Corso di Laurea Magistrale in "Economics"

An Agent-Based Simulation Model of a Healthcare System: the Specialistic Network in the Area of Turin

Supervisor:

Prof. Pietro Terna

Co-supervisor:

Prof. Sergio Margarita

Candidate:

Alberto Baldovino

Academic Year 2013–2014

Contents

Introduction

There exist a great amount of studies about healthcare industry, especially because the great role played by this sector within modern economies. It is indeed one of the world's largest and fastest growing industries, using up over 10 percent of gross domestic product of most developed countries. It is a complex system, made up of different sub-sectors providing goods and services to treat patients, involving a set of heterogeneous stakeholders such as treatments-providers, payers and, of course, patients. In recent years the healthcare theme has gained fame, in particular as a consequence of the emergence of many problems, ranging from the need for rearrangement of some areas to the increasing costs characterizing such domain. Policy-makers are striving for finding solutions, or at least for easing problematic situations.

Given this situation, we consider useful to take a deeper look at this domain from a different point of view: the simulation perspective. The main advantage is that it allows us to re-create a real-world process or a system in a controlled environment, even despite an incomplete set of information on the real situation we are investigating. It is what can be called a "Proof of Concepts", aimed at verifying that some concept or, hopefully, the entire model we created has the potential of being used even for real applications.

We have decided to analyse this topic for two main reasons: firstly because of the key role played by the healthcare industry over the entire economy of every countries all around the world, with stronger weights in those labelled as "developed". A second reason of interest lies on the consequences that nowadays restructuring healthcare systems policies such as revising hospital

and diagnostic tariffs, reducing the number of bed spots within healthcare institutions, increasing co-payments for pharmaceuticals, and finally reducing human resources through limits on staff turnovers, can have on the patients lives.

All in all, our goal is trying to provide you with a simulation instrument, developed with a simulating software called "NetLogo", which is able to evaluate different scenarios given different assumption time by time set by the user of the platform.

We have adopted a clear structure while developing our work and, as a result, we could split our work in two main areas, corresponding to the two main chapters:

i) Literature Review: it is a huge part and it lays the theoretical foundations of our entire work. It consists on a set of knowledges that must be taken into account in order to get a better understanding of the area our model is aimed at. In particular, such chapter is developed as follows. Paragraph 1.1 define the general purpose of simulation: a tool in support of a vast variety of stakeholders such as health economists, health policy makers and health insurers to name a few. The reason why such tools turn out to be effective is the degree of intrinsic uncertainty the healthcare environment is affected by. Computer modelling and simulation represent a competitive alternative to the trial-and-error process or to empirical research. Paragraph 1.2 deals with the obstacles affecting the implementation and uptake of the modelling and simulation by the healthcare community. In particular five elements have been pointed out as main alleged obstacles: costs, simulation awareness, skills and experience, organizational factors, technical factors. Paragraph 1.3 analyses in detail the methods for validation of population-based disease simulation models. Paragraph 1.4 represents a first breakthrough within this first part since it goes in detail in the simulation field, considering the particular case of "Agent Based Social Simulation". It can be intended as the modelling or simulation of social

phenomena or objects (for example societies, organizations, markets, human being and so on) which is normally performed by a computer. The reason why it is so worthy to rely on this type of simulation while tempting to analyse societies is that they, with particular emphasis on human societies, are often complex adaptive systems characterized by a lot of non-linear interactions between their members or between people. As a consequence, traditional computational and mathematical models can hardly represent these kinds of complex systems since complex social processes can not be represented in equation terms (at this point agent-based social simulation turns out to be effective and useful). In addition, agents based-models create agents that can have a one-to-one correspondence with the individuals (or organizations, or other agents) that exists in the real social world being modelled, with the interactions between the agents likewise corresponding to the interactions between the real world individuals. The paragraph is then concluded by saying that this particular type of social simulation represents a methodological approach that allows a rigorous testing, refinement, and extension of already existing theories that have proved to be difficult to formulate and evaluate using standard statistical and mathematical tools together with a much more deeper understanding of fundamental causal mechanisms in multi-agent systems whose study is currently separated by artificial disciplinary boundaries. Paragraph 1.5 deals with the topic of "Complexity". It begins by saying that a unique definition does not exist, even among scientists. It is then tackled the topic concerning the areas of applicability of such concept, showing that such issue can be found in several aspects of our everyday life. Then it is explained the distinction existing between the terms "Complex" and "Complicated". Finally, the paragraph is concluded with a set of different meanings of Complexity, different according to the domain each time the term "Complexity" is referred to. The last paragraph, 1.6, gives an insight to the concept of "Networks", which is really important when it comes

to deal with agent based simulation. In this section we start with some examples where we can find out the presence of these networks, to get familiar to those, and then we consider the particular case of social networks, describing their structures and their main features.

ii) The Model: this part represents the core business of our thesis. It consists in a model we developed by means of a simulating software (NetLogo), which will allow us to make a vast thinking given some initial condition and some rules. We have inserted some specific commands that, run in sequence by the software, constrain the Agents we have created to act (born, move, die) in a not-random way. The technique we have used is that of Agent Based Modelling. As we have seen in the literature part in the section committed at explaining ABM, this particular way of modelling allows to built "bottom-up" models, characterized by a set of agents acting and interacting among themselves and with the environment they are located in. In paragraph 2.2 we have described the procedure used to create the environment of our simulated world, which is aimed at representing the district of Turin. Paragraph 2.3 is then devoted to the creation of the Agents we have included in our simulation: Hospitals, Specialists, Professionals, Familydocs and Patients. Paragraph 2.4 goes in detail analysing the rules behind the actions the agents are supposed to endorse during the simulation. Paragraph 2.5 describes the last type of agents, the "Links", identifying a connection between the other Agents previously mentioned. The last part, included in paragraph 2.6, is the more significant since it allows the user to experiment directly the potentials of our model.

As already said, the goal of our work is trying to provide the best reasoning tool we are able in order to reproduce in a fair way what can happen in a real district healthcare environment. However there is a limitation. Since our work relies only partly on real data, a more developed and sensitive tool can be developed in future works.

However, despite the limitations just mentioned, we hope that its discussion will serve as motivation for more detailed research on this topic.

Chapter 1

Literature Review

1.1 Modelling and Simulation

In the following pages we analyse the literature about simulation modelling in healthcare domain. In particular, we want to create a clear framework to understand better a field we will work in by constructing a model of our own.

The use of simulation plays a central role in assisting decision maker in the field of healthcare, as Kirchhof and Meseth (2012) points out. This fact is supported by Fackler and Spaeder (2011), where it is stated that a variety of stakeholders, such as health economists, health policy-makers and health insurers, to name a few, have been made use of population-based models.

Healthcare needs, together with demands and outcomes of healthcare, share a high degree of intrinsic uncertainty. Indeed, since they require that healthcare policy should be based on evidence, they must be designed to cope with complex systems. So computer modelling and simulation should be helpful in providing evidence on how to cope with such problems, as a competitive alternative to the trial-and-error process or to empirical research, as it emerges in Fone *et al.* (2003).

A similar idea comes out from Escudero-Marin and Pidd (2011); indeed, Escudero-Marin and Pidd state that simulation methods have long been used to model elements of healthcare with a view both to understanding and im-

provement. Depending on the level of understanding about the real system, it is assessed the role of a simulation either as a

(i) *Generator*

(ii) or *Mediator*

(iii) or *Predictor*.

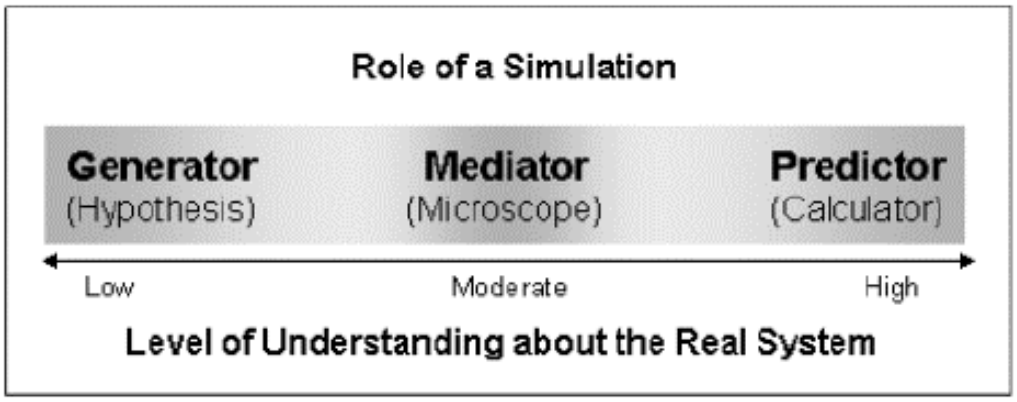


Figure 1.1: The role of simulation models depends on the the level of understanding the real world it provides

A simulation can be used as a "Predictor" when it provides a high level of understanding about the real system. It means that it produces clear predictions about the system's behaviour it is expected to explain under defined conditions. When, instead, the level of understanding is low, the simulation acquires a "Şgenerator" role, i.e it can generate hypotheses and theories about system behaviour, but not precisely. Finally, when a simulation shows a moderate level of understanding, it is endowed of a "Şmediator" role: though providing insight into the behaviour of the system, it does not offer a complete representation of that behaviour.

As stated in Kirchhof and Meseth (2012), academic literature in simulation is massive and increases constantly. There are nearly 200,000 journal papers on the simulation or modelling of care delivery processes, and more

than 30 appearing each day, as it emerges from Fackler *et al.* (2012). However there is still a surprising lack of adoption, proved by rare records of applications to real-life problems. Still, the gap in the healthcare domain is bigger than those recorded in sectors like manufacturing or logistics. Regarding these differences, a useful graph should be taken into account. It represents the results obtained in Eldabi (2009), where a cross-sectoral study is carried out in order to compare academic work in different sectors: healthcare, defence/aerospace, industry/business. This work is aimed at showing the differences in these sectors, underlying the prevalence of theoretical work in contrast to practical applications in healthcare simulation. Figure1.2 shows the results. It is made up of three bars associated to the three sectors previously mentioned. Each bar is then split in three subcategories, whose dimensions represent the percentage of each category within each sector. The first subcategory includes those articles inspecting real problem with real stakeholders. The second one contains papers addressing real-life problems without engagement for real stakeholders. The last category is filled by entirely theoretical papers.

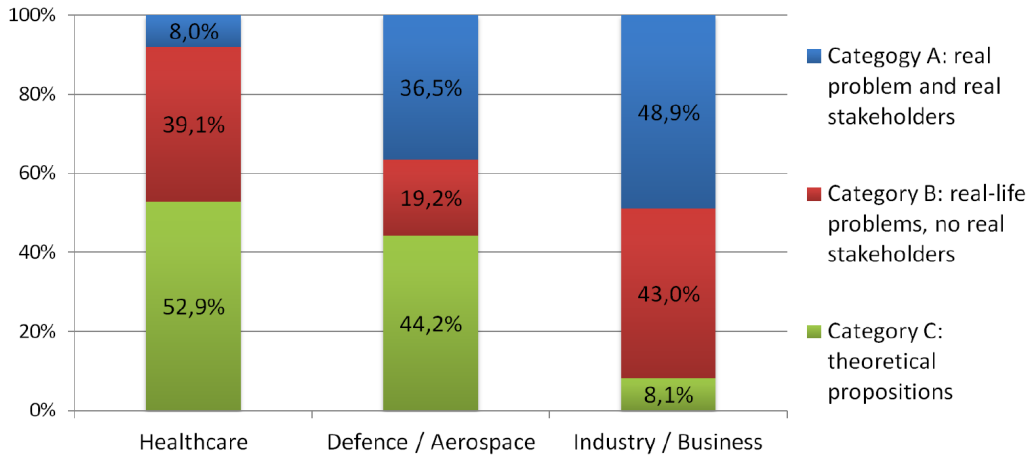


Figure 1.2: Cross-sectoral comparison of practical simulation applications. Strong and clear evidence in favour of the mainly theoretical feature associated to research papers in the healthcare field.

Only 8% of healthcare simulation articles deal with real problem, which is a small percentage if compared with that emerging in manufacturing and business sectors, 48.9%. If, instead, we consider the theoretical papers only, the situation is overturned. A great dominance of healthcare related works over the industry/business' ones; respectively, 52.9% and 8.1%. Clearly this suggests that there are some barriers for what can be termed as Şsimulation successŤ within healthcare domain, particularly that modelling in this field has started since the early beginnings of the other two domains. Such barriers, that in the following section will be analysed, lead to a scarce use of simulation in this sector.

A further step can be done in order to evaluate the quality and value of computer simulation modelling in population health and healthcare delivery. There exists another study (Fone *et al.*, 2003) which helps us in doing so properly. The authors carried out a systematic review of world literature from 1980 to 1999. Papers were included within such review if containing two elements: a computer simulation model of individuals in a stochastic system and a setting related to population health or health service delivery. Among all the papers, 182 met the inclusion criteria only. However, the quality of published papers was variable, and few reported on the outcomes of implementation of models. So, since no formal investigations has been carried out to establish the extent to which findings have been translated to policy, the authors have concluded that the value of such research works modelling-based could not be assessed. However, simulation modelling still remains a powerful tool, applied to a vast range of situation within the healthcare domain. Moreover, they found out that, even if the quality of papers is variable, it is improving over time. So, although the lack of implementation is one of the most characteristic problem associated to modelling, still the potential of simulation tools in providing aid to policy development is clear.

Another research study concerning the level of implementation of research project into practical healthcare situations is due to Brailsford *et al.* (2009). His literature review constitutes, together with seven other reviews,

the ground on which the RIGHT (Research Into Global Healthcare Tool) project builds on its efforts. RIGHT is a collaborative research venture between six UK universities, whose main aim is to assess the feasibility of applying to decision making in healthcare some of the most successful modelling and simulation methods used to support decision-making in other domains, such as defence and manufacturing industry. The type of modelling taken into account by such project is the one understood as a structured approach to understanding, and possibly solving, a real-world problem by developing a simplified version of the real world. The other most important aim of RIGHT is to provide an explanatory framework that will suggest the most suitable method to use, for a certain type of healthcare problem and available resources at the user's disposal. This project is based on a set of 342 research works, whose publication dates ranges from 1952 to 2007, selected if containing a genuine application of modelling or simulation to a healthcare problem. The 82% of the papers has been published after 1990, with the following percentages concerning the country in which the research study was carried on: North-America (60.2%), Europe (24.6%), Asia (9.1%), Africa (2.9%), Australia (1.8%), South and Central America (0.3%). It is almost clear that the vast majority of studies has been undertaken in North America and Europe. This represents the first result of such review. An other important emerging outcome concerns the analysis of publications. First of all they have been divided into six subcategories, according to the type of modelling prevailing, which are: qualitative modeling, mathematical modeling, statistical modeling, statistical analysis, simulation and other, which is a residual subcategory. A smaller but significant number employ mathematical modelling, very few belong to the residual category. It interesting to note that where qualitative methods are used, they are very often a subsidiary method, while where mathematical methods are used, they almost represent the primary instrument. This result is clear if we look at Figure 1.3.

Another important result is shown into Figure1.4. It represents the distribution of methods by year of publications. This graph clearly indicates

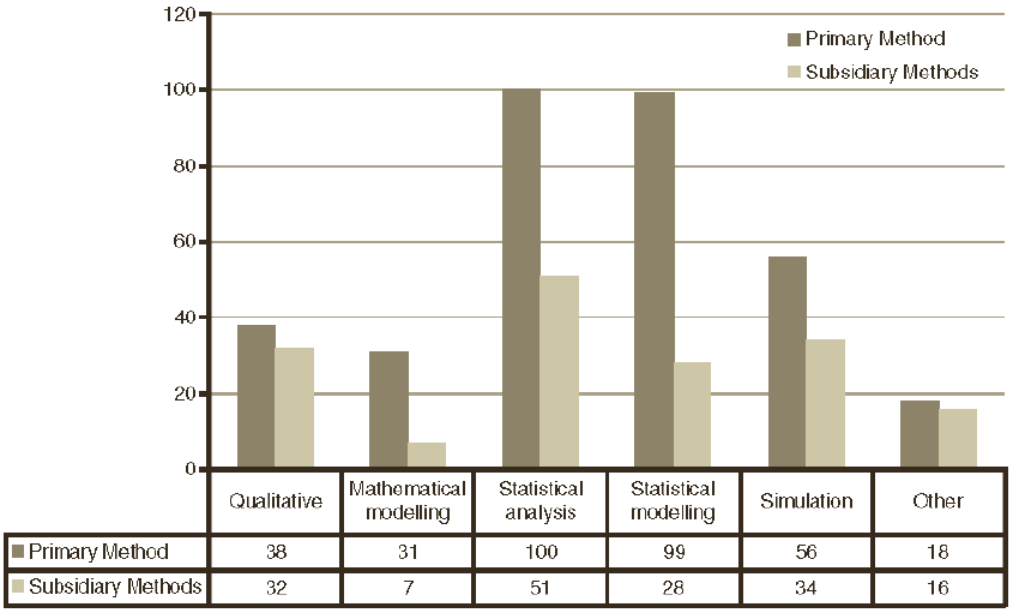


Figure 1.3: Classification of research works according to the prevailing type of modelling. Further distinction between Primary and Subsidiary methods, measuring the extent to which each type of modelling is used.

that simulation methods are currently increasing in use, supporting the ideas of Fackler *et al.* (2012). By contrast, other methods appear to have a similar uptake to the previous decade, with mathematical modelling methods possibly in relative decline.

The level of implementation is the last point the research project has focused its attention on. The extent to which the model has actually been used in practice for its stated purpose represents a key aspect of any study. Each modelling study was rated according to a three-level scale of implementation:

(i) Suggested (theoretically proposed by the authors)

(ii) Conceptualized (discussed with a client organization)

(iii) Implemented (actually used in practice).

The results of the research are shown in Figure1.5. The articles have been rated into the three categories as follows: Suggested(50%), Conceptu-

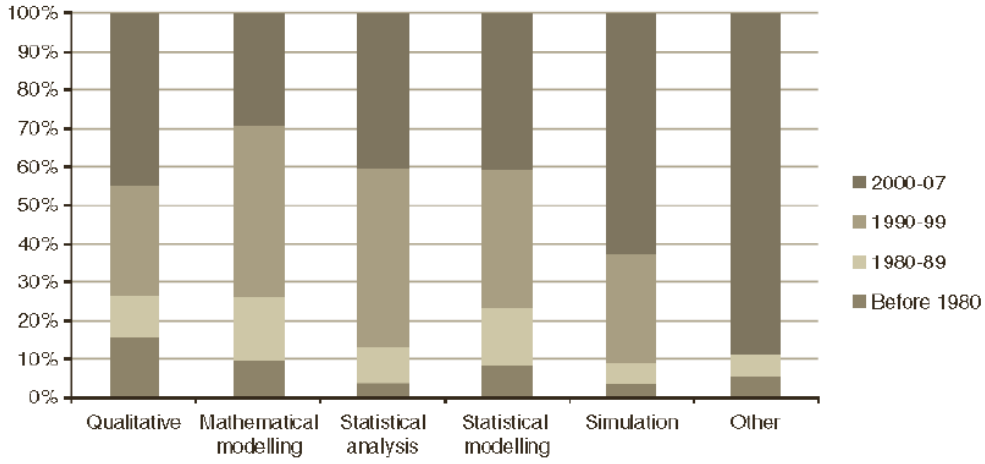


Figure 1.4: Distribution of methods by year of publications: increasing use of simulation methods as time goes by.

alised(44.7%), Implemented(5.3%). The emerging situation is quite negative; almost one out of twenty research works in health-care domain is actually used in practice. Further differences rest among each method. Statistical analysis had no istances of implementation, because these methods are perhaps very difficult for the lay person to understand. They often do not need a "client" as such, since they may simply involve the application of statistical methods to secondary data derived from the literature. By contrast, qualitative works cannot be used without interacting with human beings, and so they require a client. Therefore, such methods report a comparatively high level of implementation.

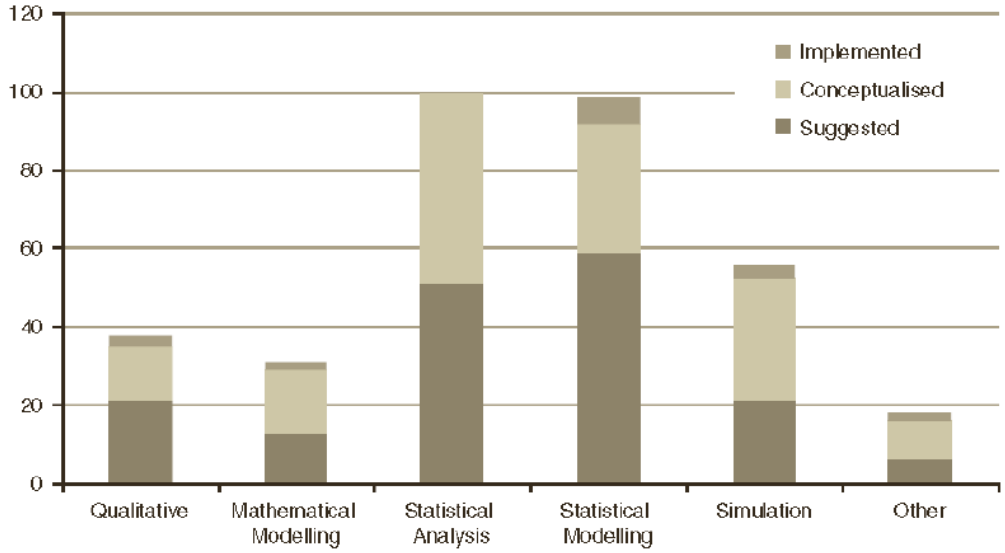


Figure 1.5: Level of implementation of research works. Quite negative results: only 5.3% of publications in the healthcare domain is actually used in a practical way.

1.2 Obstacles Affecting the Use of Simulation

Since the beginning of XXI century, modelling and simulation in healthcare domain has registered unprecedented attention from the academic community. Much of this literature has focused its work on barriers facing implementation and uptake of modelling and simulation by the healthcare community. As pointed out in Kirchhof and Meseth (2012), there are several possible obstacles to a wider use of simulation in healthcare institutions. A survey has been carried out, trying to highlight which of the five categories of supposed hurdles has the greater influence on the lack of simulation adoption. The alleged obstacles are:

i) costs,

ii) simulation awareness,

iii) skills and experience,

iv) organizational factors,

v) technical factors.

The sample was composed of 32 health German institutions. Even if German healthcare system is quite different from the Italian and other countries' systems too, the results are significant.
Figure1.6 helps in understanding the results provided by the survey. In the left side it shows the five categories of potential hurdles previously mentioned, each of them containing some items analysing the category more in detail. The participants to the survey have been asked to answer to all of them. Next to each item we can see the percentage of the sample considering whether such item is a major reason for failing the use of simulation or not. At least the 63.3% of the firms respondent to the questionnaire used to build up such survey ascribe a major role to *costs* in determining the failure of

a simulation. Only a maximum of 10% do not. Almost 67% of the sample maintains that a major reason is the *lack of knowledge of benefits of simulation.* Another problem are *missing skill.* The 80% of the institutions consider as a major reason the shortage of internal simulation skills, while only the 3.3% do not consider such point as relevant. This result proves the fact that more than three out of four (76.7%) institutions consider costs for external consultancy as a major cause, since external support should be looked for in order to compensate the internal lack of skills. An other obstacle concerns the *organizational level.* 30% consider as a major use against the use of simulation a low confidence in its capability of providing results that are able to lead towards a change. Moreover, 40% consider a major reason the preference towards an intuitive approach to decision making rather than an analytical one. It could show that, even if decision maker regards simulation as a way to solve their problems, as 46.7% among all respondents do not consider the lack of confidence in simulation as a critical reason against its use, there could be other causes preventing them from using it, such as costs. So organizational issues are not as important as other categories. Finally, *technical obstacles* have great importance too. 60% consider the development of simulation models too slow for the decision making process, and so it is regarded as a major reason. More than one out of two (53.3%) think that a simulation model requires too much maintenance to be used for decision making. Moreover, 43.3% seem to prefer other techniques to gather the information they need.

