A Modified Deffuant Model: Multiopinion and Network Structure

Pietro della Briotta Parolo, Daniela Perrotta, Stefania Pezzoli
Università degli Studi di Torino,

22/06/2011

Abstract

Our project consists in the simulation of a model based on the one proposed by Deffuant et al [1] . We then proceed by adding to the model a multiopinion interaction and a Network Structure. Our goal is to study the variations of the results according to the values of the different parameters and to interpretate the new results originated by the modifications introduced.

1 The Original Model

The starting point is the complete mixing model which consists of a continuous representation of opinion as they spread through a social structure. Randomly chosen agents interact in order to mix up their opinion and the mixing interaction will take place only if the distance between opinions is short enough according to a threshold parameter d in range (0:50) and with interaction strength μ in range (0:0,5). The equations of the interactions are:

$$\widehat{x_1} = x_1 + \mu(x_2 - x_1) \tag{1}$$

$$\widehat{x_2} = x_2 + \mu(x_1 - x_2) \tag{2}$$

Values of opinions are initially randomly generated across a uniform distribution on (0,100).

2 New Ideas

2.1 Extremists

The first modification is based on the introduction of turtles with "extreme" opinion, i.e. opinion equal to 100 or 0. They do interact with other turtles according with the previously mentioned parameters but they do not change their opinion value. They are meant to represent an element of constant influence on the system.

2.2 Multiopinion

The second modification consists in the addition of 3 topics of discussion represented by variables still in range (0,100). We intend to implement the model of interaction in two ways:

- The first gives different priorities to the topics. The agents start interacting on the first topic if the conditions are met. They will then move to the second topic and so on with the same structure. Therefore "less important topics" will be discussed only if all the previous positions are close enough.
- The second instead considers a mean degree of accordance between the agents. They will interact only if two topics are close enough and they will interact in a randomly chosen topic.

2.3 Place

The third modification consists in the introduction of a spatial structure in the movement of the turtles. The full mixing model is no longer used, turtles interact only with other nearby turtles in a subworld we create.

2.4 Network

A parallel setup has been created where the interaction structure is based on the better known networks studied in complex systems:

- Random Graph: a slider allows to choose the number of nodes and the average link degree. The setup links randomly the nodes. The result is the creation of many interconnected isles according to the chosen link degree. If such value is low the result may be the creation of separated groups.
- Small World: the nodes are created and rearranged in circle. They are linked to the first and second neighbors. Following a clockwise chain, links are then rewired to randomly chosen nodes according to a rewiring probability given by slider. The result is the creation of shortcuts that shortens the path between two nodes.
- Preferential linking: this network is based on the idea that new members of the group will be linked more likely to the mostly linked nodes, thus creating a preferential selection criteria in the linking process. Initially a linked couple is created and then at each step a new node is added and asked to be linked to the previously existing network. The result is the creations of hubs that keep growing, causing a power law distribution [2] of the degree. Eventually there will be a few nodes with a significant number of links and a huge number of nodes with very few links compared to the network population.

2.5 NETLOGO

The setup initializes the agents, the number of which is chosen with a slider, and sets them on the diagonal according to the value of their opinion. Also the color is changed according to the opinion of the turtles since Netlogo uses numbers from 0 to 140 to label the colors. A chooser allows to choose among the possible different types of interactions. In order to take into account the possibility of multiple opinions, opinion2 and opinion3 are as well initialized, but they are kept negative when the basic model is chosen in order not to show them on the plots.

- Place: we introduce a ”subworld” represented by the turtle variable called place. When the *Place?* switch is on the turtles interact only with turtles in a certain range (given by slider) of the place variable.
- Extremists: a switch allows the existence of turtles with fixed opinion 100 or 0. The number is given by slider. They are placed at the ends of the diagonal in order to show their extreme opinion.

2.5.1 Plots

There are two plots for each topic. In each case the opinion interval is divided into ten intervals. In the first graph the y-axis represents the number of turtles and each interval is represented by the color at the center of the opinion interval and the x-axis represents time. In the second graph a histogram is created where each column represents the turtle population in each opinion interval represented on the x-axis.

3 Code Comments

3.1 Standard Model

3.1.1 setup

```
setxy opinion1 opinion1
```

In order to give a better visual representation of the opinion distribution we position the agents on the diagonal with x and y coordinates equivalent to the value of their opinion. Such command is repeated after each interaction in order to update the coordinates of the agents.

```
[set opinion2  -10  set opinion3  -10]
```

This line of code has to be introduced due to problems with the plotting. Now *opinion2* and *opinion3* are out of range and will not be plotted when the basic model is running. Instead of switching off the graph, we choose to simply hide the opinion on other topics.

3.1.2 go

```
if InteractionType = "Basic"
```

To have a single go procedure we choose to insert three *if* conditions that define three different types of interactions according to the selection from the chooser.

```
set temp1 opinion1
set templace place
interact
set opinion1 temp1
```

We save on temporary variables the values of the first interacting turtle. Instead of giving a name to that turtle and using the command *[variable] of turtlename* we simply choose to use a global variable that can be used in the later steps of the program. After the interact procedure, which modifies the values of the opinions of the two turltes, the values of the first one are updated with the opposite method.

```
ask extremists100 [set opinion1 100 setxy 100 100 set color white]
ask extremists0   [set opinion1 0 setxy 0 0 set color white ]
```

We want the extremists never to change opinion. Therefore we have two choices as a matter of keeping the values fixed: either checking before the interaction whether one of the turtles is an extremist and in case write a single equation or to allow temporarily the modification of the opinion of the extremist and then later bring it back to its original value. We choose the latter and thus we have to introduce two breeds of extremists.

3.1.3 interaction

```
[let partner one-of other turtles with [place < templace + range and place > templace - range]
```

This line of code allows to choose the second interacting turtle among a specific agentset. In this case the *Place?* switch is on and therefore we look for a different turtle (*other*) among the turtles whose place variable is within a range chosen by slider.

```
ifelse partner != nobody [

       ask partner [

         ifelse  abs (temp1 - opinion1) < d [

           let tempopinion1 opinion1
           set opinion1 (opinion1 + mu *(temp1 - opinion1))
           set temp1 (temp1 + mu * (tempopinion1 - temp1))

           setxy opinion1 opinion1
           set color opinion1

           set interactions1 (interactions1 + 1)


      ]

         [set place random 100]
     ]
    ]
   [ set place random 100]
```

This is the core of the interaction process for the basic model as well as for other ones. We have already looked for a partner and if such partner exists the interaction takes place. Otherwise the first turtles is asked to move to another place of the world because it didn't find any other turtle to interact with. The interaction needs another condition to take place: we check whether the opinion of the two turtles are close enough (parameter d) and in case we update the opinion according to equation (1). We change the opinion of the second turtle first in order to use the temp1 variable, which is later modified. After the interaction we increase by one the interaction counter.

