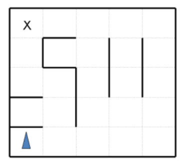
CS 188
Spring 2023

Regular Discussion 2 Solutions

1 Search and Heuristics

Imagine a car-like agent wishes to exit a maze like the one shown below:



1. The agent is directional and at all times faces some direction $d \in (N, S, E, W)$.
2. With a single action, the agent can *either* move forward at an adjustable velocity v *or* turn. The turning actions are *left* and *right*, which change the agent's direction by 90 degrees. Turning is only permitted when the velocity is zero (and leaves it at zero).
3. The moving actions are *fast* and *slow*. *Fast* increments the velocity by 1 and *slow* decrements the velocity by 1; in both cases the agent then moves a number of squares equal to its NEW adjusted velocity (see example below).
4. A consequence of this formulation is that it is **impossible** for the agent to move in the same nonzero velocity for two consecutive timesteps.
5. Any action that would result in a collision with a wall or reduce v below 0/above a maximum speed $V_{\max}$ is illegal.
6. The agent's goal is to find a plan which parks it (stationary) on the exit square using as few actions (time steps) as possible.

As an example: if at timestep t the agent's current velocity is 2, by taking the *fast* action, the agent first increases the velocity to 3 and move 3 squares forward, such that at timestep $t + 1$ the agent's current velocity will be 3 and will be 3 squares away from where it was at timestep t. If instead the agent takes the *slow* action, it first decreases its velocity to 1 and then moves 1 square forward, such that at timestep $t + 1$ the agent's current velocity will be 1 and will be 1 squares away from where it was at timestep t. If, with an instantaneous velocity of 1 at timestep $t + 1$, it takes the slow action again, the agent's current velocity will become 0, and it will not move at timestep $t + 1$. Then at timestep $t + 2$, it will be free to turn if it wishes. Note that the agent could not have turned at timestep $t + 1$ when it had a current velocity of 1, because it has to be stationary to turn.

(a) If the grid is M by N, what is the size of the state space? Justify your answer. You should assume that all configurations are reachable from the start state.

The size of the state space is $4MN(V_{max}+1)$. The state representation is (direction facing, x, y, speed). Note that the speed can take any value in $\{0, ..., V_{max}\}$.

(b) Is the Manhattan distance from the agent's location to the exit's location admissible? Why or why not?

No, Manhattan distance is not an admissible heuristic. The agent can move at an average speed of greater than 1 (by first speeding up to V_{max} and then slowing down to 0 as it reaches the goal), and so can reach the goal in less time steps than there are squares between it and the goal. A specific example: A timestep 0, the agent's starts stationary at square 0 and the target is 9 squares away at square 9. At timestep 0, the agent takes the *fast* action and ends up at square 1 with a velocity of 1. At timestep 1, the agent takes the *fast* action and ends up at square 3 with a velocity of 2. At timestep 2, the agent takes the *fast* action and ends up at square 6 with a velocity of 3. At timestep 3, the agent takes the *slow* action and ends up at square 8 with a velocity of 2. At timestep 4, the agent takes the *slow* action and ends up at square 9 with a velocity of 1. At timestep 5, the agent takes the *slow* action and stays at square 9 with a velocity of 0. Therefore, the agent can move 9 squares by taking 6 actions.

(c) State and justify a non-trivial admissible heuristic for this problem which is not the Manhattan distance to the exit.

There are many answers to this question. Here are a few, in order of weakest to strongest:

(a) The number of turns required for the agent to face the goal.

(b) Consider a relaxation of the problem where there are no walls, the agent can turn, change speed arbitrarily, and maintain constant velocity. In this relaxed problem, the agent would move with V_{max}, and then suddenly stop at the goal, thus taking $d_{manhattan}/V_{max}$ time.

