A Decision-Making Agent for Open-Water Swimming

Hugo Kitano
Department of Computer Science
Stanford University
hkitano@stanford.edu

Taide Ding
Department of Computer Science
Stanford University
tding@cs.stanford.edu

Abstract

Open-water swimmers must deal with difficulties such as unpredictable currents, cold waters, and hostile wildlife in order to reach their destinations. Often, the water conditions are hazardous enough that they would be better off not attempting to swim there at all. Thus, for our final project, we implemented a decision-making agent that finds a feasible or optimal path for an open-water swim. Using a discretized gridworld that maps out the open water, we first used value iteration over a fully observed MDP to find an optimal path for the swimmer. Then, we added randomness to the transition functions and introduced partial observability for the state space to convert the problem into a POMDP, and solved for a feasible, approximately optimal policy using fast informed bound. In the future, such methods may be transferable to similar path-finding situations over uncertain, hazardous terrain.

1 Introduction

Open-water swimming is an inherently dangerous sport. Swimmers brave a variety of hazards, including cold waters, rough waves, shifting currents, and unfriendly marine life, to reach their destinations. Swimming alone in the open water is a particularly dangerous endeavor — without any assistance from a fellow swimmer or a boat, an unassisted swimmer can easily become disoriented and/or misjudge their ability and endurance to complete a swim. If a lone swimmer turns back for shore too late or gets caught in a particularly strong rip current, there is little hope of survival. Even open waters in densely populated areas can be dangerous - for example, Ocean Beach in San Francisco is notorious for its freezing waters and strong currents: eight swimmers drowned there between 2014 and 2018.[1]

Thus, we aimed to implement a decision-making agent to make solo open-water swimming safer by planning the swimmer's course in uncertain water conditions. The agent would advise a swimmer of whether swimming from point A to point B is feasible or not, given the swimmer's energy levels; if it is, the agent would propose an optimal or feasible route for the swimmer to take. Such a system would be generalizable to other inherently life-threatening endeavors, such as mountaineering, spelunking, diving, sailing, and manned space exploration.

Pathfinding for robots using reinforcement learning and POMDP's is an open-research problem [2] [3]. Our project is in the same vein, though attacks a more specific problem than the more general pathfinding algorithms described in these papers.

[1] https://www.sfgate.com/bayarea/article/California-s-deadliest-beach-is-in-the-Bay-Area-13245629.php

[2] Meuleau, Nicolas & Plaunt, Christian & Smith, David & Smith, Tristan. (2010). A POMDP for Optimal Motion Planning with Uncertain Dynamics

[3] Lauri, Mikko & Ritala, Risto. (2015). Planning for robotic exploration based on forward simulation. Robotics and Autonomous Systems. 83. 10.1016/j.robot.2016.06.008

2 Modeling Approach

The state of the swimmer was modeled as a finite health bar that describes their energy, along with other statistics (e.g. how fast they can swim). Our model consists of a two-dimensional gridworld that maps over the open water. Each node of our grid is a discretized region of water, which contains information regarding water temperature, direction and strength of current, and chance of encountering dangerous wildlife. Given a start point and an end point, the agent maps out an path between nodes for our swimmer to take. Transitioning between nodes incurs some variable cost depending on conditions at that node: for instance, if currents are strong, the swimmer will expend more energy if they swim against the current, and if currents are faster than the swimmer's swimming speed, they will likely not end up in the direction they are aiming for.

We consider two cases of transition and observation probabilities:

1. In the fully observed case, the current direction and water temperature are fully observed for each state in the grid, and the swimmer knows their location in the water with 100% certainty at all times. Moreover, transitions depend deterministically on the observed current direction at each state, the action chosen by the agent, and the power output of the swimmer. This problem can be formulated as an MDP, and value iteration can be used to solve for the optimal policy and path. [4]
2. In the partially observed case, the farther out from shore (either the start or the end state), the less certain the swimmer is about their exact location, since the swimmer's orienteering becomes increasingly less accurate. The transitions are also stochastic to reflect shifting, unpredictable currents: the prevailing current direction and swimming direction merely informs the weighting of the transition probabilities from a state to each of the neighboring states. Since we would like the agent to inform a swimmer of whether a route is feasible prior to the swimmer actually getting into the water to attempt it, we solve this POMDP offline, using the fast informed bound approach. [5]

The gridworld is composed of states S with the following attributes: current velocity $V_s \in \mathbb{R}^2$, water temperature $t_s \in \mathbb{R}$, $isStart \in \{0, 1\}$, $isEnd \in \{0, 1\}$, and $isShark \in [0, 1]$. The swimmer has the following attributes: body temperature t_b, energy store E, power p, and insulation factor k.