3.1.4 Do-plots

```
set-plot-y-range 0 count turtles
```

set-plot-y-range is used in order to modify the range of the y axis on the plot. With this line of code we set it to the total number of agents present in the world. We are interested in a static y axis and a running x axis.

```
let partial 0
set-current-plot-pen "5"
plot-pen-up plotxy ticks partial
set partial partial + count turtles with [opinion1 >= 0 and opinion1 <=  10]
plot-pen-down plotxy ticks partial
```

This is the first cycle of the plotting sequence of the *Topic1* graph. We want the opinion distribution to be shown on the y axis by the width of the color of the relative opinion interval. In order to do this we use a temporary variable *partial* that memorizes the number of turtles with already drawn opinion. The following pen starts to draw from the end point of the previous one. The cycle is repeated for all the ten intervals.

3.2 Multiopinion

```
if  abs (temp1 - opinion1) < d [
    if  abs (temp2 - opinion2) < d [
        if  abs (temp3 - opinion3) < d [
```

The multiopinion models follow the same structure of the basic model with a few modifications. In the MultiOneByOne scenario the turtles discuss one topic at a time, moving on to the following one only if the previous conditions on the opinions are met. Therefore we create an if chain that is broken at the first divergence of opinion.

```
ask partner [
    let agreement 0
    if abs( temp1 - opinion1) < d [ set agreement (agreement + 1)]                                        ;
    if abs( temp2 - opinion2) < d [ set agreement (agreement + 1)]
    if abs( temp3 - opinion3) < d [ set agreement (agreement + 1)]
    if agreement >= 2 [

       let numberopinion random 3

if numberopinion = 0 [ "interaction on opinion1"
if numberopinion = 1 [ "interaction on opinion2"
if numberopinion = 2 [" interaction on opinion3"
```

In the MultiRandom scenario instead the discussion takes place on a random topic if there is agreement on at least two previous ones, which we check with the first four ifs. It is important to notice that in this case the interactions take place with a parallel if structure, since the topics are discussed independently.

3.3 Network

3.3.1 Setup

```
if NetworkType = "RandomGraph"[
if NetworkType = "SmallWorld" [
if NetworkType ="Preferential"  [
```

Within the setup procedure we have three ifs that initialize a different network according to the chooser.

3.3.2 Preferential Linking

```
make-node nobody
make-node turtle 0

repeat number-of-nodes - 2
make-node find-partner
```

We start with two nodes linked together and we need to create the remaining n-2 nodes using the functions make-node and find-partner.

```
to-report find-partner
  let total random-float sum [count link-neighbors] of turtles
  let partner nobody

  ask turtles  [
    let nc count link-neighbors
    if partner = nobody [
      ifelse nc > total
        [ set partner self ]
        [ set total total - nc ]
    ]
  ]
  report partner
end
```

The *find-partner function* is a *to-report* function. *Report* immediately exits from the current *to-report* procedure and reports value as the result of that procedure. *Report* and *to-report* are always used in conjunction with each other. We need such structure since this function is used to find a turtle that is passed as a variable to another function. The preferential partner is chosen in the following way: a random number (*total*) is generated between 0 and the total number of links. Turtles are then asked one at a time to check whether their number of links is greater than *total*. The first turtle satisfying the condition is the partner. In this way a node with many links is more likely to be chosen. It can be shown that this algorithm creates a power-law distribution of the links as expected from the preferential linking model. The reported partner is introduced in the network with the function *make-node*

```
to make-node [old-node]
  crt 1
  [
     if old-node != nobody
      [ create-link-with old-node [ set color green ]

to layout
    repeat 3 [
    let factor sqrt count turtles
    layout-spring turtles links (1 / factor) (7 / factor) (1 / factor)
    display
]
```

The first part of layout (if selected from switch) rearranges the position of the turtles. *layout-spring* arranges the turtles, as if the links are springs and the turtles are repelling each other. The first value passed is *spring constant*: it is a measure of the ”tautness” of the spring. It is the ”resistance” to change in their length. The second value is *spring-length*: it is the ”zero-force” length or the natural length of the springs. This is the length which all springs try to achieve either by pushing out their nodes or pulling them in. The last value is *repulsion-constant*: it is a measure of repulsion between the nodes.

```
  let x-offset max [xcor] of turtles + min [xcor] of turtles
  let y-offset max [ycor] of turtles + min [ycor] of turtles

  set x-offset limit-magnitude x-offset 0.1
  set y-offset limit-magnitude y-offset 0.1
  ask turtles [ setxy (xcor - x-offset / 2) (ycor - y-offset / 2) ]
```

The second part is needed to rescale the world . *x-offset* measures the asimmetry of the world on the x-axis. The function *limit-magnitude* reports x-offset up to the limit chosen (in this case 0.1). Then we ask the turtle to be repositioned in order to balance the maximum and minimum values of x. The same applies for the y coordinate.

3.3.3 Setup-spatially-clustered-network

```
let num-links (average-node-degree * number-of-nodes) / 2
  while [count links < num-links ] [
    ask one-of turtles   [
       let choice (min-one-of
                    (other turtles with [not link-neighbor? myself])
                   [distance myself])
      if choice != nobody [ create-link-with choice ]
```

Given the total number of links we start a while cycle that randomly connects two nodes until all the links have been placed. Starting from a randomly chosen turtle,a second turtle is chosen: it is the closest turtle that has not already been linked to the first one. It is an aesthetic trick that gives priority to the nearby nodes and avoids links between far away turtles.

