Swarm Intelligence

NWI-IMC042 Natural Computing

Gijs Cornielje s1038844
Max May s1004443
Tim Wiesner s1032181

April 15, 2023

1 Problem Description

Crowd dynamics can be studied by a graphing method called the fundamental diagram (FD). This 2D scatterplot projects the speed of individuals against the local density. Resulting in a plot where the difference in speed of herds, flocks or groups can be seen for different densities. Baglietto and Parisi (2011) studied the behaviour of humans in groups and made an FD for human group dynamics.

Reynolds (1987) designed an algorithm to simulate the movements of birds in a computer program. These simulated birds were called boids. By having each individual boid contain a position and direction and follow simple behaviour of alignment, cohesion and separation, similar behaviour to flocking birds can be achieved. By implementing this algorithm and placing these boids in an experimental setup, the circular racetrack described by Seyfried et al. (2005), we will attempt to use the FD to analyse the motion of the boids and compare this to the motion of humans. Furthermore, we will study the effects of the individual ablation mechanisms alignment, cohesion and separation.

2 Experimental design

To execute the experiments, we created a circular racetrack, consisting of lines and arcs, as shown in the assignment, see Figure 1. Every meter was replaced with 100 pixels. Resulting in a straight part of the racetrack of 400 pixels long and 80 pixels wide. The corners connecting these straights are a pair of half ellipses designed to connect to the straights.

The boid algorithm implementation is based on the version from the lecture. It consists of a canvas of 800 by 600 pixels, on which an amount of boids is placed. These particles consist of a position and a direction, which are both initialized randomly. On every frame the boids are updated based on three forces: alignment, cohesion and separation. These three forces are calculated based on the neighbouring boids. We decided that boids with a distance smaller than 50 were considered neighbours and would therefore influence the movements of a particular boid. The alignment force works by having the direction of the original boid be changed into the average direction of the neighbouring boids. The cohesion force works by adding up a fraction of the difference between the original boid's position and the average position of all neighbours to the original position. The separation force works against this by adding to the direction of the boid, a force scaled by a weight that separates the boid from its neighbours. The parameters and weights of the algorithm were chosen so that the boids performed seemingly natural behaviour. The cohesion and seperation for example were tweaked

