# Multi-Scale Swarm Dynamics

**Alex Dembélé, Mathilde Bonin, Dmitrii Timkin**
Institut Polytechnique de Paris
{alex.dembele, mathilde.bonin, dmitrii.timkin}@ip-paris.fr

## 1 Introduction

The purpose of this project is to extend swarm-formation models to multiple abstraction levels, or scales.

Generally, a swarm is formed by having each mobile agent follow a set of basic rules – e.g. staying within a distance range from its neighbours and moving in a similar direction to its neighbours.

Our main goals are:

- Provide a metric for determining when a swarm is formed, determine when several swarms are formed
- Provide a method to determine a swarm's centroid and movement trajectory
- Analyse the scalability of swarm formation with the number of agents with various rule configurations

We will answer the following questions:

- As compared to the basic model, does the multi-level approach help a single child model to converge towards a single swarm when the number of agents increases?
- Does the multi-level approach allow to converge to a single swarm from all children swarms when multiple children are created? how does this scale?
- Can you envisage an extended model with more than two levels?

To implement this model we use NetLogo[1], a programming language and integrated development environment for agent-based modeling.

Our work can be found at https://github.com/alexdembele/MultiScaleSwarmDynamics

## 2 Base Project

We built our model upon the flocking model from NetLogo's models library [2]. This model implements flocking of birds.

The birds follow three rules: "alignment", "separation", and "cohesion".

- "Alignment" means that a bird tends to turn so that it is moving in the same direction that nearby birds are moving.
- "Separation" means that a bird will turn to avoid another bird which gets too close.
- "Cohesion" means that a bird will move towards other nearby birds (unless another bird is too close).

When two birds are too close, the "separation" rule overrides the other two, which are deactivated until the minimum separation is achieved. The three rules affect only the bird's heading. Each bird always moves forward at the same constant speed.

## 3 Launch of the project

To launch simulation open Centroids.nlogo with NetLogo app. Then click on "setup" button and after than on "launch" button. You can adjust parameters and speed by using controls in top and bottom layers of simulation.

## 4 Project Architecture

We added some features to the base project in order to perform multi-scale swarm dynamics.

### 4.1 Swarm detection

The first step is to detect swarm. The swarm detection relies on the distance between agents. If two agents are close enough, they belong to the same swarm. But this is not enough, we need to deal with the case where two swarm cross each other. They do not form only one swarm. Hence the need to consider the heading of agent.

$$\textbf{If two agents have close headings and are close enough},\ \textbf{then they belong to the same swarm}.$$

The two parameters can be adjusted. By default, the delta in headings (deltaDirection) should be inferior to 12° and the distance between the agents (distanceSwarm) inferior to 4 pixels for two agents to belong to the same swarm.

Then we need to assign the same Id to each agent in the same swarm. To do this we create a link between to agent that respect the previous proposition. Then we assign an Id to each agent in the simulation; Finally, for each link, we assign to each agent attached to the link the min Id of the two. We repeat the last step 300 times to be sure of the good propagation. For consistency matter, we compute the swarm detection every 5 steps.

### 4.2 Centroid computation

We call a centroid the barycenter of a swarm. We compute the mean of coordinates and headings to describe the centroid.

$$\left(\sum_i^n \frac{x_i}{n}; \sum_i^n \frac{y_i}{n}\right) \qquad \arctan\left(\frac{\sum_i^n \sin(h_i)}{\sum_i^n \cos(h_i)}\right)$$

The coordinates of the centroid — The heading of the centroid

Centroids are recomputed every tick by using the Id of the flock to determine the members.

### 4.3 Level-Space

To deal with the multi-scale aspect, we used an extension of NetLogo : Level-Space [3].

It enables to create parent-child simulation. In our case, the child simulation is our custom Flocking and the parent simulation is like flocking but with centroid. We need to send from the child to the parent the state of centroïds. Then we can compute the flocking dynamics on centroïds in the parent simulation. The parent simulation dynamics is basically a copy paste from Flocking.

### 4.4 Feedback

After running the dynamics for the parent model, we need to give a feedback from this simulation to the child simulation. We create a function that tends to turn agent, corresponding to a centroïd, in the child toward the new heading of the centroïd in the parent simulation. To have a clear impact on child, we perform 5 step of parent dynamics before giving a feedback.