3.3.4 Small World

```
 layout-circle turtles max-pxcor - 1
  let n 0
  while [n < count turtles]  [
    make-edge turtle n
              turtle ((n + 1) mod count turtles)
    make-edge turtle n
              turtle ((n + 2) mod count turtles)
    set n n + 1  ]
```

layout-circle arranges the turtles on a circle The while cycle links every node to its two neighbors on the right. The *mod* function is needed when the nth turtle is the last or second last one since a *make-edge turtle n turtle n+1* command would eventually ask a turtle to be linked to a non existing one.

```
ask links [ if (random-float 1) < rewiring-probability
   [ let node1 end1
   if [ count link-neighbors ] of end1 < (count turtles - 1)    [
   let node2 one-of turtles with [ (self != node1) and (not link-neighbor? node1) ]
   ask node1 [ create-link-with node2 [ set color cyan  ] ]
             set rewired? true ]]
      if (rewired?)  [ die ]]
```

This is the rewiring algorithm. A slider indicates the rewiring-probability of each link. We ask every link, if it is a lucky one, to tell us the node at its end. We check if this node is not already fully linked. In case it isn't, we look for another node among the ones not already linked to *node1* and links them. After the creation of the new link we delete the "rewired" link through the boolean variable *rewired?*.

3.3.5 go

```
set protagonist one-of turtles
```

We give a name to the first interacting turtle.

3.3.6 interact

```
to interact [ turtle1 turtle2]
  ifelse turtle2 != nobody  [
         ask turtle1 [
              set opinion1 (opinion1 + mu * ( ([opinion1] of turtle2) - opinion1))
              set color opinion1 ]
          ask turtle2 [
             set opinion1 (opinion1 + mu * (temp1 - opinion1))
             set color opinion1]
          set interactions (interactions + 1)  ]
  []
```

This time we use a function structure for the interactions using the interacting turtles as variables. The *ifelse* structure is needed to avoid an error caused if the program asks nobody. If the second turtle does not exist, the program simply moves on.

```
to-report find-neighbour
    set neighbour nobody
    set neighbour one-of other turtles with [ link-neighbor? protagonist]
    if neighbour != nobody and abs (temp1 - [opinion1] of neighbour) < d
    [report neighbour]
     report nobody
```

The second turtle is chosen with the lines of code above:we initially set *neighbour* to nobody in order to avoid the value to be memorized from previous cycles. We look for a turtle linked to a specific other one (*protagonist*) and we check whether the Deffuant condition is satisfied and in case reports as *neighbour* the found turtle or *nobody*.

4 Results and Intepretation

4.1 Basic Model

The fundamental parameters of the model are μ and d. μ determines the speed of convergence: a value of 0,5 means that the two discussing agents will meet halfway. It has been proven that d = 50 is the threshold above which all agents share the same opinion, independently of the underlying social topology [3] . We chose to use d in a (0:50) range because we want to study the non single-opinion convergence. d determines the final opinion-distribution and in particular the different number of final opinions. Deffuant in his paper shows sperimentally that the number of final opinion is given by 100/2d. At first we supposed a superior limit of 100/d as it is logical since d represents the minimum distance between two different final opinions. We did find some counterexamples proving our hypothesis but we noticed that we needed to use a very small number of agents, while Deffuant worked with a significantly superior number of agents.

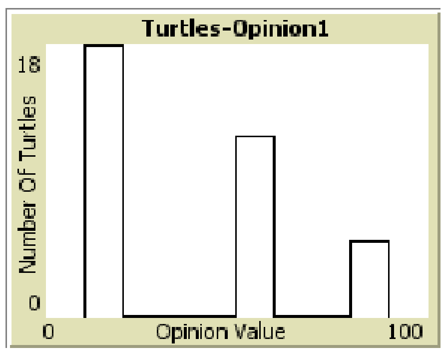


Figure 1: N= 35, d = 25 and μ = 0,5

We believe the reason for this exception is due to the limited population size. With many interacting agents the space between two peaks is significantly populated and therefore it causes the smoothing of the peak and eventually a convergence at one single opinion. With few agents and with strong uncommunicativeness (d = 25) instead it is possible that small groups of agents quickly agree on a isolated opinion that cannot be attracted by the other existing opinions. A real life parallel can be seen with mountain communities where people are isolated and are reticent to accept external ideas. Being scarcely affected by external news/information, they tend to reach a accordance independently from the opinion of the rest of the population which can be significantly distant from the ”mainstream” position.

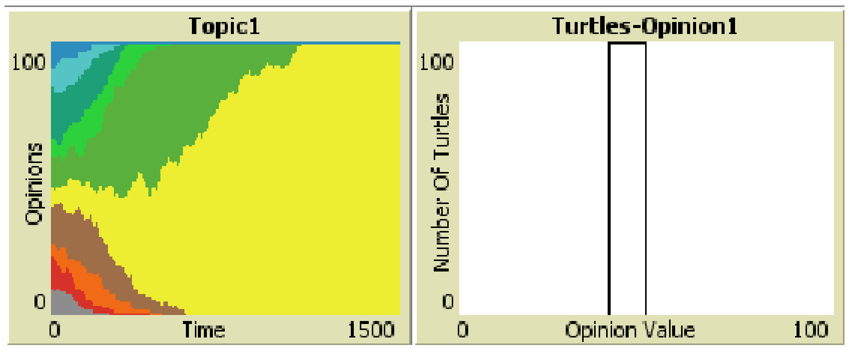


Figure 2: N= 100, d = 50, μ = 0,5, ticks = 2894, interactions = 2862

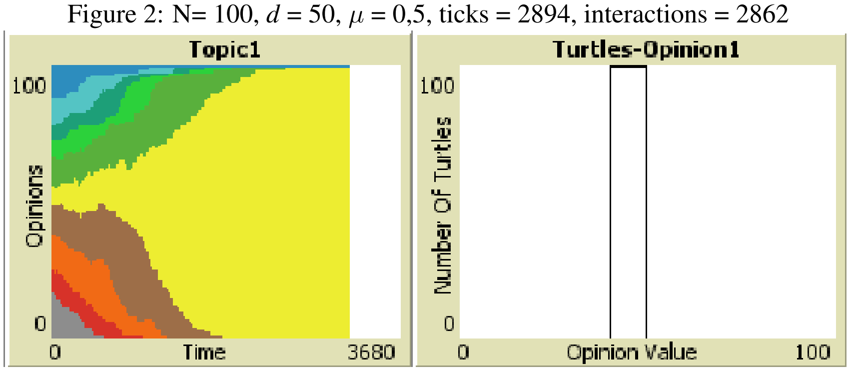


Figure 3: N= 100, d = 50, μ = 0,1, ticks = 3149, interactions = 3000

With fixed d and population size it can be clearly seen how μ affects the time of convergence. In the first simulation convergence occurs in the area of 1000 ticks, whereas in the second we have to wait for 3000 ticks.