The action space is $A \in \{-1, 0, 1\}^2$ for all states S. The reward at each state is a function of the water temperature t_s, the body temperature t_b, the action A, the current velocity V_s, and whether it is the final end state such that:

$$R(s, a) = \frac{1}{k}(t_s - t_b) + (A^T V_s) + isShark * R_{shark} + isEnd * R_{end} \tag{1}$$

with $R_{shark} << 0$ and $R_{end} >> 0$. This reward model accounts for the negative impact of colder water on the swimmer's performance that is attenuated by increased insulation (e.g. by wearing a wetsuit). The reward model also accounts for the increased energy expenditure caused by swimming against the current (i.e. when $A^T V < 0$).

3 Fully Observed Problem with Deterministic Transitions

In the fully observed case, the transition at each state is deterministic. We calculate the resultant of the current vector and the swimmer's action vector (multiplied by the swimmer's power p) and bin it to the most similar direction to a valid adjacent state $s' \in Adj(s)$, by calculating the neighboring state with the smallest norm as such:

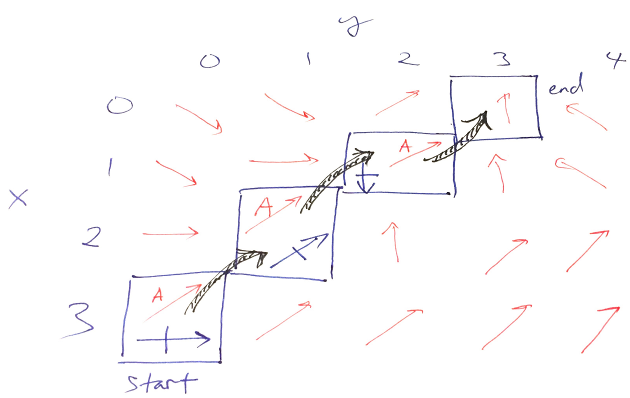
$$T(s' \mid s, a) = \begin{cases} 1 & s' = \arg\min_{s' \in Adj(s)} \| \frac{V_s + pA}{\|V_s + pA\|} - \frac{s' - s}{\|s' - s\|} \| \\ 0 & o.w. \end{cases} \tag{2}$$

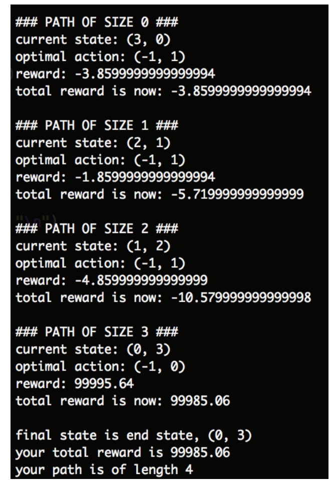
For example, if the swimmer is at a non-edge state $s = (2, 2)$, the current velocity is $V_s = (0, 3)$, the swimmer's action is $A = (1, 0)$, and the swimmer's power is $p = 3$, then $T((3, 3)|A, s) = 1$.

[4]Kochenderfer: Decision Making Under Uncertainty - Ch. 4, pg. 82

[5]Kochenderfer: Decision Making Under Uncertainty - Ch. 6, pg. 144

Value iteration with discount rate of 0.9 was run on a synthetic $(4, 5)$ gridworld, with $p > \|V_s\|$, i.e. the swimmer can outswim the current at each point. The optimal path was calculated as below:

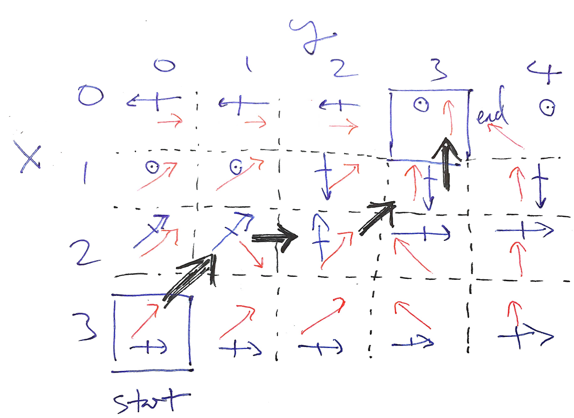


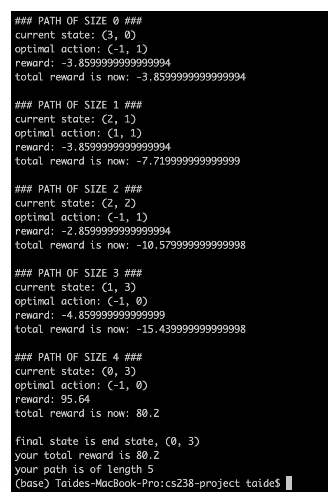


```
### PATH OF SIZE 0 ###
current state: (3, 0)
optimal action: (-1, 1)
reward: -3.8599999999999994
total reward is now: -3.8599999999999994

### PATH OF SIZE 1 ###
current state: (2, 1)
optimal action: (-1, 1)
reward: -1.8599999999999994
total reward is now: -5.719999999999999

### PATH OF SIZE 2 ###
current state: (1, 2)
optimal action: (-1, 1)
reward: -4.859999999999999
total reward is now: -10.579999999999998

### PATH OF SIZE 3 ###
current state: (0, 3)
optimal action: (-1, 0)
reward: 99995.64
total reward is now: 99985.06

final state is end state, (0, 3)
your total reward is 99985.06
your path is of length 4
```

Figure 1: Optimal policy and path for fully observed gridworld when the swimmer can outswim the current at each point. The collection of red arrows represents the optimal policy. The black arrows represent the optimal path. The blue arrows represent the current directions along the optimal path (all components are 0 or ± 1).

As shown above, if the swimmer can outswim the currents, the optimal path is a straight line, as expected. However in reality, strong currents (especially rip currents) can flow as fast as 2.5 m/s, which is substantially faster than any human can swim.[6] Thus, if the swimmer's power is dialed down such that they are unable to outswim the currents at certain states, the optimal path computed by value iteration adjusts to take advantage of favorable currents. Indeed, if the swimmer's power is reduced to a sufficient level, value iteration correctly calculates that no path exists and recommends that the swimmer not take the plunge.





```
### PATH OF SIZE 0 ###
current state: (3, 0)
optimal action: (-1, 1)
reward: -3.8599999999999994
total reward is now: -3.8599999999999994

### PATH OF SIZE 1 ###
current state: (2, 1)
optimal action: (1, 1)
reward: -3.8599999999999994
total reward is now: -7.719999999999999

### PATH OF SIZE 2 ###
current state: (2, 2)
optimal action: (-1, 1)
reward: -2.8599999999999994
total reward is now: -10.579999999999998

### PATH OF SIZE 3 ###
current state: (1, 3)
optimal action: (-1, 0)
reward: -4.859999999999999
total reward is now: -15.439999999999998

### PATH OF SIZE 4 ###
current state: (0, 3)
optimal action: (-1, 0)
reward: 95.64
total reward is now: 80.2

final state is end state, (0, 3)
your total reward is 80.2
your path is of length 5
(base) Taides-MacBook-Pro:cs238-project taide$
```

Figure 2: Optimal policy and path for fully observed gridworld where maximum current velocities exceed the swimmer's maximum velocity. The collection of red arrows represents the optimal policy. The black arrows represent the optimal path. The blue arrows represent the current directions along the optimal path. Note how the agent routes the optimal path away from state $(1, 2)$ to avoid the unfavorable current direction.

[6]For comparison, the world record for the 1500m swim (set in the controlled environment of an Olympic swimming pool) is 14:31, for an average speed of just over 1.7 m/s.

