

# **Simulation Model for Economics – Andrea Salute N. 736757**

## **A comparative analysis of network creation and development with NetLogo**

### **Abstract**

The purpose of this project is to create a model that compares the dynamics of network creation and development in a cluster scenario and in a free-market scenario. ClusterMode analyzes the formation of linkages using the acknowledged idea of proximity (Radius), which sees obtaining increasing Economies of Scale or Scope as the main goal of companies. MarketMode analyzes the formation of linkages in a scenario where companies have their core capabilities and look for external capabilities through partnership and M&A.

### **PART I Introducing the model**

A network is a group of interconnected nodes with different kind of linkages. These linkages can be knowledge flows, collaboration, innovation, friendship, familiar relationship and almost everything that one can imagine. The model simulates the creation of a network between agents that represent different companies in different scenario: a cluster and a free-market one. The purpose is to analyze and create one model that involve two different “modes” with different assumptions.

#### Cluster Mode

The research field on localized knowledge spillover<sup>1</sup> asserts that co-localization in clusters, and so proximity, is the reason why companies benefit of externalities. Geographers instead, believe that geographical proximity is not sufficient by itself; a different kind of proximity is required<sup>2</sup> (like social or cognitive proximity).

These theories are the pillar of Cluster Mode: there are different agents that represents our companies. Each agent can try to create a link with another agent given its distance (LINK PHASE). While the simulation goes, agents keep trying to create linkages and increase their radius if they are not able to find a partner. Last but not the least, Cooperation Propensity of the competitive environment influences the possibility to create links: higher level means a collaborative attitudes and so more willingness of actors to link with others. Because agents do not have any kind of information about the others, after the link phase, agents will have to evaluate the effectiveness of their link.

---

<sup>1</sup> Literature of industrial clustering: see Krugman, Breschi and Lissoni, Audretsch and Feldmann

<sup>2</sup> See for example Granovetter, Malberg, Maskell and Howells.

Many authors acknowledged that companies join a cluster to get economies of scale, speed or scope. Therefore, through the EVALUATION PHASE each link is successful only if both agents are looking for economies of scale and the distance between them is not too high<sup>3</sup>.

### Market Mode

Theories on market coopetition, state that companies can collaborate with other competitor to get what they need. Enter into an agreement allowed the parties to share the risk, cut the cost, share knowledge and information, acquire technology and market access and many other advantages, and that is way collaborations have surged in recent years. Collaboration can have many different shapes, but the most important are:

- Partnership: two companies can collaborate to produce collectively, develop a common technology, share licenses, distribution platform and so on.
- Acquisition: a company can decide to buy another.
- Merger: two company can decide to merge their company creating a brand new one.

The Market Mode is a simulation of the coopetition in a free-market environment, where companies can grow internally or externally<sup>4</sup>. Differently from Cluster Mode, there is no evaluation phase here, but only a LINK PHASE: agents look around (without radius) to find a partner and create a link. If the market need<sup>5</sup> matches other agent's market properties<sup>6</sup>, the two agents link themselves. If the collaboration has success, the two agents become allies and increase their properties; else, they keep looking for another possible partner.

Another research field of cluster literature states that "Absorptive Capacity"<sup>7</sup> influences network creation and development. Absorptive Capacity is a function of knowledge base of the firm, meaning the ability of firm to absorb and exploit external knowledge. Empirical results show that firm with higher absorptive capacity will play a more important role in the network and tend to have more linkages than the average<sup>8</sup>. Market Mode uses this approach to create the M&A PHASE:

A) Acquisition: each agent can purchase another agent only if its absorptive capacity is higher than the other one.

B) Merge: each agent can merge with another only if the difference between absorptive capacities of both agent are lower of a certain range (DeltaAC) and are compatible (Compatibility) that can be modified by the user.<sup>9</sup>

---

<sup>3</sup> The proxy for proximity is the average radius of agents

<sup>4</sup> External growth means M&A

<sup>5</sup> "Market need" represents what the agent is looking for: market access or technology ?

<sup>6</sup> "Market properties" represents the core capability of each agent: market or technology ?

<sup>7</sup> See Elisa Giuliani

<sup>9</sup> The idea is that if the distance between absorptive capacities is lower, probability of a successful acquisition is higher in terms of compatibility.

In this mode, there is a LIMIT RULE based on the idea of antitrust. Usually, companies cannot perform M&A when there are anti-competitive effects and the creation of a dominance position. If the resulting new agent will get too much power, Antitrust Authority will block any M&A operation.<sup>10</sup>

## Part II: Code and Model Description

The first part of the code concerns the declaration of variables and breeds:

**Global Variable** are variables that have a unique value, accessible by every agents in every part of the code

**Breeds** are special kind of turtles, like a subgroup

**Turtles-own Variables** are variables that have own-value for every turtle

### Global Variables

- *TotalSize*: is the sum of size of all agents
- *Regulation*: is the number of antitrust authority blocks
- *Matches*: is the number of link established
- *Failures*: is the number of times that a link is not established or that do not pass the evaluation phase
- *Acquisitions*: number of acquisition deals
- *Alliances*: number of alliances deals
- *Mergers*: numbers of mergers deals

### Breeds

- *Breed [clusters cluster]*: turtle that has established a cluster-link
- *Breed [allies ally]*: turtle that has established an alliance
- *Breed [AcqCompanies AcqCompany]*: turtle that has acquired another turtle
- *Breed [mergeds merged]*: turtle that has merged with another turtles

### Turtles-own Variables

- *R*: is the radius used to look for a partner
- *Explored?*: if true, the turtle has already tried to link with this turtle
- *PossiblePartner*: turtle with whom has established a link
- *EconomiesOfScale*: binary attribute that describe if the level of economies of scale in ClusterMode
- *Market*: attribute that represents market capability or search in MarketMode
- *Technology*: attribute that represents technology capability or search in MarketMode
- *Market\_Need*: what kind of capabilities the turtle is looking for in MarketMode
- *Market\_Properties*: what kind of capabilities the turtles can offer in MarketMode
- *Compatibility*: describe if two agent are able to merge with each other or not in MarketMode

---

<sup>10</sup> Size / Total Size have been used as a proxy of market power. Dominance threshold is set to > 40%

## Setup

Clear all results, the interface and values of variables and create a number of turtles that is defined by the user with the slider “nTurtles”. Then each turtle set its turtles-own variables randomly (the user define the value of R through the slider “RadiusRange?”). To *ifelse* statement defines what are the capabilities that each turtle offers, and what those which looks for. Then resets the ticks counter and the turtles position themselves randomly in the environment.

Finally another *ifelse* statement switch off the MarketMode button if the ClusterMode is on and vice versa, showing some information about radius and cooperation propensity in ClusterMode and Size, Absorptive Capacity, market need, market properties in MarketMode.

