Flocking on a 2-D Triangular Graph

Ahmad Reza Cheraghi, Asma Ben Janete, Kalman Graffi
Technology of Social Networks, Heinrich Heine University Düsseldorf, Germany
Honda Research Institute Europe GmbH, Offenbach am Main, Germany
ahmad.cheraghi@hhu.de, asma.ben.janete@hhu.de, kalman.graffi@honda-ri.de
https://tsn.hhu.de, https://www.honda-ri.de

***Abstract*—Flocking behavior in swarms is phenomenal. Many small entities are interacting to form a large shape and to protect themselves from bigger entities. An examples are fish, they become a flock to be safe from predators. The way of how flocks move without getting separated, having a constant speed and not interfering each other is known as the flocking problem. In this paper, we build up a the three-zones flocking model on a two dimensional triangular graph to solve the flocking problem and evaluate the impact of changes in the separation, alignment and cohesion zone sizes on the flocking behavior.**

I. Introduction

In recent years, the field of multi agent systems (MAS) and swarm robotics in particular have captured significant attention. They have been extensively studied since computer simulations allow the detailed investigation of these complex systems. Swarm robotic systems have different properties on the macro as well as on the micro level. On a macro level all MAS are reliable, efficient, flexible and cheap. Swarm robotic systems are a specific type of MAS that are additionally scalable, decentralized and homogeneous. Considering the micro level, the agents of the swarm are mobile, autonomous, proactive, robust, adaptive, and social. All these properties make swarm robotics the a suitable solution for sophisticated problems like autonomous surveillance, environment investigation, post-disaster relief and many other application fields.

One of the most promising and intriguing sub-fields of swarm robotic systems is flocking - the art of cohesion and harmony. A flock is a group of agents that hold together and move harmoniously as one unit. They have common goals such as foraging or migration. Flocking is an emerging behavior of a group of agents that arises by applying three simple rules, namely cohesion avoidance, velocity matching, and flock centering. The way how these rules could be applied depends on many factors. It makes a difference whether an agent is able to rotate or has only the possibility to move in certain directions. The first step to solve the flocking problem is to specify a flocking model. A flocking model -similarly to game rules- describes the reaction of an individual entity in every possible situation that it may be confronted with. Then those rules have to be translated to agent actions dependently of agent abilities and environmental properties. Care should also be taken to ensure that the agent properties are not violated such as its simplicity.

The aim of this work is to solve the flocking problem with applying the Three Zones Model [1] on a 2D triangular grid. The capabilities of the agents are very limited. They are just able to move step-wise in one of six directions and have a constant velocity. Instances are measured as number of steps. We implemented and evaluated this approach in the Swarm-Sim simulator. In this paper, we present the analysis of the impact the various zone sizes zones of the flock.

II. Related Work

Swarm robotic systems and flocking behavior are very intriguing fields that are in the process of being intensively researched. Based on observations in nature, several flocking models were created. A flocking model is intended for abstracting the flocking problem. In 1952, John Emlen [2] investigates behavior of birds. He finds that the attraction and repulsion between birds determines the density and the form of the flock. Attraction is due to the well-being and safety feelings of the bird within the flock. Whereas, the rejection is a result of dislike of overcrowding. In 1987, Reynolds [3] presents a a simulated flock, starting with a boid model that supports geometric flight. He sets the most famous three flocking rules, namely, collision avoidance, velocity matching and flock centering. In 2002, Couzin et al. [1] gives a flocking model where the surrounding environment of agents is divided into three zones. In each zone, one of the three flocking rules is adhered to. In 2018, Khan et al. [4] adopts the same model to investigate how attraction, repulsion and orientation can affect the patterns of agents in flocking movements. He finds that a slight change in parameters had a big impact on flocking patterns and that the movement of the swarm could be controlled by setting the attraction, repulsion and orientation forces. In 2011, Moeslinger et al. [5] presents a minimalist flocking algorithm. The focus is on agent simplicity. Thus, the algorithm is developed for robots without communication abilities, memory equipment or global information. The flocking behavior is generated by using different distance thresholds for attractions and repulsions in the different sensor fields of the robots. Thus, the alignment resulted in emergent effect. In 2014, Witkowski et al. [6] uses an original asynchronous genetic algorithm to model swarm agents which interact using only signals and move in a 3D environment looking for an invisible resource. It is found that they always form temporary local leaders allowing them to flock together. In 2018, Shimada et al. [7] uses the multi-agent reinforcement learning MARL to predict individual behavior from the observed collective behavior. It is shown that it is possible for agents to learn flocking, together creating a stable collective behavior despite initial inhomogeneity. In this previous work, the impact of the cohesion, alignment and separation fields around a member of a flock are not analyzed in detail and thus, with this paper, we aim to add this analysis on the overall flock behavior.

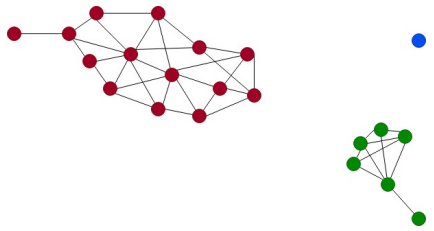


Fig. 1: Three flocks as connected components in a graph

III. Definitions

In order to understand the mathematical approach of the problem, we introduce the relevant terms.

A. Swarm-Sim Related Terms

Swarm-Sim is a 2D/3D swarm simulator developed at the Heinrich Heine University in Düsseldorf/Germany. The simulator provides a

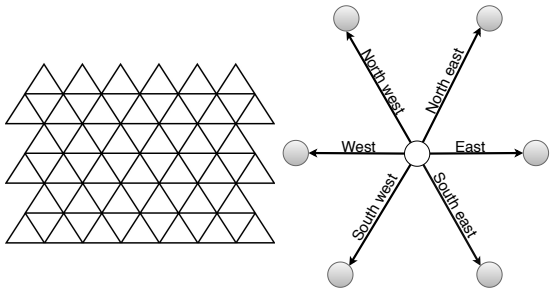


Fig. 2: Swarm-Sim World and Directions

core and is open-source [8]. Considering the structure of the Swarm-Sim simulator, definitions for some basic components and relations are given as follows.

