ARTIFICIAL INTELLIGENCE (E016350)
GHENT UNIVERSITY
AY 2024/2025
Prof. Aleksandra Pizurica
Asst. Prof. Nicolas Nadisic
E-mail: ai@lists.ugent.be

Solutions: Solving problems by searching

1. [R&N, 3.17] Which of the following are true and which are false? Explain your answers.
   (a) Depth-first search always expands at least as many nodes as A^* search with an admissible heuristic.
   (b) $h(n) = 0$ is an admissible heuristic for the 8-puzzle.
   (c) A^* search is of no use in robotics because percepts, states and actions are continuous.
   (d) Breadth-first search is complete even if zero step costs are allowed.
   (e) Assume that a rook can move on a chessboard any number of squares in a straight line, vertically or horizontally, but cannot jump over other pieces. Manhattan distance is an admissible heuristic for the problem of moving the rook from square A to square B in the smallest number of moves.

Solution:

(a) *False:* Depth-first search (DFS) is neither optimal nor complete, so it may come with a sub-optimal solution at an earlier stage than A^* (expanding less nodes). For instance, DFS could expand 1 node, with a higher cost than A^*, while A^* would expand more nodes than DFS to find an optimal path to the goal (with a lower cost but a "longer" path). A^* largely dominates any graph-search algorithm that is guaranteed to find optimal solutions.

(b) *True:* For a heuristic function $h(n)$ to be admissible, we require that it never overestimates the real cost to reach the goal. If we consider the path costs to be non-negative (as it is usually done), then $h(n) = 0$ satisfy mentioned requirement. Note, however, that using this heuristic is the same as using Uniform-cost search, an uninformed search strategy, so we are not using any advantage of an informed search.

(c) *False:* It is a fact that the world is continuous, but it is also true that usually we use discretization at some extent to solve problems that otherwise would be infeasible or harder to manage. Also, A^* search is often used in robotics; the space can be discretized or skeletonized and A^* is often used for navigation. As a reference, in the DARPA Challenge 2007, A^* was one of the options used for planning routes to achieve all checkpoints in the real-world road.

(d) *True:* if the other conditions for completeness are met. In breadth-first search (BFS), we care about the number of steps the path has rather than the path cost. So, for this search to be complete, the conditions are that the goal is at a finite depth and also that the branching factor is finite.

(e) *False:* A rook can move across the board in move one, although the Manhattan distance from start to finish is 8.
Extra: What would be an admissible heuristic for this problem?

2. [R&N, 3.33] n vehicles occupy squares $(1,1)$ through $(n,1)$ (i.e., the top row) of an $n \times n$ grid. The vehicles must be moved to the bottom row but in reverse order; so the vehicle i that starts in $(i,1)$ must end up in $(n-i+1,n)$. On each time step, every one of the n vehicles can move one square up, down, left or right, or stay put; if a vehicle stays put, one other adjacent vehicle (but not more than one) can hop over it. Two vehicles cannot occupy the same square.

   (a) Calculate the size of the state space as a function of n.

   (b) Calculate the branching factor as a function of n.

   (c) Suppose that vehicle i is at (x_i, y_i); write a nontrivial admissible heuristic h_i for the number of moves it will require to get to its goal location $(n-i+1,n)$, assuming no other vehicles are on the grid.

   (d) Which of the following are admissible heuristics for the original problem and why? (Note: h_i is the heuristic you came up with in the previous part.)

   A. $h = \min\{h_1, \ldots, h_n\}$ B. $h = \max\{h_1, \ldots, h_n\}$ C. $h = \sum_{i=1}^{n} h_i$

Solution:

(a) n^{2n}. There are n vehicles in n^2 locations, so roughly (ignoring the one-per-square constraint) we obtain $(n^2)^n = n^{2n}$ states. (If we don't ignore the one-per square constraint, we obtain $\frac{(n^2)!}{(n^2-n)!}$. *Think why this was not given as the solution here. Compare the corresponding outcomes for large n, say $n = 80$ or $n = 100$*)

(b) 5^n

(c) Manhattan distance, i.e., $h_i(x_i, y_i) = |(n-i+1) - x_i| + |n - y_i|$. This in fact gives the exact cost of moving the car to the goal, so it is also admissible. If there were other cars on the grid, this would no longer necessarily be an admissible heuristic because car i could potentially jump over other cars, allowing it to get to the goal in less steps than just using Manhattan distance.

