coût estimé du chemin le moins coûteux de n au but, on a :

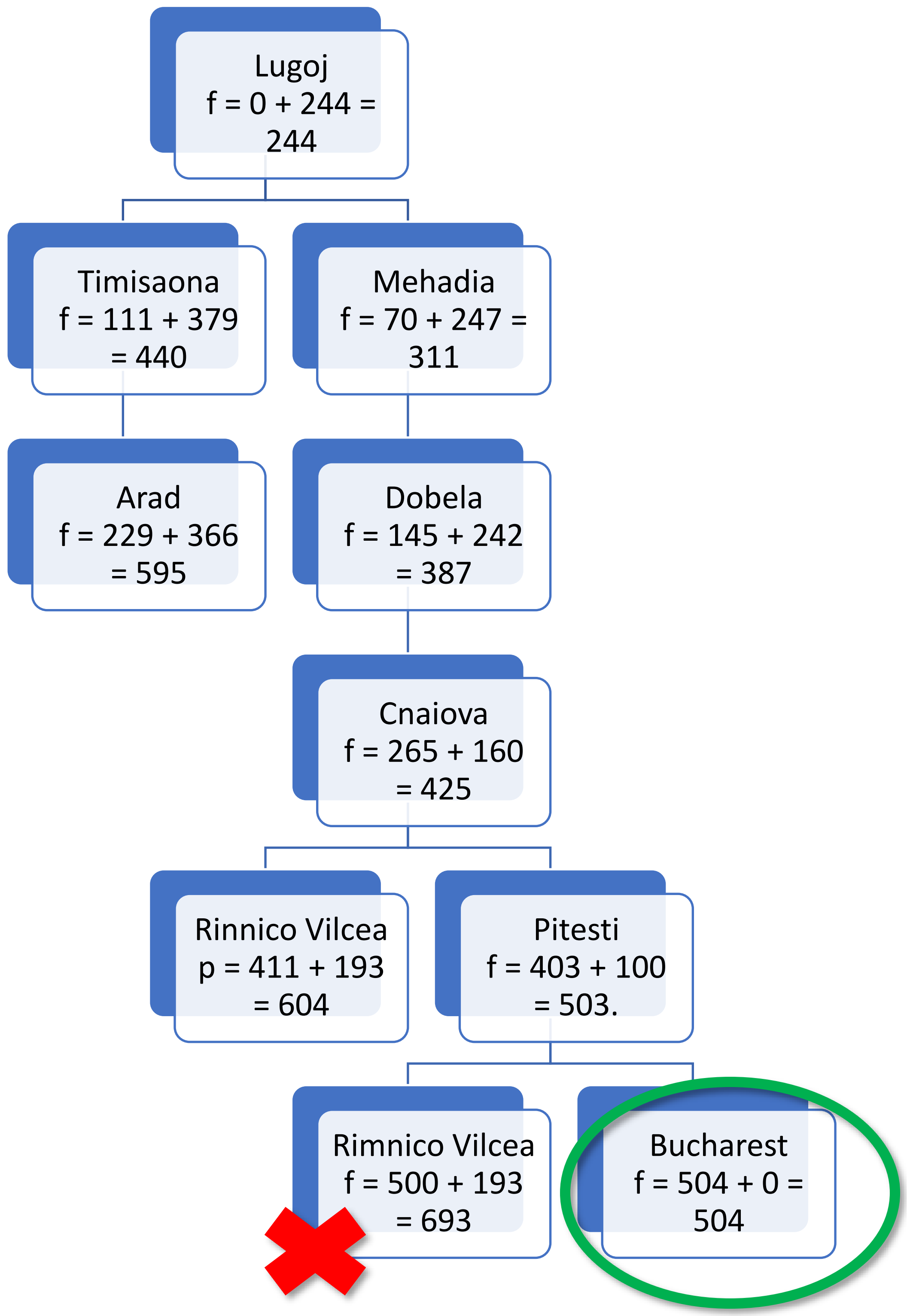
$$f(n) = \text{ coût estimé de la solution la moins coûteuse passant par } n.$$

Ainsi, si nous essayons de trouver la solution la moins coûteuse, il est raisonnable d'essayer d'abord un nœud avec une valeur de $g(n) + h(n)$ minimale. Cette stratégie est même fort avantageuse : si la fonction heuristique $h(n)$ remplit certaines conditions, l'exploration A* est à la fois complète et optimale. L'algorithme est identique à EXPLORATION-COÛT-UNIFORME, excepté que A* utilise $g + h$ au lieu de g.

