University of Ljubljana
Faculty of Computer and Information Science

Reimplementing the FRIsheeping herding algorithm using fuzzy logic

Veljko Dudić, Noah Novšak, Petra Kuralt, and Timotej Košir

Collective behavior course research seminar report

December 17, 2023

Iztok Lebar Bajec | professor | mentor

This study explores the integration of fuzzy logic into herding algorithms as an innovative approach to enhance adaptability and nuanced decision-making within simulated group dynamics. Initially, we build upon the Strömbom algorithm, implemented in a Unity environment, introducing fuzzy logic to refine the behavior of entities within the herd. Afterward, our investigation extends to the implementation of herding mechanisms using genetic algorithms. This utilization of fuzzy logic and genetic algorithms aims to create a comprehensive and robust framework for simulating herding behavior in dynamic environments.

fuzzy logic | herding algorithm | genetic algorithm

Herding algorithms, at their core, strive to model the behavior of a shepherd (sheepdog) endeavoring to control extensive groups of unwilling individuals (sheep). Traditional approaches rely on predefined rules to govern the individual behavior of agents within a group, influencing their movements based on factors like proximity and alignment with neighbors [1]. While these methods capture the fundamental aspects of collective behavior, their deterministic nature might fall short in handling the inherent uncertainty and imprecision of real-world scenarios. In this context, our approach introduces another dimension to herding algorithms by incorporating fuzzy logic. This enables a more flexible and adaptive decision-making process by allowing degrees of truth between absolute true and false, thus helping the individuals in a group respond more effectively to altering environments.

As a starting point, we take the algorithm proposed by Strömbom et al. [2], which has already been implemented by our predecessors in a Unity environment. Strömbom's model employs a weighted sum of various forces to determine the sheep's movement. At the same time, the shepherd alternates between *collecting* dispersed sheep and *driving* the group towards a predefined goal.

In this paper, we discuss our approach to reimplementing Strömbom's algorithm and its suggested improvements using fuzzy logic. Additionally, we explore the viability of the sheep's learning behavior through a genetic algorithm, whereby sheep incrementally improve their tactics through evolution.

Methodology

The testing environment consists of an enclosed pasture with a barn at its edge. The N agents (sheep) are randomly placed in the field with a shepherd (sheepdog), aiming to collect and drive all the sheep into the barn within a limited time. Several different algorithms can be selected to govern the behavior of each agent in the simulation.

Strömbom model. Agents attempt to move away from the shepherd while staying close to their neighbors. Each agent decides on its next course of action based on the locations of itself, the shepherd $\bar{S}$, and each of its n nearest neighbors $\bar{A}_i$. The agents are attracted to their neighbors' center of mass (LCM) and repelled from other agents within a boundary distance r_a. The shepherd repels them as well if it is closer than r_s.

The following components sum up all of these contributions:

1. Repulsion from the shepherd: If the agent is within a distance r_s of the shepherd, it is repelled in the opposite direction $R_i^s = A_i - S$.
2. Attraction towards the local center of mass: The vector representing attraction is denoted at $C_i = LCM - A_i$, where $LCM = \frac{1}{n}\sum A_i$ for the agent's n nearest neighbors.
3. Repulsion from Other Agents: If there are k other agents within r_a, the combined repulsion vector is defined as $R_i^a = \sum_{j=1}^{k} \frac{(A_i - A_j)}{|A_i - A_j|}$.
4. Inertia and Noise: In each timestep, the agents maintain their previous direction H_i, and some random noise ϵ is added to account for unpredictability.

Combining all these components, the movement vector in each iteration becomes

$$H_i' = hH_i + cC_i + a\hat{R}_i^a + s\hat{R}_i^s + e\epsilon.$$

A herding algorithm that uses fuzzy logic

This study explores the integration of fuzzy logic into herding algorithms as an innovative approach to enhance adaptability and nuanced decision-making within simulated group dynamics. Initially, we build upon the Strömbom algorithm, implemented in a Unity environment, introducing fuzzy logic to refine the behavior of entities within the herd. Afterward, our investigation extends to the implementation of herding mechanisms using genetic algorithms. This utilization of fuzzy logic and genetic algorithms aims to create a comprehensive and robust framework for simulating herding behavior in dynamic environments.