What clearly emerges is the dominant role played by costs and lack of skills as main obstacles for the use of simulation (roughly three out four respondents rated any of the reasons cost-related as major roles, while four out of five say that lack of skills among their staff is a problem). Unawareness of the benefits from simulation comes immediately after, registering 66.7% of the respondents supporting it. Organizational reasons are not, instead, considered as a huge obstacle to overcome.

A different set of barriers to implementation of modelling and simulation projects in healthcare is offered by Eldabi (2009). The author has categorized the obstacles developed by the literature in healthcare simulation domain into three main classes of barriers: (i) conflicting interests, (ii) lack of relevant tools and (iii) the mismatch of expectations between users and modellers.

(i) *Conflicting interests of stakeholders:* the majority of publications have highlighted the complexity of decision-making process within healthcare field. It is rare to find out a modelling exercise seeking a unified goal. Such lack of common interest will lead to the model failure

(ii) *Lack of relevant tools:* healthcare sector has borrowed most of its modelling tools from other domains such as manufacturing. Consequently, problems in capturing the essence of healthcare simulation could emerge. Indeed, even if these tools have provided aid to healthcare, they still had been constructed with other systems in mind

(iii) *Mismatching expectations:* some healthcare professionals do not usually have a proper idea concerning modelling or regarding the type of modelling that have to be adopted for the problem they face. On the other hand, modellers usually fail to provide a better explanation of the benefits of the model, or end up blaming the system for lacking relevant data that can be used for modelling.

The author goes further, taking into account the existence of a mismatch between the wicked nature of healthcare problems and the tame approaches proposed by the modelling community. The concept of 'wicked problems' is usually referred to as a problematic situation for which a solution does not exist. Having this in mind, he then argues that the main cause of lack of implementation or lack of prevalence of modelling in healthcare is because most people attempt to solve wicked problem using tame approaches. There are not linear solutions to such problems. So, if we continue to use tame approaches to healthcare problems, we will not get the expected results.

Possible solutions are the following: healthcare modellers need to start identifying different values of their models, providing good and implementable recommendations, better insights about the nature of interactions within the system, departing from the the thinking that model's value is the final result. So, in order to establish appropriate taming practices, modellers need to change their views about what is meant by success out of their model. *Increasing their communication skills* is a first task they are required to. By doing so, they should be capable of keeping the stakeholders interested in the modelling. Another feature they have to develop is the *ability of managing stakeholders*: since most wicked problems have multiple stakeholders, modellers need to be able to manage different interests and differences that each stakeholder bring into the model. Finally, they should be able to *identify any intermediate outcomes* that are beneficial to solving the problems. So they need to develop the skills necessary to spot such results and convey them to stakeholders. This view supports what has already been stated previously. The need for ways to covert theories into practice, since has not been given enough attention so far (Kirchhof and Meseth, 2012).

Q	Reasons for non use / obstacles	Not a reason	Major reason
	Costs		
1	Initial Costs for software, hardware & training	6,7%	76,7%
2	Costs for model development	6,7%	63,3%
3	Costs for data collection	10,0%	66,7%
4	Costs for external consultancy	6,7%	76,7%
	Simulation Awareness		
5	Lack of knowledge where simulation can be applied	20,0%	26,7%
6	Lack of knowledge of benefits of simulation	6,7%	66,7%
7	Assumption that simulation is not suitable for problem	23,3%	46,7%
8	Heard of negative experiences from others	73,3%	16,7%
	Skills and Experience		
9	Lack of staff skills in simulation development	3,3%	80,0%
10	Lack of staff skills for statistical analysis	16,7%	60,0%
11	Negative experiences in past use of simulation	75,0%	16,7%
	Organizational		
12	Low confidence that results of simulation will lead to change	26,7%	30,0%
13	Fear of loss of decision sovereignty through simulation	70,0%	6,7%
14	Intuitive (non analytical) approach to decision making preferred	23,3%	40,0%
15	Lack of confidence in simulation	46,7%	26,7%
	Technical Limitations		
16	Other techniques provide sufficient information	23,3%	43,3%
17	Time to develop simulation is too slow for decision making	6,7%	60,0%
18	Lack of flexibility in reusing simulation models	26,7%	16,7%
19	Simulation model requires too much maintenance	6,7%	53,3%
20	Assumptions in modeling make results unrealistic	30,0%	30,0%
21	Lack of connectivity of simulation with databases & other IT	26,7%	40,0%
22	Components of a simulation model cannot be reused	26,7%	20,0%

Figure 1.6: Alleged obstacles to the usage of simulation.

1.3 Validation of population-based disease simulation models

As it has been previously pointed out, computer simulation models are used increasingly to support public health research and policy thanks, in particular, to the increasing power and the decreasing cost of computer, together with the growing availability of population health data; however, questions about their quality still persist(Fone *et al.* (2003)). In this section we will analyse in detail the methods for validation of population-based disease simulation models. To do so, we will follow the structure of the research work of Kopec *et al.* (2010), exploiting the results they have found out. Their work is aimed at reviewing the principles and methods of validation applicable to population-based disease simulation models. Population-based models have a wide range of application, from explanation to the prediction of trends in disease frequency. The authors propose a comprehensive framework for the validation of such models, addressing several gaps in the published validation guidelines. The presence of these gaps allows them to formulate specific recommendations for conducting model validation studies(such recommendations are first explained one by one and then summarised in table1 and table2).

Model validation framework

Quoting Kopec *et al.* (2010):

> In computer modeling, *validity* has been defined as "the degree to which a model or simulation is an accurate representation of the real world from the perspective of the intended uses of the model or simulation". However, validity defined in this way is often difficult to prove. It has been pointed out that model validation must be conducted continuously and should never be considered entirely complete. From a practical perspective, models gain *credibility* among potential users by virtue of being carefully

developed and thoroughly tested. In assessing model credibility, the key issue is the amount of evidence, both theoretical and empirical, in support of the model's intended use. Consequently, we consider model validation broadly as the process of gathering such evidence.

The evidence supporting a given use of a model can be obtained by examining:

(1) *the process of model development:* including underlying theories and assumptions, definitions of key concepts, the parameters, the implementation of the model in software.

a. Conceptual models

b. Parameters

c. Computer implementation

(2) *model performance:* a comprehensive analysis is needed, starting with subjective plausibility, going further with internal consistency, a parameters analysis and a comparison between models.

a. Plausibility

b. Internal consistency

c. Parameter sensitivity

d. Between-models comparison

e. External-data comparison

(3) *quality of decisions based on the model:* the consequences of decisions based on the model are fundamental.

Evidence from examining the process of model development

a. Conceptual model The development of a disease simulation model usually begins with a conceptual description of the relationships between

the condition of interest, its causes and its consequences. The proof of conceptual model validity tends to be qualitative. In particular, it relies on the opinion of experts in their fields. Moreover, the conceptual description of the model should be based on an accepted theory of the phenomena under study. In case a favoured theoretical basis is missing, there will be serious limitations that may compromise the model's credibility. Concerning the variables of the model, their definition from a conceptual and operational point of view is needed and it should be justified. The task for model developers is to determine those aspects of the causal web that are necessary and feasible to simulate. There should be a clear presentation of model assumptions, including a description of the expected strengths and limitations of the model for a range of potential applications. Some assumptions are most critical than others. This depends on the final purpose of the model, which can be prediction, explanation or decision analysis. In order to assess whether the model is sufficiently complete and the relationship between its own variables is correctly specified, evidence should come from both theory and empirical data. If some risk factors or interactions are omitted, explanation is necessary to explain why this omission is acceptable, if so, and does not invalidate the results.

b. Parameter validation The second necessary step able to establish model credibility is providing evidence of parameter validity. It should include considerations of possible bias as well as uncertainty in estimating a given parameter. The main sources of parameter values are:

1 ***Expert opinion:*** Expert opinion is a legitimate method of obtaining parameters, however the decision to use experts to estimate a parameter should be justified and the process of obtaining such parameters should be described. Moreover such experts should be independent from the team working at the research project. Their task consists in assessing the plausibility of these parameters.

2 ***Published estimates:*** Parameters may be obtained from published or unpublished sources (such as government reports). However, simply providing a reference to a published source is generally not sufficient. A further step is needed: the quality of the study should be ascertained and the selection of the source justified. Moreover, a comparison with alternative sources or data should be made and discrepancies explained or determined to be acceptable.

3 ***Analysis of existing data:*** Since descriptive parameters in population health models may be derived directly from an existing database, the validity of such database must be ascertained. In fact the accuracy of the data varies across different variables and types of parameters. Evidence of validity may come from previous validation studies or from new studies conducted specifically to assess validity against other sources, including other databases, the literature, as well as new analyses of the data.

4 ***Collection and analysis of new data:*** Parameters could be derived through an analysis of existing or newly collected data. In this case, evidence of validity should be equivalent to that required for a publication in a scientific peer-review journal. Requirements are: proper statistical methods of analysis, considerations of selection and measurement bias in the data.

5 ***Model calibration:*** procedure involving the estimation of unknown model parameters so that the aggregate output from the model is consistent with external target. It is worth noting that model calibration methodology is evolving. Furthermore there is no consensus at this time on how calibration should be carried out. Nevertheless, a detailed description of the calibration procedures would lead towards an increased model credibility.

c. Computer implementation Simulation models can be developed using different "languages":

i) a general purpose programming language such as C++ or Java or

ii) more specialized languages such as MATLAB or R or

iii) software toolkits specifically designed to facilitate the construction of simulation models by providing graphical interfaces to other programming languages, such as NETLOGO

Current model validation guidelines do not address the issue of selecting the most appropriate simulation platform. Advantages of using specialized simulation software for model development are evident: greater model transparency and less opportunity for mistakes, leading to an improvement of model credibility. Lastly, model developers should provide information on the programming language and software used, and the reason(s) driving their choice.

Evidence from examining model performance

If we were certain that the conceptual model, its parameters and computer implementation were all free of errors, there would be no need to examine model output as part of model validation. Unfortunately, no model is perfect. Indeed by definition, all models involve assumptions and simplifications. So there will be discrepancies between the model output and the real world. This is why examining the output is an integral part of model validation.

a. Plausibility Assessment of output plausibility usually is the first step in examining model performance. It consists in asking experts (of the fields the output refer to) if the model output appears reasonable and makes intuitive sense. This involves comparisons of model output with expectations based on general knowledge and understanding of the modelled phenomena.

b. Internal consistency Internal consistency is assessed by considering functional and logical relationships between different output variables. The relationships between trends in the health outcomes generated by the model should be consistent with theory.

c. Parameter sensitivity The definition of sensitivity analysis varies between authors. Usually it refers to the study of how the uncertainty in the output of a model can be apportioned to different sources of uncertainty in its inputs. In most published guidelines, sensitivity analysis is regarded as a method of assessing the impact of parameter uncertainty on model output. In particular, Kopec *et al.* (2010) use the term parameter sensitivity when discussing sensitivity analysis as a method of quantifying the impact of parameter uncertainty. The impact of parameter uncertainty is usually evaluated by running the model repeatedly while varying the values of the parameters. If there is a high amount of variability, it may cause the model's results to be questionable or seriously limit their practical utility. Particular attention should be devoted to stochastic error. They can be reduced by increasing the size of the simulated population, option that is becoming increasingly viable with continuing progress in computing resources.

d. Between-models comparison There is agreement in the literature that comparing the results of different models provides important evidence of validity and increases model credibility. Alternative model-structures and assumptions are increasingly considered a source of variation in model output that needs to be evaluated and quantified in a systematic way. Indeed between-model comparisons can provide important insights into the impact of different approaches to model building on simulation results. If we want to assess the sensitivity of model output to alternative model-structures, it may be useful to modify different aspects of the model one at a time. Nevertheless, it is worth noting (and quite obvious) that models should only be compared when they generate comparable outputs.

e. Comparisons with external data Regarding the comparison of model output with external data, published guidelines are not entirely consistent. We should distinguish between decision-analytical models, whose

sole purpose is to help with decision making, and explanatory or predictive models, that may be used to explain or project trends in health outcomes. Ideally, prospective data should be used for external validation. If prospective validation is not feasible, ex-post forecasting and back-casting based on historical data should be used to support predictive validity. Population-based models are examples of models expected to produce outcomes that are useful for explanatory and predictive purposes.

Evidence from the consequences of model-based decisions

Sculpher et al considered the question whether cost-effectiveness models can be regarded as scientific models. They argued that randomized trials (even ideal pragmatic trials) and observational studies do not provide a valid test of model predictions. The reason is that cost-effectiveness models are developed to improve decision-making, not to predict future events. However, they concluded that such models are scientific because they could be falsified, at least in principle, by comparing the consequences of decisions that are based on models and decisions that are not. The above argument essentially equates model validity with usefulness. Yet, how usefulness of a model should be defined and measured is not clear. While usefulness is related to the accuracy of projections generated by the model, the level of accuracy needed for the model to be useful will depend on the specific application. In principle, the impact of models on the quality of decisions could be evaluated directly in a variety of ways, including subjective and objective measures. More research on how these types of evaluations should be conducted is needed.

Conclusions

Exploiting the efforts and the results of Kopec *et al.* (2010) we reviewed the three types of evidence that can be used to support the use of population-based simulation models in the healthcare domain. Although a number of check-lists for model validation have been published, important gaps remain

in the validation literature. Their framework for model validation includes: the assessment of all aspects of model development and implementation, examination of model performance, and evaluation of decisions based on the model. Recommendations for model validation based on the proposed framework are presented in Figure1.7 and Figure1.8. These recommendations are intended to be used primarily as general guidelines rather than a quantitative assessment tool.

Evidence from examining model development process	
Conceptual model	
Underlying theories	The conceptual model should be based on an accepted theory of the phenomena under study. The lack of an adequate theoretical basis is a serious limitation that may compromise the model's credibility.
Definitions of variables	Definitions of the variables in the model should be justified. Evidence that the definitions are acceptable should be provided (e.g., a reference to published and/or generally accepted clinical criteria or results from validation studies).
Model content and structure	Evidence should be provided that the model is sufficiently complete and that the relationships between the variables in the model are correctly specified. If some variables or interactions are omitted, explanations should be given why this is acceptable and does not invalidate the results.
Parameters	
Parameters obtained from experts	The process of parameter elicitation should be described (number of experts, their areas or expertise, questions asked, how the responses were converted to a parameter). Plausibility of the parameter value(s) should be assessed by independent experts. Comparisons should be made with other sources (if available) and the differences explained.
Parameters obtained from the literature	Quality of the source should be ascertained. If available, a published meta-analysis should be used, but a single high-quality study may be an alternative. If information from several sources is combined, the methodology should be explained. Comparisons should be made with alternative sources and discrepancies explained. If alternative sources are not available, plausibility of the parameter values should be assessed by independent experts.
Parameters obtained from data analysis	Validity evidence regarding the data and methods of analysis should be equivalent to that required for a publication in a scientific peer-review journal. The results should be compared with estimates from other sources and (if not available) expert opinion. Evidence to support generalizability of the parameters to the population modeled should be provided.
Parameters obtained through calibration	Calibration methodology should be reported in detail (target data, search algorithm, goodness-of-fit metrics, acceptance criteria, and stopping rule). Plausibility of the parameters derived through calibration should be evaluated by independent experts and their values compared with external data (if available).
Computer implementation	
Selection of model type	A justification for the selected model type should be provided (stochastic vs deterministic, micro vs. macro-level simulation; discrete vs. continuous time models, interacting agents vs non-interactive models, etc). Whether or not the type of model is appropriate should be determined by independent experts.
Simulation software	Information should be provided on the simulation software and programming language. The choice of software/language should be justified.
Computer program	Independent experts should evaluate the key programming decisions and approaches used. The results of debugging tests should be documented and the equations underlying the model should be made open to scrutiny by external experts.

Figure 1.7: Recommendations for model validation based on the proposed framework: Evidence from examining model development process.

Evidence from examining model performance	
Output plausibility	Plausibility (face validity) should be evaluated by subject-matter experts for a wide range of input conditions and output variables, over varying time horizons.
Internal consistency	Internal consistency should be assessed by considering functional and logical relationships between different output variables. Internal consistency should be tested under a wide range of conditions, including extreme values of the input parameters.
Parameter sensitivity analysis	Model validation should include uncertainty and sensitivity analyses of key parameters. Screening methods should be used to select the most influential parameters for more extensive analysis. If feasible, probabilistic uncertainty/sensitivity analysis is recommended. If parameters are estimated through calibration, the model should be recalibrated as part of uncertainty/sensitivity analysis. In probabilistic models, the Monte Carlo error should be estimated.
Between-model comparisons	Comparing the results of different models provides important evidence of validity. Between-model comparisons should take into account the extent to which models are developed independently. If feasible, the impact of different elements of model structure, assumptions, and computer implementation on the results should be evaluated in a systematic fashion.
Comparisons with external data	Ideally, prospective data should be used for external validation. If prospective validation is not feasible, ex-post forecasting and backcasting based on historical data should be used to support predictive validity. Data used for validation should be different from data used in model development and calibration. Cross-validation and bootstrap methods can be considered as an alternative. Criteria for model acceptability should be specifed in advance.
Evidence from examining the consequences of model-based decisions	
Quality of decisions	Quality of decisions based on the model should be evaluated and compared with those based on alternative approaches to decision making, using both subjective and objective criteria.
Model usefulness	Uptake of a given model by policy makers should be monitored to assess model usefulness.

Figure 1.8: Recommendations for model validation based on the proposed framework: Evidence from examining model performance and Evidence from examining the consequences of model-based decisions

1.4 A particular case: Agent Based Social Simulation

1.4.1 Introduction

In this subsection we will analyse in detail the concept of that particular type of social simulation which is usually known as "agent-based social simulation". We will organise our speech following the structure of two different sources, respectively: Gilbert (2004), in which the author highlights the potential and the value of agent-based models to the social sciences and Li *et al.* (2008), containing the authors' aim at reviewing research and applications of agent-based social simulation and modelling in recent years from a social computing perspective. The first step we have to take is to provide a definition of "Social Simulation". It can be intended as the modelling or simulation of social phenomena or objects (such as society, organizations, markets, human beings and so on) which is normally performed by a computer. One major computational modelling approach for social simulation is agent-based social simulation (ABSS). As we will see in the following subsection, societies, and in particular human societies, are often complex adaptive systems characterized by a lot of non-linear interactions between their members or between people. As a consequence, traditional computational and mathematical models can hardly represent these kinds of complex systems since complex social processes can not be represented in equation terms (at this point agent-based social simulation turns out to be effective and useful). In addition, agents based-models create agents that can have a one-to-one correspondence with the individuals (or organizations, or other agents) that exists in the real social world being modelled, while the interactions between the agents can likewise correspond to the interactions between the real world individuals. We can then conclude that this particular type of social simulation represents a methodological approach that could contribute to two different aspects:

1) a rigorous testing, refinement, and extension of already existing theories that have proved to be difficult to formulate and evaluate using standard statistical and mathematical tools

2) a much more deeper understanding of fundamental causal mechanisms in multi-agent systems whose study is currently separated by artificial disciplinary boundaries

Even if the influence played by computer simulation on most areas of science is quite old, it took until the 1990s for it to have a significant impact in the social sciences. Quoting Gilbert (2004):

> The breakthrough came when it was realised that computer programs offer the possibility of creating "artificial" societies in which individuals and collective actors such as organisations could be directly represented and the effect of their interactions observed. This provided for the first time the possibility of using experimental methods with social phenomena, or at least with their computer representations; of directly studying the emergence of social institutions from individual interaction; and of using computer code as a way of formalising dynamic social theories.

A different but similar idea is supported by Li *et al.* (2008), in which it is possible to read what follows:

> Although agent-based social simulation has been proposed since 1970s, it is getting more popular in recent decades with the development of artificial intelligence and computational theory. A lot of new approaches to ABSS have been proposed by researchers and the application domain is expanding rapidly. It thus becomes rather confusing for the ABSS researchers to adopt proper agent-based simulation approaches for their specific modeling and simulation problem. However, few researchers have

tried to summarize the research and development of ABSS in recent years.

When facing the ABSS issue, the starting hypothesis is that the usage of computer programs simulating aspects of social behaviour can contribute to te understanding of social processes. The majority of social sciences either develops or uses some kind of theory or model. Such theories could be stated in:

a) textual form (which means that the form the theory is expressed by is based on or conformed to a text)

b) analytical form (this is the case in which a theory is represented, for example, by an equation)

c) computer-programs language

When such theories are modelled by computer programs, a straightforward consequence is the possibility of social processes to be simulated in the computer. Such practice implies advantages and drawbacks too. The main benefits are:

a) the possibility of simulating experiments that would be impossible or unethical to perform on human population

b) sometimes a simulation is able to give insights into the emergence of macro level phenomena from micro level actions (for example, taking the case of a program simulating interacting individuals, if we shift the focus from individuals to the societal scale we could perhaps observe the emergence of clear patterns of influence)

c) it is necessary to think through one's basic assumptions very clearly in order to create a useful simulation model. Every relationship to be modelled has to be specified exactly. Moreover, every parameter has to be given a value, for otherwise it will be impossible to run the

simulation. Such requirements open the way to inspection by other researchers.

These advantages highlight the main features characterizing agent-based social simulation: clarity and precision. However, drawbacks do still exist. The hardest one to overcome is the need for estimation of many parameters, which happens when it is impossible to obtain adequate data.

1.4.2 A glance at Complexity

As Gilbert (2004) notices, the physical world is full of systems that are linear or approximatively so, meaning that the properties of each whole system are a fairly aggregation of the parts (let us just think at the properties of the galaxy, a massive system composed of millions of stars, whose properties can be precisely predicted by means of the basic equations of motion). Societies, in particular human societies, are instead different. They usually show rather unpredictable features, so it turns out to be risky to make exact predictions concerning their alleged future development. So human societies, as well as institutions and organizations, could be defined as "complex systems", where the term "complex" is used to stress the fact that the behaviour of the system as a whole cannot be understood and then determined by partitioning it and understanding the behaviour of each component part separately. The reason why human societies are "complex" is due to the fact that they are the result of many non-linear interactions between their components, i.e between people. Such interactions are used to "emerge" as a consequence of knowledge or materials transmission among individuals. But, since such transmissions are unpredictable, the behaviour of societies cannot be analysed as a whole by studying the individuals within it and, so, it is unpredictable too. Gilbert (2004) defines as "emergent" a phenomenon which can only be described and characterized using measurements and terms which are impossible to apply to the component units (we can take as an example the creed of a church. Such term cannot be applied to individuals).

We can now conclude this short digression by highlighting and summarizing the features characterizing human societies all around the world:

a) Complexity

b) People are able to recognize "emergent" features (individuals are often used to cluster in segregated neighbourhood, which are then named and are able to affect the behaviour of those living there)

c) Societies are the result of dynamical processes, they emerge from the constant change characterizing the individuals, which are constantly in motion.

d) Differently from the vast majority of physical systems, composed of similar or identical units, all the components of human societies are different from the others. They show different capabilities, desires, knowledge and needs, whom twist make each human society unique.

1.4.3 The starting point: data

As we have previously stated in the introduction of this section, having available an adequate dataset is fundamental in order to understand complex and ever changing societies. However, acquiring such information is very difficult. The main distinction within such issue is between qualitative and quantitative data. We still will follow Gilbert (2004) to characterize both of them:

a) qualitative Gathering such kind of data represents the traditional methods of analysis in sociology. They are obtained by means of interviews, or from documents and records. Even if they are able to illustrate very effectively the emergence of institutions from individual action, most of the analysis based on such data remain somewhat impressionistic.

b) quantitative Apparently, more precision is provided by studies based on quantitative data, but if we bear in mind the idea that societies are complex and their features are emergent, the limitations we are

undertaking if we use them are huge. Indeed, as Gilbert (2004) points out, much quantitative sociology is based on data that turns out to be inappropriate for understanding social interactions.