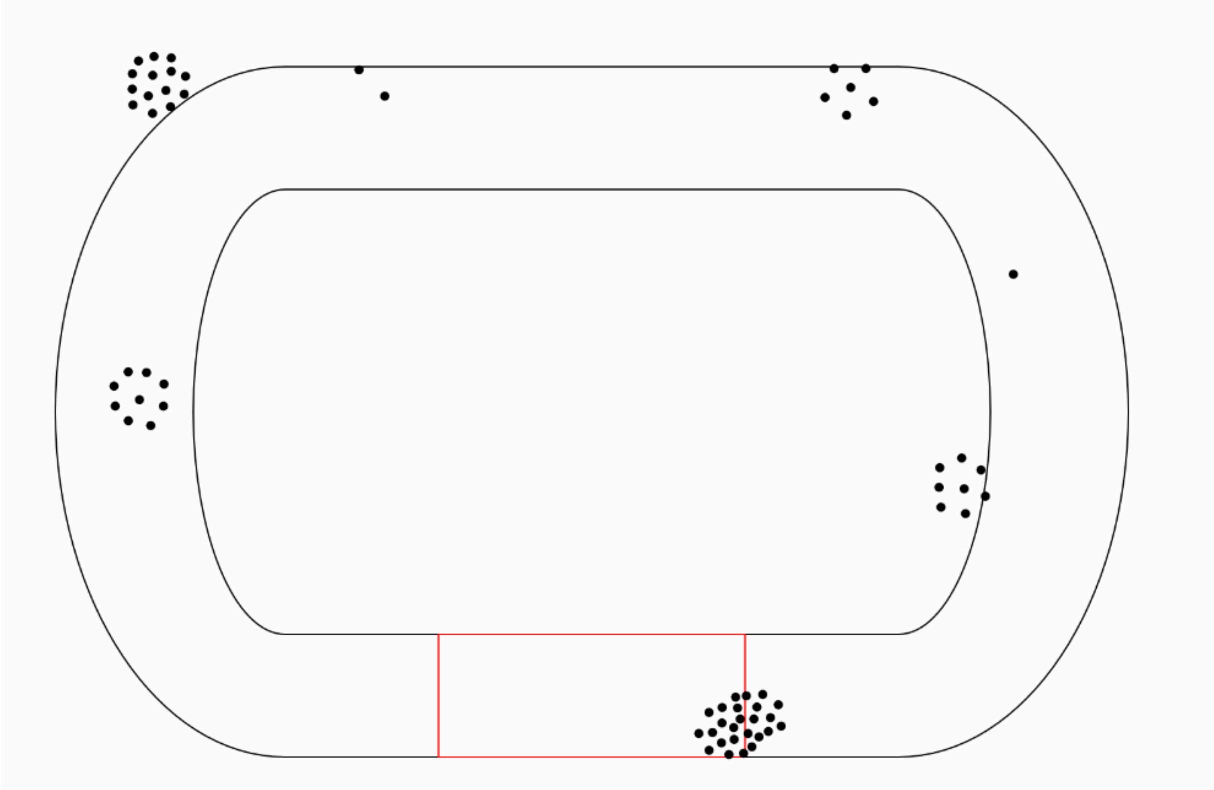

Figure 1: Experimental setup. The boids are following the circular racetrack. The measured section of the racetrack is shown in red.

until the boids would group together without overlapping too much as that would allow a large amount of boids into a space they should not fit, diminishing the effects of crowding.

In addition to these forces a small amount of stochasticity is introduced by adding noise to the direction and a number of extra "forces" are added to make the boids roughly stay on the racetrack and follow its course in a clockwise circular motion. More details on that are described in section 2.1.

For experimental purposes we simulate different amounts of boids (50, 100, 200 and 300 boids), to include a larger range of boid densities. Moreover we do not use every individual boid's passing of the measurement section to generate datapoints for the FD graph. Instead, we take a measurement every 10 timesteps and get the number of boids in the measurement section, as well as their speeds. In real life, the speed of a person cannot be measured by looking at only one timestep, but with the simulated boids, this is trivial.

We thus only take a subset of the boids going through the measurement zone. We find this reasonable, since it makes the computations and thus simulation speed much more manageable. Although there are quite a few "0-samples", meaning that there were no boids in the measurement zone at this time, we still get more than enough valid data points, since we leave the simulation running for a while[1]. The stochasticity of whether a boid happens to be in the measurement zone at a given timestep is also taken care of by running the simulation for long enough, and the reasonable size of the zone.

[1]2000 timesteps was our standard setting, creating 200 samples with a varying number of boids.

2.1 Implementation details

We chose to implement the racetrack in terms of "soft" constraints by adding extra "forces" pushing the boids onto the racetrack and creating a circular motion around it. This way, the randomly spawning boids naturally find their way onto the racetrack and follow its path. We found this implementation more interesting, also because the other forces acting on the boids can pull them off the racetrack for a moment, leading to more interesting interactions in the swarm.

It took us quite some time to fine-tune the boundaries and intensities of these extra forces to reach a satisfactory result. While the boids do not perfectly stay on the racetrack all the time (sometimes also because they are affected by the ablation forces), they roughly follow its shape and deliver, in our opinion, interesting results.

The code snippet for that is shown in the Appendix (section 6).

3 Initial simulations

From some initial simulations we plotted the fundamental diagram describing the boids' movement in terms of their speed plotted against their density (in boids/m^2). This is shown in Figure 2. We see that the density of the boids varies between almost 0 and 35 boids/m^2 in our experiments. The speed varies more with smaller densities, but stays almost constant when densities get higher. This can be explained by the fact that large density samples mostly consists of large groups, which move with a very consistent speed, as discussed below. The smaller densities are often made up of individual or small groups of boids, which move more freely.

When observing the boid algorithm execute, we notice a few things happening in every run:

1. **The boids spawn randomly over the whole area.** Because of the forces we implemented, they are gradually pushed upon the racetrack and start to move in the aforementioned circular fashion.

2. **The forces keep the boids on the racetrack, but not perfectly.** Especially those that move close to the edge of the racetrack will often overshoot a corner and then bounce back onto it. Implementing the forces around corners was the most difficult part, but after some fine-tuning, the boids stay on the track more or less consistently.

3. **The boids form circular groups quickly.** Having the same forces act on them and being encouraged to move in the same direction by the alignment mechanism, the boids quickly cluster into groups. We suppose that it is the balance between the cohesion and separation parameters that makes the boids keep the optimal distance from each other, resulting in a circular shape. Because our racetrack is not implemented as a hard constraint, the boids do not bounce off of its boundaries, but can move over the line. This usually retains a group's shape, and a group of boids almost acts as if it was an individual boid. So a group may also overshoot a corner as a whole, and then get pushed back onto it.

4. **Without any other boids or groups in the neighbourhood, the speed stays very consistent.** Only when two groups get close to each other, their speed is affected and they tend to merge into one.

