1 Model

Our environment consists of a larger starting area surrounded by a fence - the pasture and a smaller goal area positioned at its edge - the barn. At the start of the simulation, N agents - sheep and the shepherd - sheepdog are positioned randomly inside the starting area. The goal of the shepherd is to collect the agents and drive them into the barn within a certain time limit.

The behavior of the agents is implemented based on the model by Strömbom et al. [1].

1.1 Strömbom sheep model

In the model by Strömbom et al. [1], the agents' logic drives them to move away from the shepherd while staying close to their neighbors. On each step of the simulations, each agent decides on its actions based on its current position $\bar{A}_i$, position of the shepherd $\bar{S}$ and its n nearest neighbors.

Each agent is attracted to the local centre of mass (LCM) of its n nearest neighbors, repelled from other agents at distances smaller than r_a, and repelled from the shepherd, if the shepherd's distance from the agent is less than r_s.

If the agent's distance from the shepherd is less than r_s, the agent is repelled away from the shepherd in the direction $\bar{R}_i^s = \bar{A}_i - \bar{S}$.

The LCM is calculated with the equation

$$L\bar{C}M = \frac{\sum_{i=1}^{n} A_i}{n},$$

where $A_1, \ldots, A_n$ are the positions of the agent's n nearest neighbors. The agent will be attracted towards the $L\bar{C}M$ with the vector $\bar{C}_i = L\bar{C}M - \bar{A}_i$.

If there are k other agents within r_a distance from the agent and $A_1, \ldots, A_k$ are their positions, the combined repulsion vector from other agents $\bar{R}_a^o$ is defined by

$$\bar{R}_a^o = \sum_{j=1}^{k} \frac{1}{|A_i - A_j|}(A_i - A_j).$$

The desired heading of the agent is then calculated as a linear combination of these three vectors, weighted according to parameters ρ_a, c and ρ_s, plus small inertia and noise terms. The combined equation for the agent's desired heading in the next time step $\bar{H}_i{}'$ can be written as follows:

$$\bar{H}_i^{'} = h\hat{H}_i + c\hat{C}_i + \rho_a \hat{R}_i^a + \rho_s \hat{R}_i^s + e\hat{\epsilon}_i.$$

If the distance is smaller than r_s, the agent moves by δ in the direction of the desired heading, otherwise it grazes, which means that it moves with probability p, otherwise it remains stationary.

1.2 Changes to the Strömbom sheep model

As our environment uses sheep with a 1 unit diameter instead of 2 units in the original paper, the r_s and r_a parameters (shepherd detection distance and agent to agent interaction distance) have been divided by 2 to adjust for this change.

To prevent our agents from walking into trees or the surrounding fence, we introduce repulsion from obstacles R^o, which is active when an agent's distance

parameter	description	values	original values
N	total number of agents	**101**	1-201
n	number of nearest neighbors	**7-100**	1-200
r_s	shepherd detection distance	**22.5**	45
r_a	agent to agent interaction distance	**1**	2
ρ_a	relative strength of repulsion from other agents	2	2
c	relative strength of attraction to the n nearest neighbors	1.05	1.05
ρ_s	relative strength of repulsion from the shepherd	1	1
h	relative strength of proceeding in the previous direction	0.5	0.5
e	relative strength of angular noise	**0.1**	0.3
δ	agent displacement per time step	1.5	1.5
p	probability of moving per time step while grazing	0.05	0.05
r_{sS}	shepherd detection distance - strong repulsion	12.25	/
ρ_o	relative strength of repulsion from obstacles	2	/
r_o	fence detection distance	3	/
$maxTurn$	maximal turn rate in degrees per second	225	/
δ_r	agent displacement per time step when running	7.5	/

Table 1: Parameters of our implementation of the agent model based on the Strömbom model and the original implementation.

from a fence segment or tree is less than r_o. This repulsion force is multiplied by parameter ρ_o and added to the weighted linear combination of the original three forces. Similarly to the equation for the combined repulsion from nearby agents, the combined vector for repulsion from obstacles is calculated as

$$\bar{R}_i^o = \sum_{j=1}^{k} \frac{1}{|A_i - O_j|}(A_i - O_j),$$

where $O_1, \ldots, O_k$ are the positions of the closest point on bounds of each obstacle within r_o distance from agent A_i.