(d) Only A. The explanation is non-trivial as it requires two observations.
   - First, let the work W_{tot} in a given solution be the total distance moved by all vehicles over their joint trajectories; that is, for each vehicle, add the lengths of all the steps taken. We have $W_{tot} \geq \sum_i h_i \geq n \min\{h_1, \ldots, h_n\}$.

- Second, the total work we can get done per step is $\leq n$. Indeed, note that for every car that jumps 2, another car has to stay put (move 0), so the total work per step is bounded by n. Thus we can conclude that $W_{step} \leq n$.

Hence, completing all the work requires at least

$$Num_{steps} = \frac{W_{tot}}{W_{step}} \geq \frac{n \cdot \min\{h_1, \ldots, h_n\}}{n} = \min\{h_1, \ldots, h_n\}$$

steps, so $h = \min\{h_1, \ldots, h_n\}$ is admissible.

3. If $f(s)$, $g(s)$ and $h(s)$ are all admissible heuristics then which of the following are also guaranteed to be admissible heuristics:

   (a) $f(s) + g(s) + h(s)$

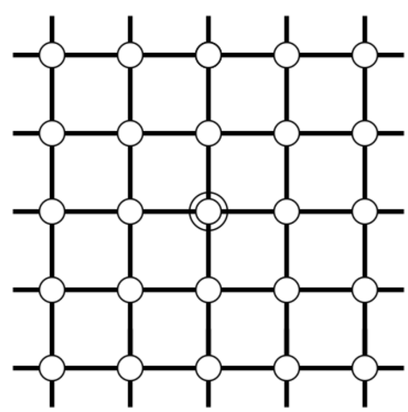
   (b) $f(s)/6 + g(s)/3 + h(s)/2$

   (c) $\min(f(s), g(s), h(s))$

   (d) $\max(f(s), g(s), h(s))$

Solution: In order to guarantee that a function of admissible heuristics is still admissible, the expression must be less than or equal to the max of the heuristics. Sums and products do not satisfy these, so (a) immediately fails. (c) and (d) work because the max of admissible heuristics is still admissible, as is the min of an admissible heuristic and anything else. Lastly, (b) is a weighted average, and is thus also less than the max, and is thus admissible.

4. **(Old exam question)** Consider the unbounded version of the regular 2D grid. The start state is at the origin $(0, 0)$, and the goal state is at (x, y).



   (a) What is the branching factor b in this state space?

(b) How many distinct states are there at depth k (for $k > 0$)?
A. 4^k B. $4k$ C. $4k^2$

(c) Breadth-first search *without* repeated-state checking *expands* at most
A. $(4^{m+n+1} - 1)/3 - 1$ B. $4(m + n) - 1$ C. $2(m + n)(m + n + 1) - 1$
nodes before terminating, where $m = |x|$, $n = |y|$.

(d) Breadth-first search *with* repeated-state checking *expands* at most
A. $(4^{m+n+1} - 1)/3 - 1$ B. $4(m + n) - 1$ C. $2(m + n)(m + n + 1) - 1$
nodes before terminating, where $m = |x|$, $n = |y|$.

(e) Define the heuristic h as a Manhattan distance for a state at (u, v) and write the corresponding expression. Explain!

(f) *True/False*: A^* search with repeated state checking using h expands $\mathcal{O}(|x| + |y|)$ nodes before terminating.

(g) Does h remain admissible if some links are removed? If not, give a nontrivial admissible heuristic for this search problem.

(h) Does h remain admissible if some links are added between nonadjacent states? If not, give a nontrivial admissible heuristic for this search problem.

Solution:

(a) The branching factor is 4 (number of neighbors of each location).

(b) B. The states at depth k form a square rotated at 45° to the grid. Obviously there is a linear number of states along the boundary of the square, so the answer is $4k$.

(c) A. Without repeated state checking, BFS expands **exponentially** many nodes: counting precisely, we get $(4^{|x|+|y|+1} - 1)/3$.

(d) C. There are **quadratically** many states within the square for depth $|x| + |y|$, so the answer is $2(|x| + |y|)(|x| + |y| + 1) + 1$. Note that we are having repeated-state checking, so we won't visit any one of the states more than once.

(e) For the state (u, v) we define the Manhattan distance by $h(u, v) = |u - x| + |v - y|$ and obviously this is an admissible heuristic for the given problem.

