Collective Behaviour Optimal Shepherding

Franziska Weber, Franz Muszarsky, and Kimberley Frings

Final Report

January 7, 2024

Iztok Lebar Bajec | izredni professor | mentor

Herding denotes a special type of so-called shepherding behaviours in which the shepherds try to steer to flock from a starting point to a target. We investigated the problem of finding optimal herding strategies by building upon an existing agent-based shepherding model. We extended this model by adding a surrounding fence to the environment and by considering the case where multiple shepherds are controlling the flock together. We also implemented an alternative algorithm for multiple shepherds and compared the performance of this algorithm to the one of our model. Our investigations revealed that in most cases the surrounding fence does not influence the shepherding process a lot and that the effect of introducing additional shepherds depends strongly on the behaviour of the flock.
Our model is publicly available at https://github.com/ki-mberley/Collective-Behaviour.

Shepherding behaviours 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 is sheepdogs guiding flocks of sheep. *Herding* denotes a special type of shepherding behaviour in which the shepherds attempt to steer the flock from a starting point to a target. [1]

In the context of the course *Collective Behaviour*, we decided to investigate the problem of finding optimal herding strategies. This problem has many engineering applications, such as environmental protection or crowd control [2].

Our work builds upon the paper titled *Optimal Shepherding and Transport of a Flock* [3] by A. Ranganathan, A. Heyde, A. Gupta, and L. Mahadevan. This paper models herding as an optimization problem for the shepherd using an agent-based approach. We enhanced the existing model by introducing two modifications: a surrounding fence and additional shepherds. We analyzed the effects of these extensions depending on different behaviours of the flock. Additionally, we compared the results of our model to the results of an existing shepherding algorithm from the literature.

Methods

Description of the original model. The herd consists of N agents which move in a two-dimensional open field. The behaviour of the agents is based on Reynolds' boids model [4]. To be more precise, the movement of each agent depends on three agent-agent interactions, namely local alignment, repulsion, and weak attraction to the herd center, and on the repulsion from the shepherd.

This leads to the following velocity field of an agent in the herd, where α, β, γ, and δ are weights:

$$\boldsymbol{v}^{\text{net}} = \alpha \boldsymbol{v}_{a-a}^{\text{alignment}} + \beta \boldsymbol{v}_{a-a}^{\text{repulsion}} + \gamma \boldsymbol{v}_{a-a}^{\text{attraction}} + \delta \boldsymbol{v}_{a-s}^{\text{repulsion}} \qquad [1]$$

Local alignment means that agents that are close to each other align their velocity vectors. We use the formulation from the Vicsek model [5]: At each timestep, the orientation of agent i is updated to be the sum of the average $\langle\theta\rangle$ of the orientations of the other agents within a certain interaction radius $r^{\text{alignment}}$ and a uniformly distributed noise $\eta \in [-\eta_0/2, \eta_0/2]$, i.e., $\theta^{\text{alignment}}(i) = \langle\theta\rangle_{r<r^{\text{alignment}}} + \eta$. $r^{\text{alignment}}$ is set to approximately ten times the agent size l_a. The local alignment term of agent i arises as the orientation of this agent multiplied by the agent speed v_a:

$$\boldsymbol{v}_{a-a}^{\text{alignment}}(i) = v_a \left(\cos\theta^{\text{alignment}}(i), \sin\theta^{\text{alignment}}(i)\right)$$

The repulsion between the agents is modeled as

$$\boldsymbol{v}_{a-a}^{\text{repulsion}}(i) = \sum_{i\neq j} \exp\left(-\frac{\|\boldsymbol{r_{ji}}\|}{l_a}\right) \frac{\boldsymbol{r_{ji}}}{\|\boldsymbol{r_{ji}}\|}$$

with $\boldsymbol{r_{ji}} = \boldsymbol{r_i} - \boldsymbol{r_j}$ where r_k denotes the position of agent k.