1) World: The world is a set of **locations** interconnected via **Links**, forming together a graph that expands in a two dimensional coordinate system. A **location** is a fix couple of coordinates $l = (x, y)$ in the two-dimensional field. Locations are the nodes of the graph. L, the set of whole locations is defined as:

$$L = \{(x, y) \mid x \in \mathbb{R},\ y \in \mathbb{R}\}$$

The world is structured in such a way that each location is connected to exactly six adjacent locations in six directions: north-east, north-west, south-east , south-west, east and west (see Fig. 2). Let A_l be the set of the adjacent locations of the location l:

$$A_l = \{a_1 ... a_6 \mid l \in L,\ a_i \in L \setminus l,\ 1 \leq i \leq 6\}$$

A **link** connects two locations and represents thus the edges of the graph. l^2 is the set of links in the world.

$$L^2 = \{\ (a, b) \mid a \in L\ b \in A_a\}$$

2) Path: Two locations are always connected over a chain of links building a path. A lot of possible paths are between two locations, which is summarized in the set of **Paths**. The set of $Paths(l_p, l_q)$ between two locations $l_p, l_q \in L$ is a set of links $LI\ \{(l_p, l_1), (l_1, l_2), ..., (l_{q-1}, l_q)\} \subseteq LI$, so that through several links l_p and l_q are connected. Please, note that several possible paths exist. Typically, the shortest paths are interesting. For that the set of **Shortest Paths** is defined as

$$minPaths : L \times L \to \mathcal{P}(LI)$$

$$p \in minPaths(l_p, l_q) \subseteq Paths(l_p, l_q)$$

$$when\ p \in Paths(l_p, l_q) \wedge \nexists\ q \in Paths(l_p, l_q) : |q| < |p|$$

3) Distance: In this context, it is necessary to define the **Distance** between two locations. The distance between two locations is the length of the *shortest path* between these two locations. The distance $D : L \times L \to \mathbb{N}_+$ is defined as the minimum number of links between the two locations $D(l_p, l_q) = |p|\ with\ p \in minPaths(l_p, l_q)$.

4) Particle: Particles are the active agents in the world, they are modeled as moving points in the field. Each particle is identified by an ID p. P is the set of all particle-IDs of n particles:

$$P = \{p \mid p \in \mathbb{N}_+ , \forall\, i, j \leq n \quad i \neq j \Leftrightarrow p_i \neq p_j\}$$

A particle can only be located on locations and moves between adjacent locations via links. For example if a particle is on location $l \in L$, it has the possibility to move just in one of the six possible directions to reach an adjacent location $a \in A_l$
The function $loc(p)$ assigns to each particle its location $l_p = (x_p, y_p)$

$$loc : P \to L$$

$$loc(p) = (l_p \mid l \in L)$$

The distance between two particles p_i and p_j is the length of the shortest path between their locations. Due to collision avoidance, two particles cannot have the same location.

$$d(\ loc(p_i), loc(p_j)\ ) \geq 1$$

To enable simulating the communication between agents, each particle is assigned a memory, which can be read/written by adjacent particles.

5) Rounds: The dynamics in the world are modeled in rounds, in each round a particle leads an Observe-Orient-Decide-Act (OODA) loop. Every round, a particle observes its neighbors, interprets the collected information, makes its decision and moves.

B. Flock

A **flock** is a set of particles that are grouped together. It consists of at least one particle. A flock is defined as a set of all particles belonging to it (a is used for identification):

$$f_a = \{p_i \in P \mid \exists p_j \in N_{p_i} \leftrightarrow p_j \in f_a\}$$

C. Flocking Rules

In [3], Reynolds presents a simulated flock, starting with a boid model that supports geometric flight. He set the most famous three flocking rules. **Separation**: Avoid collision with nearby flock mates. **Cohesion**: Try to stay close to nearby flock mates. **Alignment**: Match speed and heading with nearby flock mates. These rules remain the basis until now in simulating or investigating the flocking behavior.

D. Neighbors

Each particle has limited visual abilities, it can observe other particles in a distance within its visual range VR. These particles are called **neighbors** of the observing one. Let p be a particle with a visual range VR. N_p is the set of its neighbors:

$$N_p = \{p_i \in P \mid d(loc(p), loc(p_i)) \leq VR,\ \ p \neq p_i,\ \ VR \in \mathbb{N}\}$$

E. The Flocks Graph

In order to model all flocks in the world, each particle is modeled as a node in a graph. For each particle, edges between it and each one of its neighbors are added. Thus, an undirected graph is obtained, which is composed of connected components. A connected component is a sub-graph in which every two nodes are connected by at least one path. There are edges between connected components of a super graph. Thus, each connected component represents a flock. There can be several flocks simultaneously in the world (see Fig. 1) and F is the set of all flocks:

$$F = \{f_a\}$$

F. Division and Merging of Flock

Flocks are entities that can divide and merge. If two particles, p_i and p_j, belonging to different flocks f_{a1} and f_{a2}, come to a distance less than or equal to their visual range, then both flocks merge.

$$p_i \in f_{a1} \wedge p_j \in f_{a2} \wedge p_j \in N_{p_i} \Rightarrow \exists f_a \in F, p_i, p_j \in f_a$$

In order to understand how a flock can be divided, the definition of cut vertices from the graph theory is adopted. A cut vertex is an edge, that by removing it we obtain more connected components, as depicted in Figure 3. If two particles, p_i and p_j, are connected with a cut vertex and move in a distance greater then their visual range VR, the link between them disappears.