Here there is the code:

```
to setup
ca
create-turtles nTurtles [set R RadiusRange? + random 10

set size 1

set color blue

set EconomiesOfScale random 3

set compatibility random 4

set explored? false

set market random 10

set technology random 10

set absorptive_capacity ACrange? + random 10

ifelse random-float 100 < 50 [set market_need market][set market_need technology]

ifelse random-float 100 < 50 [set market_properties market][set market_properties technology]]

reset-ticks

ask turtles [setxy random-xcor random-ycor]

ifelse (ClusterMode? = true)

[set MarketMode? false

output-type "Average Radius of turtles is " output-print mean [R] of turtles

output-type "Level of Cooperation Propensity is " output-print CooperationPropensity

]
```

```

[set MarketMode? true

  output-type " Average Absorptive Capacity of turtles is " output-print mean [absorptive_capacity] of
  turtles

  output-type " Average size of turtles is " output-print mean [size] of turtles

  output-type " The number of agents looking for market access is " output-print count turtles with
  [market_properties = market]

  output-type " The number of agents looking for technology is " output-print count turtles with
  [market_properties = technology]

]

```

End

### Go procedure

After the setup, ask all turtles to perform “ClusterMode”, then “MarketMode”. If all the turtles have already been explored (the turtle has tried to link with all the other turtles), then stop. If Regulation (number of Antitrust blocks) is higher than 60, then stop.

Here there is the code:

```

to go

  if not any? turtles with [not explored?][stop]

  if Regulation > 60 [stop]

  ClusterMode

  MarketMode

  tick

end

```

### ClusterMode

If the ClusterMode button is on, then perform “LookAround” and “Evaluation” procedure

```

to ClusterMode

  if ClusterMode?[LookAround Evaluation]

end

```

### LookAround (ClusterMode)

Ask all turtles, that do not belongs to clusters breed, to choose a “PossiblePartner” between one of other turtles in radius R (slider RadiusRange?) that is not been explored yet (“not explored?”). If this PossiblePartner exists and a random number in range (0-1) is lower than “Cooperation Propensity”, then create a link, increases matches and show in the command center “We’re

partners now". Finally ask your partner to set is variable "explored?" to true (the turtle will know that this turtles was a partner and will not try to link with it anymore). Else, increases the radius and print in the command center "I need to go further". Then all the turtles wait for 1 second.

Lastly, an *ifelse* statement asks the program to show the level of R in the interface if the button "show-radius?" is on.

Here there is the code:

to LookAround

ask turtles

[ if breed != clusters

[ set PossiblePartner one-of other turtles with [not explored?] in-radius R

ifelse PossiblePartner != nobody and random-float 1 < CooperationPropensity

[create-link-with PossiblePartner

ask PossiblePartner [set explored? true]

set matches matches + 1

show "We're partners now"]]]

[set R R + 1

show "I need to go further"]

ifelse show-radius?

[set label R]

[set label ""]]]

wait 1

end

### Evaluation (ClusterMode)

If:

- The "PossiblePartner" exists and
- Level of "EconomiesOfScale" is the same of the "PossiblePartner" and
- Radius of turtles is lower than average radius of all turtles

Ask turtles to change color to red, breed to clusters, print in the command center "We're a cluster now" and to ask "PossiblePartner" to do the same, except for the statement.

Else, increase failures of + 1 and print "Gone bad, keep searching". In both case, show in the output the average radius of turtles and the ratio matches/failures

Here there is the code:

*to Evaluation*

*ask turtles*

*[if PossiblePartner != nobody*

*[ifelse EconomiesOfScale = [EconomiesOfScale] of PossiblePartner and  $R < \text{mean } [R]$  of other turtles*

*[set color red*

*set breed clusters*

*show "We're a cluster now"*

*ask PossiblePartner*

*[set color red*

*set breed clusters ]]*

*[ set failures failures + 1*

*show "Gone bad, keep searching"]]]*

*output-type "Average Radius of turtles is " output-print mean [R] of turtles*

*output-type "The ratio Matches/Failures is " output-print ( matches / failures)*

*end*

### MarketMode

If the MarketMode button is on, performs "MarketInteractions", "Acquisition" and "Merge".

Here there is the code:

*to MarketMode*

*if MarketMode?[MarketInteraction Acquisition Merge ]*

*end*

### MarketInteraction (MarketMode)

Ask all turtles to choose a "PossiblePartner", if exists, and if the "PossiblePartner" is offering exactly what the turtle is looking for, then:

- Create a link
- State “We’re allies now”
- Increase absorptive capacity of + 1
- Increase alliances of + 1
- Change breed to “allies”
- Increase matches of + 1
- Ask PossiblePartner to do the same and set “explored?” = true.

Else, increases failures of + 1 and state “You do not have what I am looking for”

Here there is the code:

*to MarketInteraction*

*ask turtles [*

*set PossiblePartner one-of other turtles with [not explored?]*

*if PossiblePartner != nobody*

*[ifelse market\_need = [market\_properties] of PossiblePartner*

*[ create-link-with PossiblePartner*

*show "We're allies now"*

*set absorptive\_capacity absorptive\_capacity + 0.5*

*set alliances alliances + 1*

*set breed allies*

*set matches matches + 1*

*ask PossiblePartner [set explored? true*

*set breed allies*

*set absorptive\_capacity absorptive\_capacity + 0.5]]*

*[set failures failures + 1*

*show "You do not have what i am looking for"]]]*

*end*



### Acquisition (MarketMode)

Give to "TotalSize" the value of the sum of the size of each turtle. If PossiblePartner exists and its absorptive capacity is lower than turtle perform the *ifelse* statement: if the size after the acquisition is lower than 40% then acquire the company and set the variables, else, increase Regulation of + 1 and state "I have bought company number x"

*to Acquisition*

*ask turtles*

*[ set TotalSize sum[size] of turtles*

*if PossiblePartner != nobody and absorptive\_capacity > [absorptive\_capacity] of PossiblePartner*

*[ifelse ((size + 0.5) / TotalSize ) < 0.40 [*

*set color yellow*

*set size size + 0.5*

*set explored? true*

*set breed AcqCompanies*

*set acquisitions acquisitions + 1*

*set absorptive\_capacity absorptive\_capacity + 1*

*set market\_properties market\_properties + 1*

*show " i have bought company " print [who] of PossiblePartner*

*ask PossiblePartner [die]]*

*[*

*show " Antitrust authority blocked the acquisition"*

*set Regulation Regulation + 1]]]*

*end*

### Merge (MarketMode)

Stop if there are no more turtles with "explored?" = false (if the turtles has tried to link with every other turtle). Ask turtles, if "PossiblePartner" still exists and the difference in absorptive capacity is lower than "DeltaAC" (the user will set the value with the slider) and the two turtle are compatible, then merge with the company, change breed, absorb the properties of the other company and ask the "PossiblePartner" to die. In this way, the turtle is transformed in a new turtle

that inherit the sum of variables of the two turtle. But the merger is allowed only if the resulting size is lower than the 40% of "TotalSize". Else, state "Antitrust blocked the merger" and increase Regulation + 1. Show in the output the mean of absorptive capacity and size, the ratio matches/failures.