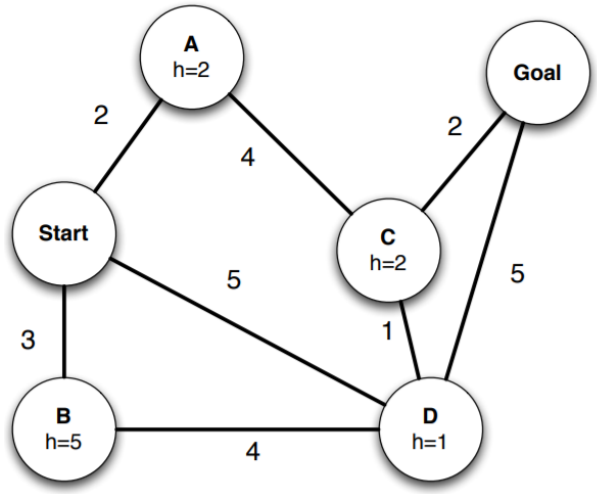
(f) *True*. For a start state $(0, 0)$ and a goal state (x, y), A^* graph search chooses exactly $|x| + |y|$ nodes for the optimal path. Note that all nodes in the rectangle defined by $(0, 0)$ and (x, y) are candidates for the optimal path, and there are quadratically many of them, all of which may be expanded in the worst case, which gives $\mathcal{O}(|x| \cdot |y|)$. But A^* is optimal, so it expands only $\mathcal{O}(|x| + |y|)$.

(g) Removing links may induce detours, which require more steps, so h is an underestimate and it will remain admissible.

(h) Nonlocal links can reduce the actual path length below the Manhattan distance, so h won't be an admissible heuristic in this case. An admissible heuristic in this case would be $h(u, v) = \sqrt{(u - x)^2 + (y - v)^2}$.

5. For each of the following graph search strategies, work out the order in which states are expanded, as well as the path returned by graph search. In all cases, assume ties resolve in such a way that states with earlier alphabetical order are expanded first. The start and goal state are S and G, respectively. Remember that in graph search, a state is expanded only once.

   (a) Depth-first search.
   (b) Breadth-first search.
   (c) Uniform cost search.
   (d) Greedy search with the heuristic h shown on the graph.
   (e) A^* search with the same heuristic h.



Solution:

(a) *Depth-first search.*
States Expanded: Start, A, C, D, B, Goal
Path Returned: Start-A-C-D-Goal

(b) *Breadth-first search.*
States Expanded: Start, A, B, D, C, Goal
Path Returned: Start-D-Goal

(c) *Uniform cost search.*
States Expanded: Start, A, B, D, C, Goal
Path Returned: Start-A-C-Goal

(d) *Greedy search with the heuristic h shown on the graph.*
States Expanded: Start, D, Goal
Path Returned: Start-D-Goal

(e) *A^* search with the same heuristic h.*
States Expanded: Start, A, D, C, Goal
Path Returned: Start-A-C-Goal

When using the A^* search algorithm, we know that if two paths have the same estimated value, then states with earlier alphabetical order are expanded first. Let us consider the following steps:

$$\textbf{Step 1:} \quad \begin{aligned} S - A &\rightarrow 2 + 2 = 4 \\ S - B &\rightarrow 3 + 5 = 8 \\ S - D &\rightarrow 5 + 1 = 6 \end{aligned}$$

Since the path $S - A$ has the smallest value we expand it further.

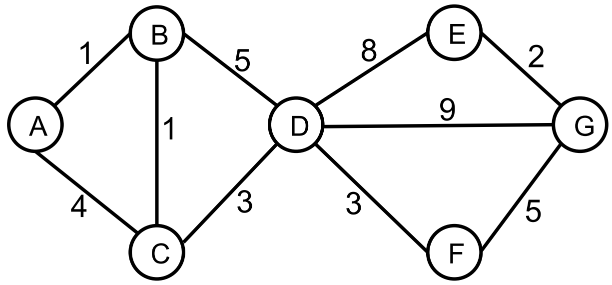
$$\textbf{Step 2:} \qquad S - A - C \quad \rightarrow \quad 6 + 2 = 8$$

Now we get the cost of 8, but the path $S - D$ has smaller value, so let us consider that one.

$$\textbf{Step 3:} \qquad S - D - G \quad \rightarrow \quad 10$$

Now we reached the goal with the cost of 10. But let us check what happens with the other two options with the lower cost: $S - A - C$ (cost 8) and $S - B$ (cost 8). Since $S - A - C$ alphabetically comes before $S - B$ we need to expand $S - A - C$ first. From here we immediately reach the goal. So the path $S - B$ won't be expanded. We always have to consider the entire path and not only one node that we take.