With the added repulsion from obstacles, the desired heading is calculated as such:

$$\bar{H}_i^{'} = h\hat{H}_i + c\hat{C}_i + \rho_a\hat{R}_i^a + \rho_s\hat{R}_i^s + \rho_o\hat{R}_i^o + e\hat{\epsilon}_i.$$

We also introduce a strong repulsion distance parameter r_{sS}, set to $r_s/2$, and a running speed δ_r to the agents' behavior. If the shepherd's distance to the agent is less than r_{sS}, the agent moves by δ_r instead of δ in each time step. Additionally, to make turns smoother, we limit the amount of degrees an agent can rotate in each step, controlled by the $maxTurn$ variable. This means that in each step before moving the agent rotates at most $maxTurn\delta_t$ degrees towards the desired heading, where δ_t is the time in seconds that passed since the last time step.

The parameters used for the experiments in the original paper and in our implementation are shown in Table 1.

1.3 Strömbom shepherd model

The shepherd model by Strömbom et al. decides between two types of behavior, collecting and herding, in each time step. The decision is based on the distances of the agents from their global centre of mass (GCM).

If all agents are within $r_a N^{\frac{2}{3}}$ of the GCM, it considers the flock collected and tries to position itself behind the flock relative to the target in order to drive the flock towards it. It does by heading towards the driving position P_d, calculated as

$$P_d = GCM + (GCM - \bar{G})\rho_a\sqrt{N},$$

where $\bar{G}$ is the goal position.

If there are agents more than $r_a N^{\frac{2}{3}}$ away from the GCM, the shepherd chooses the agent that is the furthest away from the GCM and tries to herd it back towards the herd by positioning itself behind it relative to the GCM. The shepherd heads towards the collecting position

$$P_c = \bar{A}_i + (\bar{A}_i - GCM)\rho_a,$$

where $\bar{A}_i$ is the position of the agent furthest from the GCM. In each time step the shepherd moves by δ_s in a straight line towards the target position - either P_d or P_c. If there is an agent within $3r_a$ distance from the shepherd, it moves by $0.3r_a$ instead to avoid approaching the flock at close range and causing it to disperse. The same $e\hat{\epsilon}_i$ noise as in the agent model is also added to the shepherd's movements.

The model can be implemented with global knowledge of locations of the agents, or as a local shepherd, where the agent only has information about the positions of n_s agents closest to it, and also does not receive information about agents within blind zone angle β behind it. When using local knowledge, the local centre of mass (LCM) of agents in the shepherd's visual field is used instead of the GCM.

1.4 Changes to the Strömbom shepherd model

Like the agents in our implementation, the shepherd also has an additional running state, in which it moves by δ_{s_r} each time step. The shepherd starts in running state. If there is an agent within r_s range it stops and waits, if the closest agent is between r_s and r_w away, it goes into walking state, moving δ_s per time step, and if there are no agents within r_r distance, it goes into running state. If the closest agent is between r_w and r_r away, it remains in its current state. The agent has its turn rate limited with its $maxTurn$ variable in the same way as the agents. We also implemented an option to take occlusion into account, so the shepherd can only see agents that are not occluded by other agents or objects from the shepherd's point of view.

Parameters of our implementation compared to the original model are listed in Table 2.

2 Shepherd algorithm

The Strömbom shepherding algorithm consistently succeeds in herding the agents only when the shepherd has global knowledge of the agents' positions, and when the agents are attracted to the centre of mass of at least half of the other agents. However, it becomes much less effective when the agents are attracted only to a smaller number of their nearest neighbors or when using only local knowledge for the shepherd. We attempt to modify the algorithm in a way that improves

parameter	description	values	original values
δ_s	shepherd displacement per time step	1.5	1.5
p_d	driving position	$r_a\sqrt{N}$ behind the flock	$r_a\sqrt{N}$ behind the flock
p_c	collecting position	r_a behind furthest agent	r_a behind furthest agent
e	relative strength of angular noise	**0**	0.3
δ_{s_r}	shepherd displacement per time step when running	7.5	/
e	n of nearest agents the local shepherd operates on	**7, 20**	20
e	blind angle behind the shepherd	$\pi/3$	$\pi/6$
r_s	agent distance at which the dog stops	$3r_a$	$3r_a$
r_w	agent distance at which the dog starts walking	$9r_a$	/
r_r	agent distance at which the dog starts running	$18r_a$	/
$maxTurn$	maximal turn rate in degrees per second	360	/

Table 2: Parameters of our implementation of the shepherd model based on the Strömbom model and the original implementation.

it performance with only local knowledge and makes it less dependent on the agents' behavior. The main two changes we propose are avoidance of already collected groups and a modified heuristic for choosing the target agent when in collecting mode.