The flock is divided and p_i and p_j belong then to separate flocks.

$$\exists\ p_i, p_j \in P, N_{p_i} \cap N_{p_j} = \emptyset, d(loc(p_i), (loc(p_j)) > VR \Leftrightarrow \mid F \mid > 1$$

Thus, one particle can belong to only one flock. Thus, the sum of all flock members of all flocks is equal to the number of all particles.

$$\sum_{a=1}^{|F|} \mid f_a \mid = \mid P \mid$$

G. Flock Density

Generally, the density is defined as the relationship between the mass of the substance and the space it takes up. As it is too complicated to get the area size covered by a flock, the density of the flock is defined based on distances. For this purpose, it is considered the surrounding environment of each particle in a radius of *DensityRadius*. **Density_neighbors** of one particle are the particles that are in a distance smaller equal *DensityRadius*. The set of Density_neighbors of p is given by:

$$densityNeighbors_p = \{p_j \in N_p | d(loc(p), loc(p_j)) \leq DensityRadius\}$$

The density of the flock is calculated with this formula

$$density(f_a) = \frac{\sum_{p \in f_a} \mid densityNeighbors_p \mid}{DensityRadius \times \mid P \mid}$$

H. Flocking Interaction Rings - Separation, Alignment, Cohesion

In the adopted flocking model, particles interact with their neighbors based on distances. The distance between two neighbors defines the way they react. There are three main distances to consider, namely the $min_distance$, $max_distance$ and VR.

a) $min_distance$*:* is the minimum distance that particles try to keep between each other. If a particle comes closer than the $min_distance$ to an other one, then the separation rule applies and both of them aim to move away to avoid collision.

b) $max_distance$*:* is the maximum allowed distance to the nearest neighbor. If a particle has its nearest neighbor in a distance greater than $max_distance$ it applies the cohesion rule, it moves in a direction in which it can reach the most of its neighbors.
With neighbors in a distance between $min_$ and $max_distance$ a particle applies the alignment rule. It moves in the same direction as most of its neighbors.

c) VR*:* is the visibility range of particles, they can only observe neighbors within a distance equal VR.
By applying these three definitions each particle is surrounded with three virtual rings as shown in Figure 4. The first ring is called **separation ring**. The middle one is the **alignment ring** and the third one is the **cohesion ring**.

IV. Goals

The aim of this work is to translate and apply the flocking rules for agents and analyze the impact of the three interaction rings. Agents are modeled to not be able to rotate nor change their velocity. They move stepwise in one of six directions based on the information collected by their sensors. We apply a distance based interaction model, thus the question here is what are the best settings for the three distances $min_distance$, $max_distance$ and VR. The goodness of the chosen setting is determined by the computational costs made by agents before making a choice such as inter-agent communication and data interpretation.

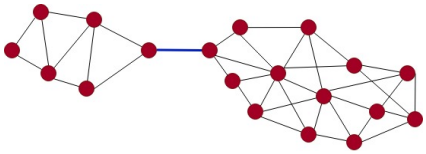


Fig. 3: A graph with one cut vertex (the blue edge)

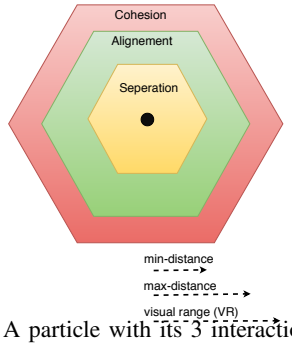


Fig. 4: A particle with its 3 interaction rings

A. Initial Situation

The initial situation is one flock in an empty world. The flock members are constantly adjusting to the movement of their neighbors and are constant motion (1 step per round).
The simulation parameters are configurable in order to have the possibility to investigate the different resulting emergent effects of those changes on the whole flock. The list of the configurable parameters of particles is depicted in Table I.

B. Contributions

The milestones of the work are fixed as follows: At first, the flocking problem is solved based on the three given flocking rules mentioned in Section III-C and following the thee interaction-rings flocking model. Thereby, the limitations of the agents must be considered. The parameters listed in Table I are configurable. Thus, changes are possible and so the emergent behavior can be investigated.

Investigation of emergent impact of local changes: Using the implemented solution, the flocking behavior is investigated with different values of the settable parameters in order to answer the following question: *What is the impact of the size of the interaction rings for separation, alignment and cohesion as discussed in Section III-H on the flock in general and on the agents as special?*

V. Methodology

In this section, a detailed presentation of how the flocking problem is solved is given. The encountered problems during this work are also explained. After that, evaluation metrics are depicted with which the goodness of the chosen settings are evaluated.

A. Three-Zones Model

Couzin et al. modeled in [1] a flock where agents adhere to three zones to make the motion decision. Their interactions with neighbors allow them to choose a turning rate to update their headings and a direction vector they adapt their velocity to. The three zones are:

TABLE I: Configurable parameters of particles

Parameters	Functionality
VR	The visual range of particles
$min_distance$	The minimum distance to keep between neighbors
$max_distance$	The maximum distance allowed to the nearest neighbor

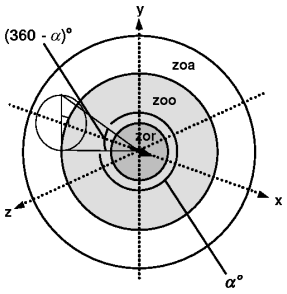


Fig. 5: Individual representation in the center of the three zones [1]

- Zone of repulsion (zor): The agent attempts to maintain a minimum distance to others within this zone.
- Zone of orientation (zoo): If there is no flock mates in zone of repulsion the agent applies the alignment rule with its neighbors withing this zone.
- Zone of attraction (zoa): The agent moves toward the positions of its flock mates within this zone if there is no other agents in the other two zones.

