Collective Behaviour - Optimal Shepherding

First Report

| Franziska | Franz | Kimberley |
| Weber | Muszarsky | Frings |

November 19, 2023

1 Introduction

Shepherding behaviors are a class of flocking behaviors in which one or more agents (called *shepherds*) try to control the motion of another group of agents (called *flock*) by exerting repulsive forces. A real-life example are sheep dogs guiding flocks of sheep. *Herding* denotes a special type of shepherding behavior in which the shepherds attempt to steer the flock from a starting point to a target. [1]

In the context of the course *Collective Behavior*, we decided to investigate the problem of finding optimal herding strategies. To begin our work, we have chosen the paper titled *Optimal Shepherding and Transport of a Flock* [2] by A. Ranganathan, A. Heyde, A. Gupta, and L. Mahadevan as a starting point. This paper models herding as an optimization problem for the shepherd using an agent-based approach.

This initial report provides an overview of the present state of our project. We start by giving a concise summary of the relevant parts of our selected paper, followed by a description of our steps involved in executing the existing agent-based model (ABM). Subsequently, we present the current state of our GitHub-Repository and describe our strategy and our goals for the further course of this project.

2 Methods

Our initial focus centered on understanding and replicating the ABM presented in our selected paper. Therefore, we start by giving a concise summary of the relevant parts of this paper.

2.1 Paper Summary

The research paper *Optimal Shepherding and Transport of a Flock* explores the techniques that a shepherd can employ to effectively guide a group of animals towards a specific destination. The investigation utilizes an ABM to simulate the behavior of both the shepherd and the flock, with the primary goal of maintaining flock cohesion while achieving the desired movement.

The paper identifies three distinct herding strategies, namely *mustering*, *droving*, and *driving*. Mustering involves the shepherd circling the flock to keep it together, while droving entails the shepherd chasing the flock in the intended direction. Driving, on the other hand, involves the shepherd positioning themselves within the flock and guiding it from the inside. The paper delves

into an analysis of the efficiency of these three strategies. The efficiency of a strategy is measured with respect to the three goals (A) moving the mass of the herd to the desired location, (B) not loosing any sheep in the process, and (C) keeping the target and the herd in alignment.

The findings of the investigation suggest that the optimal herding strategy depends on just two parameters, namely the ratio of the herd size to the shepherd repulsion length and the ratio of herd speed to shepherd speed. The paper comes to the conclusion that droving is the most efficient strategy for managing small herds, while driving proves to be the most effective approach for cohesive, tightly-knit herds. Also, as the shepherd speed increases compared to the herd speed, the optimal strategy changes from droving to mustering.

Next, we executed the existing implementation of the ABM shared by our selected paper's authors in their GitHub repository[1]. In the following subsection, we provide a detailed description of our procedure involved in running the model, along with the necessary steps to generate a plot and a video of the simulation results. All the following commands and actions were performed in a Linux operating system environment (Ubuntu 22.04.3).

2.2 ABM recreation

2.2.1 Running the simulation

The code of the ABM implementation can be found in the directory `ABM_code`. The main file used for running the simulation is `simulate.cc`. The file `params.txt` contains the parameters for the simulation (e.g., amount of steps, speed of the herd). Since for the moment we are just focusing on recreating the model, we did not adjust any of these parameters yet.

During the simulation process, the resulting data is recorded in two distinct output files, namely `data.txt` and `costdata.txt`. The former contains the positions of every agent and of the shepherd at each timestep and is the main data file for the simulation. The latter stores the values of the objective function at each timestep which is useful for analysis and debugging.

We successfully downloaded the implementation and executed the simulation by following these straightforward steps:

```
git clone git@github.com:arphysics/optimal-shepherding.git
cd optimal-shepherding/ABM_code
cp ../paper_outputs/SI_ABM_videos/config.mk ..
# replace g++-9 with g++-11 in ../config.mk
make
./simulate
```

2.2.2 Creating a plot

To visualize the simulation results with a plot, we employed the file `trajectory_plotter.py` from the directory `ABM_code`. The resulting plot is saved as `output_plot.pdf`. We started by executing the following steps to run the plotting file:

```
python3 -m pip install numpy Pillow pyparsing matplotlib
python3 trajectory_plotter.py
```

This first attempt was unsuccessful as we encountered the following error:

---

[1]`https://github.com/arphysics/optimal-shepherding/tree/main/ABM_code`

```
ValueError: setting an array element with a sequence. The requested array
would exceed the maximum number of dimension of 1.
```

Upon careful inspection, we determined that the error stemmed from an attempt to modify two elements of a list simultaneously. It appears that this operation is not allowed when working with a Python list, but is feasible when using a NumPy array. Consequently, we adjusted the code at six specific locations in the way shown below.