The main problem connected with quantitative data is that they are obtained by means of measurements made at a particular moment in time. The straightforward consequence is that such pick makes the way in which individuals change almost invisible to the analyst. The best type of data would be that able to track individuals through their life course. Even if they are very expensive to obtain and still limited in scope, such kind of data are starting to become available with large-scale panel studies. So, one of the most overwhelming problem connected with the inductive methodology of collecting data and then building models describing and summarising those data, is indeed that of data collecting. If we shift the focus towards a more deductive perspective, in which a model is created, calibrated from whatever data is available and then used to derive testable propositions, relationships or theories, we have entered the field of simulation models. The obvious advantage is that such approach places much lower demands on the data and is able to truly reflect the complex nature of societies.

1.4.4 Multi-agent models

In this paragraph we will focus on the so called "multi-agents models", i.e that particular types of models consisting of a number of software objects, the "agents", interacting within a virtual environment, depending on the software used each time. Even if it would be possible to misunderstand the relationship between Agent Based Models and Multi Agent ones, considering the latters as an "extended version" of the formers, there is a strong difference among them. In particular, an agent-based model is a class of computational models used for simulating the actions and interactions of autonomous agents (that can be both individual or collective entities such as organizations, groups etc) with a view to assessing their effects on the system as a

whole. Agent-based modelling is related to, but distinct from, the concept of multi-agent systems or multi-agent simulation, in particular with respect to the goals pursued. An ABM is indeed aimed at searching for explanatory insight into the collective behaviour of agents (which do not necessarily need to be "intelligent" differently from the multi-agent case, as we will highlight in few lines) obeying simple rules, typically in natural systems, rather than in designing agents or solving specific practical or engineering problems. By contrast, a multi-agent model is a computerized system composed of multiple interacting intelligent agents within an environment. Multi-agent systems are often used to solve problems that are difficult or impossible for an individual agent to solve. Usually the terminology of ABM is far more used in the sciences, while that of MAS in engineering and technology.

A similar idea supporting the differences among such fields is stated in Shoham and Leyton-Brown (2008). In particular the authors, quoting agent-based models in a multi-agent context, affirm that:

> There are other large-population models that provide a more fine-grained model of the process, with many parameters that can impact the dynamics. We call such models, which explicitly model the individual agents, agent-based simulation models.

Coming back to multi agents models, the features characterizing such agents are:

a. a particular degree of autonomy

b. capability of acting and reacting to the environment they live in and to other agents

c. target-aimed: agents have goals that they aim to satisfy

d. one-to-one correspondence: such agents could be characterised by an univocal correspondence with the individuals, enterprises, or other real agents they stand for within the model, and the interactions among them correspond to the interactions between the real world actors too.

Agents are credited with these features by means of object-oriented programming language (such as Java, Python, C++ and many others) or special-purpose simulation library. They are constructed using collections of condition-action rules able to "sense" and react to the situation they are in in order to achieve the targets they are given. Then, once such model has been accomplished, it should be run in order to generate output, which can be validated against observable data. Using a simulation aimed at generating patterns that one expect to find, if the model is correctly specified, and then comparing such results with targeted observations of the social world is definitely easier (and less costly too) than trying to acquire detailed data concerning social processes directly. However, there are two disadvantages that have to be taken into consideration:

a. the majority of the models and the theories on which such simulations are based are stochastic and, as a consequence, they are partly based on random chance. So, for example, agents are casually located into the simulation's "world". If agents' conditions-action depend on the proximity criterion, that original casual location will influence the entire simulation. As a consequence, re-running the simulation with a different starting (random) configuration will yield different results. So, the way we have to proceed is to run several random-based simulations and observe if some constant patterns emerge.

b. many different models may yield the same emergent patterns. So, the emergence of a correspondence between what emerges from the model and what succeeds in the social world (for example the emergence of a cluster) is only a necessary, not sufficient, condition which does not allow us to conclude that the model is correct.

A way we can proceed is the following: we have to gradually increase our confidence in the model at issue by testing it against observation in more and more ways.

Common examples

The range of social phenomena that can be studied is very wide, and by consequence the number of multi-agent social simulation models is huge. A practical way to review them is to follow the dimensions along which models can be arranged proposed by Gilbert (2004). In particular, the author depicts five different categories of social simulations, examined in turn in what follows:

a. Abstract vs Descriptive: A first differentiating factor among models is based on the degree to which they attempt to incorporate the detail of particular targets. So we can depict as "abstract" all that simulations based on essential features: agents' conditions-action ruling the model at issue each time. By contrast, a "descriptive" model is one characterized by higher degree of details: for example, models covering a particular period of time or those attaching certain details to the agents. In any case, both the categories could be intended to aid in the understanding of actual human societies.

b. Artificial vs Realistic: "Artificial" models are not intended as simulations of human societies while, by contrast, "realistic" ones are firmly focussed on modelling real social problems

c. Positive vs Normative: "Normative" models have usually clear application to policy domain; so, they are designed to make recommendations about what policy should be followed. However, the vast majority of social agent-based simulations are "positive", meaning that they tend to be descriptive about the social phenomena being investigated, providing aid to the understanding rather than providing advice.

d. Spatial vs Network: Some models (those definable as "spatial") contain agents moving and acting in a spatial environment, often a two dimensional grid of rectangular cells, sometimes instead a map of some specific landscape (in this case, the map is provided by the GIS, geo-

graphical information system). There are instead models in which the geography is not relevant. What counts more is the relationships between agents, which is often represented as a network of links between nodes (the agents indeed).

e. Complex vs Simple agents: Quoting Gilbert (2004):

> The simplest agents are ones that use a production system architecture (Gilbert and Troitzsch 2005), meaning that the agent has a set of condition-action rules. An example of such a rule could be ?IF the energy level is low, THEN move one step towards the nearest food source?. The agent matches the condition part of the rule against its present situation and carries out the corresponding action. These rules might be explicitly coded as declarative statements, as in this example, or they may be implicit in a procedural algorithm. However, it is difficult to model cognitively realistic agents using such a simple mechanism and so model-builder have sometimes adopted highly sophisticated cognitive model systems to drive their agents.

So, if the purpose of the model is to predict the behaviour of the organization as a whole, then simpler models of agents are a sufficiently satisfying device. If, instead, the aim of the simulation consists in the prediction at the individual or small group level, then more cognitively accurate models are needed.

1.4.5 ABSS, a computer scientist's perspective

In this paragraph we will analyse in detail that particular area of computer simulation previously described, the Agent Based Social Simulation, by means of a specific analysis carried out from a computer scientist's standpoint. In particular, we will refer to the work of Davidsson (2002). Following

the author's procedure, we begin with the definition of ABSS in terms of its position with respect to the three research areas that it si related to, respectively i) agent-based computing, ii) social sciences, and iii) computer simulation. Figure1.9 shows graphically the interrelationship among these three areas.

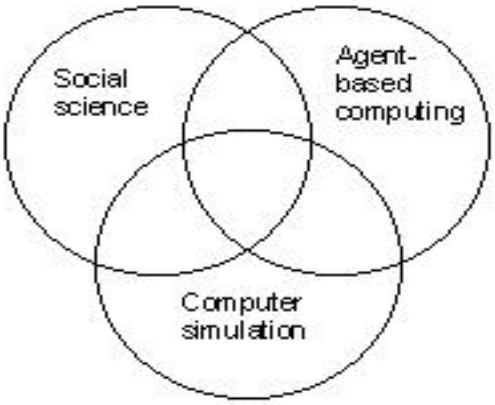


Figure 1.9: The three constituting areas of ABSS and their interrelationship

i) Agent-based computing: research area mainly within computer science, including programming, design and agent-based modelling.

ii) Social sciences: it is a set including different sciences studying the interaction among social entities, such as social psychology, management and policy. They provide theoretical foundation for the studies of social simulation and modelling

iii) Computer simulation: it consists in the study of different techniques for simulating phenomena on a computer, such as discrete event or equation-based simulation. As we have previously underlined, the phenomenon simulated could be either existing or non-existing at the time of simulation. In both cases, the aim of computer simulation is to gain a better understanding of the phenomenon at issue. Moreover, even

if computer simulation can be considered as a merely computer science sub-area, nowadays there are several applications in several fields, from physics to engineering, from biology to social sciences with new applications arising day by day.

In order to define properly ABSS we have to introduce Figure1.10, and consider the area where all the three fields intersect.

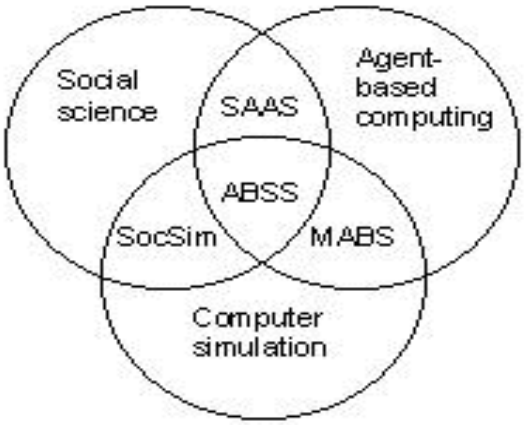


Figure 1.10: Definition of ABSS as the area shared by Social science, Agent-based computing and Computer simulation' fields. It is worth-note the emergence of three other areas, SAAS, MABS, SocSim, all analyzed in detail in the text

However, even if our focus is directed towards the ABSS area only, a brief digression on the three other emerging areas is worth note.

i) SAAS Standing for Social Aspects of Agents Systems, arises from the intersection between the social sciences and agent-based computing. Such area includes the study of norms, institutions, organizations, competition and so on.

ii) MABS Multi Agent Based Simulation, concerns the use of agent technology for simulating any phenomena on a computer.

iii) SocSim which stands for Social Simulation, is the result of the intersection between the social sciences and computer simulation. It consists in that particular type of computer simulation on social phenomena characterized by the usage of simple models of the simulated social entities, while, by contrast, the software agents used in MABS typically employs more sophisticated communication languages and interaction mechanisms. Moreover, Social Simulation studies provide an ideal opportunity for filling the gap between empirical research and theoretical work. In particular, social simulation provides not only a methodology for testing hypotheses, but an observatory of social process too.

So, if we adopt the view just presented, we can define ABSS as that research area investigating the use of agent technology for simulating social phenomena on a computer. A natural extension would be including all the intersection areas such as SAAS, MABS, and SocSim but, at the same time, we have to point out such areas that clearly do not belong to ABSS, respectively:

i) social science not including an element of either agent technology or computer simulation

ii) agent technology not including an element of either social science or computer simulation and, finally,

iii) computer simulation not including an element of either agent technology or social science.

So, according to the characterization we have just provided, the main role of ABSS consists in providing models and tools for agent-based simulation of social phenomena and, then, to apply these in different areas. A particular feature characterizing ABSS is that it shows a unique potential for providing cross-fertilization between the participating fields of research. We will now give a look to these possibilities. To a better understanding of these contributions we will introduce Figure1.11.

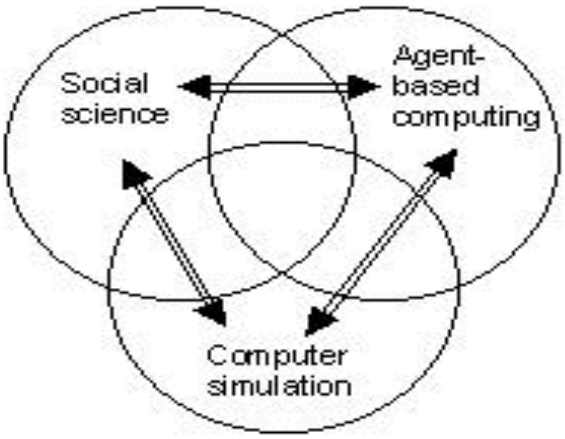


Figure 1.11: Contributions among research areas mediated by ABSS

We now analyse each of these relationships (highlighted by the double arrow lines) separately.

Connections between Social Science and Agent-Based Computing. A first source os synergy between these two areas is the provision of techniques and methodologies to built and support scale software systems aimed at studying social sciences offered by the agent-based computing research. Another synergy-source concerns the development and use of formalisms to specify software agents intended to inform a new social theory too. Indeed, the formal structure of agent based computing provides a clearly supportive environment for the application of logical formalism, which is a very useful in the specifications of agents for purposes of engineering multi-agents systems.

Connections between Social Science and Computer Simulation. Links between these two areas are largely methodological and it seems that social sciences obtains the greater benefit. As we have previously said, it took until the 1990s for computer simulation to have a significant impact in the social sciences. So, when social scientists have begun to convert social theories to computer programs, a new frontier has been opened: the possibility of analysing phenomena that it would be dif-

ficult to observe or, even, directly not possible. Computer simulation represents a useful method to clarify sociological theories too.

Connections between Agent-Based computing and Computer Simulation. We now quote Davidsson (2002) since the author stress such connection. In particular we can read that:

> Possible benefits to agent based computing from computer simulation includes methods for evaluation of multi agent systems or for training future users of the system. Many new technical systems are distributed and involve complex interaction between humans and machines. The properties of ABSS makes it especially suitable for simulating this kind of systems. The idea is to model the behaviour of the human users in terms of software agents.

So, in few words, in this paragraph we have considered ABSS as that particular field (lying at the intersection of three research areas, agent-based computing, the social sciences, and computer simulation) providing a unique potential (not totally explored yet) for cross-fertilization between these areas.

1.5 COMPLEXITY

1.5.1 Introduction

This section is devoted to the understanding of the concept of “Complexity". In particular in the first subsection we begin by introducing a definition of Complexity as it has been proposed by Neil Fraser Johnson, a physics Professor notable for his work in complexity theory and complex systems. We then consider the areas of applicability of such concept, showing that such issue could be found in several aspects of our everyday life. We will conclude this subsection by enumerating and explaining the features Complex Systems should have according to a vast part of Complexity researchers. The second subsection is devoted to the distinction between the terms “Complex” and “Complicated”, referring in particular to the work of Glouberman and Zimmerman (2002) distinguishing between three different types of problems: simple, complicated, and complex. We then conclude this section by proposing a set (not exhaustive) of different meanings of Complexity. Such differences are due to the field each time the term “Complexity” is referred to.

1.5.2 What is Complexity?

In this subsection we will follow the ideas of the previously mentioned Neil Fraser Johnson. In particular we will focus our attention on his reasoning as it is presented in his own book “A Simple Guide to the Science of All Sciences". As Johnson (2009) points out, defining Complexity is not an easy task. In particular he states that:

> Well, unfortunately, Complexity is not easy to define. Worse still, it can mean different things to different people. Even among scientists, there is no unique definition of Complexity. Instead, the scientific notion of Complexity - and hence of a Complex System - has traditionally been conveyed using particular examples

of real-world systems which scientists believe to be complex.

Areas of applicability: from technology, to health, to everyday life

The way the author follows to sum up Complexity is the usage of the phrase (which is used as title of the first Chapter of the book at issue) "Two's company, three is a crowd". The meaning is the following: Complexity Science can be seen as the study of the phenomena which emerge from a collection of interacting objects (and the crowd previously mentioned represents a perfect example of such an emergent phenomenon, since it emerges from a collection of interacting people). Examples of crowd are present in any aspect of our everyday life, ranging from the collections of commuters to that of financial market traders, from human cells to the insurgents. As a consequence, the associated crowd-like phenomena are respectively traffic jams, market crashes, cancer tumors and guerilla wars. To some extent, if we consider the collective actions of humans and, in particular, the environmental changes caused by human activity, we can call to mind the controversial emergent phenomenon of "global warming".

The fundamental idea most real-world examples of Complexity are based on is, still according to Johnson (2009), the situation in which a collection of objects are competing for some kind of limited resource, such as space or land, food, wealth or energy. If we refer to the examples previously mentioned, and in particular to that concerning the collections of commuters, a related crowd phenomenon occurs among such individuals who are competing for space on a particular road at the same time. The consequence of their action is the emergence of the so called "traffic jam". In the real life there are plenty of other examples, such as power blackouts, in which a great number of subscribers simultaneously decide to access and, as a consequence, exhaust the available resources of the power network at issue.

According to the author, the Holy Grail of Complexity Science is the understanding, prediction and control of such emergent phenomena, in particular for those having catastrophic consequences over the human race, such

as epidemics, human conflicts and environmental changes. A particular feature characterizing such phenomena is that they can arise even in the absence of any central controller or coordinator. Indeed, in such cases the collection of objects is able to self-organize itself in such a way the phenomenon appears all by itself. Even if we have so far mentioned examples referring to crowd "human-based", emergent phenomena belong to the non-human world too. The animal, insect and fish kingdoms are filled by examples of self-organization. Let us just think at bird flocks, schools of fish, the ant-trails or even the wasp swarms (see Figure1.12).

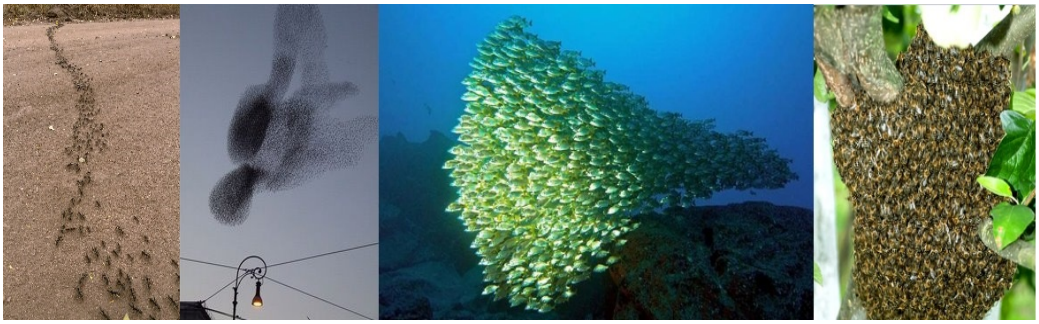


Figure 1.12: Examples of "non-human based" phenomena of self-organization. From left to right: an ant-trail, a bird flock, a school fish, a wasp swarm.

And consequently, since all these effects represent emergent phenomena, so many different disciplines are getting far more interested in Complexity (Johnson, 2009).

A prime example of Complexity in action is the area of human health and medicine. Our immune system is based on a collection of defence mechanisms for dealing with invading viruses. However, as in the case of other systems (i.e traffic or power network), it can go wrong all by itself, as in the case in which its collective response ends up attacking healthy tissues. As it clearly emerges, it turns out to be really important understanding, predicting, managing and even (and hopefully) controlling a Complex System. If we take into consideration a situation of cancer tumor as an example of a crowd effect gone wrong, any treatment involving damaging the tumor may lead to the

survival of the patient. This to remark one more time the importance of managing and to some extent controlling a Complex System.

As a consequence of the features the author attaches to Complexity Science, the interest in Complexity is not confined to natural objects, such as people animal or cells, only. For example, the ability of a collection of objects to produce emergent phenomena without the need for some central controller has attracted the attention of researchers at NASA, mainly focused on the emergent phenomena in collections of machines (robots, satellites, micro-spacecraft and so on). The reason for such interest is clear. If we consider the case of a collection of relatively simple robots which have been sent on a planet to explore its surface, if one robot within the collection were to malfunction, there would be still plenty more available while if it is the case of a malfunction in the single large machine used for the exploration, it could lead to the termination of a costly mission (an analogue reasoning holds for the case of simple satellites versus a large sophisticated one and for a collection of micro-spacecraft versus a much larger one). A further step can be made. In particular Johnson (2009) highlights that the real interest for NASA in Complex Systems belongs to the fact that it is not necessary for the machines to have local coordination in order for them to make a good job. Indeed, suitably chosen collection of such objects can work better as a group if they are not being coordinated by some single controller but if, instead, they compete for some limited resources (the same happen in the case of drivers in traffic: it is their competition for space on a road that imply the emergence of arrangements of cars that are spread out in some reasonably regular pattern).

The last example we take into consideration refers to the effects of our own collective actions on our environment and weather. In particular, the global competition for increasingly scarce natural resources is leading to increased levels of phenomena such as pollution and deforestation, which may in turn affect our climate. The weather is a complicated combination of ongoing interactions between the atmosphere and oceans, connected as they are by

currents of water, winds and air moisture. The extreme phenomena resulting from this collective behaviour at issue are floods, hurricanes and droughts. Even if scientists are now aware of the inner rules describing the air or water molecules, it is extremely complicated to build up a picture of what billions of such molecules can do once mixed together. If, moreover, we add the collective actions of human beings, we come up with the global warming, a perfect example of Complexity in action.

To briefly sum up what we have considered so far, in all of the previously presented examples the emerging precise nature of the crowd-like phenomena will depend on how the individual objects interact and how they are interconnected. What is certain is that it is extremely difficult, if not impossible, to deduce the nature of such emergent phenomena based only on the properties of an individual object. So, facing such difficulty in predicting what crowd effects will arise, under what conditions and when, we should start thinking at the Complexity science as the "Science of All sciences" (Johnson, 2009). In other words, whether an individual is interested in human health, biology, technology or other sciences, or if such individual is just aimed at avoiding a blackout, the answer is Complexity.

The constituting components of Complexity

As we have already pointed out at the beginning of this paragraph, it does not exist a rigorous definition of Complexity. However, as in the case of the concept of "happiness" for which it does not exist a unique definition but we all know what its characteristics are, we will characterize Complexity in a similar way, by describing the features a Complex System should have and then it should show. In particular, we refer to Johnson (2009) who claims that the majority of Complexity researchers would agree that any candidate Complex System should have most, if not all, of the following features:

1) **The system contains a collection of many interacting objects or "agents".** Referring to the previous examples, if we consider the case of traffic these objects are the drivers, while in the case of markets

these are traders or investors. Such objects are typically called "agents" by the scientific community. The reasons why such agents interact are numerous, respectively:

i) physically closeness among them

ii) membership in a particular group

iii) they may be linked by some public information that they share (i.e investors who are watching the same price chart for a given stock)

iv) they may be linked by some private information (take the case of two investors who happen to be friends sharing such information over the phone)

The straight consequence of the emergence of interactions among agents is the possibility of a network to arise. This is why networks have become an integral part of Complexity Science, to the extent that for many scientists the study of Complexity is synonymous with the study of agents and networks together (Johnson, 2009).

2) These objects' behaviour is affected by memory or "feedback". Two cases:

i) What happened in the past can affect something in the present (path-dependence effect) or

ii) what is going on in a particular location can affect what is happening at another one (knock-on effect).

A brief example can help us in a better understanding. Suppose an individual happened to have taken the same route work for the past few morning and it was always overcrowded, then he may choose to flip to a different route if it does exist. This is a case in which an agent is using a past information to influence its current decision. The same holds in different fields (i.e stock market investment decisions).

3) **The objects can adapt their strategies according to their history.** To put it slightly differently, an agent can adapt its behaviour by itself, in the hope of improving its performance or the outcome.

4) **The system is typically "open".** The openness feature leads to the possibility for the system to be influenced by its environment. A practical example is the case of a financial market affected by the consequences of an outside news concerning the earnings of a particular company. A closed system is, instead, one characterized by the absence of contact with the outside world. As we can guess, such closed systems are rare, and the only truly closed system is the Universe as a whole.

5) **The system appears to be "alive".** The system evolves, and the way it does so is highly non-trivial, often complicated. Such evolution is driven by a set of agents who interact and adapt in response to the feedback they share. It is common, for example, for financial analysts to talk about the market as a living, breathing object, to the extent that they usually assign to it features of pessimism or confidence.

6) **The system exhibits emergent phenomena which are generally surprising, and may be extreme.** As the author points out, in scientific terminology the system is far from equilibrium. In other words, anything can happen, and generally if we wait long enough it will. As an example consider the usual cases of markets and traffic systems: eventually the former will show some kind of crash while the latter some kind of jam. Such phenomena generally arise unexpectedly, since it is not possible a prediction based on a knowledge of the properties of the individual objects.