4 Partially Observed Problem with Stochastic Transitions

In the partially observed case, instead of transitions being calculated deterministically using the resultant of the current vector and the swimmer's action vector, we model shifting, unpredictable currents by considering the directions with the k smallest norms with respect to the resultant of the current and action vectors, and taking the softmax of the negatives of those k norms as the transition probabilities. In other words, the next state is chosen from the k most likely next states, such that next states whose directions from the starting state are more similar to the resultant direction of the current and action vectors are assigned higher probability.[7] The norms are calculated in the same way as in the fully observed case:

$$Norm(s' - s \mid V_s + pA) = \| \frac{V_s + pA}{\|V_s + pA\|} - \frac{s' - s}{\|s' - s\|} \| \tag{3}$$

In real life, an open-water swimmer will have increasing difficulty determining their precise location the farther away they are from land, due to the limits of eyesight and increased effects of visual obstructions such as fog. If an unaided swimmer is sufficiently far from land, they would effectively be unable to make any observations about their location since there would be no visible land with which to orient themselves. Instead, the swimmer would have to rely on their prior observations to construct their belief of where exactly they are. We model this uncertainty with the following observation probabilities:

$$O(o \mid s', a) = \begin{cases} \max[0, 1 - 0.2D] & o = s' \\ 0 & o.w. \end{cases} \tag{4}$$

such that

$$D = \min_{s \in \{s_{start}, s_{end}\}} (\|s - s'\|) \tag{5}$$

In other words, the probability that the swimmer knows exactly where they are is 100% at the start and end points, and decreases as the swimmer is farther away from both.

At each step, we decrement the swimmer's energy level E by the difference between the water temperature t_s and the swimmer's body temperature t_b (attenuated by increased insulation) and by the dot product of the action vector and the current vector (to account for increased energy expenditure when swimming against the current).

$$E^{t+1} = E^t - (\frac{1}{k}(t_b - t_s) - (A^T V_s)) \tag{6}$$

If the energy level drops below 0 at any point in the calculated path, the agent informs the user that the swim is unfeasible.

Because we want the agent to take partial observations into account when choosing its next action, we implemented the **fast informed bound** method for updating our alpha vectors. As shown in Kochenderfer's Decision Making Under Uncertainty, the $k+1$'th update for an alpha vector for a certain action a at state s is

$$\alpha_a^{(k+1)}(s) = R(s, a) + \gamma \sum_o \max_{a'} \sum_{s'} O(o \mid s', a) T(s' \mid s, a) \alpha_{a'}^{(k)}(s') \tag{7}$$

The POMDP solver was run on the same $(4, 5)$ gridworld as in the Fully Observed case.