The attraction to the herd center quantifies the idea that agents intend to move to the middle of the herd to avoid being captured by predators. In our model, the attraction term is independent of an agent's distance to the herd center but only depends on the agents' speed and the polar angle $\phi(i) = \tan^{-1}\left(\frac{y_{\mathrm{cm}}-y_i}{x_{\mathrm{cm}}-x_i}\right)$ between the agent's position (x_i, y_i) and the herd's center of mass $\boldsymbol{r}_{\mathrm{cm}} = (x_{\mathrm{cm}}, y_{\mathrm{cm}})$:

$$\boldsymbol{v}_{a-a}^{\mathrm{attraction}}(i) = v_a\left(\cos\phi(i), \sin\phi(i)\right)$$

Lastly, the repulsion of an agent from the shepherd is modeled similarly to the repulsion between two agents:

$$\boldsymbol{v}_{a-s}^{\mathrm{repulsion}}(i) = \exp\left(\frac{-\|\boldsymbol{r_{si}}\|}{l_s}\right)\frac{\boldsymbol{r_{si}}}{\|\boldsymbol{r_{si}}\|}$$

with $\boldsymbol{r_{si}} = \boldsymbol{r_i} - \boldsymbol{r_s}$ where $\boldsymbol{r}_s$ is the position of the shepherd and $\boldsymbol{r}_i$ is the position of agent i. Based on observations of real-world shepherds, l_s was chosen as approximately 30 times l_a.

The behaviour of the shepherd is not predefined but arises from its goal to transport the entire herd to a certain target position. This goal leads to three conditions, namely (A) the shepherd should move the herd's center of mass to the target, (B) the shepherd may not lose any agents in the process, and (C) the shepherd should keep target and herd in alignment to maintain the line of sight.

These three transport requirements are weighted with $W_{\mathrm{mean}}, W_{\mathrm{std}}$ and W_{col} respectively and linearly combined into an objective function for the shepherd:

$$C(\boldsymbol{r_s}) = W_{\mathrm{mean}}|\boldsymbol{\Delta r}| + W_{\mathrm{std}}\sigma_{r_{\mathrm{cm}}} + W_{\mathrm{col}}|\boldsymbol{\Delta R}_{\mathrm{col}}| \quad [2]$$

The importance of transporting the herd to the target is represented by $|\boldsymbol{\Delta r}| = |\boldsymbol{r}_{\mathrm{target}} - r_{\mathrm{cm}}|$, where $\boldsymbol{r}_{\mathrm{target}}$ is the position of the target. $\sigma_{r_{\mathrm{cm}}} = \left(\frac{\sum_i (r_i - r_{\mathrm{cm}})^4}{N}\right)^{1/4}$ models the objective of keeping the herd cohesive and not losing any agents. The advantage gained from keeping the flock within the line of sight of the shepherd is represented by $\boldsymbol{\Delta R}_{\mathrm{col}} = \boldsymbol{r_s} + l_s\frac{\boldsymbol{r}_{\text{cm - target}}}{\|\boldsymbol{r}_{\text{cm - target}}\|}$ where $\boldsymbol{r}_{\text{cm - target}} = \boldsymbol{r}_{\mathrm{target}} - \boldsymbol{r}_{\mathrm{cm}}$.

The actual simulation is based on a forward Euler scheme implemented in C++. At each timestep, the positions of all agents are updated based on formula 1. For the shepherd, several directions are randomly sampled from the uniform distribution on $[0, 2\pi)$ and the direction corresponding to the minimal value of the objective function is chosen.

Implementation of the surrounding fence. The original implementation provided by the paper's authors included a code base featuring a fence implementation, specifically a function for calculating the repulsion force exerted by a fence on a sheep. However, due to the lack of explanation in the paper regarding the interpretation of the fence and the difficulty of understanding the author's implementation solely through code inspection, we decided not to use their code. Instead, we opted for a basic fence implementation, leaving room for potential future extensions such as incorporating an actual repulsion force from the fence.

We introduced minimum and maximum x- and y-coordinates, defining the boundaries of the surrounding fence. When calculating the next step for a sheep or dog, if the computed value exceeds the established fence bounds, we adjust the next step's value to the corresponding minimum or maximum x- or y-coordinate, preventing the sheep or dog from crossing the fence. This modification was integrated into the parameters file (`params.txt`) to accommodate fence specifications as input. Additionally, we included a condition in the `timestepping.hh` file to update the next step in the presence of a fence. As a final step, we ensure that a fence is shown in the plot by incorporating the necessary code in `trajectory_plotter.py` and `visualizer.py`.