In [4], the previous model was adopted with the difference that a specific number of agents have a constant speed.

B. Adapted Flocking Model

In this work, a model similar to the one of Couzin et al. [1] is adopted. In this model each particle interacts based on particles within specific distances, determining the reaction type. There are three reactions respecting the three flocking rules mentioned in Section III-C

Repulsion: the particle moves away to avoid collision with its neighbor, here occurs the separation rule.

Attraction: the particle tries to get closer to its neighbors, thus it applies the cohesion rule.

Harmonization: the particle moves with its neighbors in the same direction respecting the alignment rule.

In this model the strength of the repulsion or attraction are still the same. This is due to the fact that every particle moves only one step per round, so the velocity stays constant and this is one of the main differences between this model and the one of Couzin et al. or also the boid model of Reynolds [3].

The Swarm-Sim simulator is round based. That means, in each round, every particle observes its neighbors, makes its decision and moves. The decision here is one of the six possible directions it can move to, the direction in which it achieves the goal of the actual round. Local interactions between particles are based the relative orientation to each one of the neighbors, the relative distances to them and their headings. The particle collects the required information from its observed neighbors, interprets its situation, makes its decision and then moves to the chosen direction.

C. Implementation - Collected Data

This section explains how the flocking model was implemented and tested. Different problems were faced such as simulating the real sensors which allow to get the relative direction to neighbors or making the solution configurable.

In order to make its decision, the particle has the possibility to collect information from each one of its neighbors. Those collectible information are listed in the following

1) Relative Direction: The relative direction is easy to get for robots with real sensors. To simulate this sensors the coordinates of both particles were used. The location of the observing particle is considered as an origin point and the one of the observed particle as a target point. After having a look at the different possibilities, it can be noticed that in certain situations the target point can be reached just as quickly in two directions. In order to better understand the approach, a directional axis is defined for each direction, see Figure 6. All locations, which are fastest to reach in a direction $d \in \{NE, NW, SE, SW, E, W\}$, are located on the d_axes. If a neighbor is located in the area between two directional axes d_axes and d'_axes, so the particle can be reached just as quickly in both directions d and d' In order to simplify the calculation, lets suppose that the observing particle with coordinate (x_1, y_1) is located at $(0, 0)$ and its observed neighbor with coordinates (x_2, y_2) is located at $(x = x_2 - x_1, y = y_2 - y_1)$. The field with its location coordinates is observed and it found out that the coordinates (x, y) of locations on a directional axes satisfy one of both equations: $y = 0$ or $\mid x \mid = \mid y \mid \times 0.5$. Thus, the particle learns whether its neighbors face the same direction as itself.

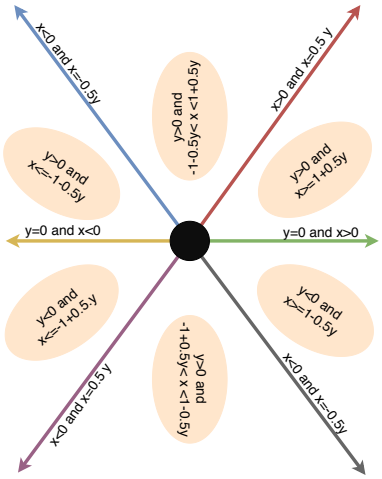


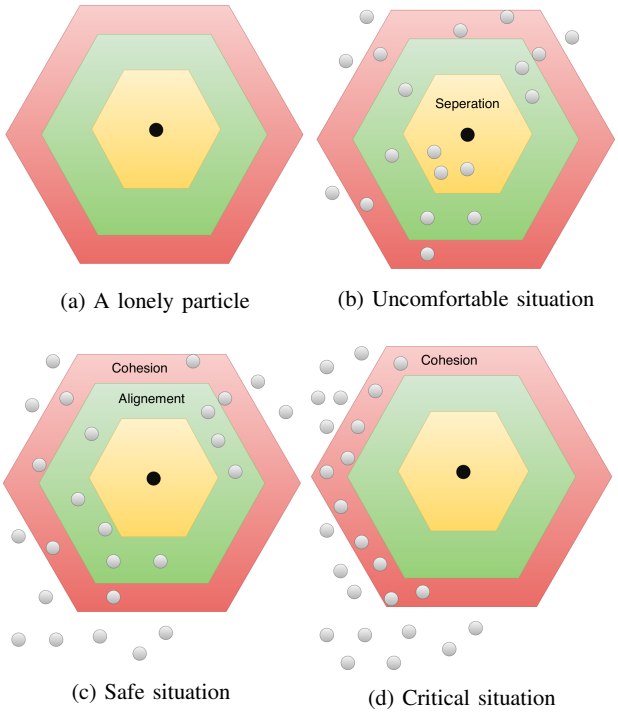
Fig. 6: Satisfied equations from locations on directional axes

2) Relative Distance: The relative distance is the distance between both, the location of the observing particle and the one of its observed neighbor. Comparing the relative distance with the three predefined distances *min_distance*, *max_distance*, *VR_* allows the particle to identify the location of that neighbor in its interaction rings, and thus the appropriate reaction type. The relative distance can be easily calculated using the scan functions provided from the Swarm-Sim simulator.

3) Previous Direction or Heading: A particle makes a step in one round, then, in the next round it is often reasonable to know in which direction it moved. The last headed direction is needed to share with neighbors in the *alignment ring* as shown in Figure 4 to allow them to apply the alignment rule. For this purpose the internal memory of particles are used. After making its decision and before a particle p moves in the chosen direction, it saves this direction in its memory. Thus, if one of the observing particles wants to get the heading of p, it can "ask", by accessing p's memory and reading the direction in which p moved in the previous round. This is how the inter agent communication is simulated in Swarm-sim. So, the frequency of memory access reflects the communication needs of the agents and helps to determine for real robots the network settings.