(c) We can improve the above relaxation by accounting for the acceleration and deceleration dynamics. In this case the agent will have to accelerate from 0 from the start state, maintain a constant velocity of V_max, and slow down to 0 when it is about to reach the goal. Note that this heuristic will always return a greater value than the previous one, but is still not an overestimate of the true cost to reach the goal. We can say that this heuristic *dominates* the previous one.

In particular, let us assume that $d_{manhattan}$ is greater than and equal to the distance it takes to accelerate to and decelerate from V_{max} (In the case that $d_{manhattan}$ is smaller than this distance, we can still use $d_{manhattan}/V_{max}$ as a heuristic). We can break up the $d_{manhattan}$ into three parts: d_{accel}, $d_{V_{max}}$, and d_{decel}. The agent travels a distance of d_{accel} when it accelerates from 0 to V_{max} velocity. The agent travels a distance of d_{decel} when it decelerates from V_{max} to 0 velocity. In between acceleration and deceleration, the agent travels a distance of $d_{V_{max}} = d_{manhattan} - d_{accel} - d_{decel}$. $d_{accel} = 1 + 2 + 3 + V_{max} = \frac{(V_{max})(V_{max}+1)}{2}$ and $d_{decel} = (V_{max} - 1) + (V_{max} - 2) + ... + 1 + 0 = \frac{(V_{max})(V_{max}-1)}{2}$. So $d_{V_{max}} = d_{manhattan} - \frac{(V_{max})(V_{max}+1)}{2} - \frac{(V_{max})(V_{max}-1)}{2} = d_{manhattan} - V_{max}^2$. It takes V_{max} steps to travel the initial d_{accel}, $\frac{d_{manhattan} - V_{max}^2}{V_{max}}$ steps to travel the middle $d_{V_{max}}$ and V_{max} steps to travel the last d_{decel}. Therefore, our heuristic is

$$\begin{cases} \frac{d_{manhattan}}{V_{max}}, & \text{if } d_{manhattan} \leq V_{max}^2 = d_{accel} + d_{decel} \\ \frac{d_{manhattan}}{V_{max}} + V_{max}, & \text{if } d_{manhattan} > V_{max}^2 = d_{accel} + d_{decel} \end{cases}$$

(d) If we used an inadmissible heuristic in A* graph search, would the search be complete? Would it be optimal?

If the heuristic function is bounded, then A* graph search would visit all the nodes eventually, and would find a path to the goal state if there exists one. An inadmissible heuristic does not guarantee optimality as it can make the good optimal goal look as though it is very far off, and take you to a suboptimal goal.

(e) If we used an *admissible* heuristic in A* graph search, is it guaranteed to return an optimal solution? What if the heuristic was consistent? What if we were using A* tree search instead of A* graph search?

Although admissible heuristics guarantee optimality for A* *tree* search, they do not necessarily guarantee optimality for A* *graph* search; they are only guaranteed to return an optimal solution if they are consistent as well.

(f) Give a general advantage that an inadmissible heuristic might have over an admissible one.

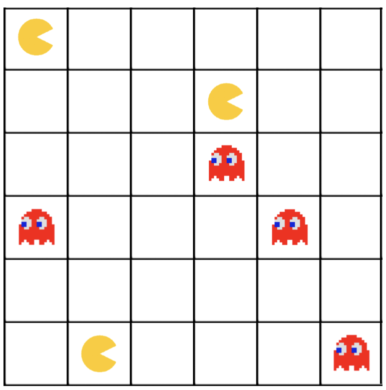
The time to solve an A* search problem is a function of two factors: the number of nodes expanded, and the time spent per node. An inadmissible heuristic may be faster to compute, leading to a solution that is obtained faster due to less time spent per node. It can also be a closer estimate to the actual cost function (even though at times it will overestimate!), thus expanding less nodes. We lose the guarantee of optimality by using an inadmissible heuristic. But sometimes we may be okay with finding a suboptimal solution to a search problem.