7) **The emergent phenomena typically arise in the absence of any sort of central controller.**

   There is no one ruling a Complex System which, instead, can evolve in a complicated way all by itself. This is why such systems are often

considered as being more than the sum of their parts.

8) **The system shows a complicated mix of ordered and disordered behaviour.**

Generally all Complex systems seem to be able to shift from order to disorder and viceversa. For example, traffic jams at a particular point in time and at a particular place on a road arise, and then disappear.

Conclusions

In this paragraph we have seen that Complexity focuses on what new phenomena can emerge from a collection of relatively simple components. In other words, it looks at the complicated side of things emerging from the interaction of a collection of objects which themselves may be rather simple. Moreover, even if we do not comprehend totally the constituent objects, Complexity allows us to understand what a collection of them might do. The essence of such important science is that a simple interaction among simple bits may lead to a rich variety of realistic outcomes. This is why Complexity is likely to be important for many areas of science, across many other disciplines and in everyday life too. Indeed we have understood that it plays an important role in making connections between previously unrelated phenomena belonging to distinct scientific disciplines. This is why the author considers Complexity as the " Science of all Sciences"

1.5.3 Complicated or Complex?

Once we have understood that Complex Systems are everywhere and that Complexity turns out to be a fundamental science, we have to proceed in a far more detailed way. In particular we have to distinguish between what is complicated and what instead is complex. Understanding such difference is nowadays becoming important for several aspects of management and policy.

Broadly speaking, a complicated issue is such that one is able to define the problem and then strategically develop proper actions aimed at obtaining the desired results. By contrast, in the case of a complex adaptive system it is difficult to predict both the cause and the effects (Allen). This topic is satisfactorily developed in Glouberman and Zimmerman (2002), so we will comply with the contents presented in there.

The starting point of such paper is the set of troubles affecting the Canadian health care system. Even if there is agreement on the presence of such problems, the same harmony does not hold concerning its nature. Indeed many different sources are identified, depending each time on the survey orientation and on the type of questions being asked by pollsters and health related surveys. The authors point out that other countries in the English world have had similar paths during the previous decade (remark: the paper has been published in 2002), with several changes hitting such systems (e.h the British National Health Service, NHS, changed its orientation from "managed competition" to "collaboration" the day the Labour succeeded to Tory). It is then argued that the majority of the approaches aimed at changing such systems are based on a rational planning. Moreover, the authors state that health care and, more generally, the systems within which it is delivered could be better understood as complex adaptive systems. But since the assumptions underlying such rational planning are not consistent with complex adaptive systems, then policies and strategies based on it can have significant unintended consequences when applied to such complex systems. So it turns out to be important the distinction made by Glouberman and Zimmerman (2002) between three different types of problems: simple, complicated, and complex.

1) simple problems: a useful example of a simple problem is cooking by following a recipe, which is essential. Problems of this type may include some issues of technique and terminology but, once these are mastered, following the recipe is sufficient to almost ensure a successful result (which is almost standardized). Moreover they do not need any

particular expertise.

2) complicated problems: differently from the previous case, formulae and recipes are fundamental in order to resolve complicated problems but, often, they are not sufficient. They contain subsets of simple problems even if they are not merely reducible to them. Their complicated nature is due to the scale of a problem, but to issues of coordination and specialized expertise too. With reference to this last element, complicated problems involve high levels of expertise in a variety of fields as a necessary condition for success. A clear example is that of sending a rocket to the moon (high level of expertise is required, formulae are critical, sending a rocket increases assurance that the next will succeed).

3) complex problems: within this set of problems formulae have a much more limited application. Expertise can contribute to solve such issues in a valuable ways, however it does not provide any necessary or sufficient condition to ensure success. The outcome is always surrounded by a veil of uncertainty. Moreover, complex problems include both complicated and simple ones, even if they are not reducible to either since they have specific requirements, such as an understanding of unique local conditions and a particular capacity to adapt as the environment change. A neat example of complex problems is raising a child (raising a child provides experience but it does not guarantee success with the next, every child is unique, uncertainty of outcome still persist, formulae have limited applications). Although the veil of uncertainty linked to the complex case, all these kind of problems can be approached with a certain level of optimism (the complexity of raising a child will not imply people to give up from this issue).

Figure1.13, which consists in a table borrowed from Glouberman and Zimmerman (2002), summarizes all the characteristics of each type of problems we have just presented, drawing attention to the differences among them.

Table 1
Simple, Complicated and Complex Problems

Following a Recipe	Sending a Rocket to the Moon	Raising a Child
The recipe is essential	Formulae are critical and necessary	Formulae have a limited application
Recipes are tested to assure easy replication	Sending one rocket increases assurance that the next will be OK	Raising one child provides experience but no assurance of success with the next
No particular expertise is required. But cooking expertise increases success rate	High levels of expertise in a variety of fields are necessary for success	Expertise can contribute but is neither necessary nor sufficient to assure success
Recipes produce standardized products	Rockets are similar in critical ways	Every child is unique and must be understood as an individual
The best recipes give good results every time	There is a high degree of certainty of outcome	Uncertainty of outcome remains
Optimistic approach to problem possible	Optimistic approach to problem possible	Optimistic approach to problem possible

Figure 1.13: Differences among Simple, Complicated, and Complex problems. Each column is devoted to one of this three issues. Starting with an example, all of these three columns sum up the main features such types of problems show.

Once introduced such distinction, the authors explain the reason inspiring it. Given the previously mentioned troubles affecting Canadian health care system, Glouberman and Zimmerman (2002) argument that many health care experts implicitly depict complex problems as they were complicated ones, and the straight consequence is the employment of solutions embedded within rational planning approaches. So, since they tend to lack of attention many aspects of complexity, there are often inappropriate ongoing solutions. This to remark that individuals should not be trapped into a narrow way of defining and, consequently, responding to current problems as if they were solely complicated. Success in answering to complex problems could represent the beginning of a revitalizing process in different fields, first of all in health care. In other words, Complex Adaptive Systems theory could provide

fresh accounts of how our systems deteriorated and lead solutions towards a different path.

1.5.4 Different Meanings of Complexity

Generally Complexity is used to describe something composed of different parts interacting with each others in various way. However, according to the specific field we are referring to, "Complexity" has a precise meaning. We now consider briefly such different meanings of Complexity.

1) Computational complexity theory: it studies the amount of resources needed in order to execute the algorithms (which is a step by step procedure used for calculations). The most popular types of computational complexity are:

- **i) time complexity of a problem:** which consists in the number of steps necessary to solve an instance of the problem as a function of the size of the input (which tends to be measured in bits).
- **ii) space complexity of a probelm:** which is equal to the volume of the memory used by the algorithm in order to solve an instance of the problem (as the previous case, the instance is solved as a function of the size of the input).

2) Information processing: in this case, complexity is considered as a measure of the total number of properties which have been transmitted by an object and consequently detected by an observer.

3) Algorithmic information theory: we have to highlight the "Kolmogorov complexity" (even known as descriptive complexity or algorithmic complexity) of a string. It is the length of the shortest binary program generating that string.

4) Software engineering: programming complexity consists in a measure of the interactions of the various elements of the software.

5) Physical systems: in this field, complexity represents a measure of the probability of the state vector of the system.

Once we have understood that there exist different meanings of Complexity (and those we have presented are not all), we now consider far more in detail such meanings of Complexity referring to the computational and algorithmic fields. In particular, we refer to Gell-Mann (1995). According to what we have just stated, the author begins his paper by claiming that:

> What is complexity? A great many quantities have been proposed as measures of something like complexity. In fact, a variety of different measures would be required to capture all our intuitive ideas about what is meant by complexity and by its opposite, simplicity. Some of the quantities, like computational complexity, are time (or space) measures. They are concerned with how long it would take (or how much capacity would be needed), at a minimum, for a standard universal computer to perform a particular task. Computational complexity itself is related to the least time (or number of steps) needed to carry out a certain computation. Other suggested quantities are information measures, referring, roughly speaking, to the length of the shortest message conveying certain information. For example, the algorithmic information content (or AIC) of a string of bits is defined as the length of the shortest program that will cause a standard universal computer to print out the string of bits and then halt.

All these quantities (which are measures of complexity for a specific entity in the real world) are to some extent context-dependent, since each time they depend on the level of detail of the description of the entity, on the language employed, on the coding method used for conversion of the language chosen into a string of bits and so on. Moreover, many of such quantities cannot be computed. Let us assume the case of the algorithmic information content (AIC) of a long bit string. It can be shown to be less then or equal to a

certain value. However, for any such value we can not exclude the possibility that such AIC could be lower, thanks to a yet undiscovered theorem revealing a hidden regularity characterizing the string at issue.

There is a different measure (which does not refer to the length of the most brief description of a set of entity, as AIC does) concerning the length of the most concise description of the regularities of the entity. The author calls it "effective complexity", and he depicts it as follows:

> Thus something almost entirely random, with practically no regularities, would have effective complexity near zero. So would something completely regular, such as a bit string consisting entirely of zeroes. Effective complexity can be high only in a region intermediate between total order and complete disorder.

Still in the computational field, there can happen situations in which such computations are difficult. For example, let us consider the case of the time involved in deducing practical predictions from a scientific theory. So, this is the case of time measures of complexity, and an example is "logical depth". For a bit string, such measure refers to the time required for a standard computer to compute the string, print it out and then halt. Then, such time is averaged over the various programs that will accomplish the task.

As Gell-Mann (1995) points out, while observing natural phenomena we frequently have to distinguish between effective complexity and logical depth. However it is not an easy task to tell whether something apparently complex really reflects a certain amount of logical depth or, instead, possesses a great deal of effective complexity.

The last measure of complexity the authors refers to in his paper is called "potential complexity". It is defined as a function of future time, relative to a fixed time, which could for example be the present. Such quantity is the effective complexity of the entity at a future time, averaged over the various coarse-grained histories of the universe between that time and the present, each of them weighted according to their probabilities.

1.6 Networks

1.6.1 Introduction

In what follows we try to give a brief explanation of Networks. The reason why we do so, is due to the fact that networks are a fundamental component of the project we have developed. The path we are going trough is characterized as follows: starting from some examples of situations in which we can find out the presence of these networks, we will then consider the particular case of social networks, by means of a definition and the description of their structures and main features. These complex interrelated systems are everywhere, from natural structures to man-made systems, as Bransburg-Zabary *et al.* (2013) points out. All-days examples are:

a. Economic networks. We can consider markets as a huge directed multi-relational network. All the elements making such markets working, such as companies, firms, financial institutions and governments, play the role of "nodes". "Links", instead, represent different interactions between such nodes. Common examples are: purchases and sales or financial loaning. The value of the transaction is captured by the weight of the links. This representation focusing on the economy as a network of interacting actors is useful to make sense of global financial meltdowns, which are provoked by a sequence of failures cascading over the highly connected and interdependent network economy.

b. Power-line and airline networks. Human-made networks. One of the main feature of such systems is that they might be involved in random failures as well as targeted attacks. Moreover, failures may have cascading effects: indeed, the failure of one node may recursively cause the failure of connected nodes, and the consequences of such events might reveal to be catastrophic.

c. Social Networks. Their main feature is linking people according to various social relationships, such as acquaintance, friendship, collaboration,

and sexual relation. The reason why they are of great importance is due to the fact that they allow to understand and anticipate the spread of ideas, innovations, as well as biological and computer viruses.

d. The World Wide Web. It is a directed network of hyper-links via URLs. Nodes represent web-pages while edges are the hyper-links between pages. In particular, there exists an edge from "pageăp"ăto "pageăq"ăif "pageăp" contains at least one hyper-link pointing to "pageăq".

e. Metabolic and protein networks. Networks structured as follows: the nodes are simple molecules like water or ATP while the links are the biochemical reactions that take place between these molecules. Furthermore, proteins can be viewed as nodes of a complex network in which two proteins are connected if they can physically interact.

Some of these networks are made by nature (e), other are built by humans (a, b, d). All of them grow in a self-organised way, without any sort of regulation coming from central authorities. We can go on with several other examples. However, these are of great importance and sufficient to our purpose, which is highlighting the presence of such networks in our all-day life. So, it turns to be fundamental to understand their growth, structure, dynamics, functioning and their mutual interrelationships in order to find out their strengths and possible weaknesses: in this case, a better understanding could lead us to discover solutions making these systems resilient against failures and attacks.

Recently, significant advances have been made in understanding the structure and function of networks. Furthermore, mathematical models of networks are now widely used to describe a broad range of complex systems (from social and technological systems to interactions among proteins). However, the other side of the coin is that until recently methods deal nearly exclusively with individual networks considered as isolated ones. Instead, an individual network is often just one component of a more complex network, the so called "network of networks". An other recent issue is the reinforce-

ment of links between existing systems (the so called "coupling between networks"). Blackouts are a useful example of such dependence between them. When long-lasting, they could result in a widespread failure of networks in several domains: healthcare, finance, railway, communication. In this cases, the failure of nodes in one network may cause failures of dependent nodes in other networks, leading to a cascading effect of failures within such networks.

Since recently developed, the debate around networks is still open, and some realistic features have not been taken into account by traditional formalism yet. Examples are: coupling between networks (versus the "isolated" feature assumption), dynamics of networks (versus the "static" feature assumption), spatial properties of networks. Providing a solution to such issues will lead to a new generation of network science, which could perhaps foster new tools to understand the world we live in (Dror Y. Kenett (Shlomo Havlin)).

In what follows we will analyse in detail what a network is, what it is made of, what types of networks exist, how they evolve as time goes by and other characteristic features referred to the particular type of "Social networks" . To do so, we refer to Cioffi-Revilla (2014).

1.6.2 Social networks, from a qualitative point of view

A social network is a social structure made up of a set of social actors (i.e individuals or organizations) and a set of ties between these actors. By means of a social network perspective we have available a set of methods that can be used for analysing the structure of whole social entities as well as a variety of theories explaining the patterns observed in these structures. Social networks are everywhere. This statement is supported by Cioffi-Revilla (2014), which says:

> Social networks consisting of actors and social relations are ubiquitous across the social science disciplines. Networks are consequential and frequent in anthropology, economics, sociology,

political science, and psychology -the Big Five social sciences- as well as interdisciplinary areas such as communication, management science, international relations, history, and geography, especially human geography.

Definition

As emerges social networks are composed by several constituent parts including:

i) entities (as actors, values, ideas, locations, attributes)

ii) relations (as links, ties, interactions, associations) and

iii) aggregations (as dyads, triads, groups, and subgroups).

More specifically, following Cioffi-Revilla (2014), we can characterize a network as follows:

Definition 1 *A Network $\mathcal{N}$ consists of a finite set $\mathbb{N}$ of entities (called nodes or vertices), denoted by $\{n_1, n_2, n_3, ..., n_g\}$ and a set of relations $\mathbb{L}$ (called lines, links, or edges), $l_1, l_2, l_3, \ldots, l_L$ defined on the set of nodes $\mathbb{N}$.*

Note that g stands for the cardinality of $\mathbb{N}$, id est it represents the total number of nodes in $\mathcal{N}$. The cardinality of $\mathbb{L}$, instead, is $L = g(g-1)$ for directional pairs, where a directional relation between node i and node j is denoted by $n_i \rightarrow n_j$ or x_{ij}. This is a fundamental concept upon which practically infinite possibilities of other kinds of network concepts, models, and methods are built on.

The benefit coming from the usage of social networks analysis as a framework is clear: the possibility to study relationships between individuals or groups, organizations or even societies. These relationships are investigated through the properties of relations between and within units and not in terms of the properties of these units themselves. This paves the way to the extension of the analysis of complex couples human-natural-artificial systems by

providing concepts, notation and applied principles that are useful in doing so.

Structures and types

Graphs are instruments providing an appropriate representation of the social networks. In particular, they are tools by means of which we can understand differences between the main structures and types of social network, with the former referring to the recognition and permanence of patterns and relationships between entities. In this paragraph we follow Cioffi-Revilla (2014) in order to specify the main items of these two categories, even exploiting its related figures to gain an appropriate knowledge about this topic.

As Cioffi-Revilla (2014) observes, there are four main types of social networks according to their social relations $\mathbb{L}$ ($l_1, l_2, l_3, \ldots$)

a. Directed network or digraph: social network characterized by directional social relations. It differs from an ordinary or undirected graph in which the links between nodes lack an exact direction, so the network is defined in terms of unordered pairs of vertices; instead, in a digraph each edge has a definite direction associated with it. Moreover, a digraph can be defined "simple" if it has no loops, and no multiple arcs. Typical examples of such networks are transaction networks, such as those consisting of flows between nodes. Trade transactions (money or goods), flows-of-people transactions (flows of migrants, tourists, international students), or other resources (imports, exports, information).

b. Signed network or valued network: social network characterized by links having valence signs: +, -, 0. An immediate example comes from politics: allies would be described by a positive sign, adversaries by a negative one, while neutrals by a zero.

c. Weighted network: social networks in which links show weight or intensity of some kind. Examples could be the strength of friendship ties or the volume of trade between countries.

d. Multiplex: multiplex networks are multilayer systems of $\mathcal{N}$ nodes that can be linked in multiple interacting and co-evolving layers. In these networks, relevant information might not be captured if the single layers were analysed separately. This is why multiple or parallel associations between node pairs should be taken into consideration in such networks. Put differently, the set of social relations $\mathbb{L}$ contains multiple social ties or links between nodes. Moreover, the interconnections between layers are only between a node and its counterpart in the other layer. An obvious example are families, but other examples could be the social networks Facebook and Twitter.

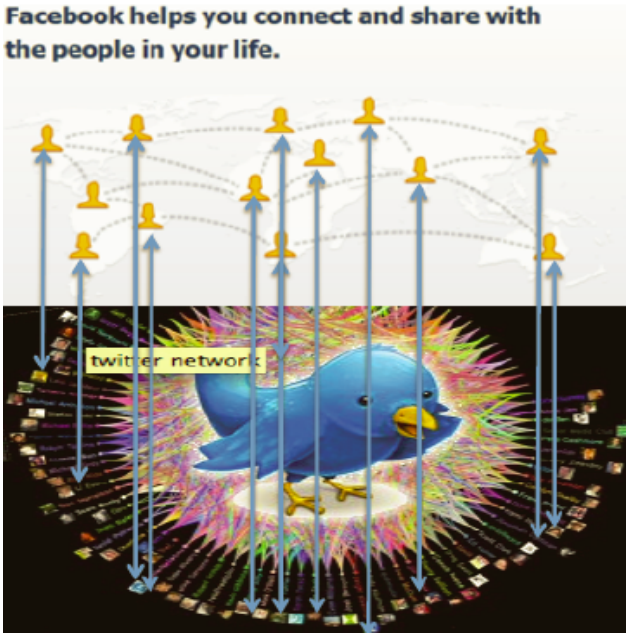


Figure 1.14: Facebook and Twitter represent two examples of a multiplex typology of social networks where users are the nodes by way of which interconnections between layers can take place. In particular, Facebook is a network of networks, i.e., the aggregate of many different social circles or subnetworks, each having its own temporal or structural patterns.

A graphical representation will enhance a good understanding of these networks' shape. Figure1.15 provides some help. In the upper left corner of

the figure a digraph is depicted. In the upper right corner a signed graph with valences. The lower left corner lodges a weighted network, while the lower right one a multiplex with various kinds of possible social relations between nodes.

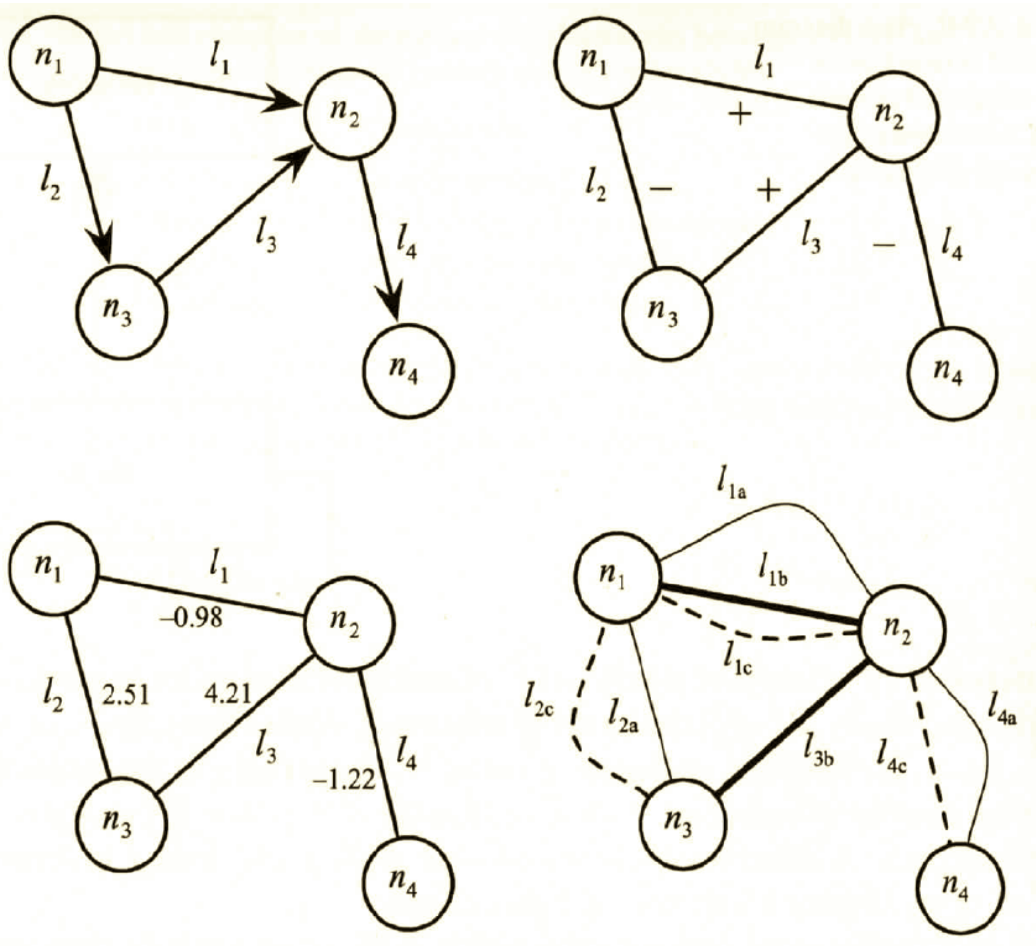


Figure 1.15: Representation of the four main types of social networks. Since we have followed Cioffi-Revilla (2014) for what concerns the types of networks we can work with, in order to be coherent we chase its graphical representation too.

Further steps can be done. In particular we have to focus our attention on the analysis of social networks at the scale relevant to our theoretical and practical question, in order to figure out possible network properties

involved. This is why the level of analysis represents a fundamental aspect in the structure of a social network. Following Cioffi-Revilla (2014), we will distinguish several levels of analysis. Starting from a micro level and pointing to the macro one (bottom up procedure), in turns we have:

Nodal level: it is the most detailed level of social network analysis. It focuses on attributes of node-entities, such as nodal degree, centrality and other significant ones.

Dyadic level: as the origin of the term reveals (in Greek "duo" means "two") this level of analysis is focused on the relationship between two units. In particular, it is possible to analyse a relational pair as a binary unit from a number of different perspective. Moreover, the qualitative type of dyads comprised in a social network can determine the very character of the network. In Figure1.16, all the networks shown contain dyads.

Triadic level: since triads can play an important role in many fields, such as balancing processes and transitive relationships, such networks triads are significant at all scales of social networks.

N-adic level: the social network analysis can, by induction, examine any aggregation of unit nodes and relations, up to the entire size of the network, so up to the condition g=$\mathbb{N}$, denoting the total number of nodes in a network.