D. Procedure

At first, each particle creates a list of neighbors to observe named *list_of_neighbors*. After that, they are categorized in three lists based



(a) A lonely particle (b) Uncomfortable situation

(c) Safe situation (d) Critical situation

Fig. 7: Possible situations of a particle in the flock

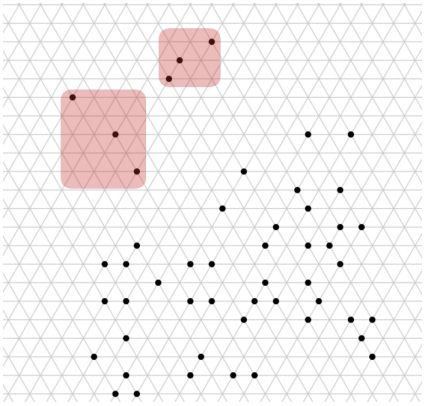


Fig. 8: An example of two groups which left the flock

on the relative distance. Three categories represent the three possible reactions mentioned in Section V-B. The first list is called *closest_neighbors* which includes all neighbors within a distance smaller than *min_distance*. The second one is *middle_neighbors*, there are all particles in a distance between *min_distance* and *max_distance* including the *min_distance* and *max_distance* themselves. The last list is for the rest of neighbors and is called *farthest_neighbors*.
After creating the three lists, the particle interprets its state. Based on the three lists four states for a particle can be distinguished as depicted in Figure 7.

1) Lonely Situation: If all lists are empty, the particle is lonely and it moves in a random direction

2) Uncomfortable Situation: It is uncomfortable for the particle when at least one of its neighbors comes closer than the minimum distance. In this case, the particle collects the relative direction to all neighbors within the separation ring and calculates the number of neighbors in each direction. After that, it goes in the direction in which the fewest neighbors are reachable.

3) Safe Situation: The particle is in a safe situation when just the *alignment-ring* and the *cohesion-ring* contain neighbors. In this case, the particle adapts its heading to the direction towards which the most of its neighbors of the alignment ring are heading. To obtain a better cohesion the farthest neighbors must also be considered. As shown in Figure 8, a mini group of particles can quit the flock together when they are just focused on each other and constantly in safe situation.

So the particle chooses the direction towards which the most of its middle neighbors are heading and in which the most of its farthest neighbors are reachable.

4) Critical Situation: It is critical for a particle when the nearest neighbor is in a distance greater than the maximum distance, which means that the *separation* and *alignment* rings include no neighbors. Thus, there is a danger to leave the flock and get lost. In this case, the particle calculates in which direction it approaches the majority of neighbors and moves toward them.

E. Termination of the Simulation

The program has two termination conditions. The first one refers to an error termination when the the flock splits up. The second one is a success termination which occurs when the predefined *max_rounds* is reached.

1) Faulty Termination: It is possible that during a simulation a flock division occurs as explained in Section III-F. This can be tested by calculating the size of one flock and comparing it with the number of particles in the world. In order to calculate the size of one flock, the flocks graph, as explained in Section III-E, is considered. Without loss of generality, one particle is picked and starting from its node a breadth-first search is applied to get all members of the flock to which it belongs. After that, the number of the flock members is compared with the total number of particles in the world. If an inequality is obtained, it means that some particles left the flock. This flock division leads to the failure termination of the program.

2) Successful Termination: If the flock holds together during the whole simulation, the program terminates successfully when the *max_rounds* is reached. *max_rounds* is a configurable parameter of the Swarm-Sim simulator. It defines the maximum number of steps that agents can make during a simulation.

VI. Evaluation Metrics

In order to evaluate the chosen settings, relevant metrics have to be identified. The evaluation is done for various values of configurable distances.

A. Success Rate

As previously stated in Section V-E, the program meets one of two termination conditions, on the basis of which the success rate can be determined. A $success_rate = 100\%$ occurs by a successful termination. Otherwise, the rate can be calculated using this formula:

$$success_rate = \frac{number_of_rounds}{max_rounds}$$

E.g., if *max_round* is set on 100 and the program terminates after 50 rounds, then the success rate is 50%.

B. Number of Collected Data

During the simulation, particles observe their neighbors and have the possibility to collect three information from each one of them. The sum of of the number of collected information is separately calculated in order to analyze the computing and communication costs.

TABLE II: Parameters of first experiment

No	min_ dist.	max_ dist.	VR	Width of the Separation Ring
1	1	3	7	0 (No separation)
2	2	4	8	1
3	3	5	9	2
4	4	6	10	3
5	5	7	11	4
6	6	8	12	5

TABLE III: Parameters of second experiment

No	min_ dist.	max_ dist.	VR	Width of the Alignment Ring
1	2	2	6	0 (No alignment)
2	2	3	7	1
3	2	4	8	2
4	2	5	9	3
5	2	6	10	4
6	2	7	11	5

TABLE IV: Parameters of third experiment

No	min_ dist.	max_ dist.	VR	Width of the Cohesion Ring
1	2	4	4	0 (No cohesion)
2	2	4	5	1
3	2	4	6	2
4	2	4	7	3
5	2	4	8	4
6	2	4	9	5

An agent uses for its decision making, the **relative distance** to each one of its observed neighbors. Furthermore depending on its situation, it needs more information to collect. If the particle is in:

- An uncomfortable situation: the agent collects the **relative directions** of its neighbors within the separation ring.
- A safe situation: the agent applies the alignment and cohesion rules. Thus, it needs the headings received through **memory reads** of neighbors within the alignment ring and the **relative directions** of neighbors of the cohesion ring.
- A critical situation: the agent collects the **relative directions** of its neighbors within the cohesion ring.