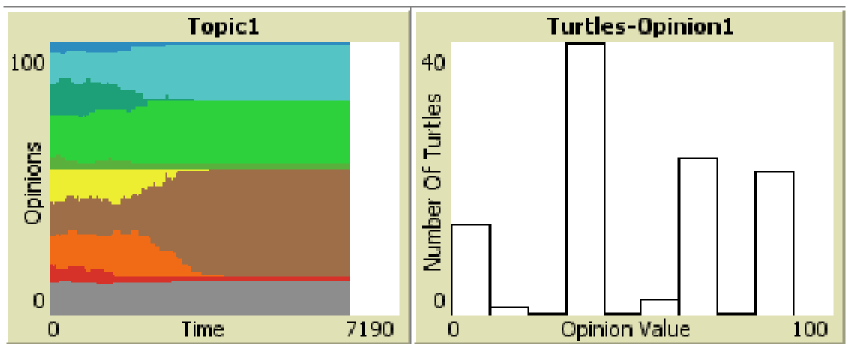


Figure 4: N= 100, d = 10, μ = 0,5, ticks = 6205, interactions = 1504

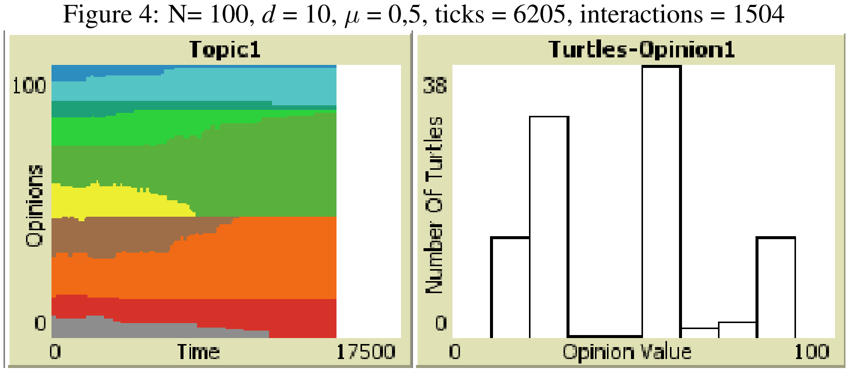


Figure 5: N= 100, d = 10, μ = 0,1, ticks = 14243, interactions = 3374

In this case d is set to 10 causing a strong diversification of the final opinion distribution. Once again it can be seen the importance of μ in the temporal analysis of the data. Both situations are static, but convergence has been reached on totally different time scales.

4.2 Extremists

The presence of extremists in the world causes a splitting in the final opinion. Agents are attracted towards the sides of the opinion interval with potential intermediate convergence according to *d*. The 0 and 100 opinions are basin of attraction for the world population. They represent static ideas that could be associated to real world opinion-makers that influence people constantly.

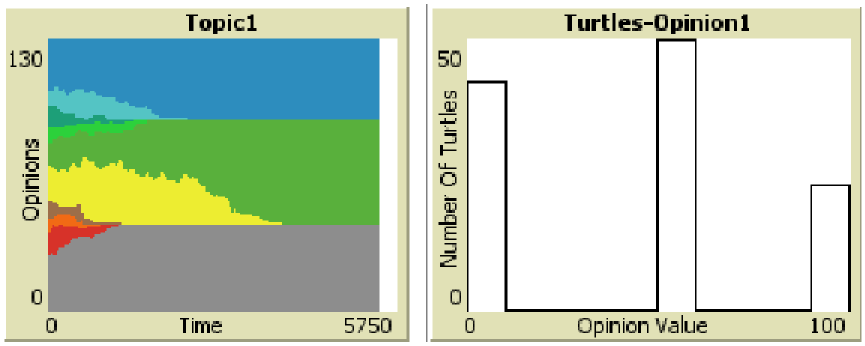


Figure 6: N=100, Extr0=15, Extr100=15, d=25, μ=0,5, ticks=5445, interactions=1838

It can been seen how the extremist attract into their opinion range the nearby population. Only a central group of agents is far enough from the opinion of the extremists and eventually survives.

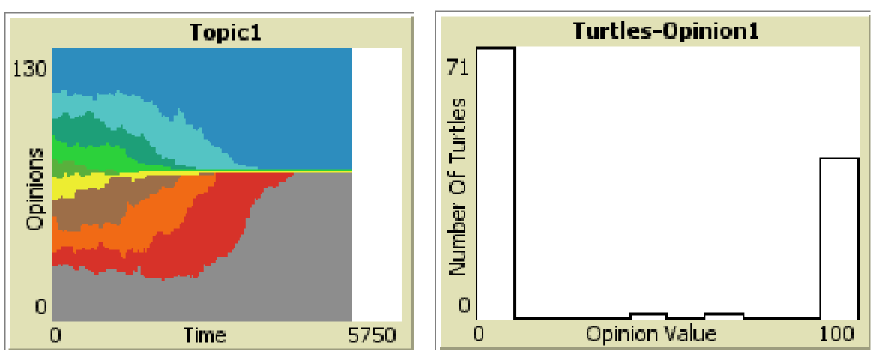


Figure 7: N=100, Extr0=15, Extr100=15, d=25, μ=0,5, ticks=4965, interactions=2231

In this case the extremist symmetrically attract the population towards them. A chain of convergence can be seen. Opinion intervals closer to 0 or 100 are gradually more populated (brown merges into orange, orange into red, red into grey) until the extreme opinions prevail. Some intermediate opinions survive, but they are statistically irrelevant.

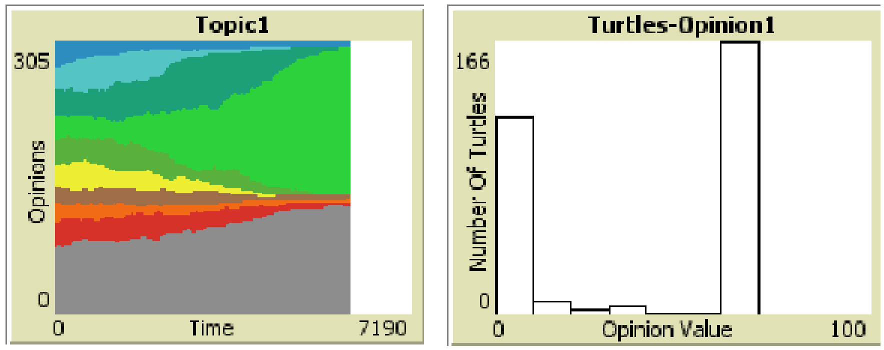


Figure 8: N=250, Extr0=50, Extr100=5, d=25, μ=0,5, ticks=5942, interactions=2669

In this case we changed the symmetry of the system, creating 50 extremists with opinion 0 and only 5 with opinion 100. The presence of so many extremists of one type causes the shifting of the "opinion barycenter" towards its side. The graph is qualitatively identical to the previous one but the 100 extremists are not strong enough to attract people to their opinion. It is an example of a majority-minority situation. The voice of the minority is still present but is influenced by the dominant opinion.