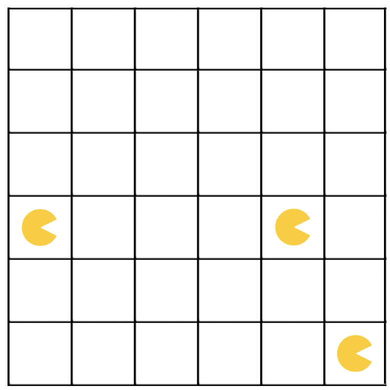
Q2. Pacfriends Unite

Pacman and his Pacfriends have decided to combine forces and go on the offensive, and are now chasing ghosts instead! In a grid of size M by N, Pacman and $P-1$ of his Pacfriends are moving around to collectively eliminate **all** of the ghosts in the grid by stepping on the same square as each of them. Moving onto the same square as a ghost will eliminate it from the grid, and move the Pacman into that square.

Every turn, Pacman and his Pacfriends may choose one of the following four actions: *left, right, up, down*, but may not collide with each other. In other words, any action that would result in two or more Pacmen occupying the same square will result in no movement for either Pacman or the Pacfriends. Additionally, Pacman and his Pacfriends are **indistinguishable** from each other. There are a total of G ghosts, which are indistinguishable from each other, and cannot move.

Treating this as a search problem, we consider each configuration of the grid to be a state, and the goal state to be the configuration where **all** of the ghosts have been eliminated from the board. Below is an example starting state, as well as an example goal state:


(a) Possible Start State


(b) Possible Goal State

Assume each of the following subparts are **independent** from each other. **Also assume that regardless of how many Pacmen move in one turn, the total cost of moving is still 1.**

(a) Suppose that Pacman has no Pacfriends, so $P = 1$.

(i) What is the size of the minimal state space representation given this condition? Recall that $P = 1$.

- ○ MN
- ○ MNG
- ○ $(MN)^G$
- ○ $(MN)^{G+1}$
- ○ 2^{MN}
- ○ 2^{MN+G}
- ○ $G(2)^{MN}$
- ● $MN(2)^G$

Since $P = 1$, we only need to keep track of the position of Pacman, as well as whether each of the G ghosts has been eliminated. We can keep track of this with MN and 2^G respectively; therefore, the size of the minimal representation is the product of the two, which is $MN(2)^G$. Note that we do not need to keep track of the ghosts' positions since they are stationary.

For each of the following heuristics, select whether the heuristic is only admissible, only consistent, neither, or both. Recall that $P = 1$.
Note that consistency implies admissibility, so just 'consistent' will never be the correct answer.

(ii) $h(n)$ = the sum of the Manhattan distances from Pacman to every ghost.

- ○ only admissible
- ○ only consistent
- ● neither
- ○ both

In this case, we can consider a dummy example, such that all the ghosts are adjacent and lined up horizontally, and Pacman is one square to their left. If there are three ghosts, then the sum of their Manhattan distances from Pacman is $1 + 2 + 3 = 6$, which is an overestimate of the actual cost 3, since the Pacman only needs to move three spaces to actually eliminate all the ghosts.

(iii) $h(n)$ = the number of ghosts times the maximum Manhattan distance between Pacman and any of the ghosts.

○ only admissible
○ only consistent
● neither
○ both

Considering the same example as in the previous part, this heuristic also overestimates the cost, since the number of ghosts × Manhattan distance between Pacman and the furthest ghost is $3 \times 3 = 9 > 3$. This question was randomized to say 'minimum Manhattan distance' as well, but the answer is still neither. Consider the case where Pacman is very far from the closest ghost, but all the ghosts are next to each other. The heuristic will give us an overestimate to collect all the ghosts, because going the minimum distance value to each ghost is an overestimate once you've reached the closest ghost.

(iv) $h(n)$ = the number of remaining ghosts.

○ only admissible
○ only consistent
○ neither
● both

Without his Pacfriends, Pacman can only eliminate at most one ghost each turn, and therefore the number of ghosts will be a lower bound on the number of moves necessary to eliminate all the ghosts, making this heuristic admissible. Additionally, since between moves, the number of ghosts can only be reduced by at most one, the heuristic is also consistent.