to Merge

if not any? turtles with [not explored?][stop]

ask turtles

[if PossiblePartner != nobody and (absorptive\_capacity - [absorptive\_capacity] of PossiblePartner) < DeltaAC and compatibility = [compatibility] of PossiblePartner

[set TotalSize sum[size] of turtles

ifelse ((size + [size] of PossiblePartner) / TotalSize ) < 0.40 [

set breed mergeds

set color red

set size size + [size] of PossiblePartner

set absorptive\_capacity absorptive\_capacity + [absorptive\_capacity] of PossiblePartner

set market\_properties market\_properties + [market\_properties] of PossiblePartner

show " I have merged with " print [who] of PossiblePartner

ask PossiblePartner [die]

set mergers mergers + 1]

[

show " Antitrust authority blocked the merger"

set Regulation Regulation + 1]]]

output-type " Average Absorptive Capacity of turtles is " output-print mean [absorptive\_capacity] of turtles

output-type " Average size of turtles is " output-print mean [size] of turtles

output-type "The ratio Matches/Failures is " output-print ( matches / failures)

output-type "The number of antitrust regulation have been " output-print Regulation

end

## Part III Results and Discussion

To evaluate results, the author have run the simulation using three different scenario: low, intermediate, high level of variables. Due to the fact that reporting every attempts will be too long, the author choose to run at least 5 simulation for each scenario and then report, for each of them, the one that was the most representative.

### Cluster Scenario 1

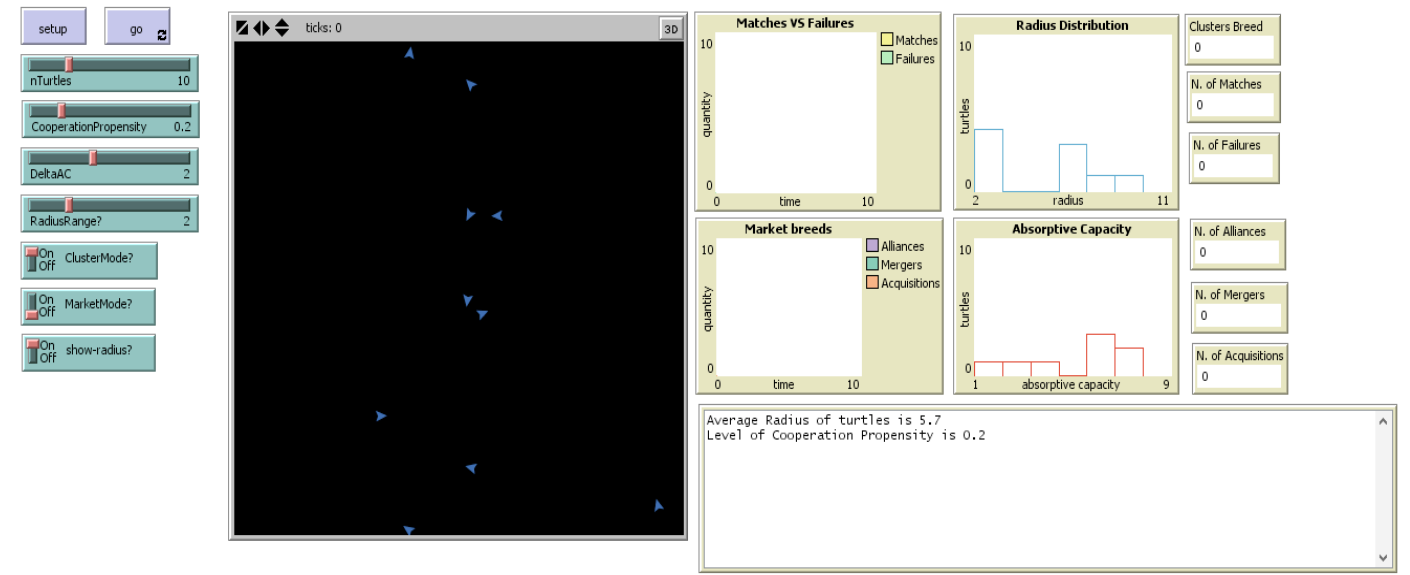


Figure 1 Cluster Scenario 1: Interface at the beginning

Scenario 1 was an attempt to run the simulation with low values of all the variables of the model. There are 10 turtles, with a Cooperation Propensity of 0.2 (about 20%), that look around with a radius of 2 + a random value that is given by the program. The radius distribution is in the interval (2, 10): four turtles have values around 2-4 and three turtles around 5-7 and other three turtle spread in the distribution. Looking at the output window (bottom-right of the picture), is possible to see that the average radius is 5.7. Figure 2 depicts the interface at the end of the simulation.