4.3 Place?

Place represents a second parameter of interaction. Not only opinions have to be close enough to discuss but agents also have to be within a specific range. Agents move around the world only if they do not find a partner with whom they agree. This leads to a correlation between opinion and position, allowing parallel peaks within range d but unable to merge due to geographical distance. This can be associated with typical provincialism (campanilismo) where residence becomes relevant in the discussion process and eventually can become an obstacle even though opinions are significantly close.

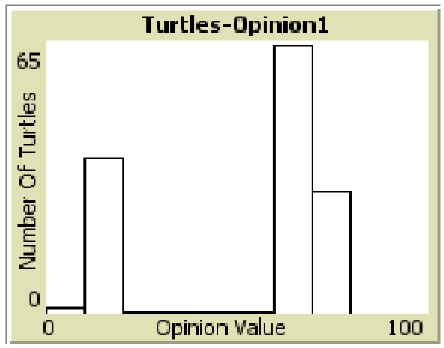


Figure 9: N=131, d=34, μ=0,5, ticks=6800, interactions=4200, range=10

It can be seen how within a d range of opinion subsets of localized agents arise. Even though the two subsets satisfy the Deffuant condition, their place variable is out of range and therefore disables discussion.

```
observer> show [opinion1] of turtle 58
observer: 60.97256263339226
observer> show [place] of turtle 58
observer: 94
observer> show [opinion1] of turtle 59
observer: 60.921933076370124
observer> show [place] of turtle 59
observer: 98
observer> show [opinion1] of turtle 69
observer: 60.83776781008007
observer> show [place] of turtle 69
observer: 93

observer> show [opinion1] of turtle 103
observer: 70.62321227064467
observer> show [place] of turtle 103
observer: 54
observer> show [opinion1] of turtle 104
observer: 71.27229364276529
observer> show [place] of turtle 104
observer: 41
observer> show [opinion1] of turtle 105
observer: 71.09015602699546
observer> show [place] of turtle 105
observer: 35
observer> show [opinion1] of turtle 107
observer: 71.20574220291257
observer> show [place] of turtle 107
observer: 30
```

4.4 Multiopinion

4.4.1 One By One

In the MultiOneByOne scenario successive topics are discussed only if there is agreement on the previous ones. The modality of convergence of higher topics determines speed and modality of convergence of the lower ones. When the first opinion-distribution converges to a single opinion, discussion is guaranteed on the lower level at a good rate. However, when the first opinion-distribution converges to more than opinion, at lower levels in order to discuss, turtles have to belong to the same "agreement group" of the final configuration of *opinion1* and at the same time be in range *d* in *opinion2*. This mechanism of conditioned probability slows down significantly the interaction process leading in some cases to absence of convergence at lower levels. This structure can be compared to meetings where a "topic list" is introduced. Discussion on the first topic is dominant and the agreement on successive topics could be influenced by the opinion on previous ones.

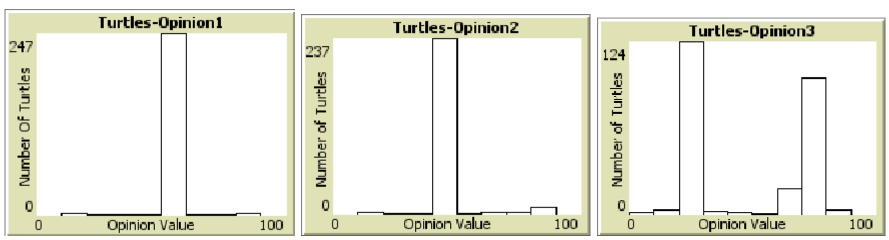


Figure 10: N=250, d=30, μ=0,5, ticks=4457, interactions1=3584, interactions2=2583, interactions3=1332

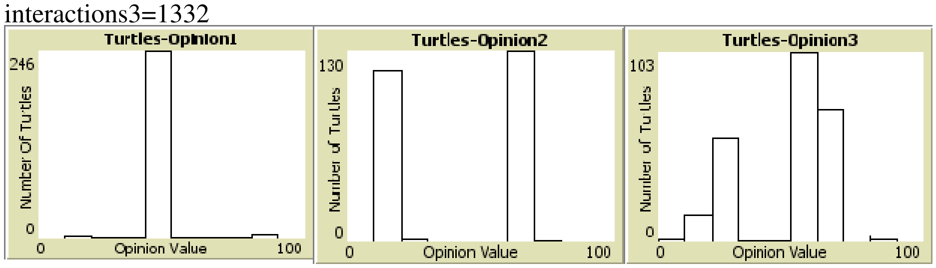


Figure 11: N=250, d=30, μ=0,5, ticks=11250, interactions1=9970, interactions2=5038, interactions3=3021

In the above graphs the correlation between convergences of different topics can be seen. Both simulations have the same parameters but lead to different results. In the first case the first two topics converge smoothly and the third one qualitatively converges to two opinions (possible given the parameters) but with "noise" effects. In the second case a split is present at the second level, causing an equilibrium situation at the third level not compatible with the given parameters: nearby peaks are unlikely to interact. Only 3000 interactions have taken place in a 11000 tick interval.