Implementation of multiple shepherds. We began our study by exploring existing literature that showcased the use of multiple dogs for shepherding [6, 7]. Through this research, we found that in order to adapt the model for multiple shepherds, we needed to make two modifications. Firstly, we aimed to include a mechanism that discourages shepherds from getting too close to each other. Secondly, we wanted to adjust the way sheep and shepherds repel each other, ensuring it considered the sum of the repulsions between sheep and each individual shepherd. Although the latter was already part of the original paper's implementation, it was not functioning correctly, and we corrected the code to ensure its proper operation.

To prevent the shepherds from coming too close to each other, we added an additional term to the cost function which models the proximity of one shepherd to the other ones. We defined the proximity as the inverse of the distance, so two shepherds that are very close to one another are penalized more heavily. The proximity penalty for one shepherd is the sum of the proximities of all other shepherds. We introduced a parameter named `shepherd_distance_penalty` which is used to balance the proximity penalty with the other terms in the cost function.

With these adjustments in place, we successfully implemented a basic version of the model with multiple shepherds.

Implementation of an agent-based shepherding model as a basis for comparison. To test whether our implementation and especially our extensions of the model were working well, we wanted to compare its performance to the one of an algorithm from the literature. During our research on existing multiple-shepherd algorithms, we determined the agent-based model from the paper *Simulating Single and Multiple Sheepdogs Guidance of a Sheep Swarm* [2] to be the most suitable for our purposes.

In this paper, the shepherds are modeled as agents that act based on the goals of keeping the herd cohesive and guiding it towards the target. Additionally, each shepherd is at the same time attracted to the center of mass of the other shepherds and repelled from other shepherds if they are coming too close.

Because the paper did not provide an implementation, we recreated a basic version of the algorithm ourselves. As the sheep model in [2] is very similar to the sheep model from our chosen paper [3], we reused our existing sheep model and only adapted the model of the shepherd.

Results

Results from the original paper. The original paper [3] identified three different emerging herding strategies, namely *driving*, *droving*, and *mustering*. 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 comes to the conclusion that the optimal herding strategy depends on just two parameters: the ratio of the herd size to the shepherd repulsion length and the ratio of herd speed to shepherd speed. We managed to recreate these three types of shepherding behaviours in our experiments. The exact parameter values that we used to make each of the three strategies emerge can be found in our GitHub repository.

Introduction of a surrounding fence. The introduction of a surrounding fence ensures that both the dogs and sheep are surrounded by a boundary and prevents them from crossing it. In Figures 1 and 2, the outcomes are depicted for the herding style driving without and with a fence. As intended, the presence of the fence confines the sheep and dog within its boundaries, while the dog still leads the sheep to the target.

Introduction of multiple shepherds. We analyzed the impact of the number of shepherds and of the parameter `shepherd_distance_penalty` on the duration of the herding process. For this analysis, we removed the randomness from our model by setting a fixed random seed.

In the case of driving, a single dog needed 35240 timesteps to successfully complete the herding process. The fastest result that we were able to achieve in the two-dog scenario was 38102 timesteps for `shepherd_distance_penalty` = 1. This parameter value leads to one dog mainly staying at the center of the herd and the other one mainly staying outside of the herd without influencing it much. With three dogs the number of timesteps reduced to 27260 for `shepherd_distance_penalty` = 1. Again, one dog mostly stayed at the center of the herd while the other two dogs stayed outside of it.

In the case of droving, a single dog needed 1568 timesteps for a successful completion of the herding process. Introducing a second dog with `shepherd_distance_penalty` = 0.01 reduced the duration to 854 timesteps. The two dogs collaborate and drove the herd together towards the target. The droving behaviours for one and two dogs are depicted in Figures 3 and 4. With three dogs the number of timesteps reduced even further to 491 for `shepherd_distance_penalty` = 0.001. Again, the three dogs cooperated and drove the herd together.

In the case of mustering, a single dog needed 9002 timesteps to lead the herd to the target. In the two-dog scenario with `shepherd_distance_penalty` = 0.01, the duration reduced to 2178 timesteps. The two dogs cooperated and used a mix of droving and mustering. The introduction of a third dog, on the other hand, led to a significant increase of the duration to 41962 timesteps for `shepherd_distance_penalty` = 0.001. The three dogs do not really cooperate but all stay at the center of the herd and drive it towards the target.