(b) Suppose that Pacman has exactly one less Pacfriend than there are number of ghosts; therefore $P = G$. Recall that Pacman and his Pacfriends are indistinguishable from each other.

(i) What is the size of the minimal state space representation given this condition? Recall that $P = G$.

○ MNP
○ $MNGP$
○ $(MN)^G$
○ $(MN)^{(G+P)}$
○ $(MN)^P 2^G$
○ $(MN)^G P$
○ $(MN)^{G+1}$
○ $(MN)^{(G+1)P}$
○ $\binom{MN}{P}$
● $\binom{MN}{P} 2^G$
○ $\binom{MN}{P}(MN)^G$
○ $\binom{MN}{P}\binom{MN}{G}$
○ 2^{MN}
○ 2^{MN+G+P}
○ $GP(2)^{MN}$

Since P can be any number of Pacman and Pacfriends, arbitrarily equal to G, we need to keep track of the position of all the Pacmen, as well as whether each of the G ghosts has been eliminated; however, since the Pacmen are **indistinguishable** from each other, we can consider states where different Pacmen have swapped positions to be the same. Therefore, we can keep track of unique Pacmen configurations using $\binom{MN}{P}$, which is the number of ways to select P Pacman positions from a grid of M by N. Therefore, the size of the minimal representation is the product of this and 2^G, which is $\binom{MN}{P} 2^G$.

For each of the following heuristics, select whether the heuristic is only admissible, only consistent, neither, or both. Recall that $P = G$.

Note that consistency implies admissibility, so just 'consistent' will never be the correct answer.

(ii) $h(n)$ = the largest of the Manhattan distances between each Pacman and its closest ghost.

○ only admissible
○ only consistent
● neither
○ both

Consider the case where we have two Pacmen and two ghosts, and one Pacman is very close to both ghosts, while the other is very far away. This heuristic would overestimate the cost since it would take the distance from the furthest Pacman to the ghosts as its metric, when the closer Pacman could just eliminate both ghosts quicker.

(iii) $h(n)$ = the smallest of the Manhattan distances between each Pacman and its closest ghost.

- ◯ only admissible
- ◯ only consistent
- ◯ neither
- ● both

At minimum, it will take the smallest of the distances between each Pacman and its closest ghost for any of the ghosts to be eliminated. Therefore this heuristic will always underestimate the true cost and is therefore admissible. To see that this is consistent, we can say that the difference between $h(a) - h(b)$ for going from any state to the next state is an underestimate of the cost between states. In fact, consider the scenario with two Pacmen and two ghosts, where one ghost is next to one Pacman and far from the other, while the other ghost is far from both. The heuristic for this state will be 1, since one of the ghosts is adjacent to one of the Pacmen, but as soon as that Pacman eliminates that ghost, the subsequent state will have a much larger heuristic value, since the minimum Manhattan distance between either Pacman and the remaining ghost will be very high; even in this edge case, $h(A) - h(B)$ will be negative, meaning that it is still an underestimate of the cost between states. Therefore, the heuristic is consistent.

(iv) $h(n) =$ the number of remaining ghosts.

- ◯ only admissible
- ◯ only consistent
- ● neither
- ◯ both

Since multiple ghosts could be eliminated in one move because there is more than one Pacman eliminating them, this heuristic can overestimate the cost to the goal state, and is therefore neither consistent nor admissible.

(v) $h(n) = \frac{\text{number of remaining ghosts}}{P}$.

- ◯ only admissible
- ◯ only consistent
- ◯ neither
- ● both

Since we are given that $P = G$, the maximum value of this heuristic is 1, since $h_6(n)$ is at most G. Therefore this heuristic is admissible since it has a value always less than or equal to 1, the cost of taking a turn, and when there are no remaining ghosts left, the heuristic evaluates to 0, so it will underestimate the cost to the goal state. It is also consistent since the heuristic value decreases fractionally in increments less than 1, and so it always underestimates the cost between two states.