5. **Some boids move off the track.** Because of the different forces acting on the boids, some of them sometimes get pushed off the racetrack and then move back onto it. When

a group of boids passes the measurement area, that also means that some boids might fall outside of it.

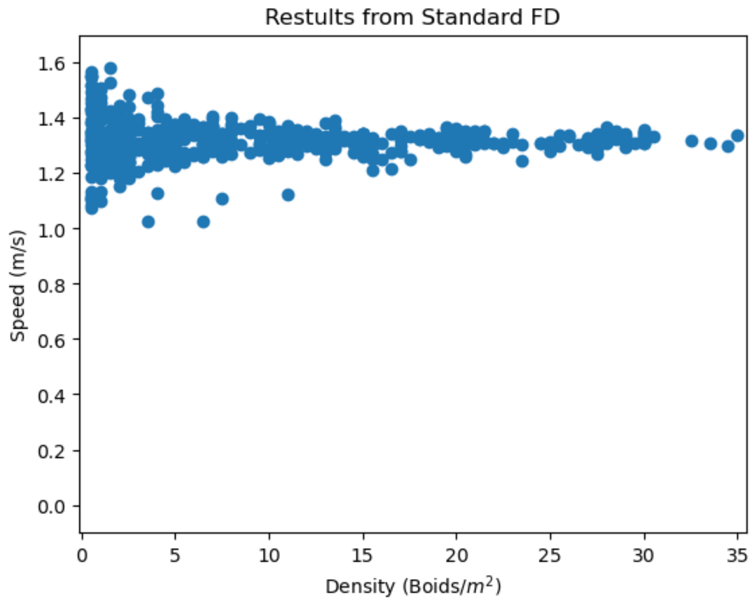

Figure 2: The fundamental diagram with alignment, cohesion and separation. Experiments run for 2000 time steps, with 50, 100, 200 and 300 boids.

To answer the question whether the boids behave in a comparable way to humans, quickly comparing the fundamental diagrams of humans (see assignment description or Baglietto and Parisi, 2011) and the one of the boids (see Figure 2) shows a clear difference. While the humans' speed is negatively correlated to their density, the speed of the boids is not really affected by their density. Large groups of boids tend to form a neat circle and move at a very consistent speed. The humans' movement, on the other hand, is impaired by large densities, while only small densities allow for quicker movement.

4 Ablation study

For the ablation study, the Alignment, Cohesion and Separation variables were switched off separately and also in pairs. The results were plotted in scatter plots, where the density (in Boids/m^2) is plotted against the average speed (in m/s) of the boids. This fundamental diagram with all three elements enabled is shown in Figure 2.

4.1 Alignment

For the alignment variable to be switched off we added an extra line of code where a vector of (0, 0) was created in its place. This was done due to the fact that the code would not run otherwise. This resulted in a clear decrease on the average speed for the boids. Moreover, a notable change can be seen in this fundamental diagram (see fig. 3a) as opposed to the aforementioned standard fundamental diagram (fig. 2), namely the fact that the density varies more strongly. In fig. 3a, the density varies between almost 0 and 12. We see that without alignment, the boids do not move together as a group in the same direction. Instead the movement is random and often causes the boids to “bump” into each other, resulting in the decrease of movement speed.

With cohesion being turned on, the groups are still being formed, but their movement is more random. Big groups do not stay together as neatly as with alignment.

4.2 Cohesion

Without cohesion the results showed that the densities varied less than the standard fundamental diagram (fig. 3b). Now they varied between 0.5 and 6 boids/m^2. This change in density could be explained by the fact that without cohesion, it is more likely that groups of boids separate. However due to the fact that the boids still have alignment and separation to each other, they will generate a more uniform distribution over the entire field. Since no boids will form a group, but they all have to try and keep distance to each other and move in the same direction. This results in decrease in varying densities.

4.3 Separation

Without separation the boids will form dense clusters. The cohesion variable allows for the boids to move to each other, however since there is no separation the boids will move on top of each other. Visually it looks like there is one boid, but this is actually a dense group of boids. This results in the spread of densities seen in figure 3c. Due to the nature of randomness in the initializing phase, it is likely that when a group is formed, this group will not separate anymore.