Comparison to the agent-based shepherding model. For the agent-based shepherding model, we again studied the driving, the droving, and the mustering scenario with one, two, and three dogs. In all cases, the herding task was successfully completed.

Interestingly, the duration of the herding did not seem to depend as much on the shepherding behaviour as it was the case for our objective-function-based model.

In the case of driving, the agent-based model needed 28123 timesteps with one, 24322 with two, and 28186 with three dogs. In the case of droving, it took 27749 timesteps with one, 34581 with two, and 33421 timesteps with three dogs. Lastly, in the case of mustering, 24986 timesteps were required with one, 25929 timesteps with two, and 28861 timesteps with three dogs.

Discussion

Introduction of a fence. Our hypothesis was that the fence could be used by the shepherd to control the flock and keep the sheep close together more easily. However, our experiments did show such behaviour only for very specific parameter choices. The impact of the fence on the herding strategies of the shepherd remains to be investigated in more detail in future work.

Introduction of multiple shepherds. The effect of the introduction of one or multiple additional shepherds differed strongly for the three shepherding scenarios. In the case of driving the dogs did not collaborate and only one of the dogs worked on leading the herd towards the target. Depending on the value of `shepherd_distance_ penalty`, the remaining dogs either ran around and disturbed the herd or stayed outside of the herd and did not influence it much. In the droving scenario, on the other hand, the dogs worked together and the introduction of additional dogs sped up the herding process. Lastly, in the case of mustering, two dogs were able to guide the herd more efficiently than a single dog. The introduction of a third shepherd, however, led to the disappearance of this cooperation and to an increase in the duration of the shepherding process.

Our results might indicate that the usefulness of having more than one shepherd depends on the behaviour of the agents in the flock. Of course, additional experiments are necessary to confirm this hypothesis.

Comparison to the agent-based shepherding model. The agent-based shepherding model completed the herding more quickly in the case of driving with one or two dogs and in the case of mustering with three dogs. In all other cases, our objective-function-based model solved the task with fewer timesteps.

This indicates that our approach works well but also that the optimal shepherding algorithm depends on the behaviour of the herd. Again, additional experiments are necessary to confirm this hypothesis, especially because we have not yet exhaustively studied the parameter space of the agent-based shepherding model.

Conclusion and outlook

In this project, we have built upon the agent-based shepherding model from the paper *Optimal Shepherding and Transport of a Flock* [3] and extended it by introducing a surrounding fence and additional shepherds. Additionally, we have compared our approach to the existing agent-based shepherding model from the paper *Simulating Single and Multiple Sheepdogs Guidance of a Sheep Swarm* [2].

Possible next steps include a more detailed analysis of our extensions. It would be interesting to study whether the three observed shepherding behaviours driving, droving, and mustering also arise for more than one shepherd and, if so, for which sets of parameters. Furthermore, one could extend the model even further, for example by introducing obstacles along the path from the herd to the target.

CONTRIBUTIONS. *Franziska Weber* took care of the GitHub repository, researched existing models with multiple shepherds, wrote the description of the original model, analyzed the effect of introducing multiple shepherds, implemented the alternative agent-based shepherding algorithm and compared it to the optimization-based approach. *Kimberley Frings* corrected and initially executed the existing implementation, implemented the model extensions with a fence and multiple shepherds, composed all three reports and analyzed the effect of introducing a surrounding fence. *Franz Muszarsky* created the presentation and the video. All three worked on understanding the implementation of the model and on the organization of the project.

Bibliography

1. Lien JM, Bayazit OB, Sowell RT, Rodríguez S, Amato NM (2003) Shepherding behaviors.
2. Baxter D, Garratt M, Abbass HA (2021) Simulating single and multiple sheepdogs guidance of a sheep swarm. *Shepherding UxVs for Human-Swarm Teaming: An Artificial Intelligence Approach to Unmanned X Vehicles* pp. 51–65.
3. Ranganathan A, Heyde A, Gupta A, Mahadevan L (2022) Optimal shepherding and transport of a flock.
4. Reynolds CW (1987) Flocks, herds and schools: A distributed behavioral model in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques.* pp. 25–34.
5. Vicsek T, Czirók A, Ben-Jacob E, Cohen I, Shochet O (1995) Novel type of phase transition in a system of self-driven particles. *Physical review letters* 75(6):1226.
6. Kubo M, Tashiro M, Sato H, Yamaguchi A (2022) Herd guidance by multiple sheepdog agents with repulsive force. *Artificial Life and Robotics* pp. 1–12.