Algorithme A*

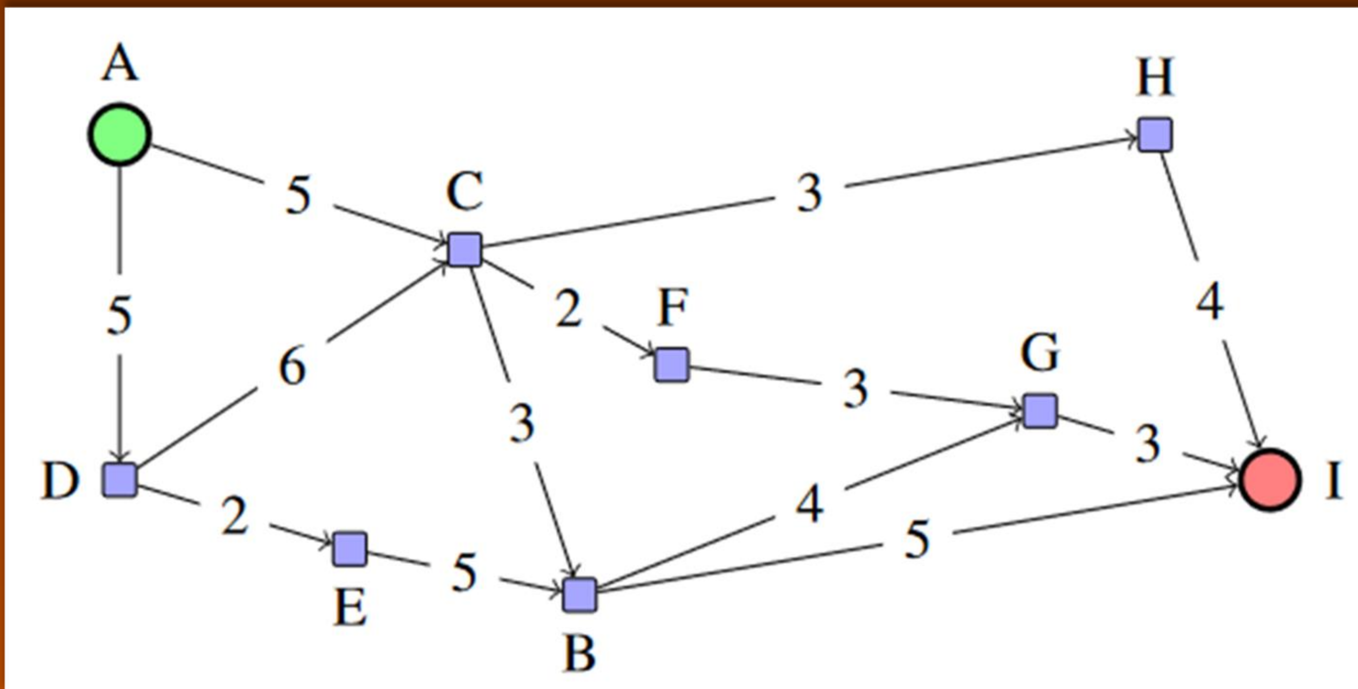
- **Idée** : Eviter de développer des chemins qui sont déjà chers
- **Fonction d'évaluation** : $f(n) = g(n) + h(n)$
  - $g(n)$ est le coût de l'état initial à l'état n
  - $h(n)$ est le coût estimé pour atteindre l'état final
  - $f(n)$ est le coût total estimé pour aller de l'état initial à l'état final en passant par n

Source : Cours UE Intelligence artificielle / E. Bonzon


Lugoj
f = 0 + 244 = 244
Timisaona
f = 111 + 379 = 440
Mehadia
f = 70 + 247 = 311
Arad
f = 229 + 366 = 595
Dobela
f = 145 + 242 = 387
Cnaiova
f = 265 + 160 = 425
Rinnico Vilcea
p = 411 + 193 = 604
Pitesti
f = 403 + 100 = 503.
Rimnico Vilcea
f = 500 + 193 = 693
Bucharest
f = 504 + 0 = 504

Exercice 2

Considérez la carte orientée suivante.



L'objectif est de trouver le chemin le plus court de A vers I.

On donne également trois heuristiques, h_1, h_2 et h_3.

Nœud	A	B	C	D	E	F	G	H	I
h_1	10	5	5	10	10	3	3	3	0
h_2	10	2	8	11	6	2	1	5	0
h_3	10	2	6	11	9	6	3	4	0

Question 1

Est-ce que h1, h2 et h3 sont admissibles ? Justifier.

Un heuristique est admissible si h(n) ≤ h* , ou h*(n) est le cout réel pour aller de n jusqu'à l'état final sans surestimer ce cout réel (heuristique optimiste).

$$\forall\ \boldsymbol{i}\ , h_i(A)\ \leq\ h^*(A)$$

$$h_i(B)\ \leq\ h^*(B)$$

$$h_2(C)\ \leq\ h^*(C)\ \rightarrow\ h_2\ non\ admissible.$$

$$\forall\ \boldsymbol{i}\ \boldsymbol{\epsilon}\ \{\mathbf{1},\mathbf{3}\}\quad h_i(C)\ \leq\ h^*(C)$$

$$h_i(D)\ \leq\ h^*(D)$$

$$h_i(E)\ \leq\ h^*(E)$$

$$h_i(F)\ \leq\ h^*(F)$$

$$h_i(G)\ \leq\ h^*(G)$$

$$h_i(H)\ \leq\ h^*(H)$$

$$h_i(I)\ \leq\ h^*(I)$$

Exercice 2

Question 2

Quelles relations de dominance existent entre ces trois heuristiques ?

$$\boldsymbol{h_1\ ne\ domine\ pas\ h_3\ car\ h_3(C) > \ h_1(C)}$$

$$\boldsymbol{h_3\ ne\ domine\ pas\ h_1\ car\ h_1(B) > \ h_3(B)}$$

Exercice 2

Question 3

Est-ce que $h4 = max(h1, h3)$ est admissible ? Justifier.