Network level: such level examines aggregate attributes such as size, diameter, connectedness, centralisation and others. Since the analysis at this level can involve aggregate properties and phenomena, the network level is commonly associated with complex system analysis.

Even if nowadays the greater part of available knowledge about social networks is at the node level and the network level, several measures are defined for each of these intermediate levels so, in principle, any social network can be described in quantitative terms, if there are sufficient data, irrespective of

the specific structure of the network. We will spend the entire next section in analysing such point of view.

Another main feature characterising networks is whether they are static or dynamic. So far we have considered social networks from a static perspective only. However there exist *dynamic networks* too. They are networks whose state changes as a function of time $\{t\}$. Common types of behaviour they exhibits are: growth, evolution, transformation, decay, among other patterns. This feature, as well as other ideas we have seen previously about social networks, does not depend on the specific structure of a network. In what follows we present a simplified scenario of social networks structures we can chance upon.

Social Networks Structures

In the real world the more intuitive way of understanding that social networks differ among each others is taking into consideration their structure. Within the huge set of possible "architectural structures" they can adopt, there are certain types that are significant for their properties and recurrence. We now present these types of social networks and try to understand better their shape by mean of a graphical representation too. All of them are presented without any reference to their relational type. Indeed they can have any relational type. Which of the four types previously seen (directed, valued, weighted or multiple) depends on the nature of its dyads. Approximating these networks by their increasing complexity we have (still following Cioffi-Revilla (2014)):

a. Chain network: also known as "line network". It is constituted by a string of nodes. Common examples are multi-stage processes of many kinds.

b. Star network: also known as "wheel network". It presents a radial form. There is a central node radially linked to all the others around it.

c. Y network: also known as "tree network". It seems a chain with split or frayed terminal path. Branching processes are a social example.

d. Circle network: it is a closed chain with nodes linked in a circular fashion. This structure is the least hierarchical seen so far.

e. Complete network: it shows the maximum degree of communication since evry node within it is connected to the others.

f. Cellular network: it is a network with one or more nodes having a complete graph attached to it.

All these structures are represented in figure1.16.

1.6.3 Social networks, from a quantitative point of view

As Cioffi-Revilla (2014) points out, there are two different classes of social network measures: one measuring the attributes of nodes ("micro-level nodal measures") and the other the features characterizing network structure as a whole ("macro-level network measures"). Since in this project-work we are involved in the interaction between agents within a single network, analysing the first set measures is sufficient to understand the framework around our work.

Even if new measures are being invented as time goes by, we have selected those we think are more significant.

(i) *Degree:* within a network, the degree of a node represents the number of connections between such node and the others. In the case of a directed network, nodes have two different degrees, the "in-degree" and the "out-degree", with the former measuring the number of incoming edges and the latter that of outgoing edges. In other words it is a measure of centrality, since measuring the number of links incident on a node.

(ii) *Distance:* it measures the number of links connecting two nodes. So $d(n_i, n_j)$=0 for all the nodes n_i within the network.

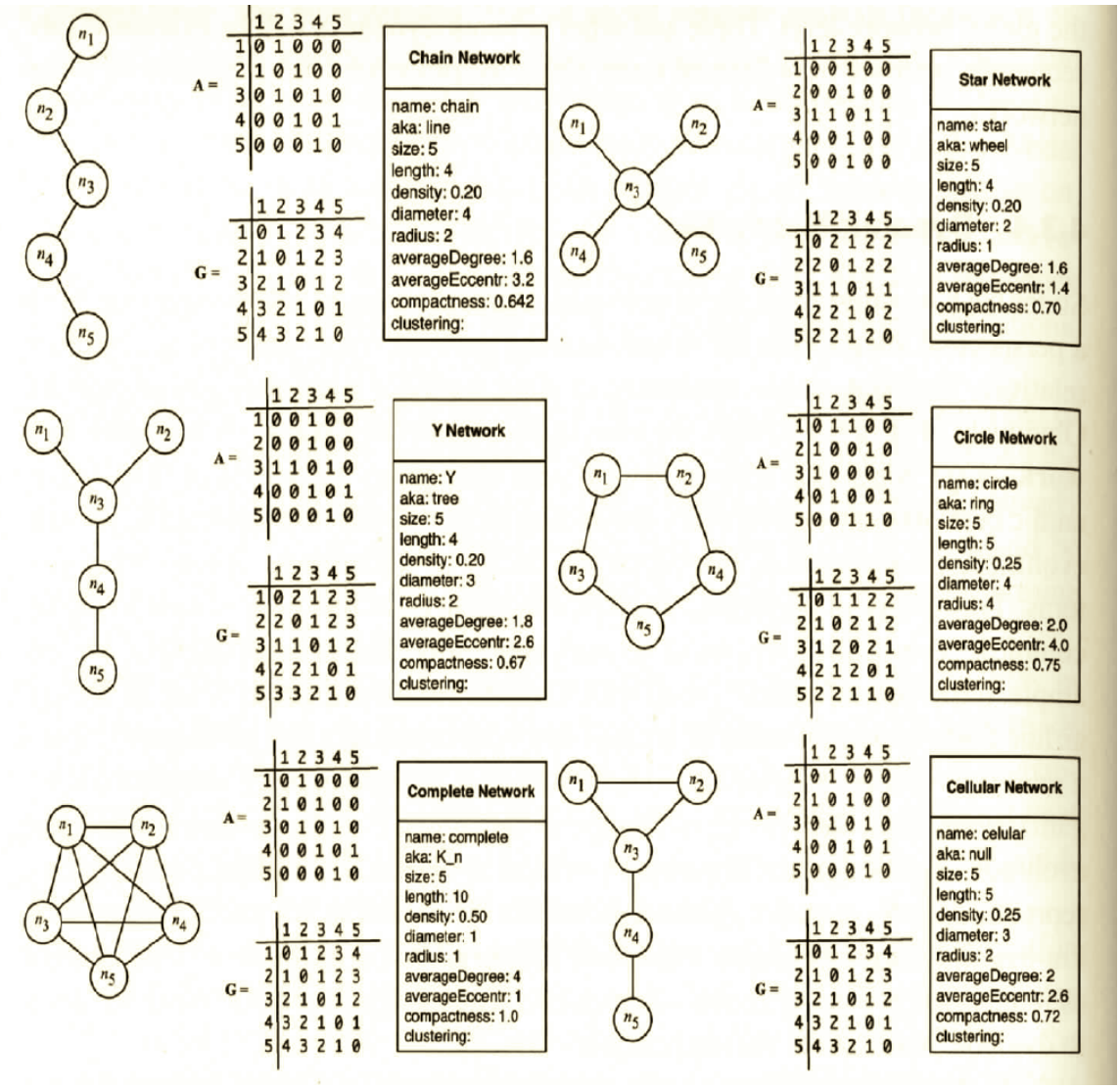


Figure 1.16: Elementary social networks structures: from the left to the right, top-down, we have examples of a Chain network, Star network, Y network, Circle network, Complete network, and Cellular network.

(iii) *Betweenness centrality:* it measures how often a node appears on shortest paths between nodes in the network. When referred to social network, this measures leads to important implications, allowing for further considerations with respect to the nodes showing the higher degree of betweenness.

Chapter 2

The Model

2.1 Introduction

This part is devoted to the presentation of the core part of our thesis. It consists in a model we developed by means of a simulating software (NetLogo), which will allow us to make a vast thinking given some initial condition and some rules. The technique we have used is that of Agent Based Modelling. As we have seen in the literature part in the section committed at explaining ABM, this particular way of modelling allows to built "bottom-up" models, characterized by a set of agents acting and interacting among themselves and with the environment they are located in. The point we have to start from is a work of Joshua M. Epstein and Robert Axtell, who have developed a simulating mechanism for all sorts of emergent behaviour within a grid of cells managed by a computer. In particular, in Epstein (1996), the authors justified the usage of model-building with the following words:

> The social sciences are also hard because certain kinds of controlled experimentation are hard. In particular, it is difficult to test hypotheses concerning the relationship of individual behaviors to macroscopic regularities, hypotheses of the form: If individuals behave in thus and such a way-that is, follow certain specific rules-then society as a whole will exhibit some particular

> property. How does the heterogeneous micro-world of individual behaviors generate the global macroscopic regularities of the society?
>
> Another fundamental concern of most social scientists is that the rational actor-a perfectly informed individual with infinite computing capacity who maximizes a fixed (nonevolving) exogenous utility function-bears little relation to a human being. Yet, there has been no natural methodology for relaxing these assumptions about the individual. Relatedly, it is a standard practice in the social sciences to suppress real-world agent heterogeneity in model-building.

In their simulations, they propose several examples of real-lyfe phenomena governed by simple rules (i.e the role of seasonal migrations, pollution, sexual reproduction, combat, and transmission of disease). What is worth note is that fundamental collective behaviours such as group formation or cultural transmission are seen to "emerge" because of the interaction of individual agents following few simple rules. In particular, referring to the way of thinking presented in their work, a model such that we have been working with should be composed by three elements:

1) The Agents. According to the definition the authors give, each agent should have two features:

- **a) internal states:** each agent is defined as an entity characterized by a set of *fixed* and *variable* states. For example, given a particular agent, its genetic characteristics (or a generic endowment of elements) are evaluated as fixed since they will not change during its life while its wealth is an example of variable state since will vary, depending on the interaction with the environment or with other agents. When the fixed attributes are different across the agent population, then the model population is considered heterogeneous.

b) behavioural rules: something the agents have to obey to.

2) The Environment. It represents the space within which the agents raise, grow, die, act and interact, both with other agents or with the environment itself. In the case of a software tool as NetLogo, it consists in a screen containing all the agents we have introduced.

3) The Rules. They consist in a set of commands regulating the agents' behaviour. As a consequence, they are the basis of agents' acting and interacting. They can create interactions among agents and elements of the environment (in NetLogo they are called "patches") or among the environment elements.

This set of features constitutes the architecture of a generic model developed with NetLogo. This is why in the following pages we will first of all analyse separately these constituting parts, in order to fully comprehend the basis of our reasoning. Once we had satisfactorily explained such elements, we will then shift focus to the potentiality our work is endowed with, and then we will carry out several experiments deriving some conclusions for each of them.

We now begin with a clear description of the element "Environment" since it represents the framework within which our reasoning takes place.

2.2 The Environment

2.2.1 The real World

In our simulation, the real piece of World we are referring to is the district of Turin in the Italian North-Western region called "Piemonte". Its shape is represented in Figure2.1.

As we can see there is a clear concentration of buildings in the Central-Eastern part of the district, corresponding to the city of Turin. When transforming such real space into our NetLogo environment, we have exploited

Figure 2.1: Georeferenced map of the Turin district within the Italian region "Piemonte".

the fact that the district could be divided into four different areas:

i) The city of Turin

ii) The first belt

iii) The second belt

iv) The residual set of municipalities

The first belt consists in the Turin metropolitan area (pursuant to Art 17 Law No 142/90, a "metropolitan area" is defined as that area, usually standing around a big city, including a set of municipalities whose settlements share strong economic, cultural and local activities with the city at issue), which includes twenty-three municipalities. The second belt is instead

composed of twenty-nine municipalities. This distinction is due to the Italian entity named "Osservatorio Demografico Territoriale del Piemonte", which is aimed at providing data concerning the main characteristics of people living in Piemonte. The set of cities constituting the first and the second belt are represented in Figure2.2, which has been taken on the website of the entity at issue ($http://www.demos.piemonte.it$).

Prima Cintura		Seconda Cintura	
Codice Istat	Comune	Codice Istat	Comune
1008	Alpignano	1002	Airasca
1018	Baldissero Torinese	1013	Avigliana
1024	Beinasco	1034	Brandizzo
1028	Borgaro Torinese	1038	Bruino
1048	Cambiano	1045	Buttigliera Alta
1063	Caselle Torinese	1051	Candiolo
1078	Chieri	1058	Carignano
1090	Collegno	1059	Carmagnola
1099	Druento	1062	Caselette
1120	Grugliasco	1068	Castiglione Torinese
1130	Leini	1082	Chivasso
1156	Moncalieri	1086	Ciriè
1164	Nichelino	1112	Gassino Torinese
1171	Orbassano	1127	La Loggia
1183	Pecetto Torinese	1168	None
1189	Pianezza	1193	Piobesi Torinese
1192	Pino Torinese	1194	Piossasco
1214	Rivalta di Torino	1197	Poirino
1219	Rivoli	1215	Riva Presso Chieri
1249	San Mauro Torinese	1220	Robassomero
1265	Settimo Torinese	1228	Rosta
1280	Trofarello	1240	San Francesco Al Campo
1292	Venaria	1248	San Maurizio Canavese
		1257	Santena
		1302	Villarbasse
		1308	Villastellone
		1309	Vinovo
		1314	Volpiano
		1315	Volvera

Figure 2.2: Set of municipalities constituting the first and the second belt around the city of Turin. The table has been imported by the website of the Italian entity aimed at providing data about the population of the district "Piemonte".

2.2.2 NetLogo Environment

Transforming such real environment into the NetLogo scenario constitutes the beginning of our simulation work, the box within which each action of the agents will then take place. In order to reproduce a simulated world, faithful to the real one, we have divided the screen into four areas, each of them corresponding to one of those real. In doing so, two assumptions have been undertaken, respectively concerning:

i) ***the shape of the district.*** Since it would take too much time to faithfully represent the district shape (but it consists in an improvement which can be done in a further developed version of our work), and since the shape we assume does not imply any limitation to the reasoning we are going to develop in the following pages, we have decided to consider the whole NetLogo screen as the entire district of Turin. So, a rectangular-shaped environment.

i) ***the shape of the belts.*** Since the real belts are more or less concentric around the city of Turin, we have depicted them as concentric rectangular around an inner one representing the city at issue. We have assigned different colours to each of them as follows: a) white colour depicts the city of Turin, b) red colour the first belt, c) yellow the second belt and, finally, d) the green colour represents the rest of the cities.

The related code is the following:

```
to setup-canvas
ask patches [set pcolor green]
set Torino2 patches with
[pxcor >= 60 and pxcor <= 170 and pycor >= 10 and pycor <= 85 ]
ask Torino2 [ set pcolor yellow]
set Torino1 patches with
[pxcor >= 100 and pxcor <= 163 and pycor >= 27 and pycor <= 75 ]
```

```
ask Torino1 [ set pcolor red]
set Torino patches with
[pxcor >= 123 and pxcor <= 145 and pycor >= 45 and pycor <= 60 ]
ask Torino [ set pcolor white]
set Torino3 patches with [pcolor = green]
end
```

In particular, such few lines represent the way through which we assigned to each area the colour, the name, and the coordinates, all together constituting the features of such areas. "Torino" is the Italian word for the city of Turin, while for simplicity we called "Torino1" and "Torino2" the belts from the inside to the outside, and finally "Torino3" is the name we attributed to the rest of the cities.

The last feature we have to focus our attention on is that referring to the coordinates. They have been attributed to each areas as follows. Given the set of cities composing the first and the second belts, for each of these we have taken the coordinates of the more external ones as edges of the belt at issue. This procedure has been firstly developed with real values, and then while transferring them into NetLogo we have maintained the proportions. The results are shown in Figure2.3.

2.3 The Agents

2.3.1 Hospitals

This is the only set of agents that is geo-referenced. In order to geolocalized such institutions we followed the same procedure as that used for building up belts (so, again, beginning from real coordinates we have then plotted such values into NetLogo in a way able to maintain the proportional distances among each buildings, in order to work in an environment well representa-

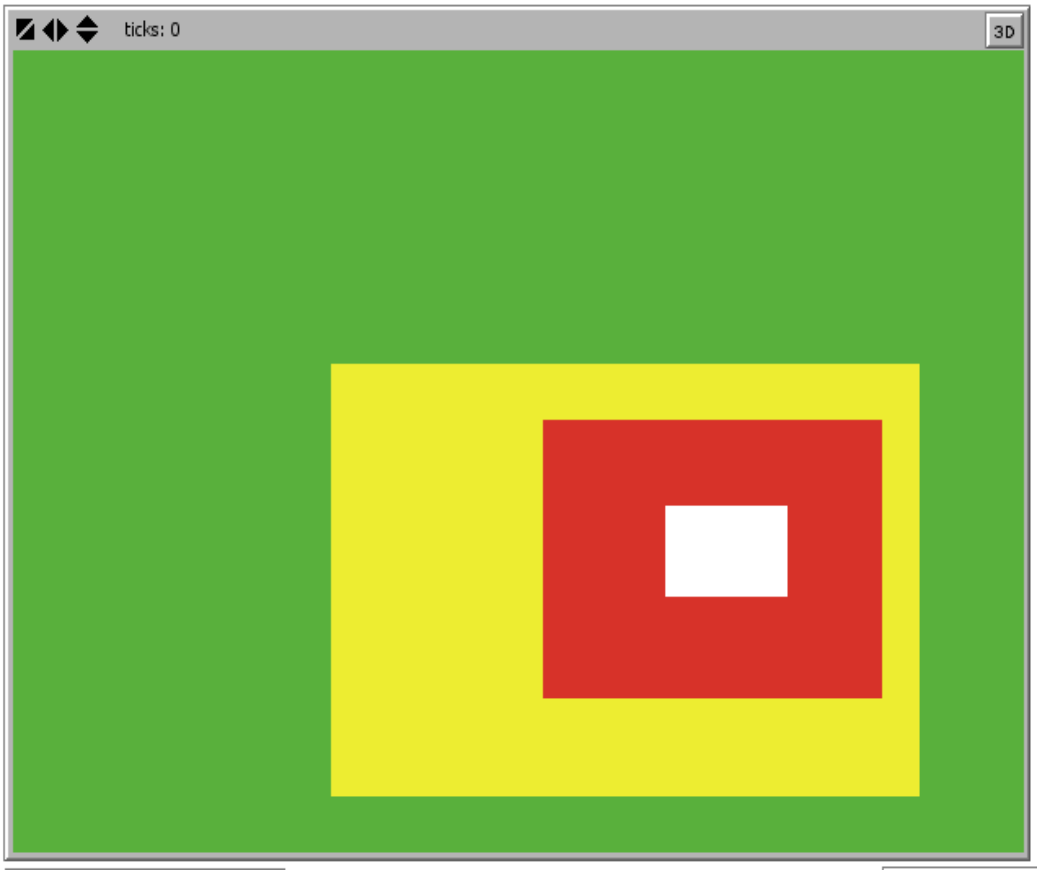


Figure 2.3: Representation of the simulated world, corresponding to the real district of Turin. It is divided into four concentric rectangular areas: i) the white one corresponds to the city of Turin, ii) the red to the first belt surrounding the city, iii) the yellow represents the second belt and iv) the green the rest of the cities within the district.

tive of the real situation). Since our work consists in the analysis of the health-care path patients walk through during their life (so, moving from family doctors to hospitals, from health-care specialists to health-care laboratories), we have chosen a set of representative (not exhaustive again) hospitals within the district of Turin. By representative we mean well-functioning hospitals that are well-known in the district because of some features they are endowed with. We now enumerate all those structures together with the

pair of coordinates (x-coordinate first, y-coordinate then) identifying them in a univocal way on the screen:

1) Hospital 0: CTO (134.6 46.8)

2) Hospital 1: Mauriziano (133 50.2)

3) Hospital 2: Regina Margherita (134.6 47.2)

4) Hospital 3: Molinette (135 48.4)

5) Hospital 4: Orbassano (135 48.4)

6) Hospital 5: Candiolo (113.2 25.94)

7) Hospital 6: Koelliker (130.8 48.4)

8) Hospital 7: Valdese (136.8 51.4)

9) Hospital 8: Cottolengo (136.2 56.2)

10) Hospital 9: Rivoli (104.6 52)

11) Hospital 10: Susa (10.6 72.4)

12) Hospital 11: Pinerolo (64.2 8.6)

13) Hospital 12: Lanzo (97.4 96.2)

14) Hospital 13: Ivrea (174.6 133.4)

15) Hospital 14: San Carlo Canavese (122.6 88.6)

16) Hospital 15: Chieri (160 42.2)

17) Hospital 16: Carmagnola (143.2 15.4)

18) Hospital 17: Moncalieri (138.2 40.4)

The related code procedure is the following:

```
to setup-hospitals

create-hospitals 18 [set shape "house" set size 4 set color black]
set cohord [[134.6 46.8][133 50.2][134.6 47.2][135 48.4][111.8 46]
[113.2 25.94][130.8 48.4][136.8 51.4][136.2 56.2][104.6 52]
[10.6 72.4][64.2 8.6][97.4 96.2][174.6 133.4][122.6 88.6]
[160 42.2][143.2 15.4][138.2 40.4]]
```

Figure2.4 depicts the graphical output.

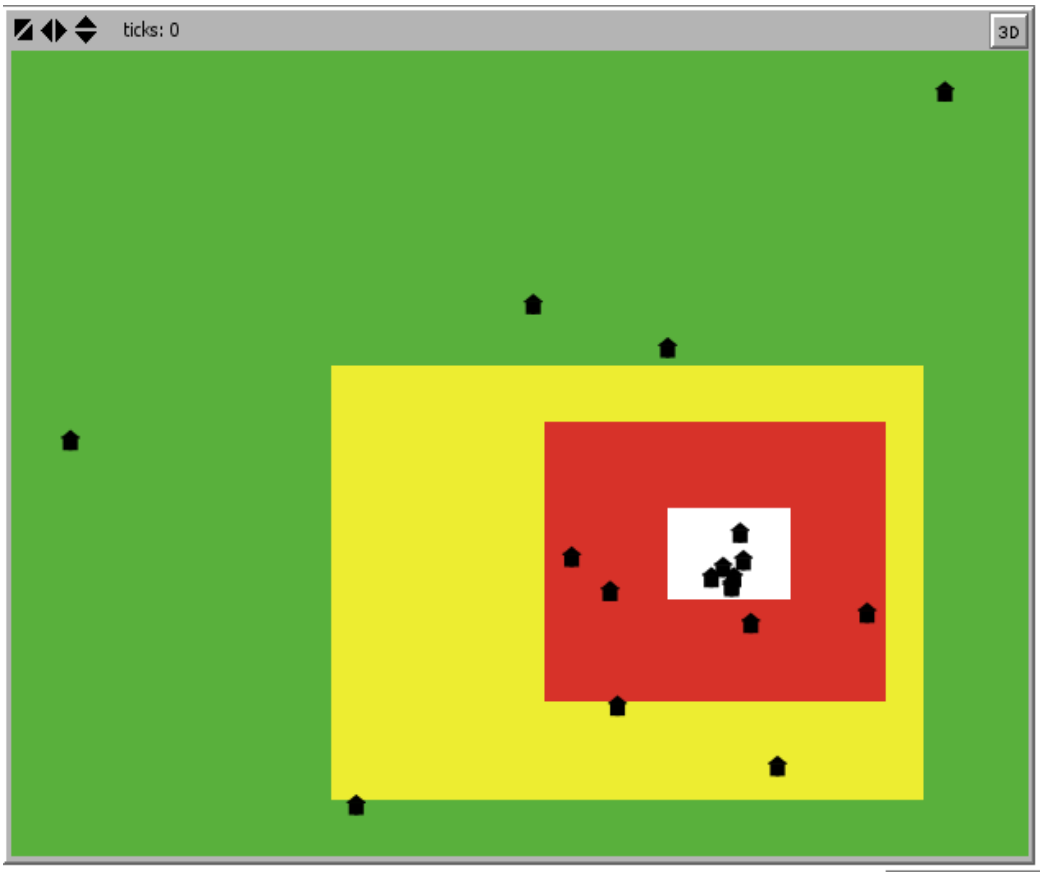


Figure 2.4: Simulated world filled by all the eighteen hospitals.

A NetLogo peculiarity consist in the possibility of inspecting agents by means of a right click on their graphical representation on the screen. This

procedure allows us to consider all the features we have attributed to the agents "hospitals". What we are saying is graphically supported by Figure2.5, in which we have reported all the characteristics of a generic hospital. The worth note features are the following: i) the ID number, labelled as "who", in this case identifying hospital 12, Lanzo ii) the colour, black iii) the pair of coordinates iv) the shape, which have been setted as *house-shape* for all the structures v) the size and, finally vi) the set of services each hospital does provide.