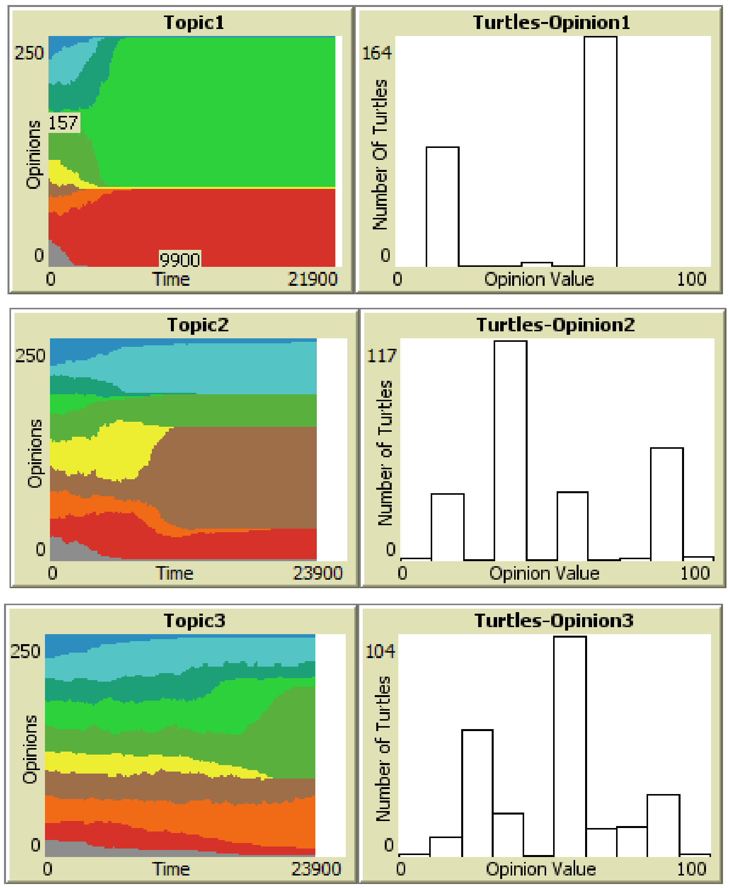


Figure 12: N=250, d=20, μ=0,5, ticks=21397, interactions1=11088, interactions2=5282, interactions3=2479

Reducing the parameter d the hindering mechanism is emphasized. The correlation between time scales can be clearly seen. A splitting is already present on the first topic after a few ticks (4000). Only after convergence of *opinion1*, the distribution of *opinion2* starts to be modified eventually leading to convergence on a very long time interval (12000 ticks). The same mechanism happens for *opinion3*: the original distribution substantially survives until convergence of *opinion2* and then is modified. However, as mentioned above, the conditioned probability (this time even more unlikely due to the higher number of peaks) leads to non convergent situation after 23000 ticks. We have observed that the number of final opinions at a certain level represents the lower bound of final opinions for the successive one.

4.4.2 Multirandom

The MultiRandom structure is similar to the basic model. As soon as one topic reaches convergence, the distribution of opinion of other topics quickly converges as well. This is caused by the fact that total agreement on one topic satisfies the multiple-opinion condition (agreement on at least two topics) and thus facilitates interaction. This interaction model describes informal situation where there is no priority in the topic discussion. Agreement on one of them however leads to a "smoothening" of relations and thus leads to faster agreement on all levels of interaction.

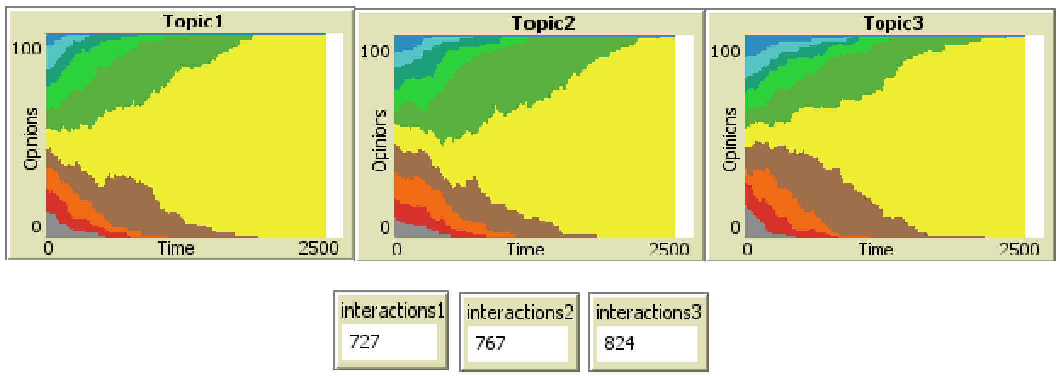


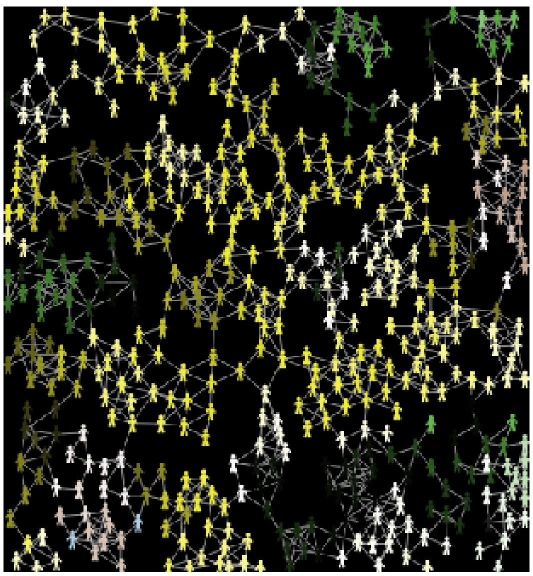
Figure 13: N=100, d=50, μ=0,5, ticks=2350

The number of interaction is very similar: this is caused by the random choice of the topic of discussion. The correlation of time of "beginning of convergence" can be seen from the graphs: the distribution of *opinion1* is the first to show a tendency to converge, represented by the widening of its yellow band; after this event, the yellow bands of other graphs, which were not growing before, start to expand and eventually all opinions converge at the same time.

4.5 Network

4.5.1 Random Graph

The network structure significantly modifies the results of the model, even though there are no further modifications. Compared to the complete mixing scenario, in this case interactions are mediated by links between agents. This causes a strong phenomenon of localization of opinions and of clash between scarcely connected areas. Results are qualitatively resembling the ones for the basic model, but full convergence is never achieved in temporal intervals up to one magnitude higher compared to the ones needed for convergence in the original version. We decided to use ”limit conditions” (d = 50, μ) in order to have quick and clear results. The typical scenario under these conditions is the presence of a long chain of agents with a strong dominant opinion surrounded by small groups of agents with different opinions that survive for a long time. Full convergence in shorter time scales is extremely unlikely due to the limited structure of interactions. Strongly connected groups that are scarcely connected with outer agents tend to slow down the process by blocking the diffusion of the mainstream opinion. This can also lead to the creation of parallel converging processes in different areas of the world that divide the agents in two huge blocks of similar opinions but that are unable to reach a definite agreement due to the constant conflict on the border between the two areas. A strong analogy with a segregation model can be seen as well as the modeling(modelization?) of interaction among family structures. The choice of an average node degree of five is comparable with a typical small social group and thus the opinion among the group has stronger impact within the members than the opinion coming from outside in spite of its global strength.