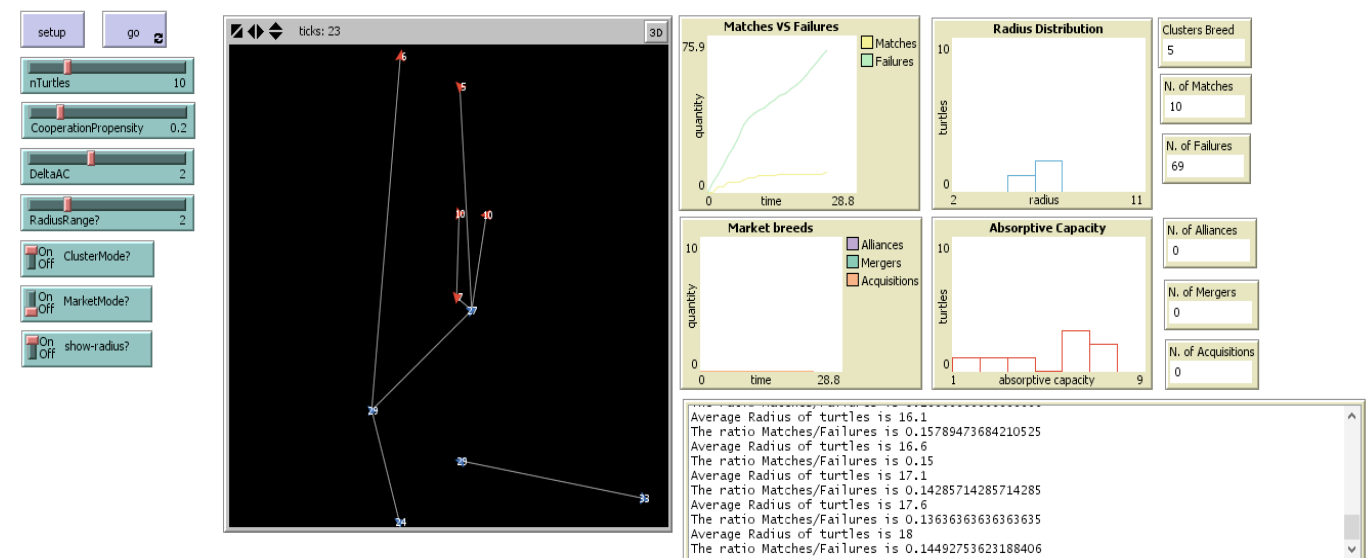


Figure 2 Cluster Scenario 1: interface at the end

Given the low Cooperation Propensity of turtles, the ratio between number of matches (10) and number of failures (69) is very low (about 0.144). Five turtles were able to satisfy the evaluation phase and fixed their links, changing breed to clusters and color to red. Despite the success and the formation of a network that enclose about the 80% of the turtles, as it is possible to see from the picture, a cluster was not formed (a cluster is a network of at least three red turtles linked together). The radius distribution is split into two part: clusters breed have a radius lower than 10, with an average value of 8, while blue turtles (normal ones) have an average radius of 27. No subgroups exist.

## Cluster Scenario 2

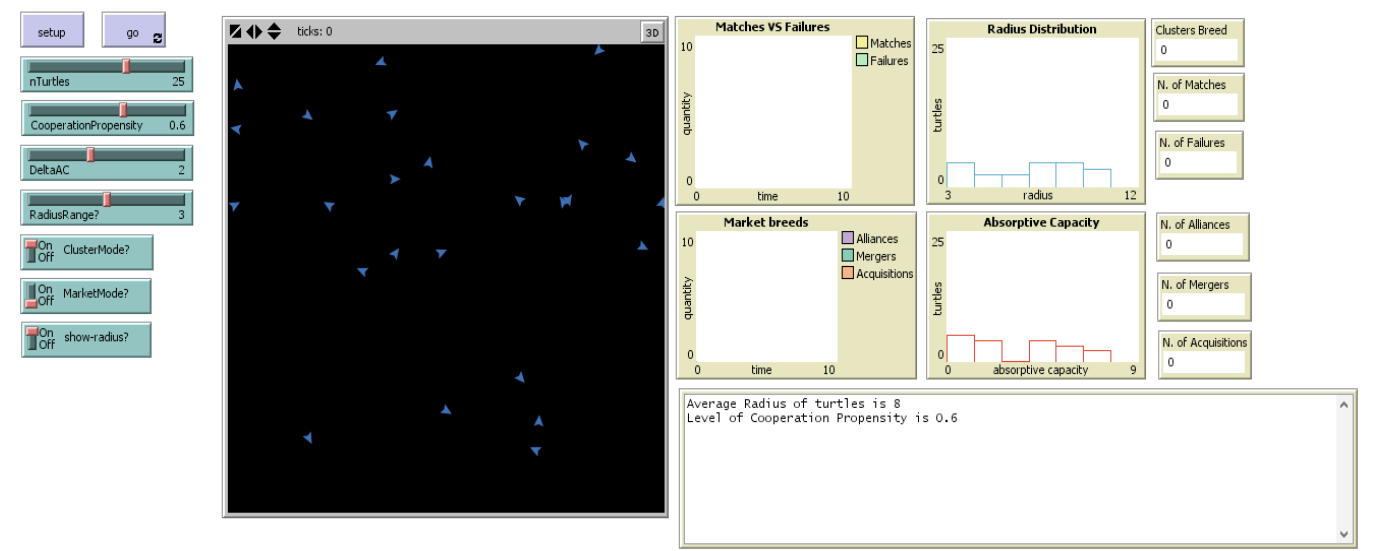


Figure 3 Cluster Scenario 2: Interface at the beginning

Figure 3 represents Scenario 2, an attempt to run the simulation with intermediate values of all the variables of the model. There are 25 turtles, with a Cooperation Propensity of 0.6 (about 60%), that look around with a radius of 3 + a random value that is given by the program. The radius distribution is in the interval (3, 12): there are three group of 5 turtles over the average value 8, and another big group and two small group below it. Looking at the output window (bottom-right of the picture), is possible to see that the average radius is 8. Figure 4 depicts the interface at the end of the simulation.

Given the medium level of Cooperation Propensity of turtles, the ratio between number of matches (25) and number of failures (42) is medium-high (about 0.61). The first difference with Scenario 1 is the dynamic of the ratio: as is it possible to see in “Matches VS Failures” chart, the two curves behave in the same way until the half of the link phase and then diverge. Another big difference is the radius distributions: the global interval is the same (3, 12) but turtles are far more heterogenous than previously. The average radius is lower ( $10 < 18$ ), and the same is for radius of blue turtles ( $10 < 27$ ). Average radius of clusters breed is an exception: the value is higher ( $9 > 8$ ). Eleven turtles were able to satisfy the evaluation phase and fixed their links, changing breed to clusters and color to red. This result is more than twice the previous ( $11 > 5$ ).