## 5 Parameters

There are different parameters for top and bottom layers of model.
Parameters of the bottom layer:

1. **population** – number of birds participating in swarms creation

2. **vision** – the distance that each bird can see 360 degrees around it
3. **minimum-separation** – minimum distance between two birds. After reaching it "separation" rule starts to work.
4. **max-align-turn** – maximum angle at which bird can rotate to satisfy "alignment" rule
5. **max-cohere-turn** – maximum angle at which bird can rotate to satisfy "cohesion" rule
6. **max-separate-turn** – maximum angle at which bird can rotate to satisfy "separate" rule
7. **distanceSwarm** – minimum distance between birds to detect them as one swarm
8. **deltaDirection** – minimum angle between birds to detect them as one swarm
9. **max-swarm-turn** – maximum angle at which birds can be rotated influenced by centroids

Parameters of the top layer:

1. **number_worlds** – number of created layers. Experimental feature, in practice use 1.
2. **vision** – the distance that each centroid can see 360 degrees around it
3. **minimum-separation** – minimum distance between two centroids. After reaching it "separation" rule starts to work.
4. **max-align-turn** – maximum angle at which centroid can rotate to satisfy "alignment" rule
5. **max-cohere-turn** – maximum angle at which centroid can rotate to satisfy "cohesion" rule
6. **max-separate-turn** – maximum angle at which centroid can rotate to satisfy "separate" rule

There are some hidden parameters that can affect dynamics but there are more related to dynamic modelization. You can change the number of step between each swarm computation in children. You can change the number of step in the parent model before giving the feedback. The current value is 5 but we tested with 25 and it have not affected the dynamics so much. Maybe, it is an effect of the others parameters. When you conduct experiment you can change the file in which the data will be writen.

# 6 Tests

## 6.1 Automatic process

We created an automatic pipeline to conduct tests. It performs the dynamics during 335 ticks of the child 20 times for a given set of parameters. It writes the result into a csv file. To improve this pipeline, we further want to use pyNetLogo to automatically choose parameters in order to do optimization and grid search.

## 6.2 Metrics

We need to define a metric to quantitatively compare our experiences. It is difficult to define good metrics, so we will use the number of centroids over the iterations. Then we will compute max, min and mean of centroïd number. Then we need to analyse the impact of parameters by varying them.

Another metric could be the headings of centroids. It will be use to analyse the dynamic of movement, how the agent move in the simulation. We could use histogram to plot headings for a frame, but it is hard to have a plot over time.

## 6.3 Difference between classic swarm dynamics and multi-scale

First, it is important to see if switching to a swarm of swarms with our feedback has an effect on the dynamics.

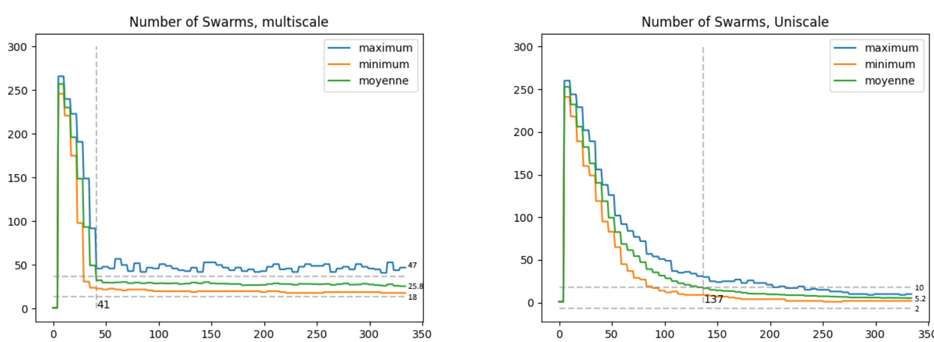
To do this we conduct to set of experience, in the first one, the feedback is activated, but not in the second one.

Here the parameters

| Bottom Layer | |
|---|---|
| population | 300 |
| vision | 5 |
| minimum-separation | 0.5 |
| max-align-turn | 4.25 |
| max-cohere-turn | 3.5 |
| max-separate-turn | 0.75 |
| distanceSwarm | 4 |
| deltaDirection | 12 |
| max-swarm-turn | 10 |

| Top Layer | |
|---|---|
| vision | 40 |
| minimum-separation | 1 |
| max-align-turn | 1 |
| max-cohere-turn | 1.25 |
| max-separate-turn | 2 |

And then curves

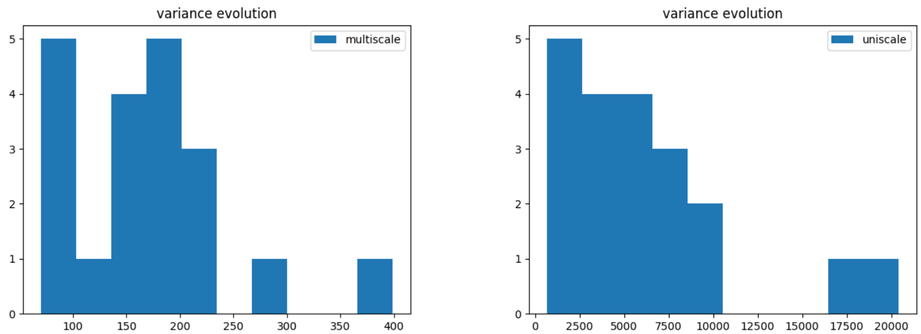



(a) Multi-scale (b) Uni-scale

Figure 1: Number of swarms in both dynamics

We notice that in multi-scale and centroids dynamics, there are more swarm at the end. It comes from the separate effect in the top dynamics. Agents are less likely to meet other swarm once a swarm structure is set. We can also see that the transitory regime is shorter without the feedback, the final state is reached faster. This behaviour may depends on the others parameters.