Appendix

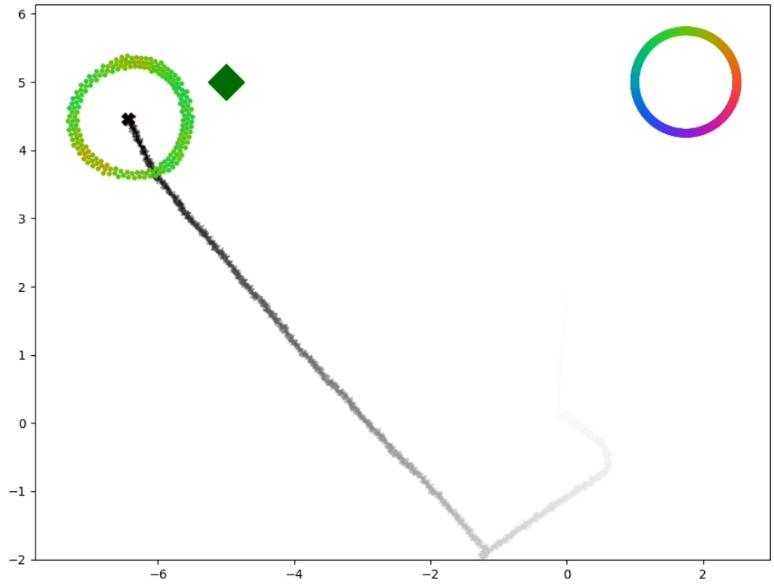

Figure 1. Trajectory plot for one shepherd in the driving scenario without a fence. The colors of the agents indicate their orientation.

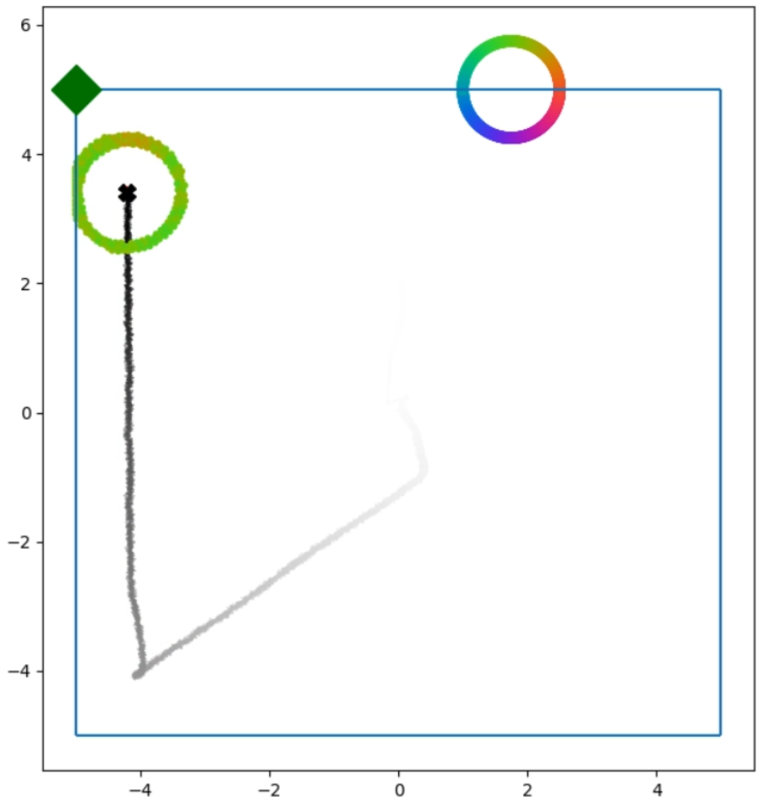

Figure 2. Trajectory plot for one shepherd in the driving scenario with a fence.