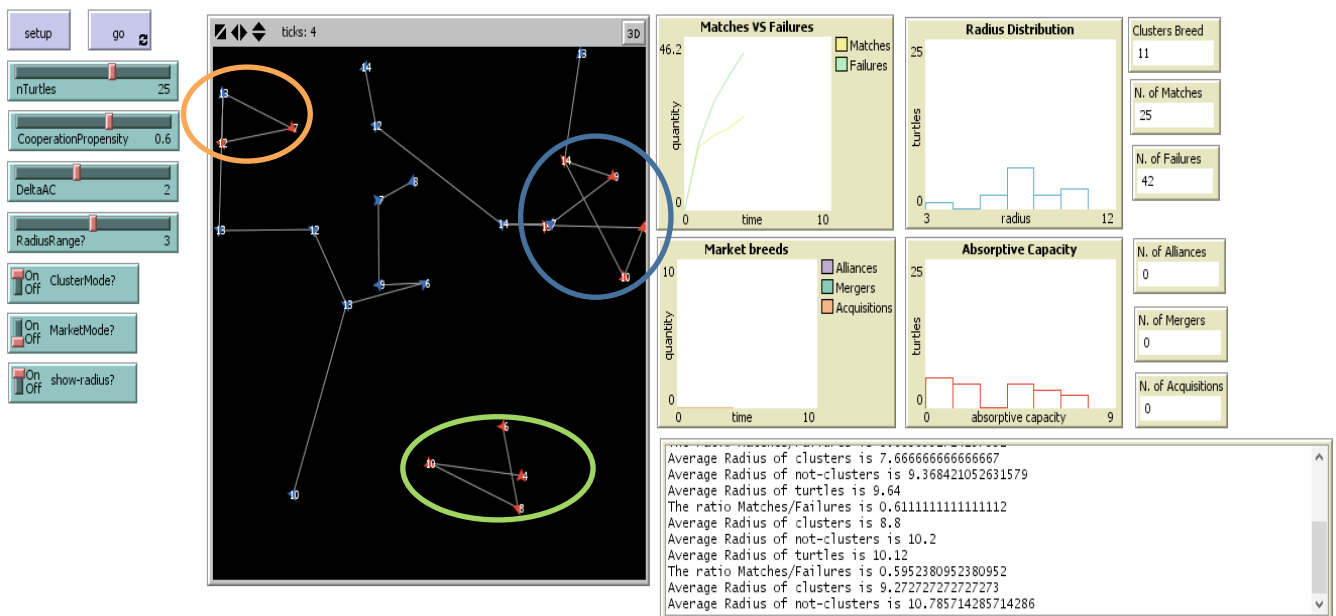


Figure 4 Cluster Scenario 2: interface at the end

Comparing the network, is it possible to observe that in Scenario 2 a real network was created, with the presence of 3 small groups (colored circle in the picture) and two real clusters (green and blue circle). These two clusters are different: the green one is isolated, while the blue one has a greater number of linkages (5 clusters turtles > 4) and in addition cooperates with three blue turtles.

### Cluster Scenario 3

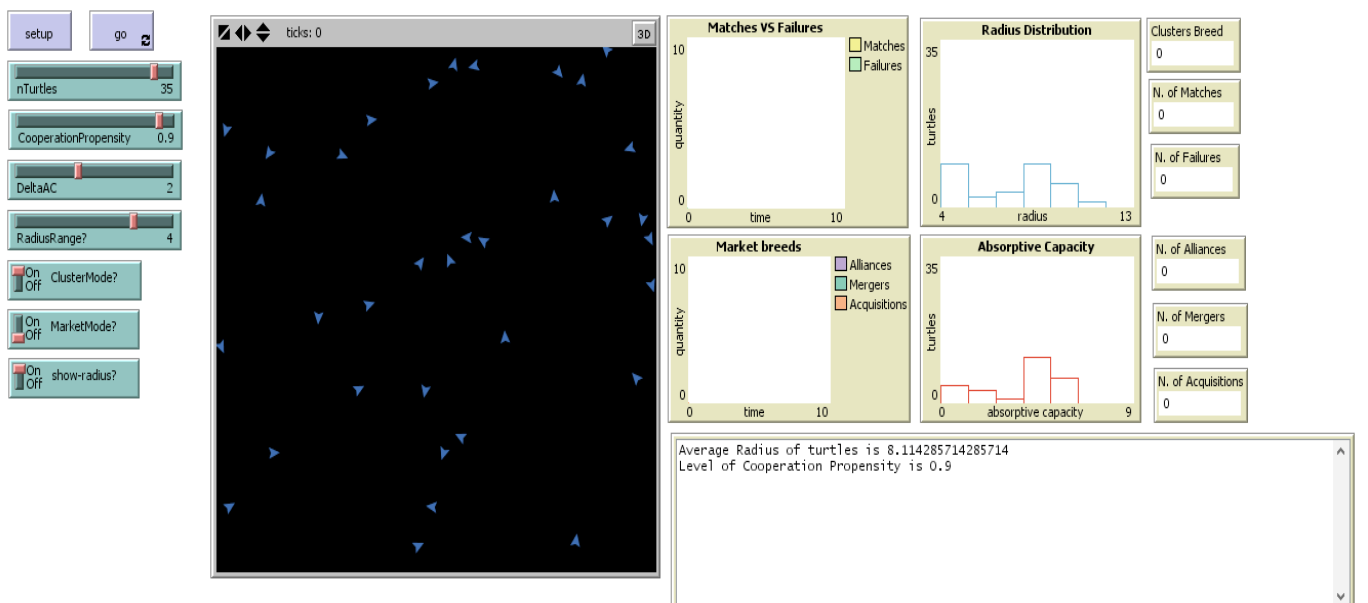


Figure 5 Cluster Scenario 3: interface at the beginning

Figure 5 represents Scenario 3, an attempt to run the simulation with high values of all the variables of the model. There are 35 turtles, with a Cooperation Propensity of 0.9 (about 90%), that look around with a radius of 4 + a random value that is given by the program. The radius

distribution is in the interval (4, 13), wider than Scenario 1 and 2. The distribution is less distributed than Scenario 2 but more distributed than Scenario 1: there are two main group of about 10-12 turtles over the average value 5 and 10, and other small groups along the curve. In some way, is it possible to state that the distribution is bipolarized. Looking at the output window (bottom-right of the picture), is possible to see that the average radius is 8.1

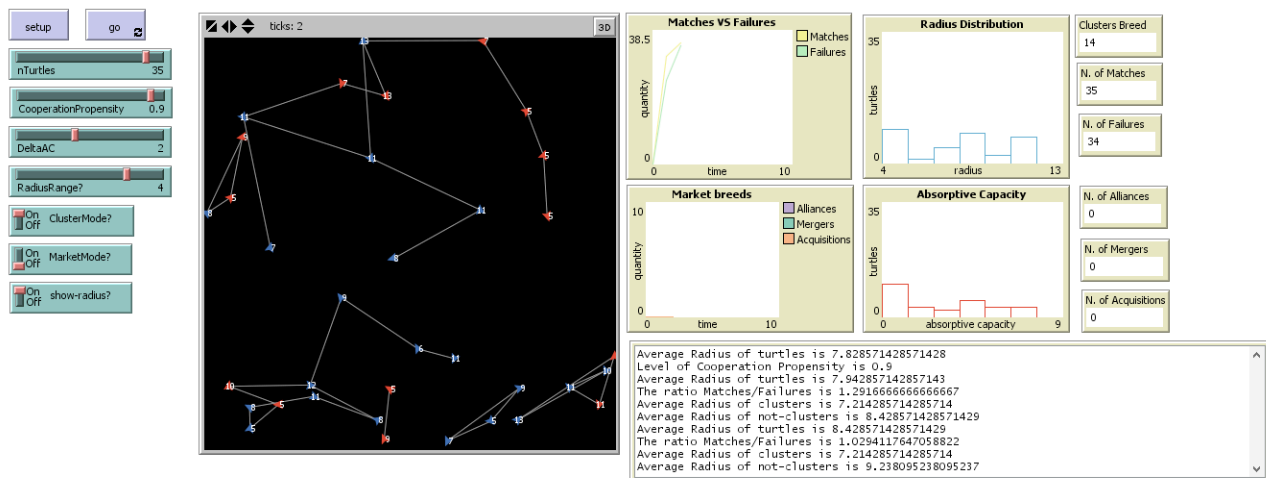


Figure 6 Cluster Scenario 3: interface at the end

The ratio between number of matches (35) and failures (34) is high (about 1.02). This result is higher than Cooperation Propensity because of the role played by the increased radius. As observed before, the radius distribution is still polarized, but the main group are three at the end. Therefore the radius distribution is tripolarized. The average total radius is lower than Scenario 1 and 2 ( $8.4 < 10 < 18$ ) and is the same for radius of blue turtles ( $9 < 10 < 27$ ). Fourteen turtles were able to fix their link ( $14 > 11 > 5$ )

