CentraleSupélec

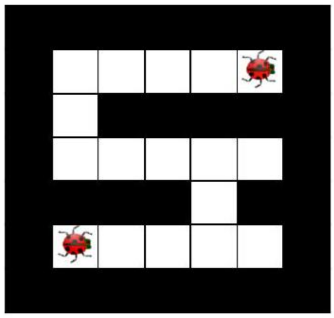
2nd Year
Elective Course

2022 – 2023

Artificial Intelligence – Tutorial Session 2
TS 2: State Space Search

1 Long Lost Bug Friends

You control a pair of long lost bug friends. You know the maze, but you do not have any information about which square each bug starts in. You want to help the bugs reunite. You must pose a search problem whose solution is an all-purpose sequence of actions such that, after executing those actions, both bugs will be on the same square, regardless of their initial positions. Any square will do, as the bugs have no goal in mind other than to see each other once again. Both bugs execute the actions mindlessly and do not know whether their moves succeed; if they use an action which would move them in a blocked direction, they will stay where they are. Bugs cannot jump onto walls. Both bugs can move in each time step. Every time step that passes has a cost of one.



(a) Give a minimal state representation for the above search problem.

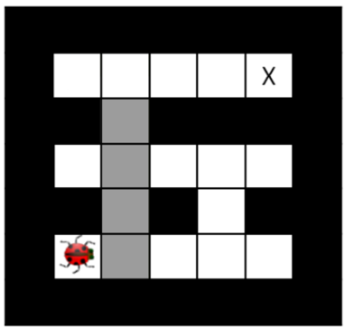
(b) Give the size of the state space for this search problem.

(c) Give a nontrivial admissible heuristic for this search problem.

a) On représente les cases où on est sûr de ne pas avoir d'insecte avec la suite de déplacement
b) Si carte de taille M*N alors 2^MN
c) On fait la division entière par 2 du nombre de cases où il peut y avoir des insectes

2 Menagerie

The hive of insects needs your help again. As before, you control an insect in a rectangular maze-like environment with dimensions $M \times N$, as shown to the right. At each time step, the insect can move into a free adjacent square or stay in its current location. All actions have cost 1.
In this particular case, the insect must pass through a series of partially flooded tunnels. Flooded squares are lightly shaded in the example map shown. The insect can hold its breath for A time steps in a row. Moving into a flooded square requires your insect to expend 1 unit of air, while moving into a free square refills its air supply.



(a) Give a minimal state space for this problem (i.e. do not include extra information). You should answer for a general instance of the problem, not the specific map shown.

(b) Give the size of your state space.

Parts (c), (d), and (e) Consider a search problem where all edges have cost 1 and the optimal solution has cost C. Let h be a heuristic which is $\max\{h^\star - k, 0\}$, where $h^\star$ is the actual cost to the closest goal and k is a nonnegative constant.

a) La position de l'insecte et la quantité d'air
b) A*M*N

(c) Which of the following statements are true?

(i) h is admissible.

(ii) h is consistent.

(iii) A* tree search (no closed list) with h will be optimal.

(iv) A* graph search (with closed list) with h will be optimal.

(d) Which of the following is the most reasonable description of how much more work will be done (= how many more nodes will be expanded) with heuristic h compared to h^*, as a function of k ?

(i) Constant in k

(ii) Linear in k

(iii) Exponential in k En effet, plus on s'approche de la fin, plus on va tester de chemins différents

(iv) Unbounded

Now consider the same search problem, but with a heuristic h' which is 0 at all states that lie along an optimal path to a goal and h^* elsewhere.

(e) Which of the following statements are true?

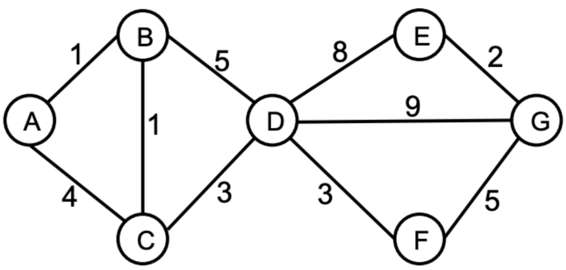
(i) h' is admissible.

(ii) h' is consistent.

(iii) A* tree search (no closed list) with h' will be optimal.

(iv) A* graph search (with closed list) with h' will be optimal. On n'est pas consistant donc a priori non

3 Search



Node	h_1	h_2
A	9.5	10
B	9	12
C	8	10
D	7	8
E	1.5	1
F	4	4.5
G	0	0

Consider the state space graph shown above. A is the start state and G is the goal state. The costs for each edge are shown on the graph. Each edge can be traversed in both directions. Note that the heuristic h_1 is consistent but the heuristic h_2 is not consistent.