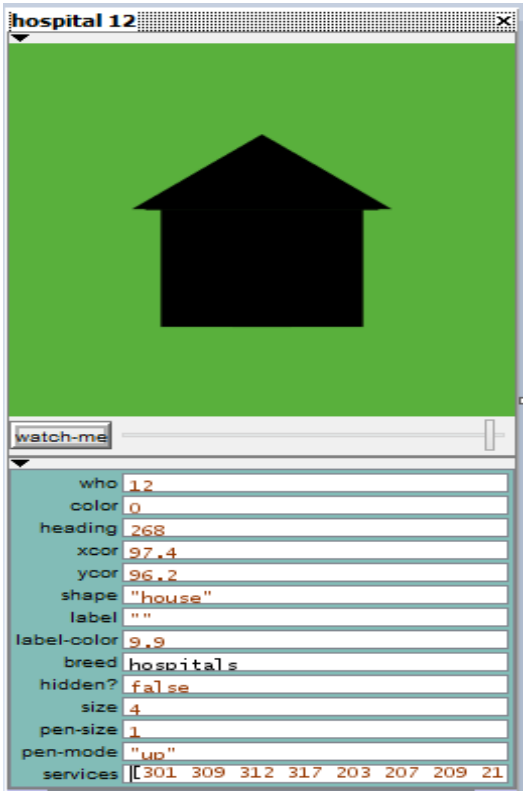


Figure 2.5: Generic hospital inspection. This window allows the user to consider simultaneously all the features he has attributed to the agent at issue. The important ones are: the ID (this is hospital 12, so the hospital of Lanzo), the pair of coordinates, the shape (house type), and finally the services offered.

The issue of services provided is particularly important. We have made the following assumption: *there can exist only three alleged typologies of services the hospitals are able to provide*, respectively:

i) Laboratory-based tests

ii) Specialist services

iii) Compelling services

We have assigned to each service a univocal three figures code. Those starting with 1 (i.e 101, 102, etc) belong to the first typology, then those beginning with 2 (i.e 201, 202, etc) refer to the specialist services (such as an orthopaedic exam, or a dental one) and finally the set of services whose first digit is 3 (i.e 301, 304, etc) are urgent type ones. A further assumption we have based our work on is that *each hospital can at most provide twenty different services (whose identification three figures codes range from 101 to 120, from 201 to 220, and from 301 to 320) for each one of the three typologies at issue.* The related code procedure is reported is shown below:

```
let cat1 []
let cat2 []
let cat3 []
let i 1

while [i <= 20] [
set cat1 lput (100 + i) cat1
set cat2 lput (200 + i) cat2
set cat3 lput (300 + i) cat3
set i (i + 1)
]
set treatments (list cat1 cat2 cat3)
set maxCat (list maxCat1 maxCat2 maxCat3)
```

On the interface of the programme we have then inserted three different sliders (which are represented in Figure2.6), tools that a generic software user can modify according to the different situation he wants to investigate each time. Those we are referring to manage the number of services per typology all the hospitals can at most provide. We said "at most" because the process assigning such services is random, meaning that the number of services attributed to each hospital will be included in a range from 0 to the number selected.

The three figures codes we have referred to up to now do not identify particular healthcare performance provided. Two exceptions have been undertaken, with respect to the healthcare service "blood test":

i) 101 taking of a blood sample

ii) 102 blood analysis

Their provision is widely spread over the healthcare institutions since it represents one of the most common practice hospitals conduct (taking a blood sample mainly). Then it is plausible to suppose that the vast majority of the hospitals is able to offer this service. But not all of them. This is why we have constructed another pair of sliders regulating the probability that an hospital is able to carry out these tasks (see Figure2.7). Again, it will be the software user that will decide the probabilities attached to such services provisions, depending each time on the particular system he wants to test. Indeed, one of our work peculiarity is that it is not district related only. Even if we attached some specific district features such as the position of the hospitals on the screen, or as we will see later on elements such as the population (reflecting real age proportions), we really intended to built up a model able to develop reasoning. In particular, an instrument that, given a set of user-modifiable variables, allows us to test some agents-based phenomena, that will be analysed in the following subsections.

Figure 2.6: Sliders regulating the maximum number of services each hospital in the district of Turin can provide. "maxCat1" refers to laboratory-based tests, "maxCat2" to specialist services and "maxCat3" to urgent ones. By modifying such values we change the maximum number of services all the hospitals can at most provide.

Figure 2.7: Sliders regulating the probability that blood tests are carried out in all the district hospitals. In detail,"bloodTakingHosp" manages the probability that each hospital conduct the taking of blood samples service, while "bloodAnalysingHosp" the probability associated to the blood analysis one.

2.3.2 Specialists

Specialists are the first set of agents effectively providing services. (Indeed, *Hospitals* act as box containing the healthcare services that are then provided by specialists). We have intended them as hospital head physicians, so we have considered plausible the following assumption: *each specialist is able to offer just one service among all those provided within the hospital he is affiliated to.* This falls down under the characteristics labelled as "specialty" each specialist does have.

Since we have them intended as public employees, we now consider all the features we have them attributed:

i) affiliation: each specialist is linked to one hospital, and the number identifying such variable corresponds to one of the eighteen hospitals.

ii) queue: it measures the time a patient has to wait for his illness to get cured. We have assumed it be always greater than zero. In particular, it ranges randomly from one to five-hundred.

iii) cost: since intended as public employees, we assume their efforts in nursing patients' illnesses are offered for free. So costs are zero.

iv) reputation: such variable measures how much the specialist is renowned for the service he does provide. It ranges, still randomly, up to one-hundred.

As a summary of the characteristics we have described so far, together with the graphical shape we have attributed to this kind of agents, let us give a look at Figure2.8.

The process for specialists creation is the following: *According to the health-care services each hospital is able to offer, it creates its specialists-affiliated agents and then they will lie in an area (the so called "radius") surrounding the hospital at issue.* A particular type of Specialists hospitals are able to create are what we have called "ers". The term "er" stands for "emergency room", and we have invented these agents as specialists providing services whose codes begin with the number 3, i.e compelling services. Figure2.9 graphically depicts a generic Er specialist. The associated procedure is now presented.

```
ask hospitals [
let h who
if (random-float 1 < erProb)
[hatch-ers 1 [set color lime set size 2 move-to one-of
 (patches in-radius specialistRadius) set affiliation h]]

foreach services [
hatch-specialists 1
[set color black set size 3 set specialty (lput ? [])
```

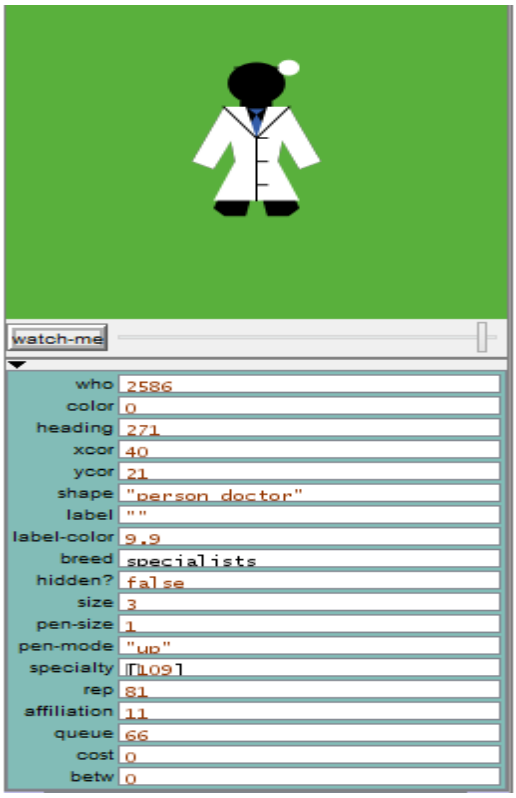


Figure 2.8: Generic Specialist inspection. The important emergent features specialists are endowed with are: the ID (this is agent 2568), the pair of coordinates (differently from the hospitals case they are not fixed, so they will change running the program another time), the shape (person doctor type), the services offered (labelled as "specialty"), and finally the four features described in the text, respectively i) reputation, ii) affiliation, iii) queue, and iv) cost.

```
move-to one-of (patches in-radius specialistRadius)
set affiliation h set queue random 500 set rep random 100]
```

The first five code lines refer to te creation of Ers, with probability lower than one in order to faithfully represent the real situation, i.e the fact that not all hospitals are endowed with emergency rooms. The four last lines instead examine the specialists creation, their capability of nursing one type of services only, their affiliation and their core features. Finally, the user

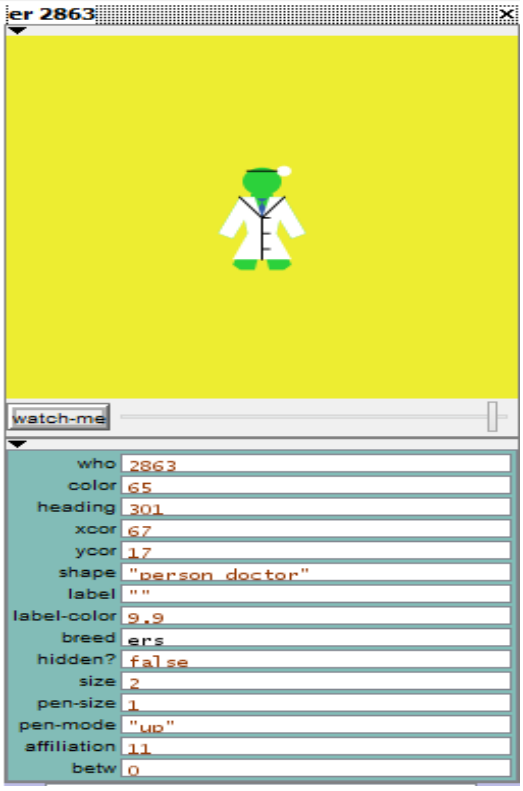


Figure 2.9: Generic Specialist (er type) inspection. The important emergent features specialists are endowed with are: the ID (this is agent 2863), the pair of coordinates (differently from the hospitals case they are not fixed, so they will change running the program another time), the shape (person doctor type) and finally the affiliation.

is able to change the radius around the hospitals (within which both type of Specialists are located) simply by modifying the slider which has been reported in Figure2.10.

Figure 2.10: Sliders regulating the dimension of the radius surrounding the hospital and containing its affiliated specialists.

2.3.3 Professionals

Another set of agents we have introduced in our work consists in those we have labelled as "Professionals". From a theoretical standpoint, we have intended them as private external laboratories which are able to offer the same services Specialists provide. However, the main difference among these two agent-sets is that, while Specialists can provide one service only since we have intended them as hospital head physicians, Professionals offer at least one service and at most ten. In particular their main features are:

i) cost: since private institutions, we have assumed cost to be always greater than zero, ranging randomly up to a maximum of one-hundred.

ii) reputation: in the same way of Specialists, such value ranges randomly from zero to one-hundred.

iii) queue: it is assumed to be zero, since we suppose private employees to be far more inclined to offer better services, in terms for example of waiting time.

So the main assumption we have undertaken is that *there is a trade-off between two variables: cost and queue. Patients are supposed to pay in order to accede to private health-care cures. However, by paying they will not have to wait. The opposite occurs in case of a public institutions as a Specialist is. The service is offered for free but the patient is likely to wait for its illnesses to get cured.*

We now report the related code:

```
to make-externals
set-default-shape professionals "person business"
create-professionals externalLab1
[set color magenta set size 3 move-to extRadius Torino
set specialty n-of ((random 9) + 1) (one-of treatments)
set cost random 100 set rep random 100]
create-professionals externalLab2
```

```
[set color magenta set size 3 move-to extRadius Torino1
set specialty n-of ((random 9) + 1) (one-of treatments)
set cost random 100 set rep random 100]
create-professionals externalLab3
[set color magenta set size 3 move-to extRadius Torino2
set specialty n-of ((random 9) + 1) (one-of treatments)
set cost random 100 set rep random 100]
create-professionals externalLab4
[set color magenta set size 3 move-to extRadius Torino3
set specialty n-of ((random 9) + 1) (one-of treatments)
set cost random 100 set rep random 100]

ask professionals [
if (not member? 101 specialty and
 random-float 1 < bloodTakingProf)
[set specialty lput 101 specialty]
if (not member? 102 specialty and
 random-float 1 < bloodAnalysingProf)
[set specialty lput 102 specialty]
]
end
```

This code procedure is divided into two blocks:

1) the first block: containing the first eighteen lines, create Professionals and then endow them with several characteristics.

- **a) colour:** magenta
- **b) dimension:** size 3
- **c) position:** Professionals are located around the hospitals (the reason supporting this localisation is that in the real world such institutions are likely to open close to the hospitals, in order to "steal" them those clients far more interested in a service for which they

do not have to wait) of the four belts within a radius, whose dimension varies from one to one-hundred (this is a theoretical difference we have introduced with respect to the case of Specialists. Indeed we assumed that *the radius Professionals stand in is greater than that of Specialists, which ranges only from one to thirty*), and it is adjustable by means of a slider, which have been reported in Figure2.11. The related code procedure is:

```
to-report extRadius[district]
let location [one-of
(patches in-radius professionalsRadius)]
of one-of hospitals-on district
report location
end
```

d) specialty: it consists in the illnesses such agents are able to nurse, and it varies from a minimum of one to a maximum of ten (since they have been intended as *institutions (private) and not hospital head physicians only.*)

e) cost: assumed *greater than zero*, it ranges randomly from one to one-hundred.

f) reputation: it ranges, still randomly, from one to one-hundred.

2) the second block: it is constituted by the last six code lines. This procedure attaches with probability ranging from zero to one the ability of taking a blood sample and that one of analysing it to the agent-set at issue. Similarly to the case of Specialists, the user is able to modify these two values directly in the program interface, since we have inserted two sliders managing such values (see Figure2.12)

The agents number is not fixed as in the case of Hospitals, but it can be modified by the user by means of other four sliders devoted to such issue.

Figure 2.11: Slider regulating the dimension of the radius containing the agent-set *Professional* standing around the hospitals.

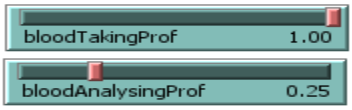


Figure 2.12: Slider regulating the probability attached to each Professional of, respectively, taking samples of and analysing blood. Since it is a probability, it ranges from zero to one.

They are four since each of them manages the number of Professional of a different belt (again, the number of emerging structures will be one included between one and the number chosen, since the creation process is driven by a probability procedure). We can see them represented in Figure2.13.

Figure 2.13: Slider regulating the number of Professionals that will be created around the hospitals (randomly) in the belt at issue.

We now conclude this paragraph by presenting Figure2.14, in which all the features of Professionals are reported.

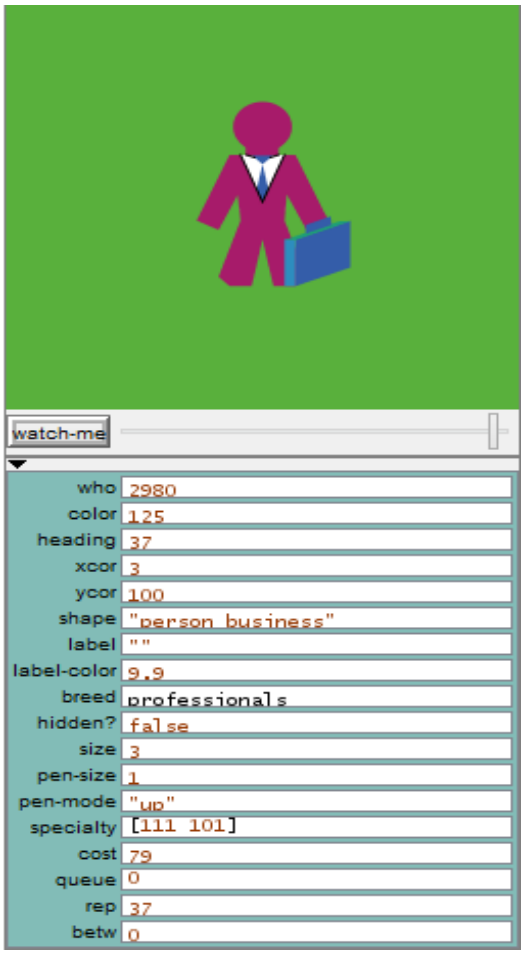


Figure 2.14: Generic Professional inspection. The main features such agents are endowed with are: i) the ID (in this case agent 2980), ii) the pair of coordinates (again not fixed), iii) the specialty (i.e the set of health-care services offered, in this case services 111 and 101), iv) cost, v) queue and finally vi) reputation.

2.3.4 Familydocs

This set of agents has a really important function in our work. As we have already mentioned, our goal is to analyse, given some preliminary conditions within a simulating environment, the health-care life of a set of patients faithfully (the most we can) representing what happens in the real world. Familydocs (intended as the real family doctors) represent the first step patients have to walk through in order to simulate this phenomenon. We have built up our program as follows: *patients, ill individuals, go to Familydocs who are the only set of agents able to attribute to patients their future movements towards Specialists or Professionals. Indeed, once observed the illnesses patients are affected by they made a calculus whose output will determine the movement to Specialists or Professionals in order to get cured* (we will see in few pages how Familydocs play their role).

The number of such type of agents varies across the belts (from a minimum of one to a maximum of one-hundred). The way they are created is similar to that for Professionals (the only difference is that they are randomly located in each belt); indeed, we have introduced four different sliders, each of them regulating the maximum number of Familydocs each belt can contain. We say "maximum number" because, once we have set the sliders, the effective agents there will be plotted on the screen is a random number ranging from zero to the value set by the user. The related code procedure we are referring to is shown below:

```
to make-doctors

set-default-shape familydocs "person doctor"
create-familydocs familydoc1
[set color pink set size 3 move-to one-of Torino]
create-familydocs familydoc2
[set color pink set size 3 move-to one-of Torino1]
create-familydocs familydoc3
[set color pink set size 3 move-to one-of Torino2]
```

```
create-familydocs familydoc4
[set color pink set size 3 move-to one-of Torino3]
```

As an example of the fact that the effective number of agents will be randomly determined in the range [0, user chosen value] we refer to Figure2.15. In this case the numbers of Familydocs user-chosen are: 1 (for the city of Turin), 5 (first belt), 1 (second belt) and finally 10 (residual set of municipalities). The effective numbers of agents plotted on the screen are respectively: 1, 2, 0 and 9.

The set of sliders ruling such creation process are represented in Figure2.16, while Figure2.17 depicts a generic Familydoc agent together with its own features.

2.3.5 Patients

This is the last set of Agents composing our work. The main assumptions concerning them are now explained in detail:

1) number: the first step we have done has been searching for the real number of population actually living in the district of Turin (approximatively 2,254,720 people). The data refer to the last update (January, 1, 2013) the national statistical institute (ISTAT) has carried out. The data are divided in five years based age classes, ranging from zero to one-hundred. We have made a further step, finding out real numbers and real proportions on a year by year basis and then we have maintained such proportions in order to faithfully represent the real population. The only assumption we have undertaken is an approximatively one thousandth scale (the program begin with 2,500 patients, but the number can be easily modified on the screen by simply inserting the desired number in the input represented in Figure2.19). Figure2.18, which is the Turin district Population pyramid updated at the beginning of 2013, captures the five years basis proportions we were talking

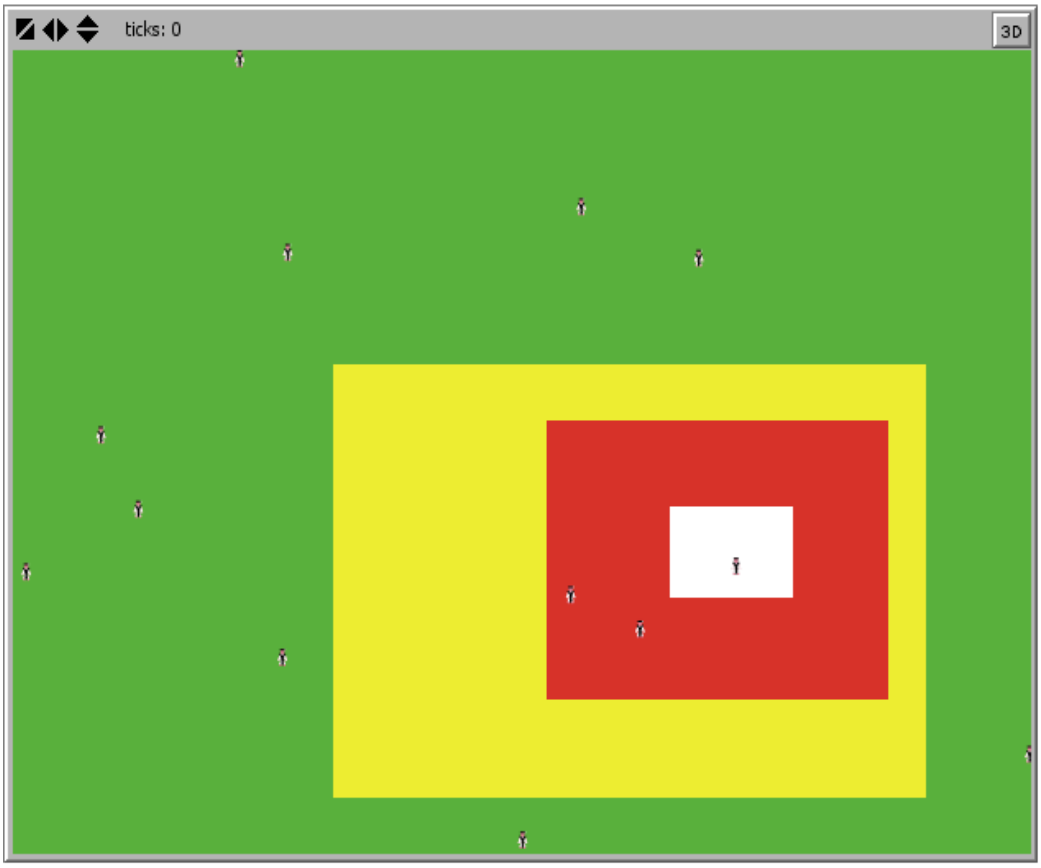


Figure 2.15: Simulated world filled by some Familydocs. This figure is aimed at explaining that there can be difference from the number of agents set by the user and that plotted on the screen because of the random process driving the creation procedure.

about. What we have to highlight is that *the program reads such year by year proportions from an external file, written down in the Excel format, and then manages with probability such number, as in the case of Professionals creation.*

2) location: patients are spread all over the simulated world with the following proportions, 40% in the city of Turin, 30% in the first belt, 20% in the second one and the last 10% in the rest of the municipalities. These proportions are quite similar to the real ones. However they can

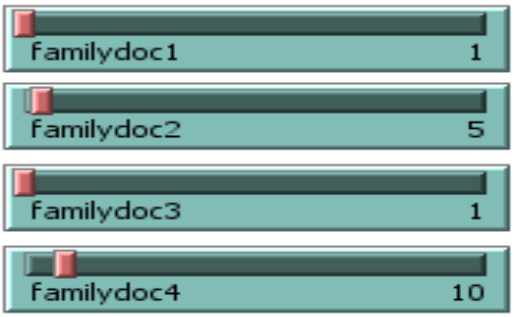


Figure 2.16: Slider regulating the number of Familydocs that, at most, will be created within the area at issue.