Comparing the network with Scenario 2, it is possible to observe that there are more subgroups ( $5 > 3$ ) but there is only one cluster (while in Scenario 2 there were two of them). In addition, the cluster is isolated and not linked with other turtles. Finally, the cluster is not a real network but looks like a supply chain. As far as networks are concerned, there is a much greater number of networks: two big networks, a small network, a clique and the cluster.

Concluding the analysis on ClusterMode, the results show that the more the values of the variables increase, the more the number and heterogeneity of networks. Clusters tend to form as networks when the scope of search (radius) is more heterogeneous.

### Market Scenario 1

Figure 7 represents the interface of the first scenario with a low level of "DeltaAC", a threshold that influences the merger procedure<sup>11</sup> and a low level of "ACrange?" which influences the level of absorptive capacity. There are 20 turtles with an average value of absorptive capacity of 6.6. Absorptive Capacity distribution is heterogeneous but with a strong group of six turtles along the

<sup>11</sup> If the difference between absorptive capacities is lower than DeltaAC, the merger is allowed

average. Nine agents are looking for “market access”, while eleven agents are looking for “technology”.

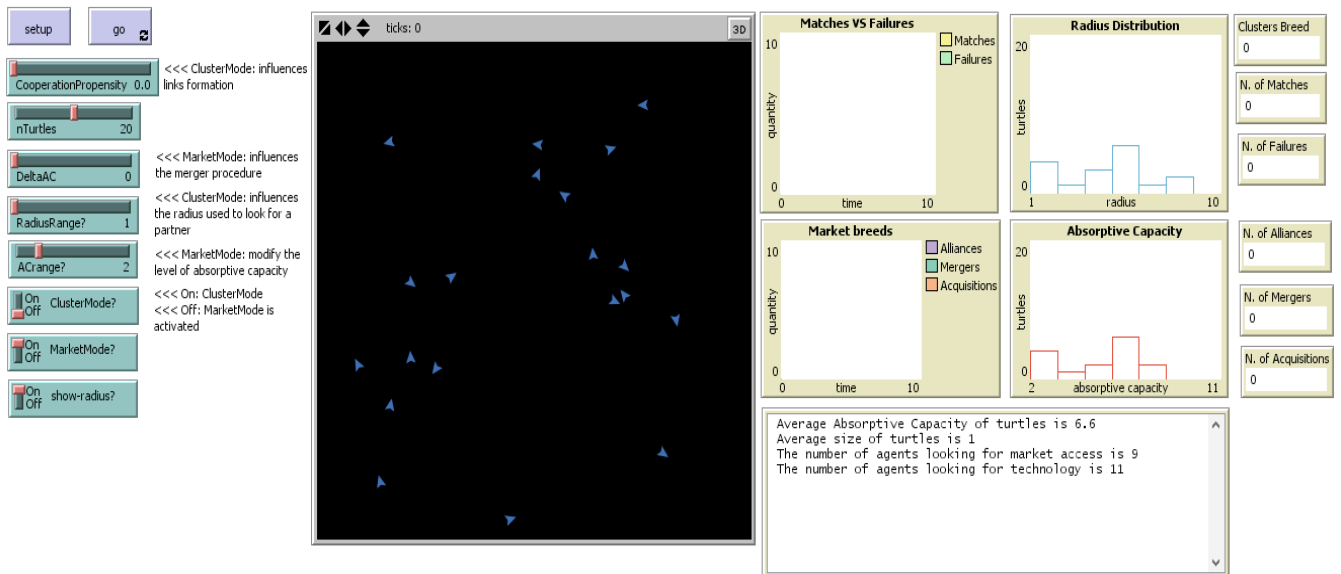


Figure 7 Market Scenario 1: interface at the beginning

Figure 8 shows a great number of alliances compares to mergers and alliances (respectively 10, 4, 1). The low level for alliances (only 1 match), results in very low level of matches/failures ratio (0.005). Average Absorptive Capacity is 13.4, while the size is 2.5. In addition, there is a very high number of Regulation (61). The graph shows the presence of 1 normal turtle, 1 merged turtle and 4 acquired turtles with different size (meaning different level of acquisitions) and no collaboration linkages.