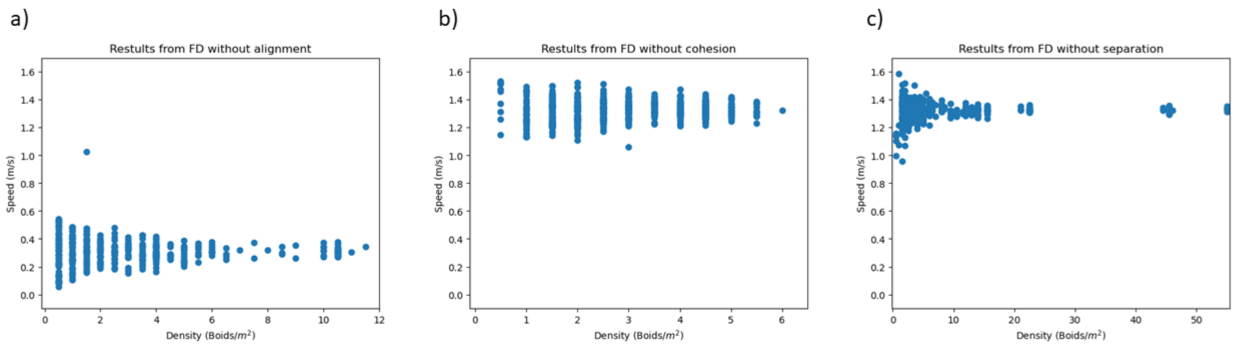

Figure 3: Fundamental diagrams without (a) Alignment (b) Cohesion (c) Separation.

4.4 Combinations

For the fundamental diagrams without combinations of the variables, we will firstly discuss alignment and separation, alignment and cohesion and lastly cohesion and separation.

Simulating the boids without alignment and cohesion, results in a fundamental diagram (fig. 4a) that represents a diagram, which is the combination of diagrams fig. 3a,b. The average speed again matches that of fig. 3a, while the variation in density lies between fig. 3a and b. This is probably caused by the fact that the boids now can move in random directions as opposed to all in the same direction and since there is no cohesion only separation, the boids will spread uniformly across the field. Resulting in less varying densities.

Secondly, simulating the boids without alignment and separation, results in a fundamental diagram (fig. 4b) that shows the same characteristics as the fundamental diagrams as shown in fig. 3a,c. This is probably caused by the fact that the boids now will form groups, but since the boids will not follow the same trajectory smaller groups will be formed that constantly "bump" into each other. Resulting in the lower average speed. Moreover since there is no separation it is less likely that boids will split off groups to form new bigger groups.

Lastly, the resulting fundamental diagram of the simulation without cohesion and separation (fig. 4c), resembles the standard fundamental diagram. This is likely caused by the fact that now all the boids will have the same trajectory and whether groups will be formed is now more random, since there is no cohesion. Moreover since there is no separation, the boids will not be punished for moving towards each other.

4.5 Without everything

For the last simulation all the variables (alignment, cohesion and separation) are switched off. The resulting fundamental diagram shows on average a lower speed and a less varying density of the boids (fig. 5), in contrast to the original fundamental diagram. This is likely caused by the fact that the boids now move more randomly. They do not share a collective trajectory anymore and whether or not they form groups is also more random. Resulting in a more stochastic distributed diagram with low speeds.

5 Conclusion

With the present experiments, we consider the boid algorithm as proposed by Reynolds in 1987. We compared our observations, summarized in fundamental diagrams, with the findings of Baglietto and Parisi (2011) on human behaviour. We have found that the boids behave fundamentally different to humans. While humans are affected by the density in their environment and tend to move more slowly with more other people around them, the boids' speed is very consistent, even (or especially) in large groups. We conclude that the boid algorithm with the ablation mechanisms of alignment, cohesion and separation, can not model human behaviour accurately.

In the ablation study, we experimented with turning individual mechanisms on or off to study the effects of alignment, cohesion and separation. We noticed that alignment allows for a collective trajectory, increasing the average speed of the boids. Turning it off resulted in more random movements. Cohesion allowed for the forming of groups, turning it off resulted in a more haphazard way of forming groups. Lastly, separation allows for the boids to keep distance to each other and turning it off resulted in the boids to move on top of each other, which is not realistic if you want to compare the boids to humans.