fuzzy logic | herding algorithm | genetic algorithm

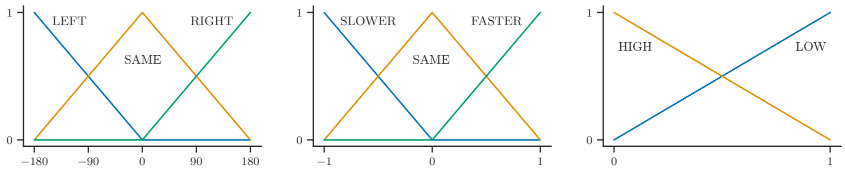
An entirely separate set of rules governs the shepherd's behavior. It dynamically switches between *collecting* and *driving*, based on the maximum distance of agents from the center of mass (either global GCM or local LCM). If all agents are within $r_a N^{\frac{2}{3}}$, it attempts to position itself behind the flock to push it towards the goal. Alternatively, if any agent is farther than $r_a N^{\frac{2}{3}}$, the shepherd instead selects the most distant agent and herds it back towards the flock. The shepherd's trajectory is thus always directly toward the desired collecting or driving position. Additionally, some noise is added to the shepherd's movement as well.

Fuzzy logic. We defined the existing *crisp* model in the previous section. Now, we will try to describe its *fuzzy* counterpart. Using fuzzy logic may introduce a more nuanced and realistic decision-making process for determining the behavior of each agent.

As a starting point, we will take the model proposed by "Boids with a fuzzy way of thinking" [3]. They find the most important aspect of flocking behavior is the alignment function. Then, we will add an additional component to model the shepherd guiding the flock. The described model uses three linguistic variables: the relative heading difference H coded by the values $LEFT$, $RIGHT$, and $SAME$ 1, the relative speed difference S coded by $SLOWER$, $FASTER$, and $SAME$ 1, and the significance SIG of each neighboring agent, coded by $HIGH$ and LOW 1. Using these linguistic variables, the alignment steering function is represented by the following rules:

```
if (SIG is LOW) or (H is SAME) then H' is SAME,
if (SIG is HIGH) and (H is LEFT) then H' is LEFT,
if (SIG is HIGH) and (H is RIGHT) then H' is RIGHT,
if (SIG is LOW) or (S is SAME) then S' is SAME,
if (SIG is HIGH) and (S is SLOWER) then S' is SLOWER,
if (SIG is HIGH) and (S is FASTER) then S' is FASTER,
```

Our proposal for adding a shepherd is to model it with more or less the same alignment rules but using different weights and membership functions.



(a) Heading (b) Speed (c) Significance

Figure 1. Membership functions

The first step in implementing this model is to fuzzify the inputs, i.e., taking the agents' positions and current headings to determine the degree to which they belong to each of the appropriate fuzzy sets via the aforementioned membership functions. The next step is to combine the fuzzy inputs according to the proposed rules into a single number, the antecedent. Then, create a new fuzzy set, the consequent, and reshape it using the antecedent. Then, finally, all of the rule outputs are aggregated into a single fuzzy set and defuzzified into a single numeric output.

OCEAN model. To further extend the fuzzy logic model, we use the OCEAN personality model, generally used to describe people, to individualize the agents [4].

As defined in social science and psychology, our personalities shape our thoughts, feelings, and behaviors, distinguishing us from other individuals. Modeling heterogeneous sheep requires considering personalities, as individuals with diverse traits react differently to the same situation. The Five-Factor personality model, known as OCEAN, categorizes personalities into five contributing factors:

1. **Openness (O)** reflects the degree of curiosity and creativity, indicating preferences for novelty and variety.
2. **Conscientiousness (C)** describes the level of organization and care exhibited in collective activities
3. **Extraversion (E)** related to the degree of energy, sociability, and outgoingness.
4. **Agreeableness (A)** represents a tendency to exhibit compassion and cooperation rather than suspicion and antagonism towards others.

5. **Neuroticism (N)** indicates the tendency to experience unpleasant emotions easily, such as anger, anxiety, depression, or vulnerability, and is the opposite of emotional stability.

Agents are modeled with these traits to achieve more realistic crowd behavior.

Fuzzy Inference Model. The first implementation is based on [3] and does not use an intermediate pathing algorithm. Each sheep has its own personality, and using the position and speed of sheep in a local environment, we use our fuzzy inference system to return a new heading and speed for all sheep. These values are directly used for the movement of the given sheep.

