Quality Diversity for Improving Agent Design
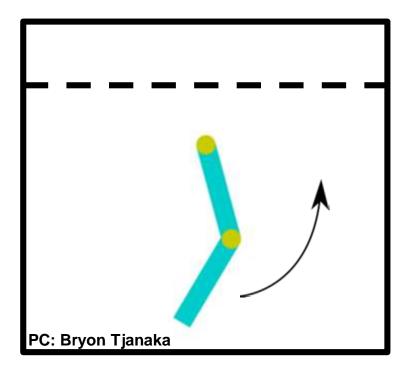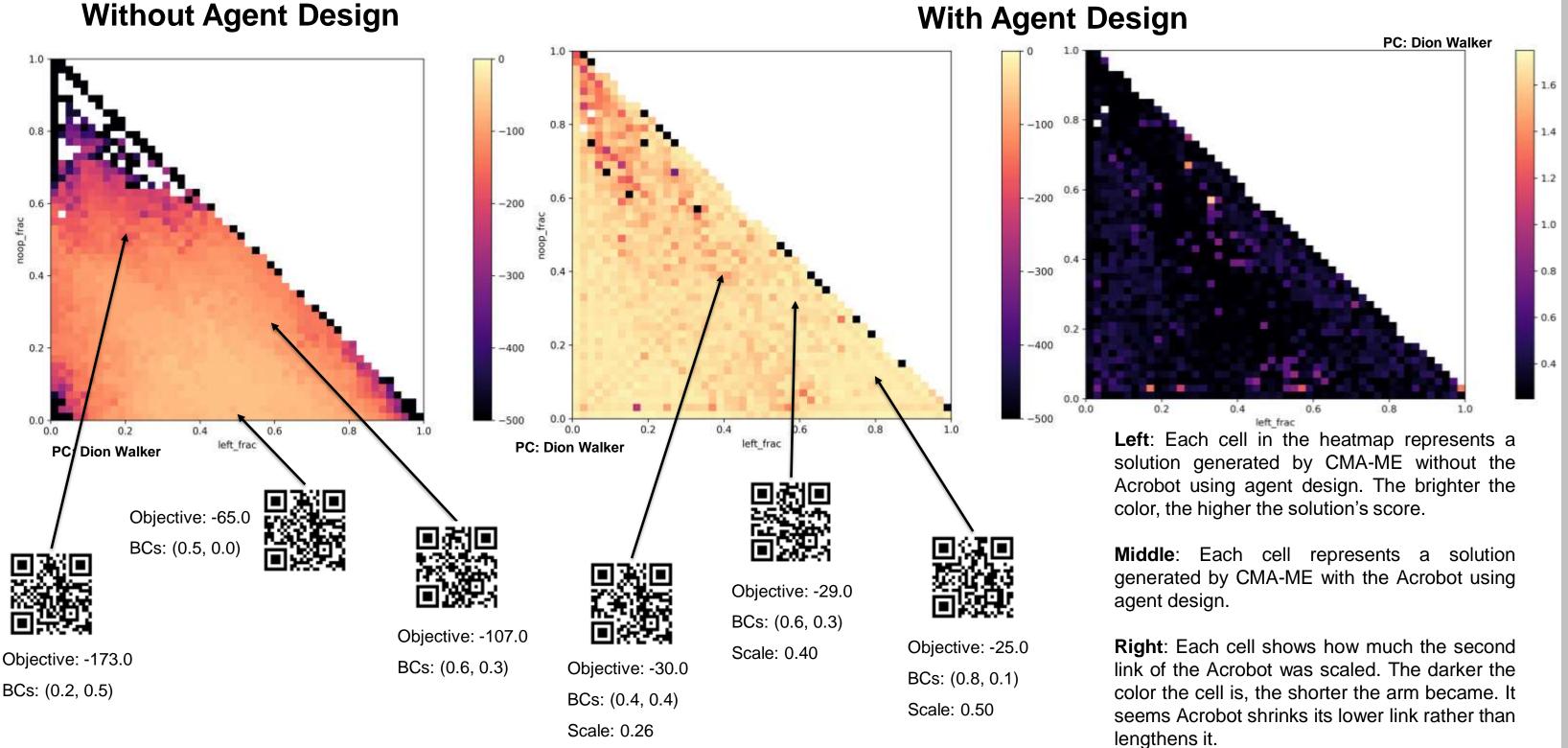
Dion Walker, raymondwalker2292@gmail.com
ICAROS Lab
Great Oak High School, Class of 2024
USC Viterbi Department of Computer Science, SHINE 2021

Introduction

Quality diversity (QD) is a form of evolutionary computation with a focus on finding a diverse set of high-performing solutions to a problem. In this project, we apply QD to the Acrobot domain from OpenAI Gym, in which the robot tries to swing one of its links up to a given height. We also experiment with agent design, where we allow the Acrobot to change the length of its second link.

PC: Bryon Tjanaka

The goal of the Acrobot is to swing up to the dotted line. The Acrobot only controls the middle joint (the one connecting the two links).

Methods

Objective: The agent is scored based on how quickly it swings the arm up. The score starts at 0, and the agent loses 1 point per timestep. The agent has at most 500 timesteps to solve the problem, so -500 is the worst score possible.

Behavior Characteristics (BCs): BCs are properties of a solution. We use two BCs:
- Ratio of left actions to total actions (left_frac)
- Ratio of no-ops to total actions (noop_frac)

Policy: The agent uses a linear policy to determine how it will interact in the environment.

Algorithm: We use CMA-ME (Covariance Matrix Adaptation MAP-Elites) to generate high-performing policies with diverse values of the BCs. In the case of agent design, CMA-ME also generates a scale for the second link.

Results

Without Agent Design

PC: Dion Walker

Objective: -65.0
BCs: (0.5, 0.0)

Objective: -173.0
BCs: (0.2, 0.5)

Objective: -107.0
BCs: (0.6, 0.3)

With Agent Design

PC: Dion Walker

Objective: -29.0
BCs: (0.6, 0.3)
Scale: 0.40

Objective: -30.0
BCs: (0.4, 0.4)
Scale: 0.26

Objective: -25.0
BCs: (0.8, 0.1)
Scale: 0.50

PC: Dion Walker

Left: Each cell in the heatmap represents a solution generated by CMA-ME without the Acrobot using agent design. The brighter the color, the higher the solution's score.

Middle: Each cell represents a solution generated by CMA-ME with the Acrobot using agent design.

Right: Each cell shows how much the second link of the Acrobot was scaled. The darker the color the cell is, the shorter the arm became. It seems Acrobot shrinks its lower link rather than lengthens it.

Next Steps

Our next steps would be to allow the agent to design more of itself, e.g., lengthen and shorten the first link as well. We could also use a physics engine such as Box2D to ensure accurate dynamics. Finally, we could extend to other environments like the bipedal walker.

References

Fontaine, M. C., Togelius, J., Nikolaidis, S., & Hoover, A. K. (2020). Covariance matrix adaptation for the rapid illumination of behavior space. *Proceedings of the 2020 Genetic and Evolutionary Computation Conference.* https://doi.org/10.1145/3377930.3390232

Ha, D. (2019). Reinforcement Learning for Improving Agent Design. *Artificial Life*, *25*(4), 352–365. https://doi.org/10.1162/artl_a_00301

Skills and Tools

- Quality Diversity
- Reinforcement Learning
- Matrix Multiplication
- Bash/terminal
- GitHub
- VSCode
- Google Colab
- OpenAI Gym
- Autoformatters / linters

Acknowledgements