C. Density

To evaluate the form of the flock, and how far the particles are from each other, the flock density is calculated in each round. For this purpose, the radius (*DensityRadius*) has to be set as needed and remains unchanged during all experiments to obtain consistent results. The *Density* is calculated with the formula explained in Section III-G.

VII. Experiments

In this section, we give a summary of the various experiments to reach the goals specified in Section IV. For all experiments, one scenario was used which contains one flock consisting of 105 particles as shown in Figure 9a. The aim here is to investigate the impact of different individual settings of the interaction rings on the global behavior.

A. First Experiment: Impact of Separation Ring

The value of the *min_distance* defines the width of the *separation ring*. In this first experiment, the widths of both external rings are fixed and the width of the first one is variable. In order to keep the same widths for the other two rings, *min_distance*, *max_distance* and *VR* are simultaneously incremented. For this experiment, the width of the alignment ring is set to 2 hops and the width of the cohesion ring to 4 hops. In Table II you find the setup of the first experiment.

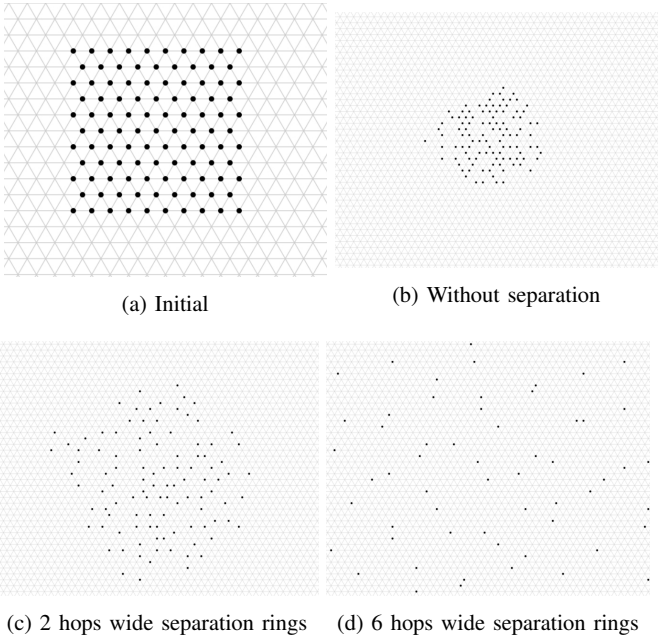
At first, there is no separation and than the separation ring incremented successively by 1 hop until it became 7 hops wide. No separation is meant here that the width of the separation ring is equal to zero. In this case, the distance between pairs of particles must be at least 1 hop. Thus, they cannot have the same location, i.e. the collision avoidance is not violated.

Results show that this ring influences mainly the density of the flock. As consequent, the number of relative distances and the rate of memory access had changed significantly.

1) Success Rate: From a width of 5 hops, the flock begins to be unstable and does not hold during the simulation any more. This can be explained by the fact that the focus of the agents on the separation is too big, that the single neighbors of a particle want to keep a large distance from each other, and thus the particle itself has less neighbors. As a result, it can get out of the flock more easily. This can be seen in Figure 11a from the number of calculated relative distances which reflects how many neighbors particles have in average.

TABLE V: Success rate for different widths of separation ring

Width of separation ring	0	1	2	3	4	5
Success rate	100 %	100 %	100%	100%	100%	98,9%



(a) Initial (b) Without separation

(c) 2 hops wide separation rings (d) 6 hops wide separation rings

Fig. 9: Flock Types with varying Separation Ring Sizes

TABLE VI: Evolution of flock density

Width of first ring	0	1	2	3	4	5
Density[*particles/DensityRadius*]	11,78	6	3,9	2,9	2,4	2,15
Density Radius	6	6	6	7	7	7
Density[*particles/6hops*]	11,78	6	3,9	2,18	1,8	1,6

TABLE VII: Average number of collected information per round

Width of separation ring	0	1	2	3	4	5
Relative distance	4979	2973	2321	2044	1952	1949
Relative direction	486	385	389	440	521	621
Memory reads (Heading)	77	48	39	34	32	29

2) Density: The wider the separation ring is, the less dense the flock is. The initial value for density of the scenario of Figure 9a is $density = 18,85\ particles\ /\ 6hops$. If there is no separation, the average value for density reaches its maximum with *density=11,48 particles/6hop*. The density becomes smaller although there is no repulsion as the particles are moving and the initial compact shape is no longer there. After incrementing the width of the separation ring with only one hop, the density is almost halved. Figure 9b shows how close particles are to each other after setting the value of separation rings width on 0. The density of the flock decreases with separation ring widths of 2 and 6 hops as shown in Figure 9.

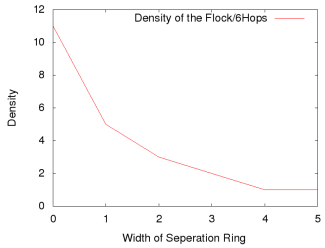
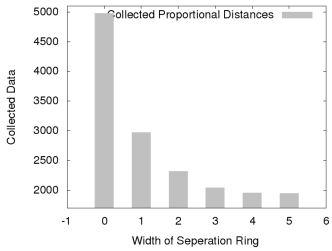
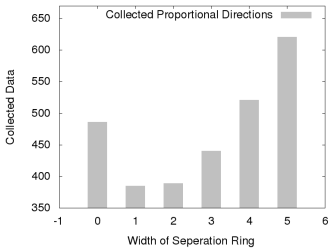


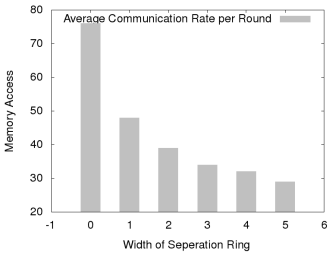
Fig. 10: Flock density with different separation ring widths