$h_4 = max(h_1, h_3)$ *est admissible car*

$\forall\, n, h_1(n) \leq h^*(n)$ *et* $\forall\, n, h_3(n) \leq h^*(n)$

Donc $\forall\, n, h_4(n) \leq h^*(n)$

ne domine pas h_3 *car* $h_3(C) > h_1(C)$

h_3 *ne domine pas* h_1 *car* $h_1(B) > h_3(B)$

	A	B	C	D	E	F	G	H	I
h_4	10	5	6	11	10	6	9	4	0

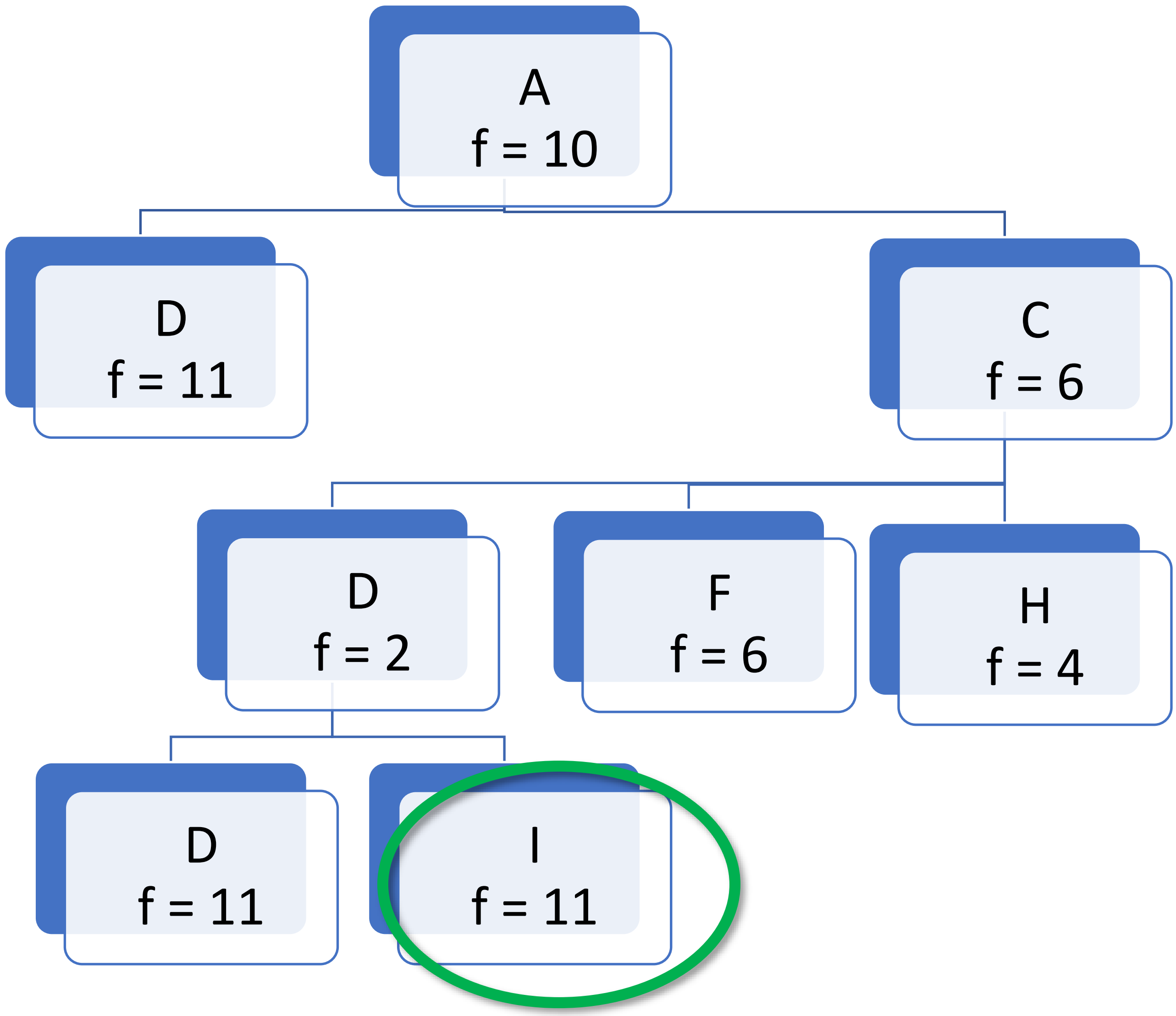
Exercice 2

Question 4

Appliquer la recherche gloutonne en utilisant h3.

Donner la suite des nœuds développés.

[redacted]



Nœud développès : A, C, B