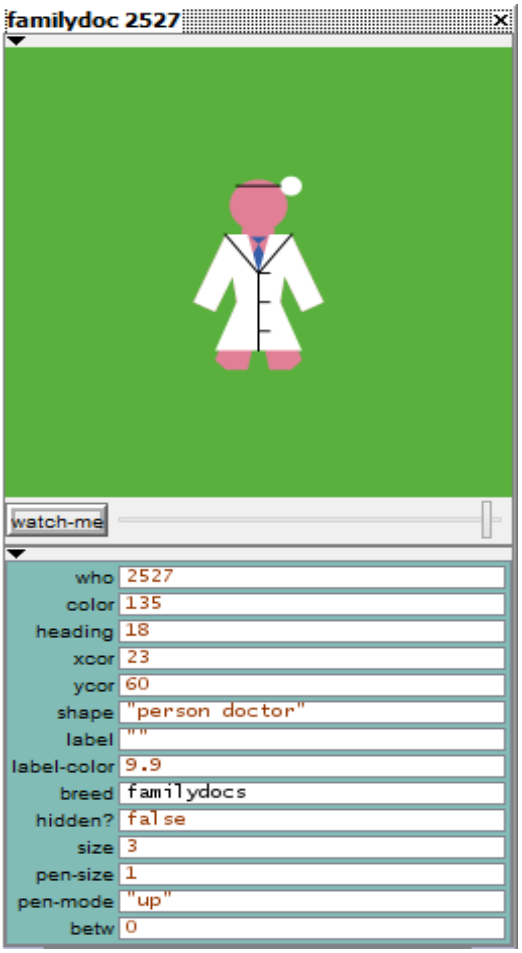


Figure 2.17: Generic Familydoc inspection.

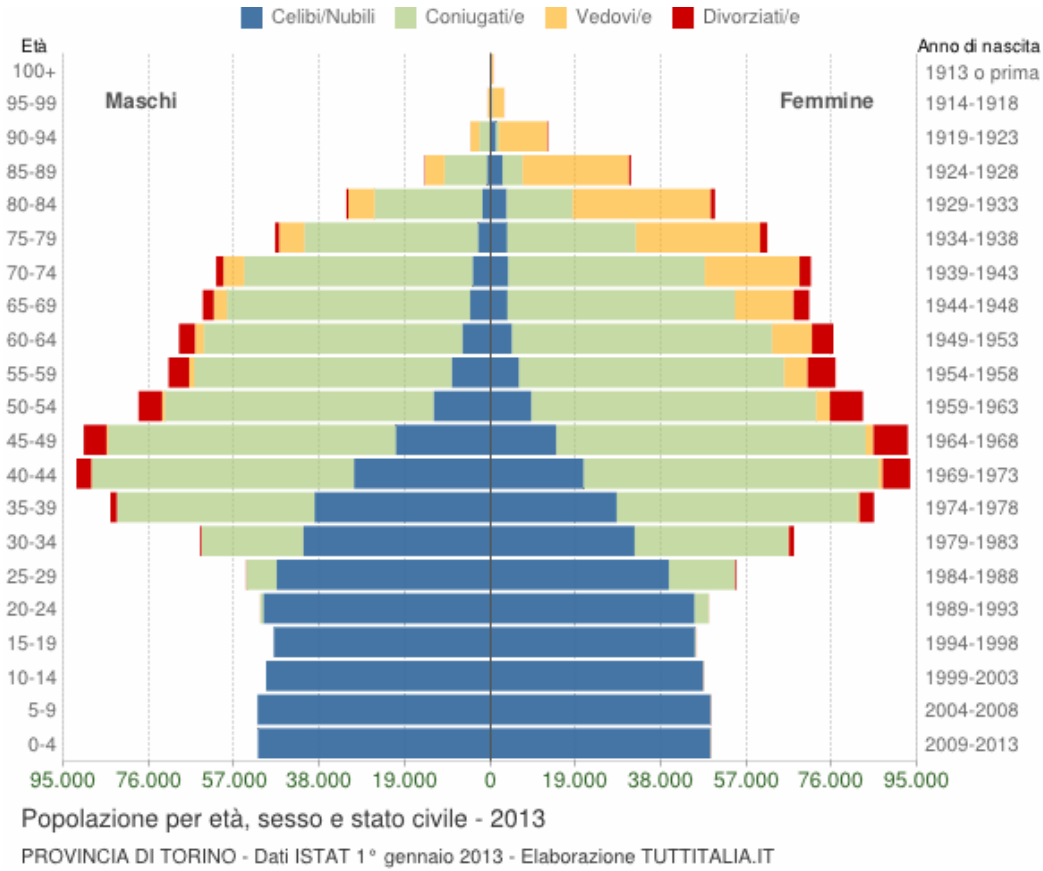


Figure 2.18: Population pyramid of the Turin district. On the left side of the pyramid we can see the proportions of the male part of the population, divided on a five-years basis. The right one is instead devoted to the feminine part. What we are interested in for our work are the proportions of males, females and their age, while we do not take into consideration their marital status.

Figure 2.19: Input managing the number of patients the user wants within the program.

be modified by an external user simply changing the values in the code procedure shown below:

```
to make-patients
set-default-shape patients "person"
let t1 round (0.4 * agents)
let t2 round (0.3 * agents)
let t3 round (0.2 * agents)
let t4 round (0.1 * agents)
create-patients t1
[ set size 2 set color black move-to
one-of patches with [pcolor = white] ]
create-patients t2
[ set size 2 set color black move-to
one-of patches with [pcolor = green]]
create-patients t3
[ set size 2 set color black move-to
one-of patches with [pcolor = yellow]]
create-patients t4
[ set size 2 set color black move-to
one-of patches with [pcolor = red]]
end
```

A further assumption we made concerns the more external of the four areas. As we can see in Figure2.20, representing a geographical representation of the district, the area at issue is characterized by the presence of mountains and mountain chains, and consequently there are some places where individuals can not live in.

Instead, we assumed *an homogeneous land, with no mountains, where all sort of agents can stand in and move through without any problem.*

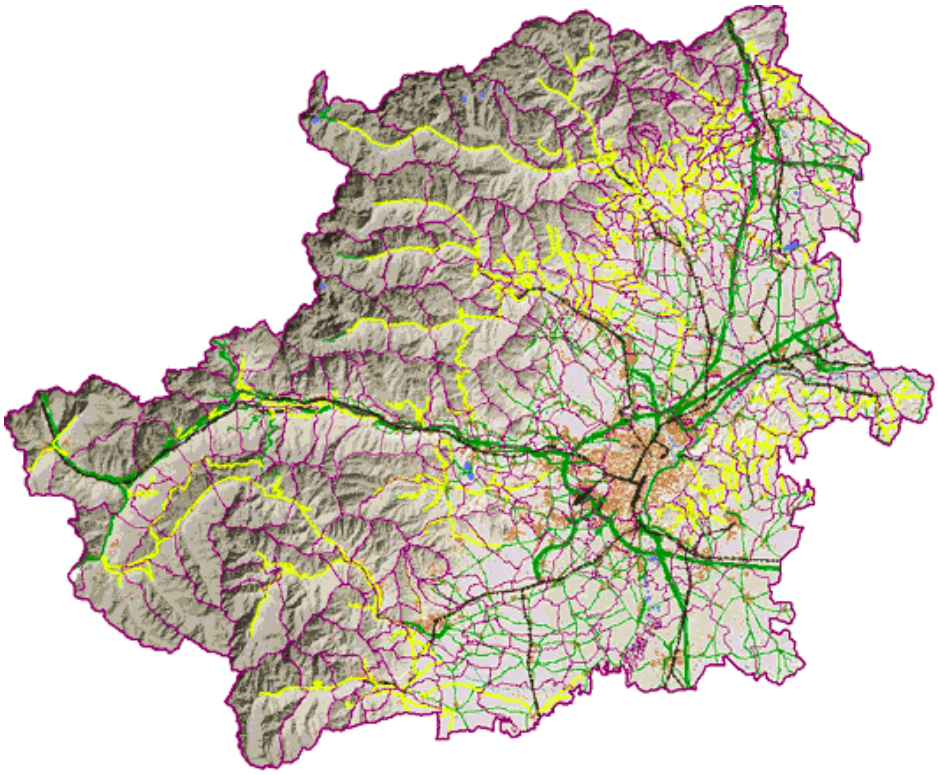


Figure 2.20: Geographical representation of the Turin district within the Italian region "Piemonte".

Obviously it is a simplification, however it will not cause great negative consequences to our goal of a faithful representation of the real world.

3) age categories: we have divided patients into five different age classes.

i) *0-4*

ii) *5-14*

iii) *15-44*

iv) *45-65*

v) *>65*

Such distinction will allow us to attribute to each of these classes health-care features more faithfully to the real situation. In particular, it

allows us to attach to each of them different probabilities of getting ill in every life-time periods, and in each of these age classes different probabilities of falling ill because of one of the three types we have previously explained when talking about the services hospitals are able to offer. What we are now discussing is strictly related to a set of sliders we have inserted on the screen of the program (see Figure2.21). There we have reported all the sliders managing the probabilities for all the five age classes. The way we have to "read" them is the following: the first number of each label of each slider tell us which age class they are referred to. We now analyse in detail the elements of the first set of sliders managing the first age class (the interpretation for the other four set of sliders is absolutely the same):

"prob1": refer to the probability that a patient belonging to the first age class, i.e a baby (0-4 years), both female and male, falls ill. So, when the slider "prob1" is set on 0.15 it means that individuals of the first age class are ill in the fifteen percentage of the cases.

"len1": consists in the number of illnesses (differently speaking, health-care services that either Specialists or Professionals will nurse) that a first age class individual is affected by. Even if "len1" is set on a zero value, but "prob1" is different from zero, then patient will be affected by one illness since "prob1" different from zero means that the patient is sick, and so at least one disease is required.

"p11", "p12", "p13": labels concerning the probabilities that each patient of the first age class will be hit by illnesses of, respectively, type one (those that need a laboratory-based test), type two (those that need a specialist-based cure) or type three (those nursed in the hospitals emergency rooms).

The code procedure used for the creation of such probabilities is the following:

```
to normalize-prop
```

```
let p11n (p11 / (p11 + p12 + p13))
let p12n (p12 / (p11 + p12 + p13)) + p11n
let p1 (list p11n p12n 1)

let p21n (p21 / (p21 + p21 + p23))
let p22n (p22 / (p21 + p21 + p23)) + p21n
let p2 (list p21n p22n 1)

let p31n (p31 / (p31 + p32 + p33))
let p32n (p32 / (p31 + p32 + p33)) + p31n
let p3 (list p31n p32n 1)

let p41n (p41 / (p41 + p42 + p43))
let p42n (p42 / (p41 + p42 + p43)) + p41n
let p4 (list p41n p42n 1)

let p51n (p51 / (p51 + p52 + p53))
let p52n (p52 / (p51 + p52 + p53)) + p51n
let p5 (list p51n p52n 1)

set probIllness (list p1 p2 p3 p4 p5)
set len (list len1 len2 len3 len4 len5)
set probAge (list prob1 prob2 prob3 prob4 prob5)
end
```

4) healthcare conditions: The set of sliders just explained are clear examples of truthfulness, since we can introduce differences among age classes depending on real information. For example, it seems to be reasonable to attribute more illnesses to the last class age individuals than that attributed to those belonging to the first one (and so the slider labelled as "len5" will show a greater value than what "len1" does). Moreover, they allow to identify patients in an univocal way. Indeed, once the sliders have been set and the user gets the program off the ground, patients are endowed with a set of codes identifying their all life long health-care services (in other words, all the illnesses that have to be nursed during their life). The structure is the following: we made use of the metaphor of the *recipes* which, from a technical standpoint, are sequences of numerical or alphanumerical codes. They are reported in vectors, and drive the movement of the patients they are attached to towards other agents. In this way events are determined and the edges of the emerging networks are generated. Such recipes are coded as strings of numbers, each of them related to a specific act. An example is now well-timed:

[44 3 101 203 301 302]

The way we have to read the string of numbers is the following:

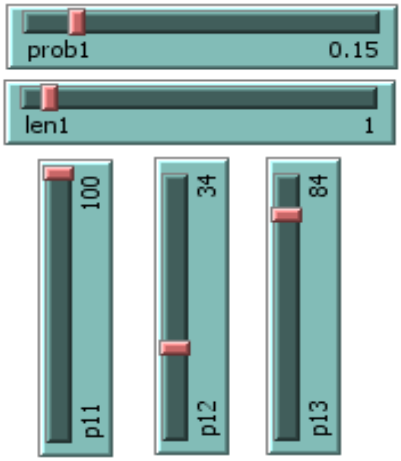
i) the first number: is the year the recipe takes place in.

ii) the second number: is a further level of detail, representing the specific month within the year at issue.

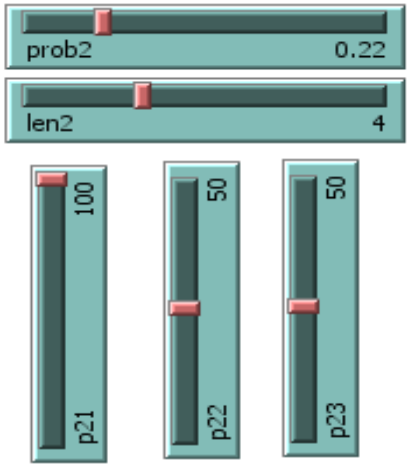
iii) the rest of numbers: is the set of illnesses affecting patients in that specific calendar month within that year.

The related code procedure is now presented:

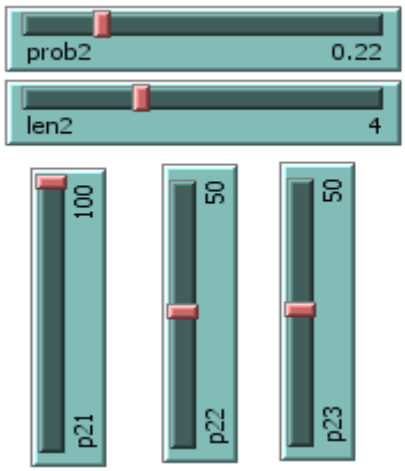
```
to giveIllness
ask patients [
set recipes []
```



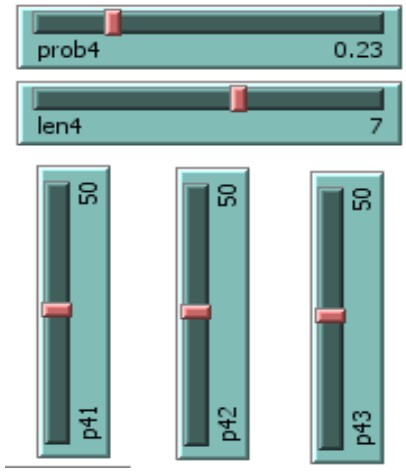
(a) first age class probabilities



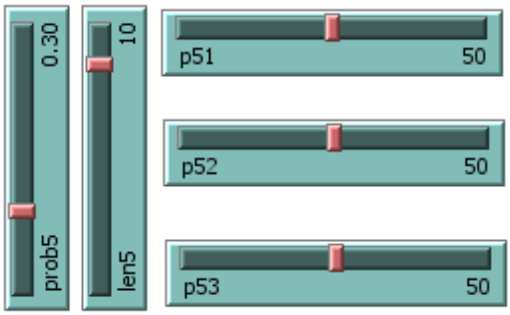
(b) second age class probabilities



(c) third age class probabilities



(d) fourth age class probabilities



(e) fifth age class probabilities

Figure 2.21: Set of sliders referred to all the five age classes of patients, respectively: a) for class one, b) for the second and so on up to the fifth class.. We briefly explain the label of each slider managing the first age class, since the interpretation for the other four classes is exactly the same. "prob1" rules the probability of getting ill. If this is the case, "len1" reveals the number of diseases affecting patients in each year they are ill, while "p11", "p12", "p13" the probabilities that each patient of such age class will be hit

```
let i age
while [i < 100] [

let class age-category i

if (random-float 1 < item class probAge)[
let year []
set year lput i year
set year lput ((random 12) + 1) year
let j 0
let howManyTreatments (item class len)
let catTreat extract-probability (item class probIllness)

while [j <= howManyTreatments] [
set year lput (one-of item catTreat treatments) year
set j (j + 1)
]
set recipes lput year recipes
]

set i (i + 1)
]
]
end
```

As a graphical summary of what we have said up to now we present Figure2.22 in which a generic patient is reported. The main features distinguishing such agent-set from the others are: i) “gender”, which can be either male or female, ii) “age”, which varies between 0 and 100, and iii) “recipes”, which reports the all life long health-care situation of each patient.

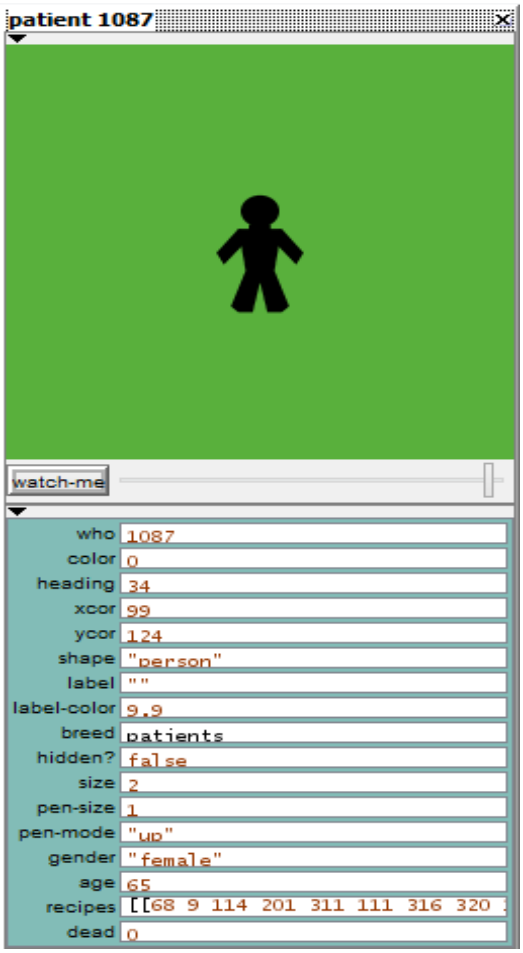


Figure 2.22: Generic Patient inspection. Emerging peculiar features, patients-related only: i) gender, ii) age, and iii) recipes.

A further step can be made. If in the lower part of the interface (the white part in which the user can insert commands to the program as an external observer) we write down this string of codes "ask patient 1087 [show recipes]" the program will give us the all life long health-care story of patient 1087, in other words all its future diseases. The output will be of the form:

```
observer> ask patient 1087 [show recipes]
(patient 1087):
  [[68 9 114 201 311 111 316 320 102 220 207 306 110]
  [73 4 111 201 214 212 302 314 307 311 111 201 315]
```

```
[77 2 302 108 220 214 218 212 303 301 316 318 115]
[79 11 209 314 117 318 314 120 211 101 310 216 320]
[95 3 307 107 207 310 303 312 104 104 217 115 110]
[99 11 312 206 111 203 118 301 306 308 208 313 209]]
```

So, it represents the health-care story of a woman who will be sick when sixty-height years old first, then when she will be seventy-three, seventy-seven, seventy-nine, ninety-five and finally when ninety-nine.

2.4 The Rules

In the two previous sections we have analysed all the Agents and the Environment within which they act and interact. This section is devoted to those commands, the Rules, driving such actions and interactions. We will follow the same procedure adopted up to now: first of all we will explain the arguments from a theoretical point of view, sometimes providing graphical explanations when they are needed. Then we will plot the related parts of code so that an hypothetical external user, if not satisfied by what we have done, is able to improve quickly the part at issue. In particular we will refer to all those commands which start functioning when the user press the button "go" on the interface, represented in Figure2.23 (if, instead, he presses the "setup" button he will cause the creation of the Environment and that of the Agents, previously explained).

Figure 2.23: These buttons should be pressed by the program-user in order to provoke certain outputs. In particular, when the "setup" button is pressed the Environment, together with the Agents it contains, emerge. When, instead, the "go" button is pushed, the actions between them start happening.

We now consider the more important procedures driving such interactions:

2.4.1 To getCured

This time we will first of all present the code associated to the procedure, and then we will proceed to the explanation. The coding lines are shown below:

```
to getCured
ask patients with [not hidden?
and length (filter [first ? = age] recipes) = 1
and (item 1 first (filter [first ? = age] recipes)
= currentMonth)] [let treats but-first but-first
(first (filter [first ? = age] recipes))
let choice floor (first treats / 100)
ifelse (choice = 3) [
let doc choose-er
if (see-er doc treats = 0)[hide-turtle set dead 1]
][
let doc choose-family
if (see-family doc treats = 0) [hide-turtle set dead 1]
]
]

end
```

The procedure works as follows: it checks the first service composing the set of diseases hitting the generic patient in the year at issue. This checking mechanism operates in this way: the code identifying the first illness the patient is affected by is divided by one-hundred and then it is rounded off to the closer integer. Two different situations may occur: i) if the resulting

number is three, then the sick individual will be nursed in an emergency room. ii) If instead the output of the calculus is one or two, then the patient will move to a Familydoc (a rare case, but possible, is that in which there are no hospitals providing cure for a particular illness. If this is the case, those who are affected by such "incurable disease" will die in the year they get sick). We now present in detail both the situation just mentioned.

i) The patient will be firstly nursed at the emergency room: in particular, we have assumed that *each patient will receive cures in the closer er,* as we can notice from the code managing such situation:

```
to-report choose-er
report min-one-of ers [distance myself]
end
```

So, once the program has pinpointed the target er, then it traces a green link (we refer to the following section to a well-detailed analysis of the links, which have to be considered as "Agents") whose metaphorical head and tail are respectively the patient and the er the sick individual goes to. The set of links connecting patients with ers (we will see later on that this rule has been applied to all sorts of links) are generated one time only. The part of code we are referring to consists in the last lines of the procedure shown below, in particular those beginning with "set process remove-duplicates".

```
to-report see-er[doc treats]
rush doc
let flag 1
ask doc [
let lab1 one-of (specialists with
[affiliation = [affiliation] of myself and member?
```

```
101 specialty]) if (lab1 = nobody) [set lab1 min-one-of
(specialists with [member? 101 specialty])
[distance myself] ] let lab2 one-of (specialists with
[affiliation = [affiliation] of myself and member?
102 specialty]) if (lab2 = nobody) [set lab2 min-one-of
(specialists with
[member? 102 specialty])[distance myself]]

if (lab1 != nobody) [
take-tests lab1
if (lab1 != lab2) [ask lab1 [take-tests lab2]]
]
let process []
foreach treats [
let targetDoc one-of specialists with
[affiliation = [affiliation]
of myself and member? ? specialty]
if targetDoc = nobody [set targetDoc min-one-of
(specialists with [specialty = ?])[distance myself] ]
set process lput targetDoc process
]
set process remove-duplicates process
ifelse not(member? nobody process) [
take-visits-to first process
while [length process > 1] [
ask last process [take-visits-from
last (but-last process)]
set process but-last process
]
][set flag 0]
]
```

```
report flag
end
```

Another important feature concerning such procedure is that it considers separately the blood tests (sampling and analysis, codes 101 and 102) and the other services (generally called "treats"). However, even if these two process are managed separately, the logic behind them is the same. If the health-care treatment required by the ill patient is not provided by the er, either it is a blood taking sample, or a blood analysis or a generic health-care service, then the patient will be moved to the *closer Specialist* providing such cure. So we can briefly sum up such rule.

1) *The program picks up the first illness (which takes the form of a three figures code) hitting the agent at a specific month in a specific year.*

2) *It divides such number by one-hundred and then it rounds the output off to the closer integer.*

3) *If the resulting number is three, then the sick individual moves to the closer er.*

4) *If the er provides all the other illnesses hitting the agent, then it will stay. If instead it is affected by something the er is not able to nurse, then it will be carried out to the closer Specialist offering such particular cure.*

ii) The patient will move to Familydoc: if this is the case, then the running procedure is the following.

```
to-report choose-family
let area pcolor
report min-one-of (familydocs with [pcolor = area] )
```

```
[distance myself]
end
```

From this procedure it emerges that the criterion leading the patient choice of the Familydoc is a "Proximity" one. We now present the two main assumptions we have undertaken to built up such ruling-mechanism and then we present the associated code procedure.