[7] $k = 4$ was used for this implementation

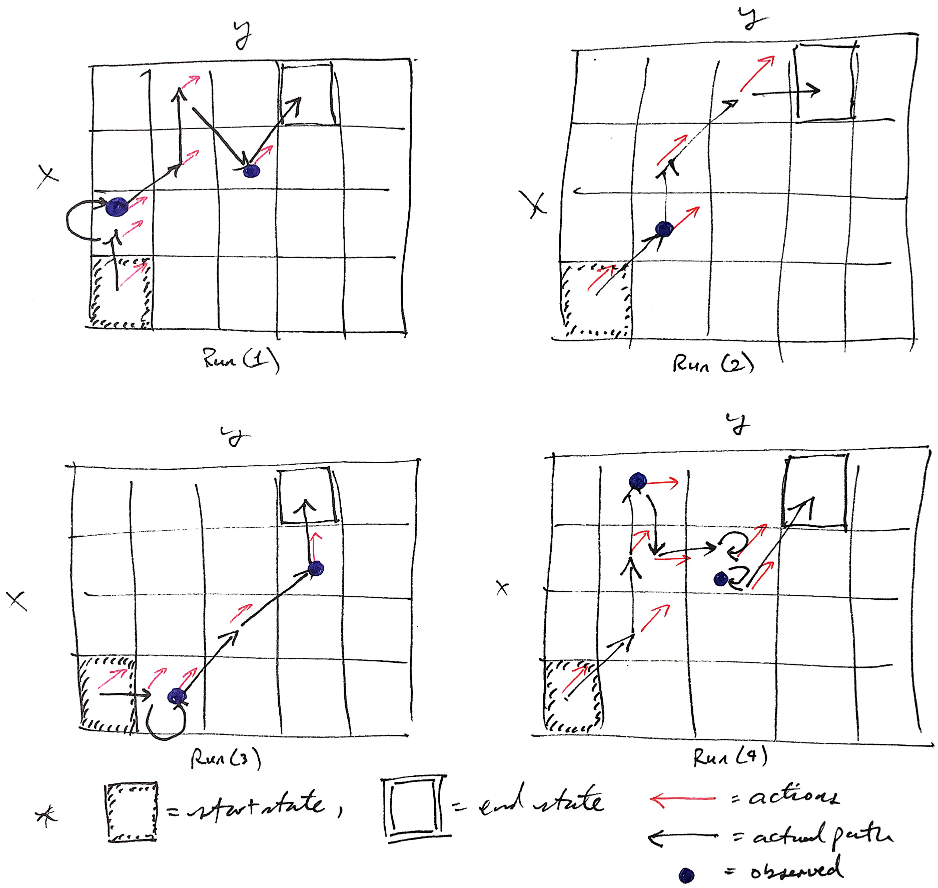


Figure 3: Four runs of the POMDP agent using fast informed bound, showing the variation in calculated paths due to the stochasticity of transitions. The red arrows mark the actions taken, the black arrows mark the path traversed, and the black circles denote observations of a state.

The stochastic transitions and state uncertainty make for more complex and realistic paths. Run (4) in particular was stymied by currents at the penultimate state, an altogether probable occurrence in real life because of rip currents. [8]

The belief vectors for part of Run (1) are printed below, showing the probability distribution across the possible locations.

[8]Compare this to the trajectories of the four crossings of the English Channel by American swimmer Sarah Thomas: https://www.npr.org/2019/09/17/761511898/american-becomes-1st-person-to-swim-english-channel-four-times-without-stopping

```
observed state: (2, 0)
### PATH OF SIZE 2 ###
current belief:
[[0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0.]]
your energy is now 88.28
optimal action: (-1, 1)
next state not observed
### PATH OF SIZE 3 ###
current belief:
[[0.         0.         0.         0.         0.        ]
 [0.18058373 0.38821581 0.         0.         0.        ]
 [0.19042215 0.2407783  0.         0.         0.        ]
 [0.         0.         0.         0.         0.        ]]
your energy is now 82.42
optimal action: (-1, 1)
next state not observed
### PATH OF SIZE 4 ###
current belief:
[[0.02135417 0.13772048 0.19737942 0.         0.        ]
 [0.05910862 0.2018103  0.21423186 0.         0.        ]
 [0.03561653 0.09007028 0.04270834 0.         0.        ]
 [0.         0.         0.         0.         0.        ]]
your energy is now 78.56
optimal action: (-1, 1)
```

Figure 4: Belief vectors, optimal actions, observations, and energy expenditures for steps 2 through 4 of Run (1). At step 3, the agent's true location was $(1, 1)$, while at step 4, the agent's true location was $(0, 1)$.

5 Conclusions

In this study, we demonstrated a decision-making agent that is able to guide a swimmer through both fully observed and partially observed open-water conditions. The agent is able to take account of current velocities to plan non-direct but energy-efficient trajectories. The agent is also able to determine whether the path is feasible for the swimmer given their energy level by tracking the fatiguing effects of cold water and swimming against the current: such an agent offers a very useful failsafe for open-water swimmers, preventing them from misjudging their abilities and running out of energy in the middle of the ocean, particularly when they are disoriented and unsure of their precise location.

Further steps could include a more interactive path visualization component, adding more attributes to the swimmer and gridworld, and developing even more realistic reward, transition, and observation functions that account for the myriad complexities of the open-water environment. Alternatively, the model can be generalized to work for other types of similar path-finding problems.

This type of open-world exploration problem with high risks associated with failure not only applies to open-water swimming, but to a variety of situations where the agent is in unfamiliar terrain affected by stochastic processes that affect their survival, such as mountaineering, spelunking, diving, sailing, and manned space exploration. The models we implemented would transfer well to other problems that benefit from planning a course of travel both before the journey is begun, and while the journey is in progress.

6 Group Member Contributions

Hugo Kitano: implemented value iteration for MDP, formulated partially observed gridworld dynamics, implemented POMDP solver and pathfinder, drafted and edited final paper

Taide Ding: proposed initial project idea, formulated and implemented fully observed gridworld dynamics, transition probabilities, and reward function; drafted and edited final paper