2.1 Avoidance of already collected groups (prejsnji algoritem z racunanjem kotov)

When the shepherd is collecting the agents, situations occur where the agent that is furthest away from the centre of mass is on the opposite side of an already collected group of agents from the shepherd. By approaching the agent in a straight line in such a situation, there is a possibility of breaking up the already collected group again.

To prevent this from happening, we make the shepherd follow an arc around the calculated GCM/LCM instead, such that it keeps its distance to the GCM/LCM constant. The shepherd will only use this path when it is further away from the GCM/LCM than the collecting position, otherwise it will head directly towards the collecting position.

While this change helps to prevent the shepherd from splitting the already collected groups again, it can significantly increase the time the shepherd takes to reach the desired collecting position in certain cases, such as when the shepherd is much further from the GCM/LCM than the collecting position.

2.2 Avoidance of already collected groups and obstacles (nova verzija - utezena vsota)

When the shepherd is collecting the agents, situations occur where the agent that is furthest away from the centre of mass is on the opposite side of an already collected group of agents from the shepherd. By approaching the agent in a straight line in such a situation, there is a possibility of breaking up the already collected group again.

To avoid this, the shepherd should circle around the herd in an arc instead in such situations. We simulate this behavior by calculating the shepherd's desired heading as a weighted combination of a vector pointing directly towards

parameter	description	value
r_r	distance at which to start transitioning to arc movement	22.5
r_o	fence detection distance	3
ρ_o	strength of repulsion from obstacles	0.1

Table 3: Additional parameters used in our model.

its target position, $\bar{H}_d$, and an arc movement vector $\bar{H}_a$, calculated as

$$\bar{H}_a = \sum_{i=1}^{n} \frac{1}{(|A_i - S|)^2} \hat{H_i^{\perp}},$$

where S is the shepherd's position, $A_1, \ldots, A_n$ are the positions of agents the shepherd has knowledge of and $H_i^{\perp}$ is a vector that is clockwise perpendicular to $\hat{A}_i - \hat{S}$ if the shepherd's target position is to the right of the GCM/LCM from its point of view, and counter-clockwise perpendicular to $\hat{A}_i - \hat{S}$ otherwise.

The shepherd's desired heading in the next step is computed as

$$\bar{H}_s' = \rho_d \hat{H}_d + \rho_a \hat{H}_a + \rho_o \bar{R}_s^o.$$

The weights ρ_d and ρ_a are based on the distance d between the shepherd and the nearest agent:

$$\rho_d = \begin{cases} 1, & \text{if } d \geq r_r \\ \frac{d}{r_r}, & \text{otherwise,} \end{cases}$$

$$\rho_a = \begin{cases} 0, & \text{if } d \geq r_r \\ \frac{r_r - d}{r_r}, & \text{otherwise,} \end{cases}$$

When not close to any agents, the shepherd moves straight towards the target, then transitions to move in an arc when it gets closer than r_r to an agent. We set the r_r parameter to the same value as the r_s parameter of agents, assuming that the shepherd would notice the agents reacting when it moves closer than that distance and start adjusting its path accordingly.

To help the shepherd to navigate the field, we also added repulsion from obstacles $\bar{R}_s^o$, defined in the same way as for the sheep:

$$\bar{R}_s^o = \sum_{i=1}^{j} \frac{1}{|S - O_i|} (S - O_i),$$

where O_1, …, O_i are positions of the closest point on bounds of each obstacle within r_o distance from the shepherd. This vector is weighted by a small constant ρ_o. The additional parameters used in this model are presented in Table 3.

2.3 Selection of target agent when collecting

One of the problems we encountered was the shepherd getting stuck in the centre of mass between multiple groups of agents or repeatedly switching targets when collecting and walking back and forth between them as a consequence. To counteract this, we modified the heuristic for choosing the target agent when collecting them. Instead of choosing the agent that is furthest away from the GCM/LCM, we choose the agent with the highest priority value, using the equation

$$priority = d_{sC}(1 - |\alpha/180|),$$

where α is the angle between the shepherd's current heading and the vector from its position towards the agent's position. This makes the shepherd more likely to stay focused on a single group of agents.

References

[1] Daniel Strömbom, Richard P Mann, Alan M Wilson, Stephen Hailes, A Jennifer Morton, David JT Sumpter, and Andrew J King. Solving the shepherding problem: heuristics for herding autonomous, interacting agents. *Journal of the royal society interface*, 11(100):20140719, 2014.