The second implementation is based on [4] and builds on top of the Strömbom algorithm. Instead of directly obtaining the speed and angle of the sheep, we derive new parameters for Strömbom from the personality of each individual agent and assign the parameters from defined fuzzy rules, such as:

```
if (Extraversion is POS) and (Agreeableness is POS) then k neighbors HIGH
if (Neuroticism is POS) then added noise is HIGH
```

Each sheep's personality is based on the OCEAN model but excludes the conscientiousness aspect. The inputs of the fuzzy inference system are, in this case, the personality of a sheep and the previous Strömbom parameters, and the outputs are the new parameters.

Genetic algorithm. The second focus of the article is making sheep dynamics less scripted but learned. We plan to achieve this by representing the sheep as autonomous agents that utilize a genetic algorithm (GA) to evolve their herding strategies incrementally.

GAs are optimization algorithms inspired by the concept of natural selection. The process starts by generating an initial population of potential solutions. Each solution is assessed using a cost function, and solutions with higher fitness are more likely to be selected for reproduction. The chosen individuals undergo crossover and mutation, eventually replacing the less fit individuals in the population. This cycle repeats for multiple generations, refining the solutions over time.

Regarding collective behavior, we need to modify the traditional GA by adding some changes presented in Thomas Miconi's "A Collective Genetic Algorithm"[5]. Since we can not evaluate a single agent but rather the population as a whole, we randomly select two sheep and separately remove them to evaluate their impact on the whole herd. We then create offspring through crossover and mutation. The less valuable sheep, as determined by their effect on the collective herding fitness, are then replaced by the offspring. The sheep's genotype will be represented as a vector so that we can use it as our sheep's decision model. Currently, the plan is to use a simple three-layer artificial neural network. The input layer will consist of an input vector containing information about distances to other sheep and shepherds, and the output layer will return the angular direction and speed of the sheep's movement.

The initial suggestion for the cost function is a weighted sum between the time it took for all the sheep to enter the barn and the distances separating them in a herd. The optimization strategy is designed to encourage sheep to form a cohesive pack while simultaneously maximizing their evasiveness. Examples of similar implementations using ANN models can also be observed in simulating the collective behavior of zebrafish [6] and robots [7], albeit using supervised training of the neural network agent's models instead of genetic algorithms.

Results

We have extended the FRIsheeping Unity project to include the proposed Fuzzy Logic models, moving away from planned pathing algorithms to mimic the erratic behavior of the sheep herd. The improvement allows for more dynamic and unpredictable collective movements, as each sheep now exhibits individualized responses influenced by unique personality traits.

Figure 2. An example of our setup in Unity environment

Discussion

The two proposed fuzzy logic inference models give us insight into different fuzzy logic applications for simulating herding dynamics. The first approach, stemming from [3], embraces simplicity by directly incorporating fuzzy logic into the movement of each sheep. The personalized nature of sheep behavior, dictated by OCEAN personality, allows for a decentralized decision-making process. In contrast, the second implementation, inspired by [4] and building upon the Strömbom algorithm, introduces a layer of complexity by deriving new parameters based on individual personality traits. In both approaches, a fuzzy inference system determines the relationships between personality traits and decision preferences.

Bibliography

1. Lien JM, Bayazit O, Sowell R, Rodriguez S, Amato N (2004) Shepherding behaviors in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004.* Vol. 4, pp. 4159–4164 Vol.4.
2. Strömbom D et al. (2014) Solving the shepherding problem: Heuristics for herding autonomous, interacting agents. *Journal of The Royal Society Interface* 11.
3. Bajec IL, Mraz M, Zimic N (2003) Boids with a fuzzy way of thinking in *Proceedings of ASC*. Vol. 58, p. 62.
4. Xue Z et al. (2017) Fuzzy logic-based model that incorporates personality traits for heterogeneous pedestrians. *Symmetry* 9(10).
5. Miconi T (2001) A collective genetic algorithm in *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*. pp. 876–883.
6. Beckerleg M, Zhang C (2016) Evolving individual and collective behaviours for the kilobot robot in *2016 IEEE 14th International Workshop on Advanced Motion Control (AMC)*. pp. 263–268.
7. Cazenille L, Bredeche N, Halloy J (2018) Modelling zebrafish collective behaviours with multilayer perceptrons optimised by evolutionary algorithms. *arXiv preprint arXiv:1811.11040*.