(a) Possible paths returned For each of the following graph search strategies (do not answer for tree search), mark which, if any, of the listed paths it could return. Note that for some search strategies the specific path returned might depend on tie-breaking behavior. In any such cases, make sure to mark all paths that could be returned under some tie-breaking scheme.

Search Algorithm	A-B-D-G	A – C – D – G	A – B – C – D – F – G
Depth first search	(i)	(ii)	(iii)
Breadth first search	(iv)	(v)	(vi)
Uniform cost search	(vii)	(viii)	(ix)
A * search with heuristic h_1	(x)	(xi)	(xii)
A * search with heuristic h_2	(xiii)	(xiv)	(xv)

(b) Heuristic function properties

Suppose you are completing the new heuristic function h_3 shown below. All the values are fixed except $h_3(B)$.

Node	A	B	C	D	E	F	G
h_3	10	?	9	7	1.5	4.5	0

For each of the following conditions, write the set of values that are possible for $h_3(B)$. For example, to denote all non-negative numbers, write $[0, \infty]$, to denote the empty set, write $\emptyset$, and so on.

(i) What values of $h_3(B)$ make h_3 admissible? [0, 12]

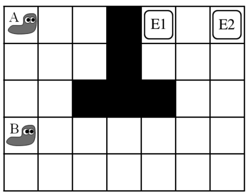
(ii) What values of $h_3(B)$ make h_3 consistent? [9, 10], car il faut vérifier les arrêtes dans les 2 sens

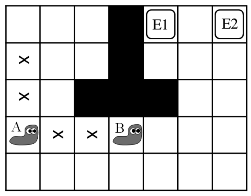
(iii) What values of $h_3(B)$ will cause A* graph search to expand node A, then node C, then node B, then node D in order? [12, 13]

4 Slugs

As shown in the diagram on the left, two slugs A and B want to exit a maze via exits E_1 and E_2. At each time step, each slug can either stay in place or move to an adjacent free square. A slug cannot move into a square that the other slug is moving into. Either slug may use either exit, but they cannot both use the same exit. When a slug moves, it leaves behind a poisonous substance. The substance remains in the square for 2 time steps; during this time, no slug can move into the poisonous square.

For example, if slugs A and B begin in the positions shown above on the left, and slug A moves Down 3 steps while slug B moves Right three steps, then the world becomes as shown above on the right, with ×'s marking the poisonous squares. Note that slug A does have the option of moving Right from its current position, since the trail from B will evaporate by the time it arrives.





You must pose a search problem that will get both slugs to the exits in as few time steps as possible. Assume that the board is of size M by N. While all answers should hold for a general board, not just the one shown above, you do not need to generalize beyond two slugs, two exits, or two timesteps of decay for the slug trails.

(a) How many states are there in a minimal representation of the space? Justify with a brief description of the components of your state space. (5*M*N)^2

(b) Let $d(x, y)$ denote Manhattan distance between x and y. Consider three heuristics, defined in terms of slug locations A and B and exit locations E_1 and E_2. Remember that either slug can use either exit, but they must use different exits.

Definition	Explanation
$h_1 = \max_{s \in \{A,B\}} \min(d(s, E_1), d(s, E_2))$	Return the maximum, over the two slugs, of the distance from the slug to its closest exit.
$h_2 = \max(d(A, E_1), d(B, E_2))$	Assign slug A to exit E_1 and B to E_2; then return the maximum distance from either slug to its assigned exit.
$h_3 = \min_{(e,e') \in \{(E_1,E_2),(E_2,E_1)\}} \max(d(A, e), d(B, e'))$	Return the max distance from a slug to its assigned exit, under the assignment of slugs to distinct exits which minimizes this quantity.

(i) For each heuristic, check the box if the heuristic is admissible

☐ h_1 ☐ h_2 ☐ h_3

(ii) For each pair of heuristics, check the box which correctly describes their dominance relationship. Note: dominance is defined regardless of admissibility.

☐ h_1 dominates h_2	☐ h_2 dominates h_1	☐ $h_1 = h_2$	☐ none
☐ h_1 dominates h_3	☐ h_3 dominates h_1	☐ $h_1 = h_3$	☐ none
☐ h_2 dominates h_3	☐ h_3 dominates h_2	☐ $h_2 = h_3$	☐ none