We can also analyse the composition of swarms at the end with the number of agent in each swarm. For this, at the end of each iteration we compute the variance of agent number in each swarm.




(a) Multi-scale (b) Uni-scale

Figure 2: Variance of agent number in swarms

We notice that despite there are less swarm in uni-scale, the distribution of agent in swarms are far from being equal, in the multi-scale dynamic, the distribution is most equal although there are more swarm.

### 6.4 Multi-scale stabilize headings

After some experiences, we noticed that multi-scale with proper parameters can stabilize dynamics even if the child parameters are badly chosen.

For example, if we consider the $\mathbf{max-separate-turn}$, by increasing it the swarm have difficulties to form and move in random direction. But if we activate the feedback of multi-scale dynamics, then swarms are more likely to form and end going in the same direction.

To analyse that, we plotted the variance of headings over iterations

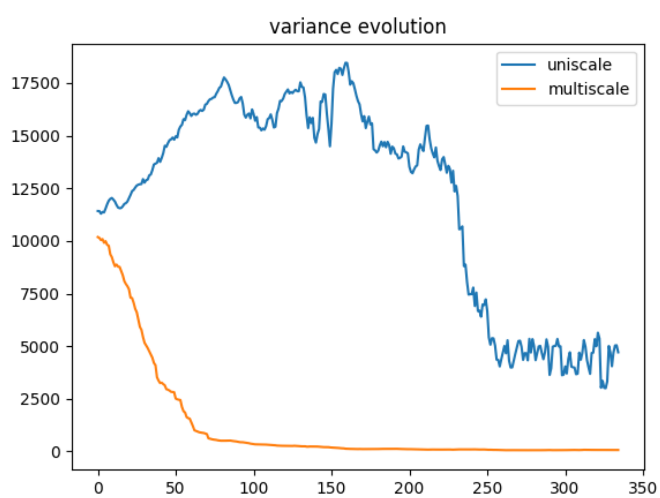



Figure 3: Heading variance over iteration

We can see that in multi-scale dynamics, headings are quickly decreasing and have a lower variance at the end. All agent end to turn towards the same direction.

### 6.5 Dynamic switch

We will test both dynamics effect when we activate/desactivte multi-scale during a simulation. We set the parameter at the boundaries of stability for uni-scale, meaning that if we increase a certain parameter by 0.25, no swarm will form. In this case, the uni-scale dynamics create a lot of swarms and the mutli-scale decrease the number of swarm. We can see that on this figure.

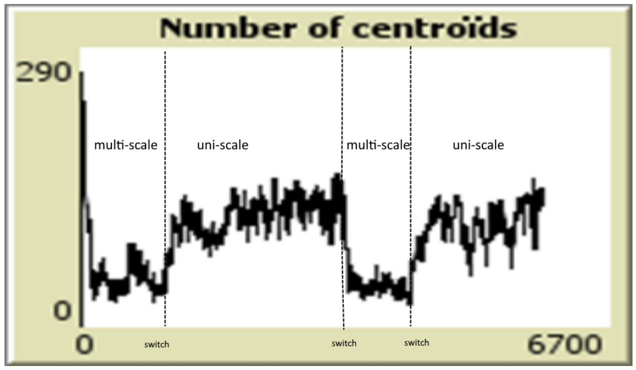
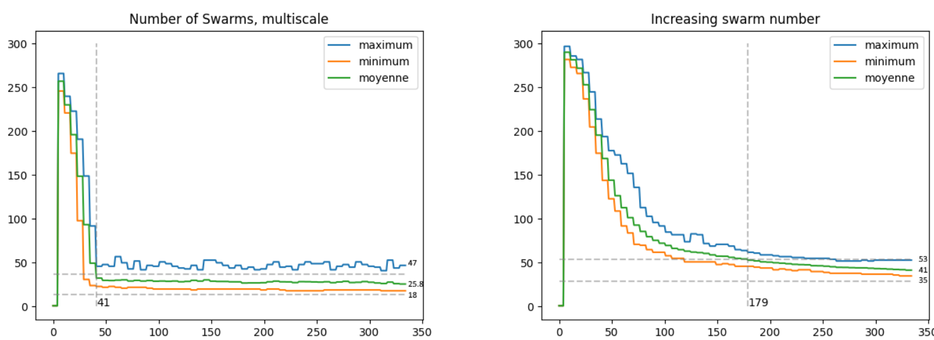



Figure 4: Dynamic Switch

## 7 Increasing the number of swarm

Given a certain set of parameters, we tried to increase the number of swarm by changing some parameters. The best parameters to increase swarm number is vision in parent model and distance swarm in the child model.