6. Consider the state space graph shown below.



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

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) 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			
Breadth first search			
Uniform cost search			
A^* search with heuristic h_1			
A^* search with heuristic h_2			

(b) 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.

1. What values of $h_3(B)$ make h_3 admissible?
2. What values of $h_3(B)$ make h_3 consistent?
3. 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?

Solution:

(a) The return paths depend on tie-breaking behaviors so any possible path has to be marked. DFS can return any path. BFS will return all the shallowest paths, i.e. $A-B-D-G$ and $A-C-D-G$. $A-B-C-D-F-G$ is the optimal path for this problem, so that UCS and A^* using consistent heuristic h_1 will return that path. Although, h_2 is not consistent, it will also return this path.

Search Algorithm	A-B-D-G	A-C-D-G	A-B-C-D-F-G
Depth first search	✓	✓	✓
Breadth first search	✓	✓	
Uniform cost search			✓
A^* search with heuristic h_1			✓
A^* search with heuristic h_2			✓

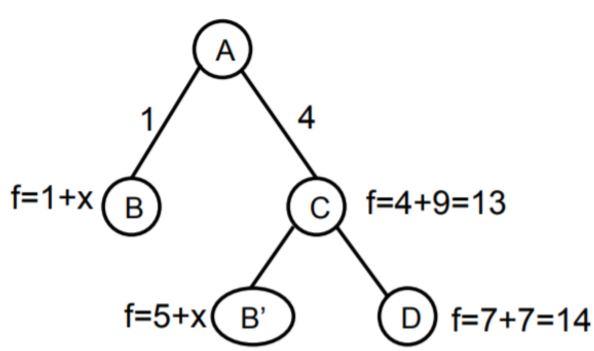
(b)
1. To make h_3 admissible, $h_3(B)$ has to be less than or equal to the actual optimal cost from B to goal G, which is the cost of path $B-C-D-F-G$, i.e. 12. The answer is $0 \leq h_3(B) \leq 12$.
2. All the other nodes except node B satisfy the consistency conditions. The consistency conditions that do involve the state B are:

$$h(A) \leq c(A,B) + h(B) \qquad h(B) \leq c(B,A) + h(A)$$
$$h(C) \leq c(C,B) + h(B) \qquad h(B) \leq c(B,C) + h(C)$$
$$h(D) \leq c(D,B) + h(B) \qquad h(B) \leq c(B,D) + h(D)$$

   Filling in the numbers shows this results in the condition: $9 \leq h_3(B) \leq 10$.
3. The A^* search tree using heuristic h_3 is represented below. In order to make A^* graph search expand node A, then node C, then node B, suppose $h_3(B) = x$, we need

$$\begin{aligned} 1+x &> 13 \\ 5+x &< 14 \quad \text{(expand } B'\text{)} \quad \text{or} \quad 1+x < 14 \quad \text{(expand } B\text{)} \end{aligned}$$

   so we can get $12 < h_3(B) < 13$.



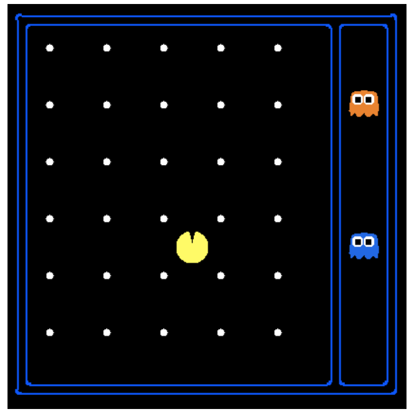
Remark

As mentioned in the theory lesson not all admissible heuristics are consistent. In practice it is usually assumed that admissible heuristics are consistent, implying that consistency is a desirable attribute. In the AI textbook "Artificial Intelligence: A Modern Approach", Russell and Norvig write that "although consistency is a stricter requirement than admissibility, one has to work quite hard to concoct heuristics that are admissible but not consistent". In the following exercise we will present an example of an admissible heuristic that is not consistent. Apart from that we will show that if a heuristic is consistent, then it also has to be admissible.

7. **(State space size)** For the Pacman world shown in the image on the right, assume

   - Agent positions: 120
   - Food count: 30
   - Ghost positions: 12
   - Agent facing: NSEW

   How many world states are there? How many states for route finding? How many states for "eat-all-dots"?