Initial code   `xs[i] = [x[i], x[i] + unit_perp[0] ]`
Updated version   `xs[i] = np.array([x[i], x[i] + unit_perp[0] ]).flatten()`

This solved the issue and we obtained the plot shown in figure 1.
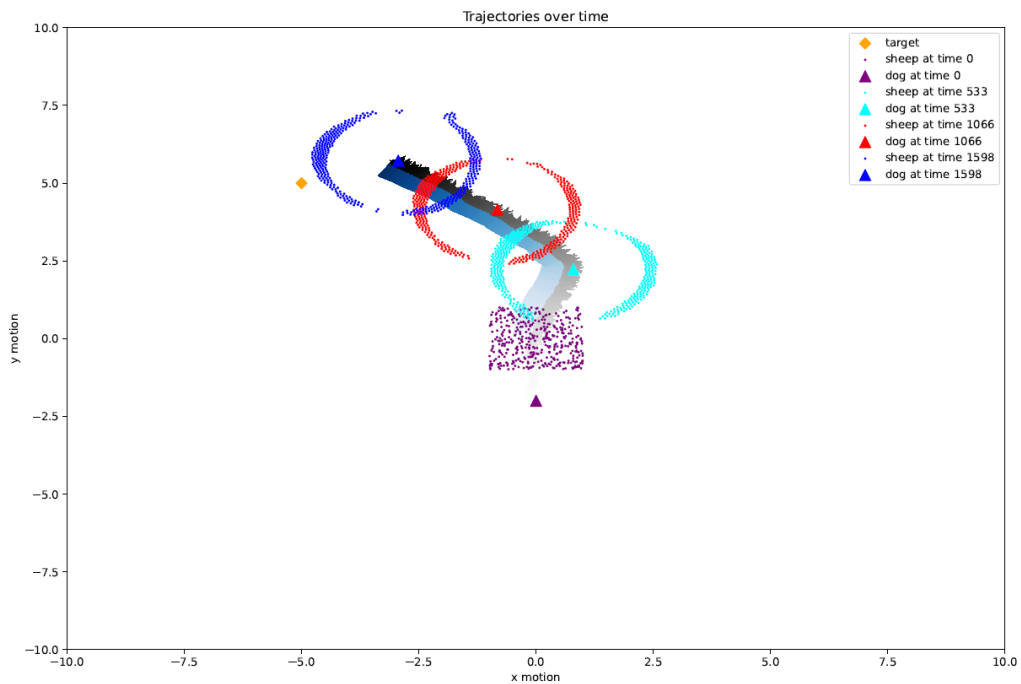

Figure 1: Trajectory of the agents and the shepherd over time

2.2.3   Creating a video

We also visualized the simulation results through a movie by using the file `newest_visualizer.py` from the directory `ABM_code`. The resulting video is saved as `output_movie_clone.mp4`. We performed the following steps to create a video of the results:

```
mkdir test_plots
sudo apt install -y ffmpeg
python3 newest_visualizer.py
```

3

We encountered the same error here as we did when creating the plot and were able to use the same solution to resolve it. The resulting video can be found in our GitHub repository[2].

3   Results

As described in the previous section, we have successfully run the existing implementation of the ABM and used it to visualize the simulation with a plot and a video. Afterwards, we identified the parts of the implementation that are relevant for our project and pushed them to our GitHub repository. While doing so, we also corrected the encountered errors as explained above and organized the code files in subfolders to make the structure of the implementation easier to understand. A more detailed description of our reduced version of the ABM implementation can be found in the README of our GitHub repository.

4   Discussion

In this report we have described the overall topic and the current state of our project about optimal shepherding strategies. We have understood the ABM presented in the paper *Optimal shepherding and transport of a flock* and plan to use it as starting point for our own investigation. We have already run the existing implementation of the ABM and identified the code files which are relevant for our project.

Our initial plan involved the idea of translating the original implementation from C++ to Python. We felt that Python, being a language we are more comfortable with, would facilitate modifying and adjusting the code. However, upon further examination and considering the feedback from fellow students, we have reconsidered this approach. We now believe that such a translation is not necessary, as we can reuse and extend the existing code with a manageable degree of effort. Additionally, the conversion to Python could potentially have a negative impact on the performance of the simulation.

For the kick-off, we did not yet have concrete ideas for extending our chosen paper. Now, as our comprehension of the ABM has improved, we have identified the scenario of having multiple shepherds as a a suitable extension of the paper's content. This scenario is highlighted as an open question for future study by the authors of our selected paper and was also proposed by fellow students.

Until the second report deadline, we will try to extend the ABM implementation such that it supports multiple shepherds. A possible goal for the final report is to investigate which optimal herding strategies emerge in the case of more than one shepherd.

References

[1] Jyh-Ming Lien, O. Burchan Bayazit, Ross T. Sowell, Samuel Rodríguez, and Nancy M. Amato. Shepherding behaviors, 2003.

[2] Aditya Ranganathan, Alexander Heyde, Anupam Gupta, and L. Mahadevan. Optimal shepherding and transport of a flock, 2022.

---

[2]`https://github.com/ki-mberley/Collective-Behaviour/tree/main`