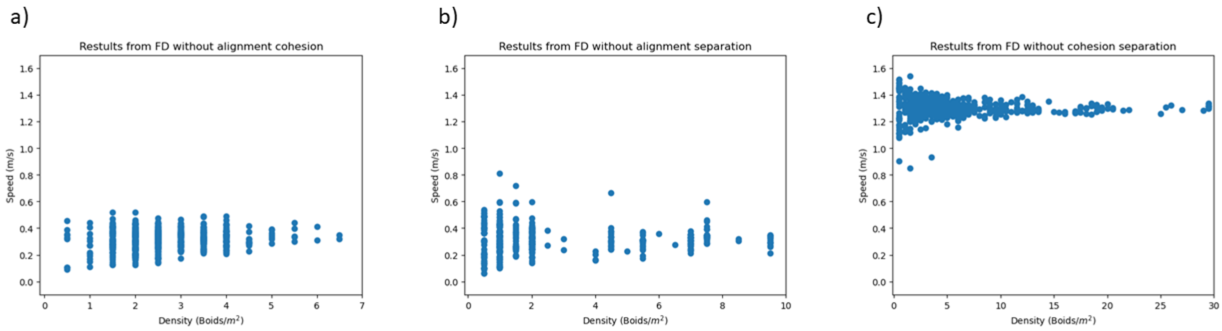

Figure 4: Fundamental diagrams without (a) Alignment and Separation (b) Alignment and Cohesion (c) Cohesion and Separation.

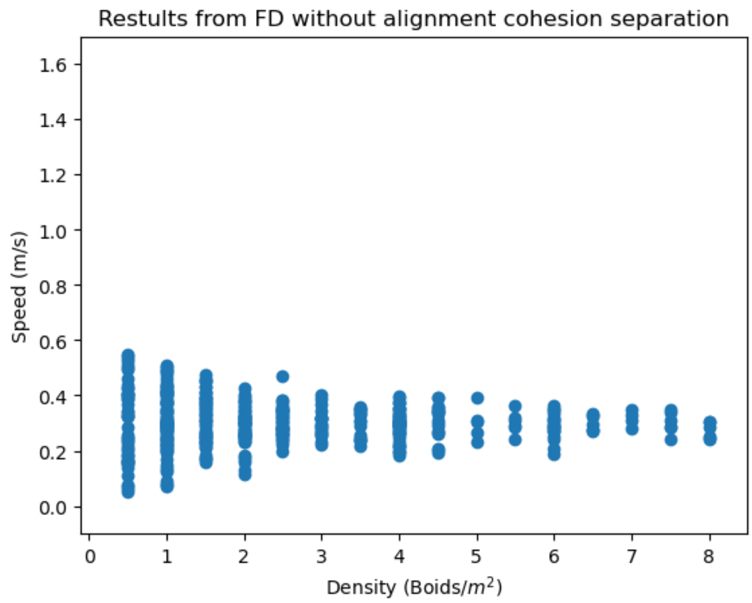

Figure 5: Fundamental diagrams without Alignment, Cohesion and Separation.

References

Baglietto, G., & Parisi, D. R. (2011). Continuous-space automaton model for pedestrian dynamics. *Phys. Rev. E*, *83*, 056117. https://doi.org/10.1103/PhysRevE.83.056117

Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, 25–34.

Seyfried, A., Steffen, B., Klingsch, W., & Boltes, M. (2005). The fundamental diagram of pedestrian movement revisited. *Journal of Statistical Mechanics: Theory and Experiment*, *2005*(10), P10002.

6 Appendix

The following code is used to steer the boids onto the racetrack and make them move in a circular movement around it.

```
// circular motion
if (150 < this.pos.x && this.pos.x < 650){
  if(230 < this.pos.y){
      this.dir.add(createVector(-0.3, 0))
  }
  if(this.pos.y < 420){
     this.dir.add(createVector(0.3, 0))
   }
}

if(100 < this.pos.y && this.pos.y < 550){
  if(this.pos.x < 200){
    this.dir.add(createVector(0, -0.1))
  }
  if (this.pos.x > 600){
    this.dir.add(createVector(0, 0.1))
  }
}

// push particles onto racetrack
if (100 < this.pos.x && this.pos.x < 700){
  // straight lines
  if (this.pos.y < 100){
    this.dir.add(createVector(0, 0.1))
  }
  if(180 < this.pos.y && this.pos.y < 325){
    this.dir.add(createVector(0, -0.1))
  }
  if(325 < this.pos.y && this.pos.y < 470){
    this.dir.add(createVector(0, 0.1))
  }
  if(550 < this.pos.y){
    this.dir.add(createVector(0, -0.1))
  }
}else{
  // curves
  let middle = createVector(400, 325)
  let dist = abs(p5.Vector.dist(middle, this.pos))

  if(this.pos.x < 400){
    // left half
    if(dist < 280){
      this.dir.add(createVector(-0.3, 0))
    }else if(dist > 330){
      this.dir.add(createVector(0.3, 0))
    }
  }else{
    // right half
    if(dist < 280){
```

```
      this.dir.add(createVector(0.3, 0))
    }else if(dist > 330){
      this.dir.add(createVector(-0.3, 0))
    }
  }
}
```