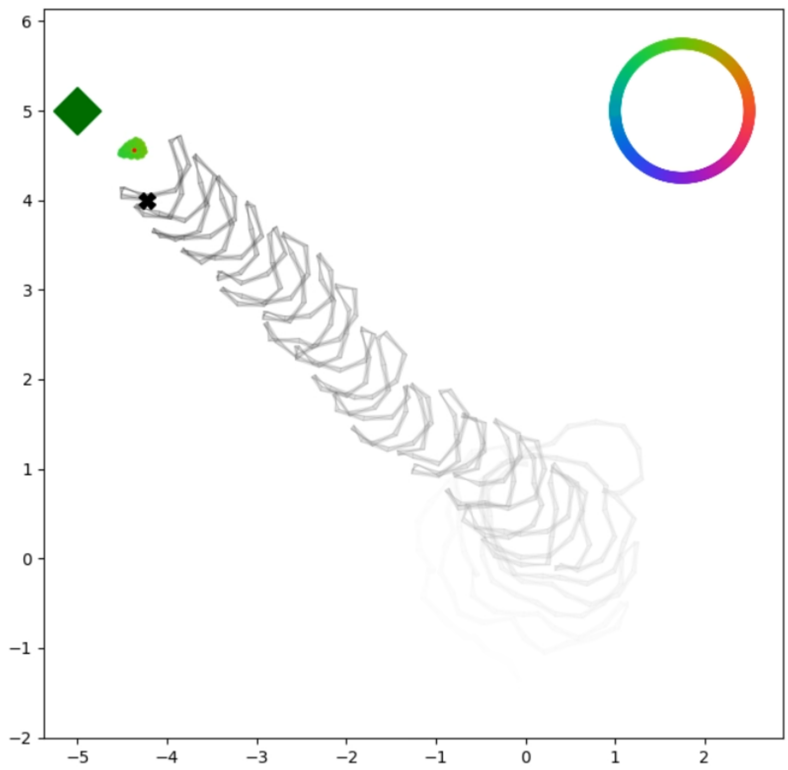

Figure 3. Trajectory plot for one shepherd in the droving scenario.

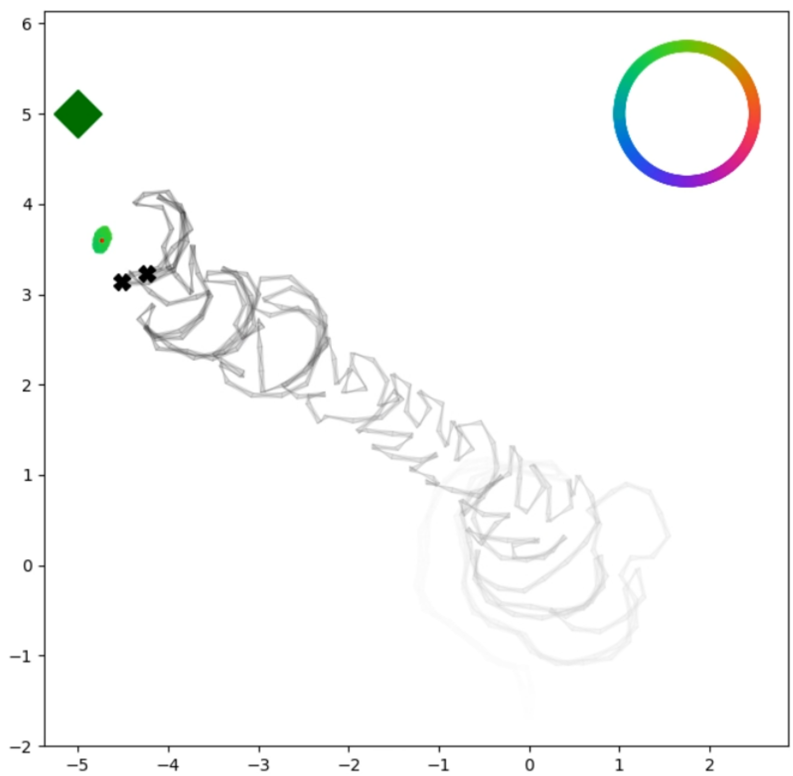

Figure 4. Trajectory plot for two shepherds in the droving scenario.