Determining an optimal walking pattern for a unicellular walker

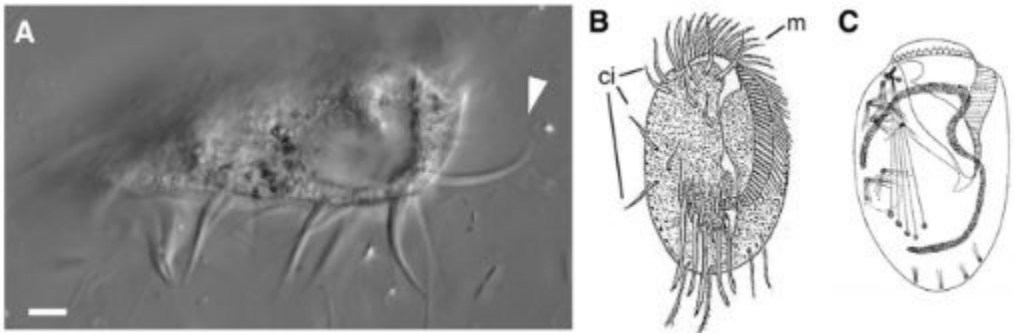
Eric Carson

Agenda

1. Background
2. Goals
3. Model Definition
4. Methodology
5. Analysis
6. Results/Demo
7. Conclusion/Next Steps

Background

- Research project with Professor Julius and Ben Larson (Berkeley)
- Single celled organism Euplotes eurystomus has 14 leg-like appendages called cirri controlled by microtubules
- Previous research determined cirri movement could be represented as a finite state machine



Project Goal

- Find a sequence of cirri movements which maximizes distance travelled in the x-direction
- A Matlab simulator developed by Larson currently generates the next states of each leg and calculates the distance travelled as a result of the changes

Model Definition - State Space

- Note: the organism is assumed to be moving in a substrate which the cirri can attach to
- Individual leg states: 0 (attached to substrate, not moving) or 1 (free from substrate and moving to starting position)
- State space: 0 to 2^14-1 to represent all possible positions of legs at any given time
  - Ex: [0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1] -> b01100100001011 -> 6411

Action Space & Transition Model

Assumptions made:

- A leg movement is attempted is successful => P(switch)=1
- One leg moves at a time

Action space: [a_1, …, a_{14}] correspond to switching the state of an individual leg

Transition model: Modifies the leg corresponding to the action

- Ex: if s=6411 => [0, 1, 1, **0**, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1] and a=leg 4,
  s’ = [0, 1, 1, **1**, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1] => 7435

Reward Model

- Reward model: magnitude of the distance travelled in the x-direction
- Note: a single step out of context will not give an accurate representation of the distance travelled (note: a leg movement may not move the organism but instead set up for a bigger movement)
- Input: current sequence of leg movements
- Output: total distance travelled as a result of those leg movements

Methodology

- Starting state: all 14 cirri will be attached to the substrate (state 0)
- The simulation will be run for 15 iterations (steps), each iteration focuses on a specific leg and update the state using the transition model
- Monte Carlo Tree search was be the learning method used since the number of iterations would be equivalent to the depth parameter and good to use for exploring different cirri movements (exploration vs exploitation)

MCTS Modifications

- Current “best path” given to mcts as a parameter instead of a predefined reward model
  - Reason: unreasonably computationally expensive to preemptively determine all possible distances travelled (at that point this is a brute force problem)
- When calculating R(s’), the potential new state temporarily added to the current sequence of leg movements to determine the reward

MCTS Analysis

- At times, the reward value for determining which leg to move next was often a tie, meaning the lowest number leg which had the maximum reward would move in that case
- Result: leg movements would mainly use legs 1-7 only
- Fix: introduce randomness when there is a tie to make the leg movements more realistic and evenly distributed

MCTS Analysis

- A depth of 4 is a good balance of computation cost vs good results
- MCTS ran with a depth value of 1-10 for 15 steps, after a depth of 4, the maximum distance did not change significantly

```
Depth: 1, Distance: 0.607274099662532
Depth: 2, Distance: 0.2133298526129473
Depth: 3, Distance: -0.02739009004908355
Depth: 4, Distance: 1.5688058799017406
Depth: 5, Distance: 1.5741929361730238
Depth: 6, Distance: 1.6143900553268453
Depth: 7, Distance: 1.7136427276961492
Depth: 8, Distance: 1.568440121361488
Depth: 9, Distance: 1.605148460225943
Depth: 10, Distance: 1.6942961499919185
```

Results/Demo

WARNING: IF YOU DO NOT LIKE BUGS LOOK AWAY!!!

Conclusion and Next Steps

- Prediction for optimal movement: the general optimal pattern will be to move the cirri from a fixed to free state from front to back order
- Generally, the prediction was correct
- Extensions of the model:
  - Allow multiple legs to move at once
  - Redesign reward model to allow for faster computation time