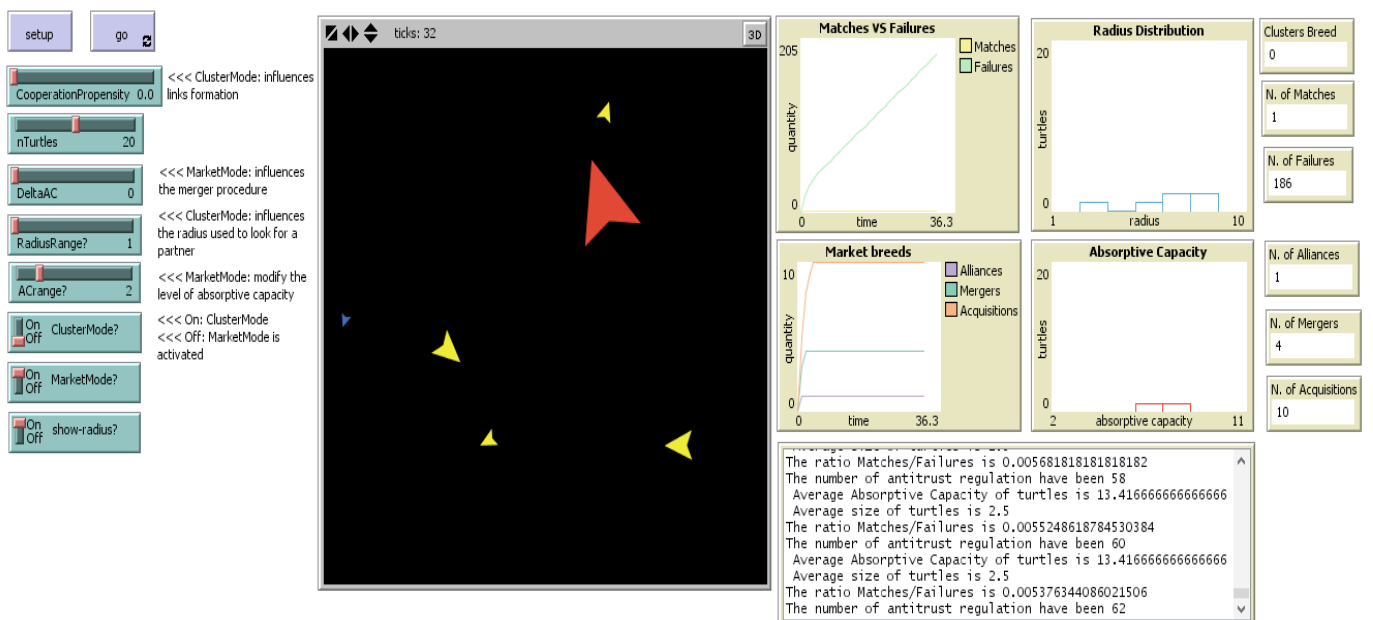


Figure 8 Market Scenario 1: interface at the end

## Market Scenario 2

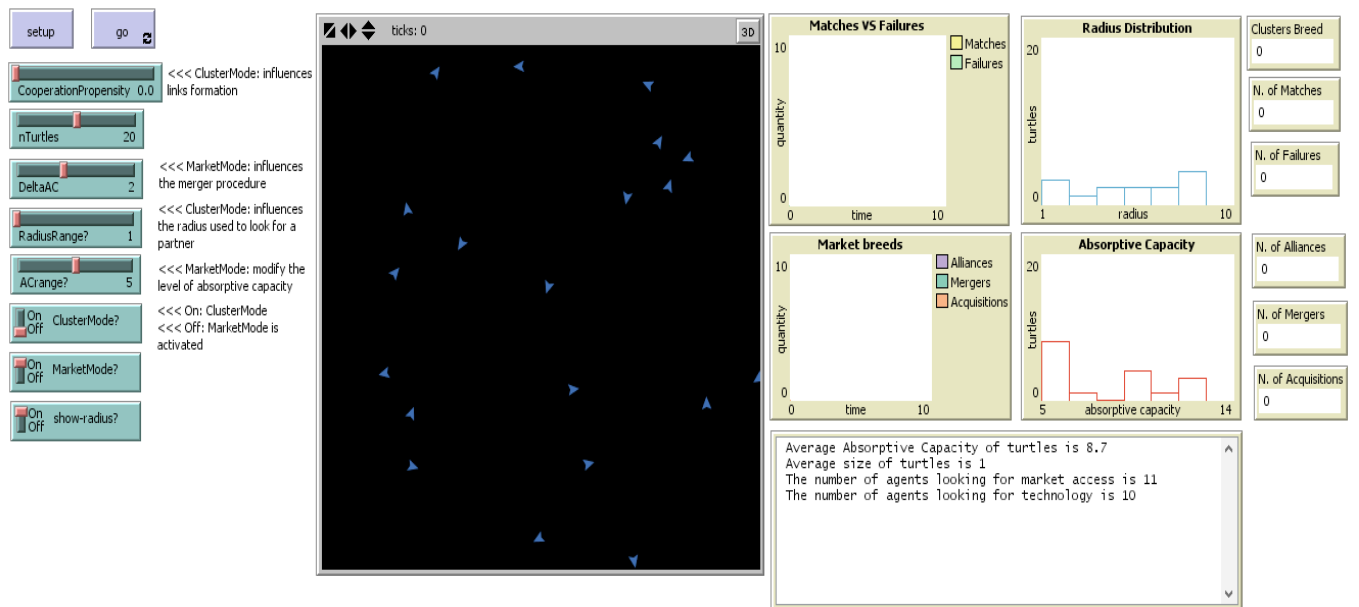


Figure 9 Market Scenario 2: interface at the beginning

Scenario 2 is an intermediate simulation, with average absorptive capacity of 8.7 (in the first scenario was 6.6) and DeltaAC at 2 (previously was 0). Absorptive Capacity distribution is extremely focus on values in range (5-7) and (8-13).

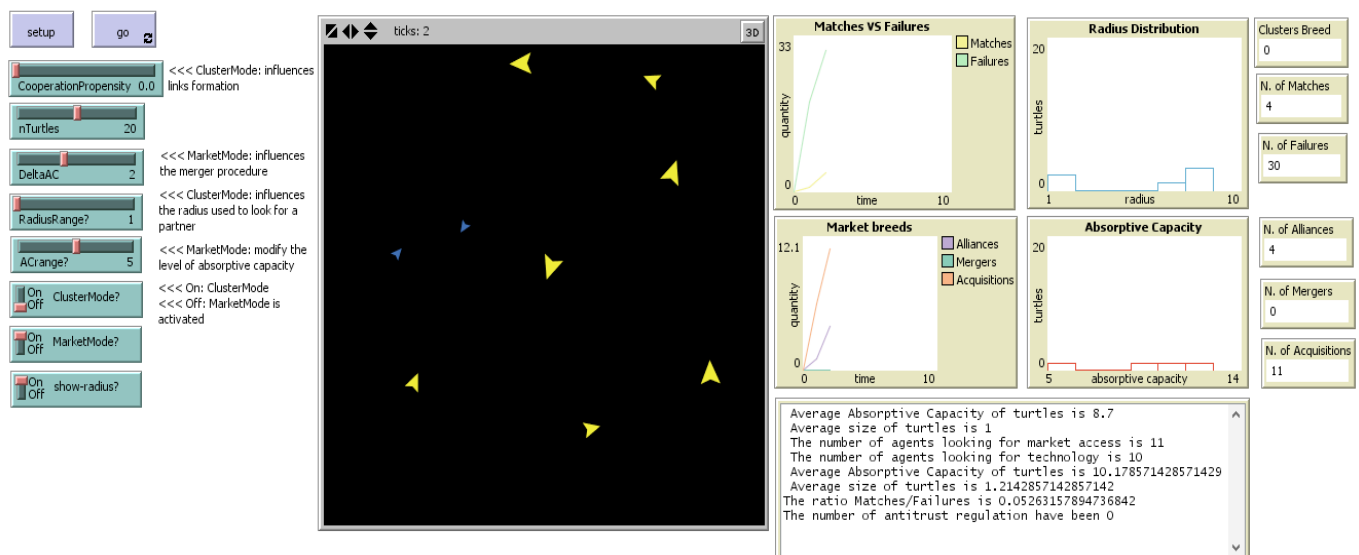


Figure 10 Market Scenario 2: interface at the end

Comparing the results with Scenario 1, the first important point is that we do not have any kind of Regulation (0) and merger. The ratio matches/failures is the same and so the number of alliances. This is strange because of the increased value of absorptive capacity and DeltaAC. A higher DeltaAC should have caused a surge in the number of mergers. Nevertheless, there is high probability that this result is due to the compatibility variable. The lack of mergers increase the number of survived turtles and increase variety amongst them. Observing the graph, we can see



that the acquisition process is more distributed than Scenario 1 and two blue turtles were able to survive. In addition, values of absorptive capacity are more heterogeneous.