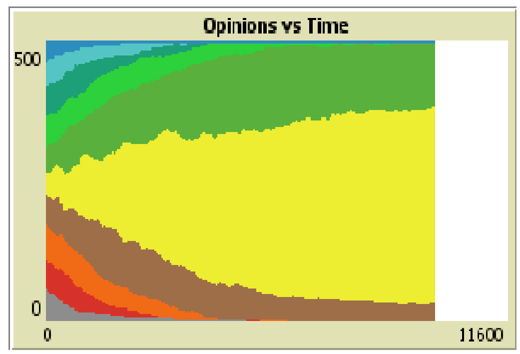


Figure 14: N=500, d=50, μ=0,5, average node degree=5

It's possible to see a long chain of yellow agents representing the dominant opinion. However small green groups survive in different areas of the world. The Deffuant model pushes towards convergence in the green-yellow-brown range and thus the color distribution is explained. The survival of different opinions however is due to the topological structure of the network.

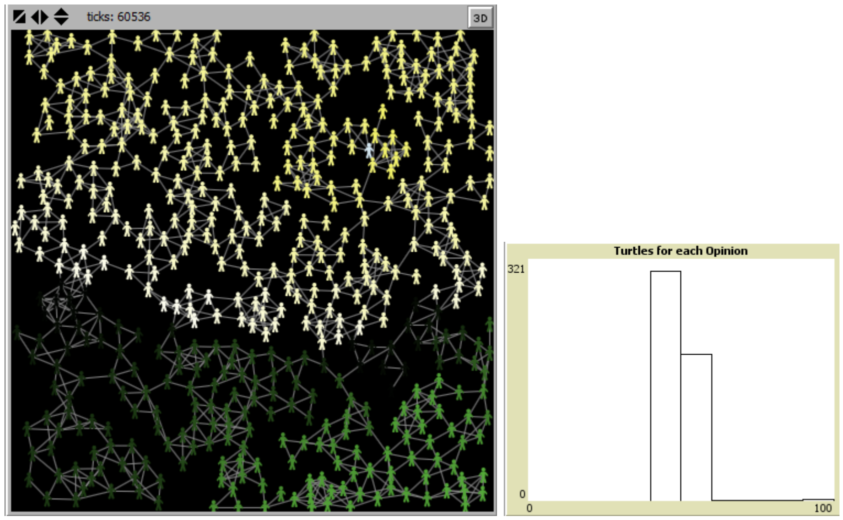


Figure 15: N=500, d=50, μ=0,5, average node degree=5

In this case we can see the fight among two macrogroups with very close opinions. The basic model worked well but convergence arises from two separate areas of the world thus causing a strong clash in the central nodes that link the two sides. They keep switching from one side to the other with "smoothed" opinion, but full convergence requires very long time.

4.5.2 Small World

Convergence occurs for very high time scales if compared to the basic model. Convergence time heavily depends on the rewiring probability. For very high values convergence is extremely fast. This might seem counterintuitive considering that a small world network with high rewiring probability is statistically equivalent to a Random Graph. However the underlying starting chain-structure survives enough to cause an absence of strongly connected clusters that block the converging process.

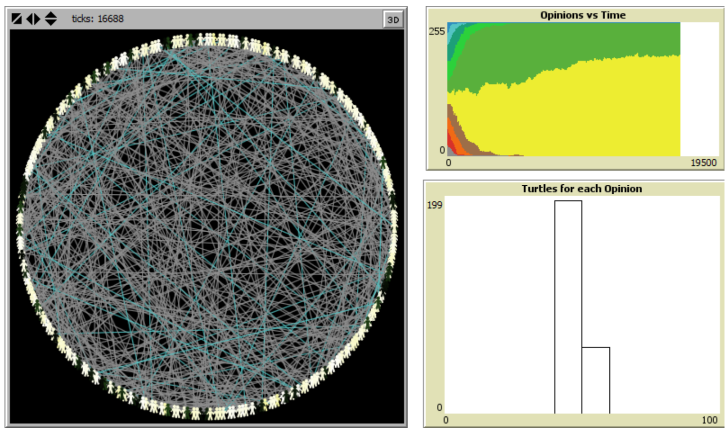


Figure 16: N=255, d=50, μ=0,5, rewiring probability=0,1

There is a prevailing opinion that will eventually converge, but after 19000 interactions no static distribution has been reached.

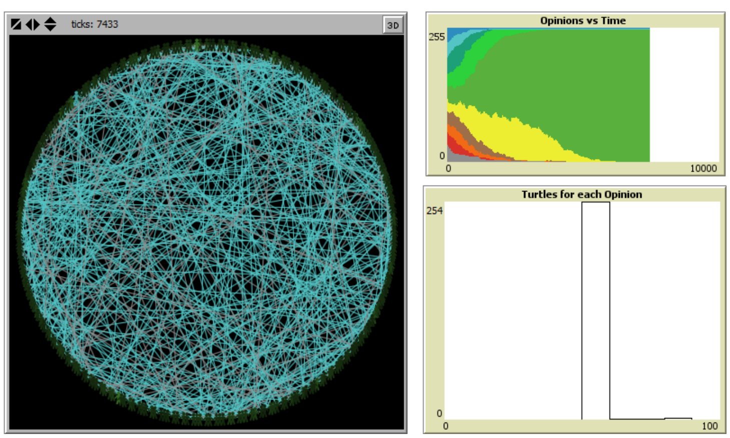


Figure 17: N=255, d=50, μ=0,5, rewiring probability=0,8

Convergence has been quickly reached.

4.5.3 Preferential Linking

In the preferential linking scenario we notice that the dominating opinions are the ones impregnating the areas of the hubs. They influence everyone around them and are influenced by all agents linked to them. Once a hub area has been ”conquered”, the opinion quickly spreads to nearby hubs. Similarly to the random graph situation, blocks of different opinions (related to the relative hub) emerge, but eventually there is convergence even though for high time scales.

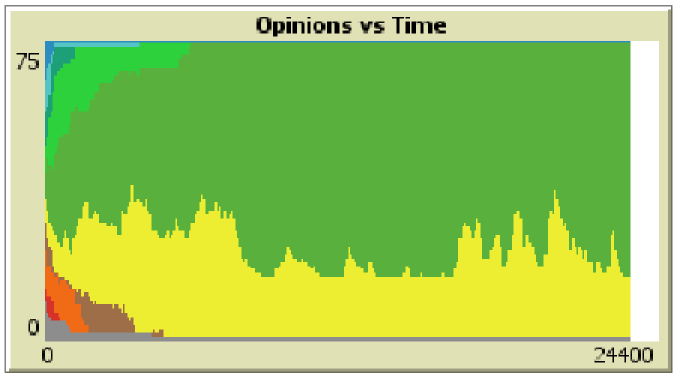


Figure 18: N=75, d=50, μ=0,5

Compared to other network structures, this one is particularly vulnerable to specific attacks to hubs. We can see here how the distribution suffers from quick fluctuations absent in other networks.