(a) Basic Multis-cale

(b) Increased swarm number

Figure 5: Number of swarms in both dynamics

We notice that indeed the number of swarm is a bit higher but also that the transitory regime is slower with a higher swarm number. It is harder to reach final state with many swarm.

### 7.1 Parameter impact

We analyse quickly the effect of each parameters on dynamics. These test are conducted on a set of parameters, consequently the results might change for another set of parameters.

We can see that some parameters do not have the same effect if they belong to the parent model or to children.

Here are a comparison table. A + mean that a increase in this parameter make swarm more likely to form. This is the opposite for -, an increase results in swarm are less likely to form.

Parameter impacts are quite obvious for the child model. An higher vision enable an agent to form swarm with further agent, as an increase max-align-turn. Increasing parameter for swarm detection result in bigger swarm in that they

## Parameter impact
(child model)

| Parameters | Effect |
|---|---|
| *max-separate-turn* | Increase => swarms are less likely to form |
| *vision* | Increase => swarms are more likely to form |
| *minimum-distance* | Increase => swarms are less likely to form<br>=> Increase spacing in swarms |
| *max-align-turn* | Increase => swarms are more likely to form |
| *max-cohere-turn* | Increase => swarms are less likely to form |
| *distanceSwarm* | Increase => bigger swarm |
| *deltaDirection* | Increase => bigger swarm |

(a) Child model

## Parameter impact
(parent model)

| Parameters | Effect |
|---|---|
| *max-separate-turn* | Increase => swarms are more likely to form |
| *vision* | Increase => big swarms are less likely to form |
| *minimum-separation* | Increase => swarms are less likely to form<br>=> Could be increase a bit after reaching balance |
| *max-align-turn* | Increase => swarms are less likely to form |
| *max-cohere-turn* | Increase => swarms are more likely to form (not too high)<br>=> Increase time to stabilize |
| *max-swarm-turn* | Increase => depend on others parameters ! |

(b) Parent model

Figure 6: Parameters impact

## Parameter impact comparison

| | Child | Parent |
|---|---|---|
| *max-separate-turn* | **-** | **+** |
| *vision* | **+** | **-** |
| *minimum-separation* | **-** | **-** |
| *max-align-turn* | **+** | **-** |
| *max-cohere-turn* | **-** | **+** |

Figure 7: Parameter comparison

restrain less the detection of swarm. Max-separate-turn, minimum-distance making swarm less likely to form is also quite obvious in that they constrain more the swarm formation. But max-cohere-turn is strange cause we could expect that it improve the agglomeration of agent.

Parameter impacts are more difficult to understand for parent model. The vision make swarm less likely to form in that swarm start seeing each other further and the separate effect make swarm dodging each others. Max-separate turn and max-align-turn make swarm less likely to form because they stabilize parnt dynamics and prevent swarm to meet each others. This is the contrary for max-cohere-turn.

Concerning max-swarm-turn, the higher is this parameter, the more agents will tend to turn towards the same direction cause each agent of a swarm will head in the same direction and the align effect will spread and average directions through swarms.

### 7.2 Adding weight on parent dynamics

We tried to add weight on parents dynamics to take into account the number of agent in each swarm. We weighted some function of parent model by the ratio of swarm weigh. However, this change have not changed the dynamics in term of swarm number. We can have an intuition on that owing to the fact that a weighting basic function will just change the attraction of each swarm to others. The swarm with less agent will be absorbed by the bigger one in all case. To improve our weighted model, we could use a model similar to gravitation to add an interaction between swarm.

## 8 Conclusions

We improved a swarm dynamic model into a multi-scale swarm dynamic model. This process require a feedback from top level dynamic to low level. Multi-scale dynamic significantly change the system behaviour. We implemented a procedure to conduct test and we carried out a study that compare classic swarm dynamic and multi-scale one.

## References


[1] U. Wilensky. Netlogo. http://ccl.northwestern.edu/netlogo/, 1999. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

[2] U. Wilensky. Netlogo flocking model. http://ccl.northwestern.edu/netlogo/models/Flocking, 1998. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

[3] A. Hjorth, B. Head, and U. Wilensky. Levelspace netlogo extension. http://ccl.northwestern.edu/rp/levelspace/index.shtml, 2015. Evanston, IL: Center for Connected Learning and Computer-Based Modeling, Northwestern University.