(a) Collected distances



(b) Collected relative directions



(c) Communication rate

Fig. 11: Evaluation of the first experiment: Impact of the Separation Ring

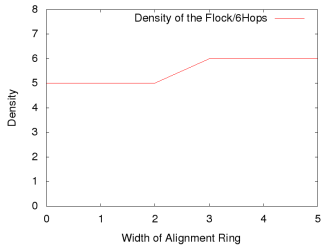
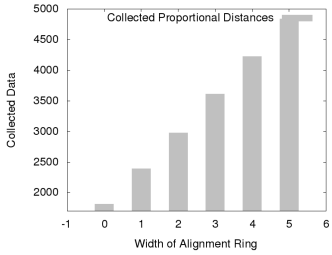
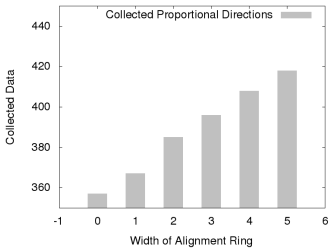


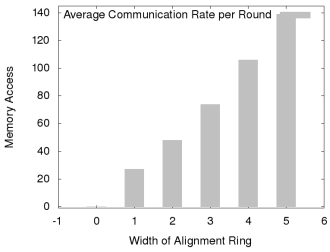
Fig. 12: Flock density with different alignment ring widths



(a) Collected distances



(b) Collected relative directions



(c) Communication rate

Fig. 13: Evaluation of the second experiment: Impact of the Alignment Ring

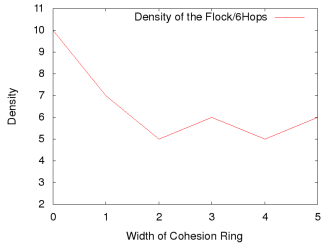
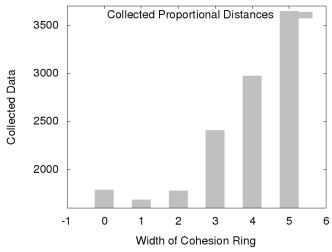
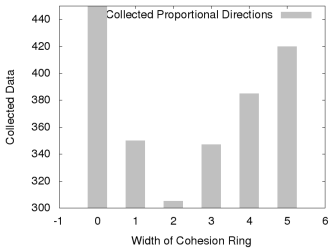


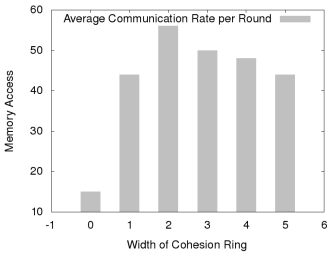
Fig. 14: Flock density with different cohesion ring widths



(a) Collected distances



(b) Collected relative directions



(c) Communication rate

Fig. 15: Evaluation of the third experiment: Impact of the Cohesion Ring

3) Collected Information: Incrementing the width of the separation ring decreases the rate of communication through memory reads. Just to remember, a particle communicates through memory reads only with its middle distant neighbors. With a zero hop wide separation ring, the average value for memory reads per round is 81 accesses per round. After setting the width to 1 hop, it becomes equal to 50 reads per round. This change is due to the fact that the middle neighbors of each particle tries to keep more space between each other. Therefor, they becomes fewer as the width of the alignment ring does not change.

An important decrease in the number of relative distances collected during the simulations is also to notice. A particle calculates the relative distance to all neighbors within its visual range. Although the visual range is getting bigger during this experiment, particles have fewer neighbors due to of density changes.

The number of collected relative directions reaches its minimum when the separation ring is 1 hop wide. Then it increases fast to reach its maximum with an average value of 621 for a separation ring width of 5 hops. This is due to an increase in the number of neighbors within the separation ring, for which the relative direction is needed to avoid collisions with them. Table VII summarizes the resulting aggregated values of the three metrics.

B. Second Experiment: Impact of Alignment Ring

The second experiment investigates the impact of the *alignment ring* size. As in the first experiment, the width of the separation and cohesion rings were fixed respectively to 1 and 4 hops. The width of the alignment ring (WoAR) is incremented step by step from 0 to 5 hops. In Table III you find the setup of the second experiment.

The results show that increasing the width of the alignment ring leads to an increase in computing and communication costs.

1) Success Rate: The alignment ring has no impact on the flock cohesion. For this reason a 100% success is kept despite the changes that occurred during the experiment.

2) Density: The density of the flock remains almost the same as seen in Figure 12. There is only a negligible value deviation of 0,25 due to different flock shapes.

3) Collected Information: By increasing the width of the alignment ring, the value of the visibility range (VR) is automatically increased so that the cohesion ring can maintain the same width. As the repulsion remains the same here, the number of neighbors of each particle increases, as does the number of needed data.

C. Third Experiment: Impact of Cohesion Ring

The last ring or the cohesion ring is defined through max_distance and the visibility range (VR). The width of this cohesion ring

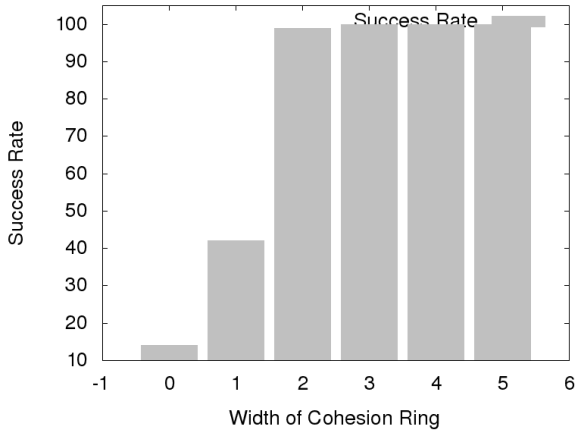


Fig. 16: Success rate with different cohesion ring widths

(WoCR) is equal to the difference between both of them. In this third experiment, the values of *min_distance* and *max_distance* are fixed respectively to 2 and 4 hops and only the VR increments to enlarge step by step the cohesion ring. In Table IV you find the setup of the third experiment.