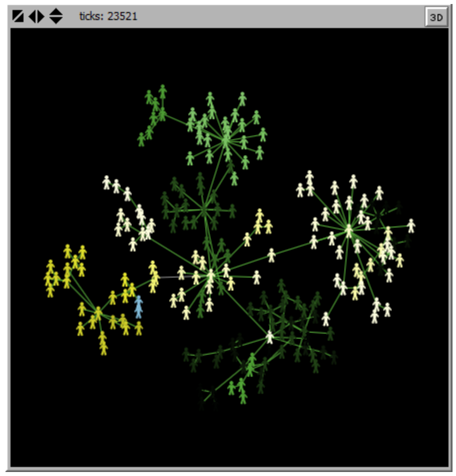
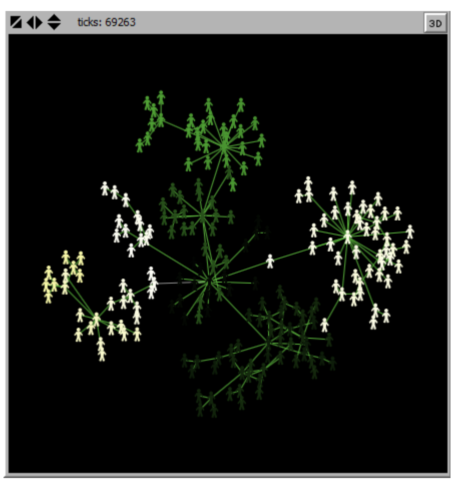
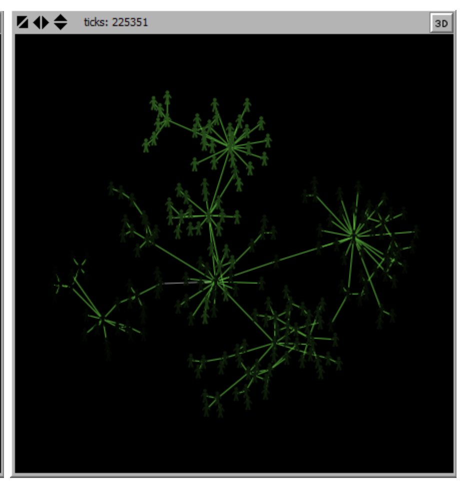


Figure 19: N=210, d=50, μ=0,5

After more than 20000 interactions no convergence has been reached. There is a dominating range of opinions but the topological structure of the network allows the existence of clusters with different opinions. It can be seen how the central hub is cromatically unstable and is linked to unicolor blocks.





We can see the importance of the central hub mentioned before. The green side has taken over the hub and eventually dominates.

The network structure turned to be useful to study their own statistical properties. However we noticed how the biggest limit in the social interpretation of the results was the excessive staticity of the links. They enhance the difficulty of discussion since they limit the possibility of opinion mixing, similarly to what the *Place?* switch does. It would be interesting to study the same model on a dynamic network that allows the relinking of nodes following a preferential re-linking law.

References

[1] Deffuant G., *Mixing beliefs among interacting agents*. Adv. Compl. Sys. 3, 87, 2000.

[2] Barabàsi A.L., Réka Albert, *Statistical Mechanics of Complex Networks* Rev.Mod.Phys. **74**,47 2002

[3] Fortunato S., *Universality of the Threshold for Complete Consensus for the Opinion Dynamics of Deffuant et al.*

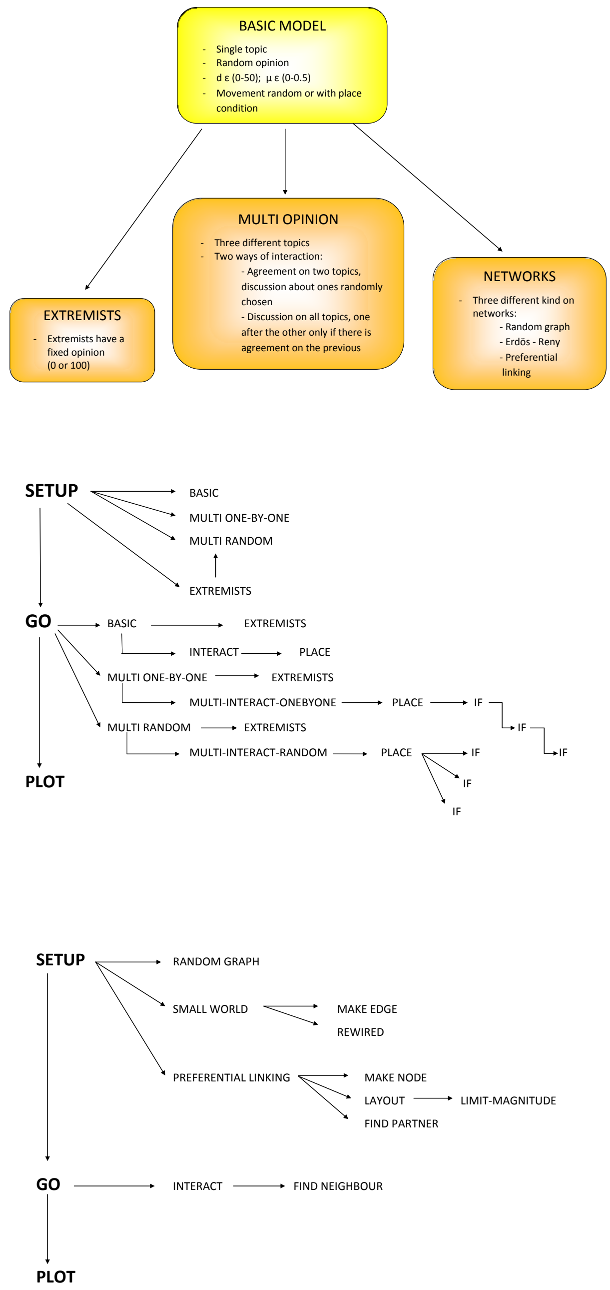


Figure 20: C-Maps of the model and of the programming