### Market Scenario 3

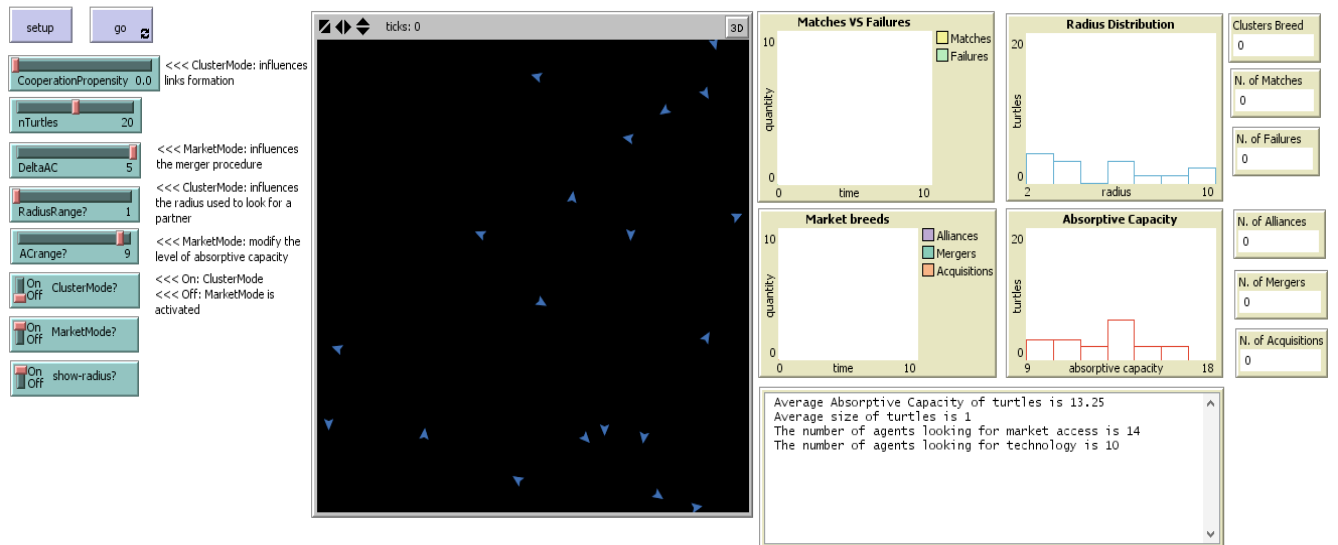


Figure 11 Market Scenario 3: interface at the end

Figure 11 reports Scenario 3 setup, with a DeltaAC level of 5 and ACrange? of 9. Absorptive Capacity is well distributed, but there is a strong group around the central value (13.25).

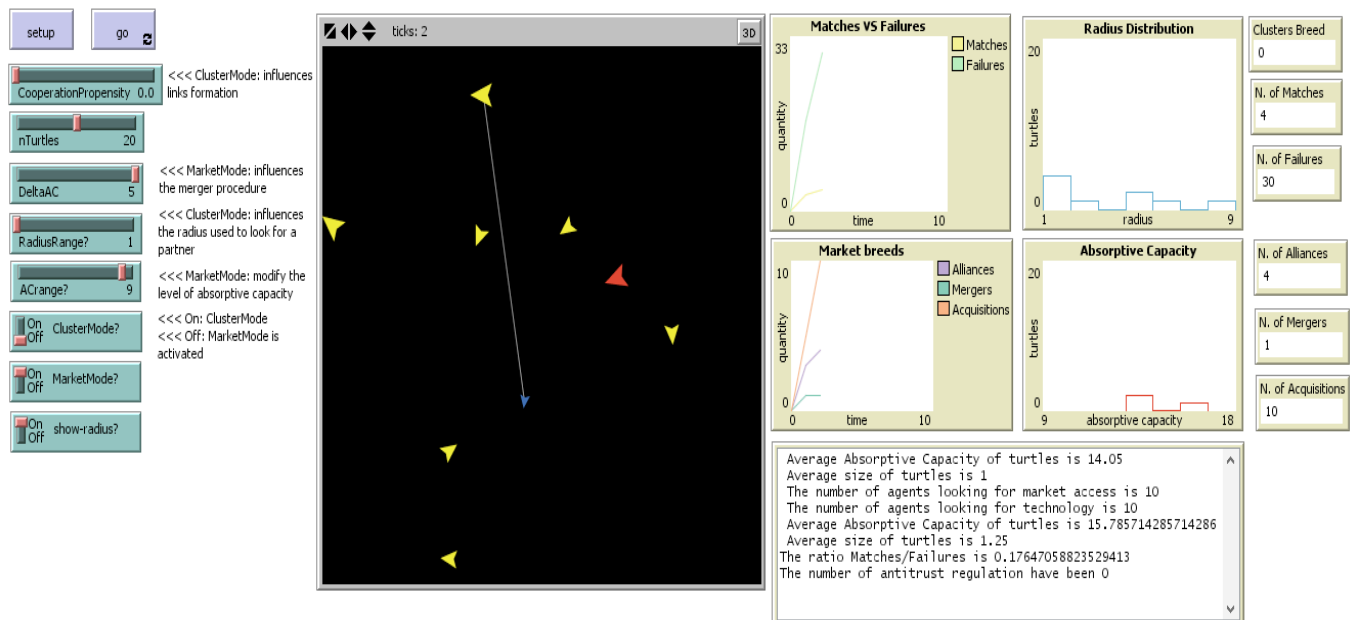


Figure 12 Market Scenario 3: interface at the end

Comparing Scenario 3 with the others scenarios, it is possible to observe that this is the simulation with the highest number of survived turtles at the end ( $10 > 9 > 6$ ) and the most heterogeneous in terms of variety of results. Indeed, there are seven acquisition turtles, one merged turtles and one normal turtle. In addition, it is the unique scenario where there is a link,

between a yellow turtle and a blue turtle. Like Scenario 1, absorptive capacity distribution is strictly polarized around the value 14. The ratio matches/failures is the highest (0.176).

Comparing all scenarios, we observed that there is not any kind of network because turtles prefer to M&A, and use collaborations only when M&A is not allowed or possible. This result is not unexpected but driven by the code nature. It is clear that M&A processes graphically obscured the presence of linkages between turtles.