Task Sheet 3 - Local sampling and anti-agents

Objective

A. Implement a Buffon's needle simulator

B. Boost a robot swarm aggregation behavior with so-called 'anti-agents'

C. Measure performance in a swarm simulation

1 Buffon's needle

Implement Buffon's needle for needle length b = 0.7 and line spacing s = 1. Note that you do not have to simulate the whole setting in a big 2D plane. Instead you can reduce the situation to a single line segment of width s because the middle of the needle always ends up between two lines. What remains is simple geometry that determines whether the needle intersects with one of the two lines based on the needle's angle.

A. Use your simulation to determine the intersection probability P. Test your program by checking whether it estimates good approximations to π according to $\pi = 2b/(sP)$.

B. An experiment consists of n needles being thrown. Run experiments with $n \in \{10, 20, \ldots, 990, 1000\}$ trials. For each setting of n, run the experiment 10,000 times (i.e., 10,000$\times n$) and calculate the standard deviation of the resulting set of intersection probabilities. Plot the standard deviation over the number of trials n.

C. For a maxium of $n = 100$ trials, plot the currently measured probability of many experiments over the trial number n along with the Binomial proportion 95%-confidence interval.[1]

D. Measure the ratio of experiments for which the true probability[2] is outside of the the 95%-confidence interval based on the measured probability. Plot this ratio over the number of trials n. Interpret your results.

[1] from the lecture we know that the Binomial proportion 95%-confidence interval is defined by $\hat{P} \pm 1.96\sqrt{\frac{1}{n}\hat{P}(1-\hat{P})}$.

[2] from the lecture we know that the true probability is $P = 2b/(s\pi)$.

2 Anti-agents in swarm aggregation

An interesting approach is that of 'anti-agents' by **?** and **?**. The idea is to have a heterogeneous swarm with a minority of robots that behave basically inversely to the majority of robots. They use it for object clustering and observe an improvement in performance once a certain percentage of anti-agents is added. The standard approach to object clustering is probabilistic with a probability P_{pick} to pick up an object and P_{drop} to drop it. An anti-agent is a robot that behaves differently from the majority of robots. 'Reverse anti-agents invert the above probabilities to $1 - P_{pick}$ and $1 - P_{drop}$. **?** found a counterintuitive effect, namely that certain numbers of reverse anti-agents increase the observed clustering effect.

In this task, you can choose from two options, either (a) or (b), but always (c):

A. Implement object clustering (aka 'wood chip clustering,' robots pick up objects and cluster them) and add anti-agents in the above sense. You can also introduce pick-up and drop probabilities that depend on densities of objects (pick-up is more likely in sparse areas, drop is more likely in dense areas). This would then need to be inverted for the anti-agents as well.

B. Implement a swarm aggregation scenario (robots cluster, no objects) and add anti-agents that we define in the following way. Anti-agents move around and can order individual aggregated robots (via messaging) to leave a cluster. It seems useful to make that The 'leave' command is dependent on the cluster/neighborhood size.

C. Test different anti-agent percentages (preferably low percentages) and measure the performance. For example, you can define the performance as the biggest observed cluster after a defined time. You should do several repetitions for each setup to get statistically relevant measurements. Can you confirm the findings of **?** and **?**? It is tricky to reproduce their result. Try different parameter sets to find situations when anti-agents help.

2.1 Your submission

- Please zip your submission to a single file
- Along with the source code, there should be a README.md, with detailed information about key notes and compilation steps
- For task 1, provide written solutions and plots.
- For task 2, provide experiment videos(if possible), written solutions, and plots where applicable.

References

Daniel Merkle, Martin Middendorf, and Alexander Scheidler. Swarm controlled emergence-designing an anti-clustering ant system. In *2007 IEEE Swarm Intelligence Symposium*, pages 242–249. IEEE, 2007.

Alexander Scheidler, Daniel Merkle, and Martin Middendorf. Swarm controlled emergence for ant clustering. *International Journal of Intelligent Computing and Cybernetics*, 2013.