The results show that the wider of the cohesion ring is, the better the flock holds together, but also leads to higher computational costs.

1) Success Rate: The results depicted in Figure 16 show high impact of the cohesion ring on success rate. As the flock centering rule is applied just with neighbors of this ring, a particle can easily leave the flock if those neighbors disappear fast from its visibility range. If the cohesion rule is not applied, the flock holds only for about 15 rounds, the time taken to reach a satisfying spacing between flock members. 100% success is reached in this experiment only with a cohesion ring width of at least 3 hops.

2) Density: When the cohesion ring is 0 hops wide, the calculated density is a little high, as seen in Figure 14. This is due to the fact that the simulations have a faulty termination after 15 rounds in average. Thus, the particles do not manage yet to maintain the desired distances from each other. In the last four simulations bundles, the success rate is over 90% and the equivalent densities has almost the same values. So, it can be deduced that the cohesion ring has no impact on flock density.

3) Collected Information: When the cohesion rule is violated, the simulations terminate in average after 15 rounds. In this case, the average number of collected distances and directions are higher than in the other simulations. This can be explained by the fact that since the density is initially high and the VR is fix, particles have a lot of neighbors to observe. Thus, they collect more data.

During the rest of the experiment, the number of collected relative distances is highly increased. It reaches its minimum where the width of the separation ring is set to 2 hops. Simultaneously, the number of memory reads reaches its maximum. This means that the particles in a cohesion ring with a width of 2 hops are in safe situations more often than with other width values. Thus, it leads automatically to a decrease in the frequency of occurrence of the critical and uncomfortable situations and consequently a decrease on the collected relative directions.

VIII. Conclusion and Future Work

Using the flocking model presented in this work, it is possible to investigate the emergent effects of changing interaction distances in the flock. The main focus of this paper is on the flocking behavior without considering any boundaries or obstacles. The size of the separation ring influences directly the flock density. The flock gets less dense when the width of the separation ring increases. Although a low density leads to the decrease of collected relative distances and memory reads, it increases the risk of flock division.

Increasing the width of the alignment ring has no influence on density or success rate but leads to enlarging the communication need of particles. Thus, each particle has more neighbors to observe and therefore more data to collect. However, the cohesion ring has a significant impact on the success rate. The wider it is, the better the flock holds together. Widening this ring leads to an increase of the number of collected data which is due to the increase of the observed particles in the visibility range with the same separation ring size.

After experimenting with different settings, the flock adopting the distance interactions shows the best cohesion by setting the minimal distance, i.e. separation ring size, to 2 hops, the maximal distance, i.e. the alignment ring size, to 3 hops and the visibility range on 7 hops. Thus, the separation and alignment rules are not violated and an average success rate of 100% was reached after running 100 simulations with 100 different seed values.

As all scientific works, we face some limitations for different reasons. At first, the obtained results still dependent on the given flocking model. Respecting another model may deliver different results. Furthermore, the Swarm-Sim simulator allows only to simulate a swarm in a two dimensional environment. Although, adding another dimension could lead to unexpected changes on the global behavior of the flock. In addition, the experiments carried out in this work, used only one initial flock placement. So, it remains open for future work to investigate whether the initial form and size of the flock has an impact on the obtained results. Moreover, the designed environment is kept simple. There are no obstacles and no boundaries which would disturb the agents during their motion. For a robot flocking behavior, we aim to improve the communication model and add a limitation in how the agents can communicate according to the opportunistic communication models defined in [9].

Acknowledgments

We thank the Arab-German Young Academy of Sciences and Humanities and Dr. Ahmed Khalil for their support in this research.

References

[1] I. D. Couzin, J. Krause, R. James, G. D. Ruxton, and N. R. Franks, "Collective memory and spatial sorting in animal groups," *Journal of theoretical biology*, vol. 218, no. 1, pp. 1–11, 2002.

[2] J. T. Emlen, "Flocking behavior in birds," *The Auk*, vol. 69, no. 2, pp. 160–170, 1952.

[3] C. W. Reynolds, *Flocks, herds and schools: A distributed behavioral model*. ACM, 1987, vol. 21, no. 4.

[4] S. Khan, S. Momen, N. Mohammed, and N. Mansoor, "Patterns of flocking in autonomous agents," in *2018 International Conference on Intelligent Autonomous Systems (ICoIAS)*. IEEE, 2018, pp. 92–96.

[5] C. Moeslinger, T. Schmickl, and K. Crailsheim, "A minimalist flocking algorithm for swarm robots," in *Europ. Conf. on Artificial Life*, 2009.

[6] O. Witkowski and T. Ikegami, "Emergence of signal-based swarmimg," in *Artificial Life Conference Proc. 14*, 2014, pp. 302–309.

[7] K. Shimada and P. Bentley, "Learning how to flock: deriving individual behaviour from collective behaviour with multi-agent reinforcement learning and natural evolution strategies," in *Proc. of the Genetic and Evolutionary Comp. Conf. Companion*. ACM, 2018, pp. 169–170.

[8] A. Cheraghi and K. Graffi. (2020) Swarm simulator. Heinrich-Heine-University Duesseldorf, Germany. [Online]. Available: https://gitlab.cs.uni-duesseldorf.de/cheraghi/swarm-sim

[9] S. Sati, C. Probst, and K. Graffi, "Analysis of Buffer Management Policies for Opportunistic Networks," in *Proc. of the 25th Int. Conf. on Computer Communication and Networks (ICCCN)*, 2016.