Solution:

Total states: $120 \times 2^{30} \times 12^2 \times 4$

States for route finding: 120

States for "eat-all-dots": 120×2^{30}

8. **(Old exam question)** For the 8-queens problem, hill climbing succeeds to find a solution in 14% of cases (and thus it fails in 86% of the cases). When it succeeds, it takes 4 steps on average and when it fails it takes 3 steps on average. How many steps will take on average to solve the problem with a random-restart hill climbing algorithm?

Solution: *Hint*: To start with, think of the following: if an experiment succeeds with a probability p, how many times you need to repeat it (on average) to reach a solution? Out of those attempts, all but one will be "fail". Hence, we have:

$$(7 - 1) \times 3 + 1 \times 4 = 22$$

Observe that this is pretty remarkable for a state space with approx. 17 million states!

9. **(Optional theoretical exercise)** [Adapted from R&N, 3.37]

(a) Prove that if a heuristic is consistent, it must be admissible.

(b) Construct an admissible heuristic that is not consistent.

(c) We know that A^* tree search is optimal if the heuristic h is admissible, while the A^* graph search is optimal if the heuristic h is consistent. Give a simple example that shows that A^* graph search with inconsistent but admissible heuristic h is not necessarily the optimal search strategy.

Solution:

(a) A heuristic function $h(n)$ is said to be consistent if, for every node n and every successor n' of n generated by any action a, the estimated cost of reaching the goal from n is no greater than the step cost of getting to n' plus the estimated cost of reaching the goal from n', i.e.

$$h(n) \leq c(n, a, n') + h(n')$$

where $c(n, a, n')$ is the step cost of getting to n' by taking action a at n.
Let $k(n)$ be the cost of the cheapest path from n to the goal node. We will prove by induction on the number of steps to the goal that $h(n) \leq k(n)$.

- **Base case:** If there are 0 steps to the goal from node n, then n is a goal and therefore $h(n) = 0 \leq k(n)$.
- **Induction step:** If n is i steps away from the goal, there must exist some successor n' of n generated by some action a s.t. n' is on the optimal path from n to the goal (via action a) and n' is $i-1$ steps away from the goal. Therefore,

  $$h(n) \leq c(n, a, n') + h(n').$$

  But by the induction hypothesis, $h(n') \leq k(n')$. Therefore,

  $$h(n) \leq c(n, a, n') + k(n') = k(n)$$

  since n' is on the optimal path from n to the goal via action a.

(b) One simple example can be found in (c). Here let us consider another example of an admissible but not consistent heuristic. Thus, consider a search problem where the states are nodes along a path $P = n_0, n_1, \ldots, n_m$ where n_0 is the start state, n_m is the goal state and there is one action from each state n_i which gives n_{i+1} as a successor with cost 1. The cheapest cost from any node n_i to the goal state n_m is then $k(n_i) = m - i$. Let us define a heuristic function as follows:

$$h(n_i) = m - 2\left\lceil \frac{i}{2} \right\rceil.$$

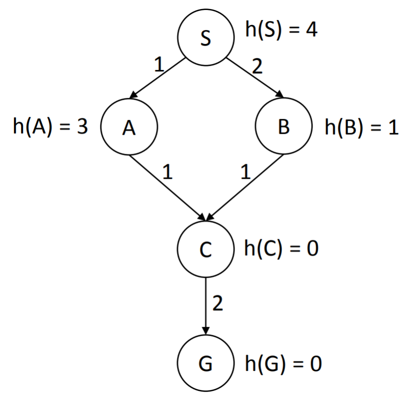
For all states n_i, holds that $h(n_i) \leq k(n_i)$, and so h is admissible. However, h is not consistent. If i is odd, then $h(n_i) = k(n_i) + 1$. Also $i + 1$ is then even and $h(n_{i+1}) = k(n_{i+1})$. From here, since the true cost is 1, we can easily conclude that

$$h(n_i) = k(n_i) + 1 = k(n_{i+1}) + 2 = h(n_{i+1}) + 2 > h(n_{i+1}) + 1.$$

Thus h is not consistent.

(c) In this example (where the heuristic is also admissible but not consistent), when searching for a path from S to G, the states are expanded in the following order (parentheses show f-values):

S	$(0 + 4 = 4)$
B	$(2 + 1 = 3)$
C	$(3 + 0 = 3)$
A	$(1 + 3 = 4)$
G	$(5 + 0 = 5)$



Note that expanding A finds a shorter path from S to C, but this is not reflected in the solution since C was expanded before A and we are not allowed to expand a state more than once in the case of the A^* graph search strategy.