1) *Each patient is considered an agent characterized by few information. However, all the patients are endowed with three different types (important remark: the weights of such variables will be the same for all patients) of "personal" preferences:*

i) Quality

ii) Distance

iii) Waiting list

These three typologies of preferences constitute the "core business" of our work. Indeed, according to the weights the external user attributes to each of them, he can test different situations (we will test such emergent scenarios in the section devoted to the experiments our program allow the user to develop). The user can set the desired values by simply modifying the three sliders we have inserted on the top left of the interface (whose weights range from zero to one). They have been reported in Figure2.24. We will have a better understanding of such weights when considering the role played by the Familydoc.

2) *We have intended the Familydoc as a rational omniscient agent. In particular he is the agent in charge of deciding which health-care institutions (whether public Specialists or private Professionals) the patients have to be sent to. The reason why we consider each Familydoc as "rational" is that they all develop the following reasoning, including the execution of a numerical calculus*:

Figure 2.24: Sliders regulating the weights of patients' personal preferences towards the quality, distance, and waiting list characterizing all the agents providing alleged health-care cures against their illnesses.

i) They first of all observe patients' preferences. They all know the exact values inserted by the user by means of the three sliders previously mentioned; indeed, they have been built up as omniscient agents.

ii) Then, they execute the following calculus:

```
[dist * (distance myself) + waitlist *  queue
+ cost - quality * rep]
```

iii) Finally, for each year they will create all the links (which show the path patients have to follow to get cured) to each health-care structure minimizing the calculus at issue. An example of the graphical output of link creation is presented in Figure2.25.

3) *In order to endow our work with a far greater level of reality, we have allowed for the possibility that patients will not follow what Familydocs decide. Indeed, in the real life, those individuals which have been visited by family doctors are then told what to do, but they are not forced to do it. They can decide whatever they want: whether going private or public, whether being nursed or not. So, we have introduced the possibility to manage this situation too. To do, we have inserted another slider on the interface, which has*

been represented in Figure2.26. The related code procedure is the following:

```
foreach treats [
let targetDoc nobody
ifelse (random-float 1 < stubborness)
[set targetDoc one-of structures with
    [member? ? specialty] ]
[set targetDoc min-one-of (structures with
[member? ? specialty])
[dist * (distance myself)
 + waitlist *  queue + cost - quality * rep] ]
```

It works as follows: the user sets the slider called "stubborness" on the desired value. Then, the program extract a floating random number within the interval [0,1]. If this value is lower then that set by the user, then patients will move randomly ([set targetDoc one-of structures with [member? ? specialty]]). If, instead, the user-chosen value is greater than that randomly extracted, then patients will follow what Familydocs have decided at their place ([set targetDoc min-one-of (structures with [member? ? specialty]), according to the minimization problem previously explained. The limit cases are the following values of the slider: 0, meaning that the Familydoc's decision is the only alleged possibility, while a value of 1 means that the Familydoc is no longer playing his decision role, and patients are moving completely random.

The whole code procedure related to the case in which the first illness hitting the patient brings him to a Familydoc is now presented:

```
to-report see-family[doc treats]
```

```
go-check doc
let flag 1
ask doc[
let structures turtles with
[breed = professionals or breed = specialists]
let lab1 min-one-of
(structures with [member? 101 specialty])
[dist * (distance myself) + waitlist *  queue
+ cost - quality * rep]

let lab2 min-one-of (structures with [member? 102 specialty])
[dist * (distance myself) + waitlist *  queue
+ cost - quality * rep]

if (lab1 != nobody) [
take-tests lab1
if (lab1 != lab2) [ask lab1 [take-tests lab2]]
]
let process []
foreach treats [
let targetDoc nobody
ifelse (random-float 1 < stubborness)
[set targetDoc one-of structures with [member? ? specialty] ]
[set targetDoc min-one-of (structures with [member? ? specialty])
    [dist * (distance myself) + waitlist *  queue
    + cost - quality * rep] ]

set process lput targetDoc process
]
set process remove-duplicates process
ifelse not (member? nobody process) [
```

```
take-visits-to first process
while [length process > 1] [
ask last process [take-visits-from last (but-last process)]
set process but-last process
]
][set flag 0]
]
report flag
end
```

2.4.2 Time framework

In this short paragraph we will analyse the program we have developed from a temporal standpoint. In particular, we firstly will consider the procedure which links the evolution of the program with the passing of time, and we have called it "To timeGoesBy". Then we will face the mechanism trough which patients get older and then "naturally" die, and we have labelled it as "To getOlder". We now begin with the first we have just mentioned.

To timeGoesBy

In this part of the code, we have linked the passage of time with the program peculiarity of being capable of evolution. In other words, we have associated to each movement of the program, which is called "tick", the passage of one month. So every twelve ticks passed away correspond to one year. So, as an example, given a patient which has been created at the age of five, after twelve ticks are gone away we expect that the patient will be six years old (if he is not dead because of a "rare disease", those for which no hospitals provide a cure). The related code is now presented.

```
to timeGoesBy
set currentMonth currentMonth + 1
if (currentMonth = 13) [getOlder set currentMonth 1
```

```
set currentYear currentYear + 1]
end
```

We now introduce an example to better understand such process. We begin with the inspection of a generic patient, focusing on his age, since it is the only time element patient are characterized by. Figure2.27 represents the starting condition. We have highlighted with red circles the elements we want to focus our attention on. The patient, his age, the ticks-counter. Since we have previously said that each tick corresponds to the passage of one month, we expect that after at most twelve ticks are passed away the patient will be one year older (if he is not death). Figure2.28 shows what happened after twelve ticks are gone. The patient is effectively one year older.

To getOlder

This code procedure represents the mechanism trough which patients get older and die. The Turin district is characterized by the following values of life expectancy at birth: 79.6 and 84.7 for, respectively, men and women (Istat 2010). Despite such real values, we have made a strong assumption: *since we have the real proportions of all the population, from zero up to more then one-hundred years, we have decided to represent a set of agents passing away for natural causes (another already mentioned death cause is the case in which there is no health-care institution providing cures for the illness the patient at issue is affected by) at the age of one-hundred.* Indeed, our goal is to built up a reasoning instrument which can be used to test the emergence of particular types of networks among particular types of agents. So, the age at which patients can at most die does not have negative effects on the result we are pursuing. We now report the code procedure we are referring to:

```
to getOlder

ask patients with [age = 100][hide-turtle set dead 1]
ask patients with [not hidden?] [set age age + 1]
end
```

2.5 Links

Links consist in a particular type of Agents. However, we have decided to postpone their explanation up to now since they start coming into view as the program goes by. In other words, they start emerging after the user has pressed the "go" button, when indeed the rules managing the agents' interactions are taking place. Even if we have built up four different types of links, they all share some characteristic features:

i) They identify a connection between Agents of the same type (with the exception of Patient type) or between different types Agents too.

ii) Every link is characterized by the feature "visits". It counts the number of transits along the same link. We have managed them so that the link is built up only the first time a transit takes place, and then every other passage increases that counter of one unit.

iii) Every link is univocal pinpointed by a head and a tail (respectively labelled as "end1" and "end2"). Links can depart from:

i) Patients

ii) Familydocs

iii) Specialists

iv) Professionals

v) Ers

Differently, links can only reach:

i) Familydocs

ii) Specialists

iii) Professionals

iv) Ers

Indeed, *since we are considering the movements towards agents which can provide cures to particular illnesses, it would be absurd a movement patient-to-patient.* Moreover, depending on the type of cures they stand for, they will be characterized by different colours. We now consider all the types of links we have introduced in our work.

2.5.1 Link typologies

Green links

Green links are those heading to an "Er". We have labelled them as "emergency". Figure2.29 shows the main elements of the generic agent belonging to this category, respectively: i) the starting point, in this case "Patient 1426". ii) The destination, "emergency room 2850". Finally, iii) the number of transits between such extremes, in this case one only.

The related code procedure is now reported.

```
to rush[doc]

ifelse (out-emergency-neighbor? doc)
[ask out-emergency-to doc [set visits visits + 1]]
[create-emergency-to doc [set visits 1 set color lime]]
```

Pink links

The logic behind this agent construction is the the same than that for the green set of links named "emergencies". The only difference is the agent-set they are directed to, i.e Familydocs. As an example, see Figure2.30. The code procedure creating such agents and attributing them a greater value of "visits" each time a passage takes place is:

```
to go-check[doc]

ifelse (out-checkup-neighbor? doc)
```

```
[ask out-checkup-to doc [set visits visits + 1]]
[create-checkup-to doc [set visits 1 set color pink]]
```

Since the logic is absolutely the same for the two remaining links categories, we just present their name, the code procedure and the associated graphical representation, highlighting on it the particular agent-set they are pointing to.

Blue links

Blue links, which have been called "Specialistics", can be either directed towards the agent-set "Specialists" or to "Professionals". Indeed, such link type stands for all specialistic cure, which can be provided by both those two Agent-sets. Figure2.31 represents a generic specialistic-type link, standing for a patient movement from a Familydoc to a Specialist, while Figure2.32 represents the case of a blue link directed to a Professional.

```
to take-visits-to[doc]

ifelse (out-specialistic-neighbor? doc)
[ask out-specialistic-to doc [set visits visits + 1]]
[create-specialistic-to doc [set visits 1 set color blue]]
```

and

```
to take-visits-from[doc]

if not (doc = myself)[
ifelse (in-specialistic-neighbor? doc)
[ask in-specialistic-from doc [set visits visits + 1]]
[create-specialistic-from doc [set visits 1 set color blue]]
]
end
```

Magenta Links

This last link-category consists in those standing for all laboratory tests. Indeed, they can be both offered by Professionals and Specialists. Figure2.33 depicts a generic "test" pointing to a Professional, while Figure**??** one pointing to a Specialist. "Test" is the name we have them attributed. In what follows we have reported the associated code procedure.

```
to take-tests[lab]

ifelse (out-test-neighbor? lab)
[ask out-test-to lab [set visits visits + 1]]
[create-test-to lab [set visits 1 set color magenta]]
end
```

2.5.2 Links management

As for the other Agents, whoever is using our work can test particular emergent phenomena given a set of modifiable conditions. We now refer to those trough which a generic user can have a graphical influence on the emergent links. Since our goal is to test particular phenomena, with particular attention to the type of emerging networks (IF they emerge), we have introduced an input labelled as "Threshold". It works as follows: the user inserts the desired value, and then all the links endowed with a number of visits lower than that inserted in the input will disappear. Three different possibilities (composing the chooser called "underThreshold") of links disappearance can be used:

1) Manual delete: links are eliminated after the button "prune" (which is shown in Figure2.35) is pushed.

2) Automatic delete: the necessary condition that has to be satisfied so that this elimination takes place is that the input labelled as "residualCycles" (which has been represented in Figure2.37) must be filled by

a number greater than zero. It says how many cycles the program is going to run in addition to the current one. If this is the case, the elimination of the links with a number of visits lower than that inserted by the user will take place before a new cycle is run (a new cycle begins when all those patients acting in the previous cycle are dead). The code procedure ruling such possibility is now presented:

```
to checkPopulation

if (all? patients [hidden?]) and (residualCycles > 0)
[
if (underThreshold = "automaticDelete")[ask links with
[visits < (threshold * cycle)][die]]
ask patients [show-turtle set dead 0]
giveAge
giveIllness
set cycle cycle + 1
set residualCycles (residualCycles - 1)
]
end
```

3) Manual hide: in this case the links disappear but not definitively. Indeed, they are hidden and so the user is able to make them appear again by means of the following command, which have to be inserted in the Command Center at the bottom of the NetLogo interface. The string of code we are referring to is "ask patients[set hidden? false]. The code procedure associated to manual hide and manual delete is now shown:

```
to modLinks

ask links [show-link]
```

```
if (underThreshold = "manualHide")
[ask links with [visits < threshold][hide-link]]
if (underThreshold = "manualDelete")
[ask links with [visits < threshold][die]]
end
```

All these possibilities are shown in Figure2.36.

2.5.3 Networks

As we have already said, one of the main goals we want to reach by means of our program is to provide help in understanding the emergence of networks between Agents, if they emerge. In particular, it is useful to study such connections among Specialist and among Professionals. The way trough which we could show them is the following:

1) *Press the button "setup"*

2) *Press the button "go"*

3) *Once the program is finished, i.e once all the patients are dead and there are no further possible movements towards institutions providing health-care cures, we can start analysing networks. To see them emerging we have to press either the button "Specialist Network" or that labelled as "Lab test Network" (which have been shown in Figure2.38. They both can be found on the program interface).*

As we can see there are two different types of networks we can call for:

i) one whose connections among Agents are links standing for cures of the second type, specialist services (and so, blue links), and

ii) one which is composed by links standing for first type cures only, i.e laboratory based tests (magenta).

Further assumptions:

i) *we have opted for a circular layout, meaning that once the network buttons are pressed the Agents at issue will be arranged in such a way.*

ii) *The Agent-sets which can compose the edges of both the networks are the following: Familydocs Specialists Ers and Professionals.*

iii) *They will be ordered anticlockwise, according to a greater value of the feature labelled as "betweenness", owned by all of these Agent-sets.*

The string of codes managing such networks creation are:

1) Specialistic-coding-network:

```
to spNetwork
ask links [hide-link]
let structures (turtle-set familydocs
    specialists ers professionals)
let connections (link-set specialistics)
ask connections [show-link]
nw:set-context structures connections
ask structures [set betw nw:betweenness-centrality
set eigvec nw:eigenvector-centrality
set weighted nw:closeness-centrality]
layout-circle sort-on [betw] structures with [betw != 0] 65
end
```

2) Test-coding-network:

```
to testsNetwork
ask links [hide-link]
let structures (turtle-set
specialists ers professionals familydocs)
let connections (link-set tests)
ask connections [show-link]
```

```
nw:set-context structures connections
ask structures [set betw nw:betweenness-centrality
set eigvec nw:eigenvector-centrality
set weighted nw:closeness-centrality]
layout-circle sort-on [betw] structures with [betw != 0] 65
end
```

So, as we can see, an important argument we are dealing with in such section is that of "Betweenness". In particular, using the words of Wikipedia (2014):

> *Betweenness centrality is an indicator of a node's centrality in a network. It is equal to the number of shortest paths from all vertices to all others that pass through that node. A node with high betweenness centrality has large influence to the transfer of items through the network, under the assumption that transfer follows shortest paths. The concept finds wide application, including computer and social networks, biology, transport and scientific cooperation.*

So, wide reasoning can then be made upon those Agents showing different levels of betweenness (i.e from a policy making perspective, it can be reasonable to shut down an hospital characterized by a really low value of betweenness instead of one characterized by a greater level or, instead, if such institution fulfils an important social role, a low value could imply a great investment of resources to improve the role played.)

We now show the graphical consequences which can be obtained by either pressing the "Specialist Network" button or the"Lab test Network" one. Figure2.39 shows the situation in which a generic simulation has finished all the commands it has been asked for: all patients are dead, and all their movements are reported on the screen. If, at this point, we press the "Specialist Network" button, we will observe a situation like that shown in Figure2.40. Agents processing blue links (i.e Professional and Specialists) are arranged

in a circular way. As we have previously said, they are ordered anticlockwise, according to a greater value of "betweenness". We have highlighted with a red circle those agents showing the highest and the lowest value, respectively the left and the right Professionals. The level of such variable they are characterized by are then shown in Figure2.41. Finally, Figure2.42 depicts the case in which, moving from the same starting condition, the user has pressed the "Lab test Network" Button.

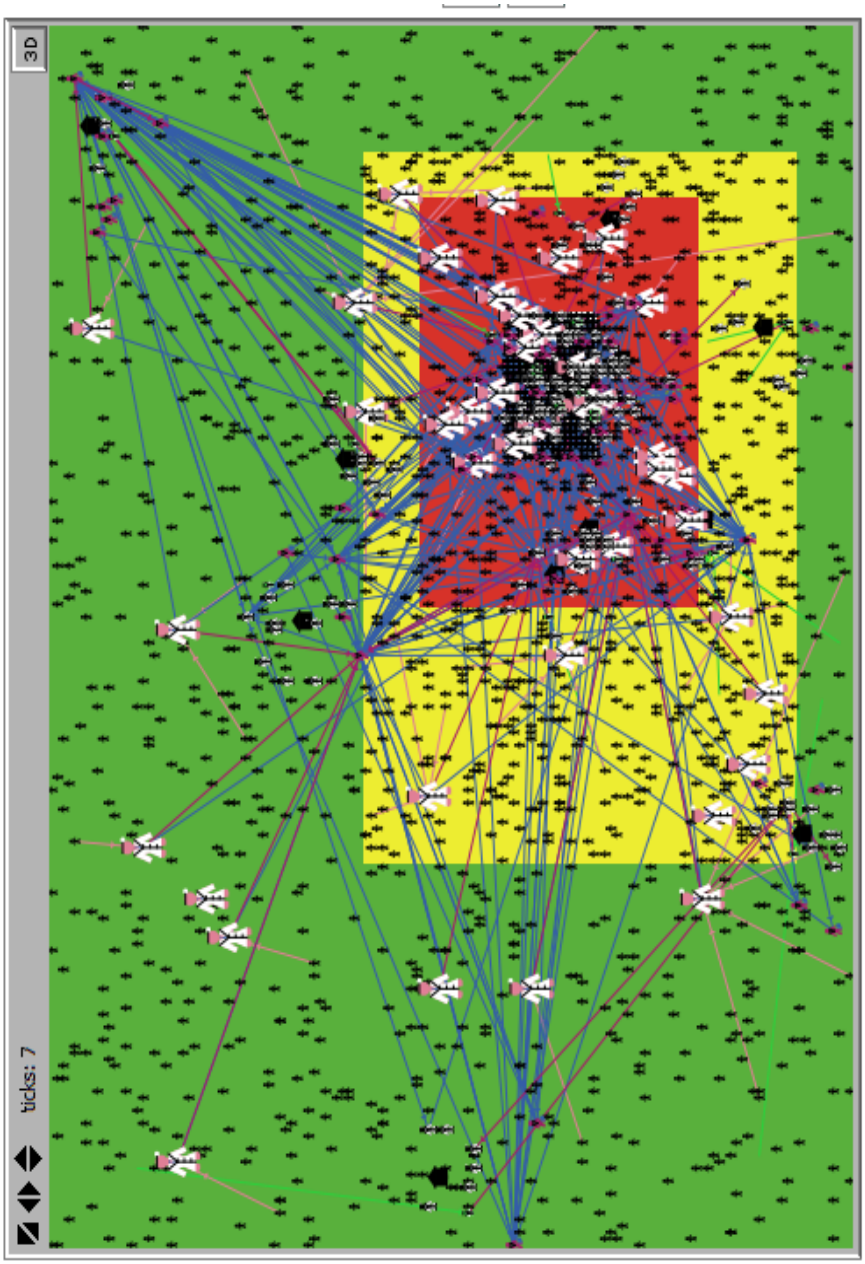


Figure 2.25: Simulated world representing the consequences of Familydocs actions: creation of different sets of links each of them driving patients to different health-care institutions according to the minimization problem explained in the text. Familydocs dimensions have been increased to allow a better identification on the screen.

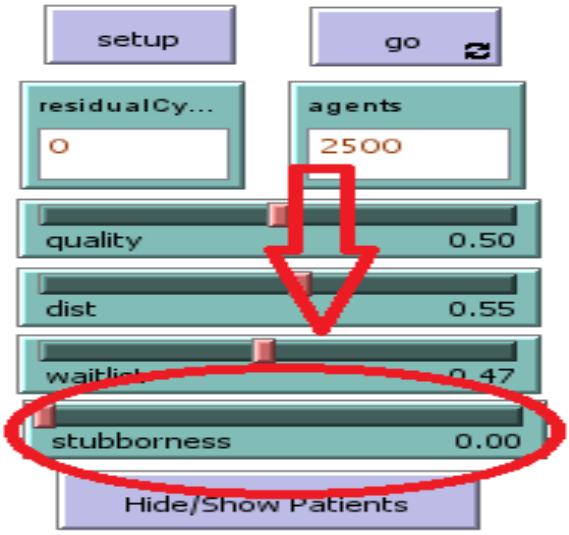


Figure 2.26: Sliders regulating the patients' stubbornness.

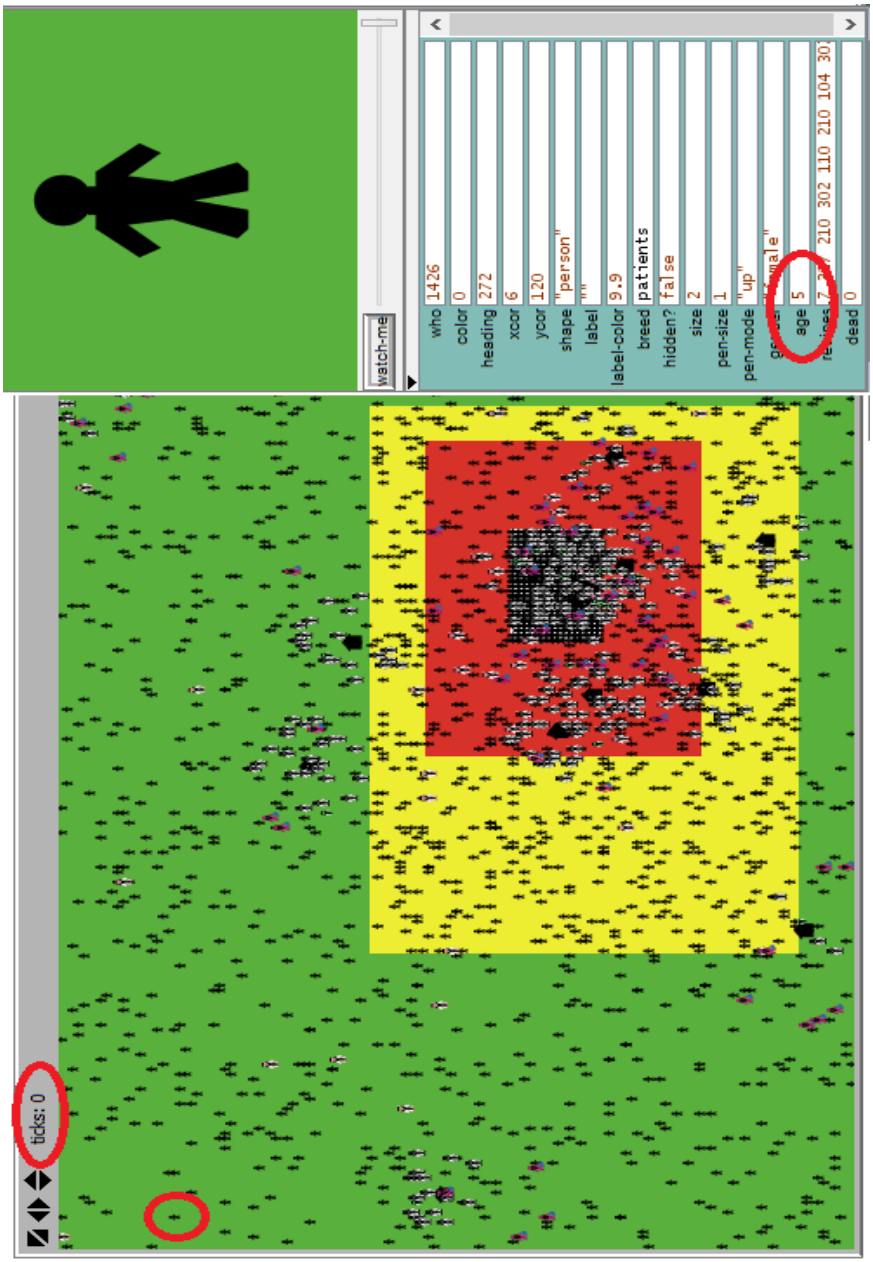


Figure 2.27: The beginning of the example: ticks 0, age 5.

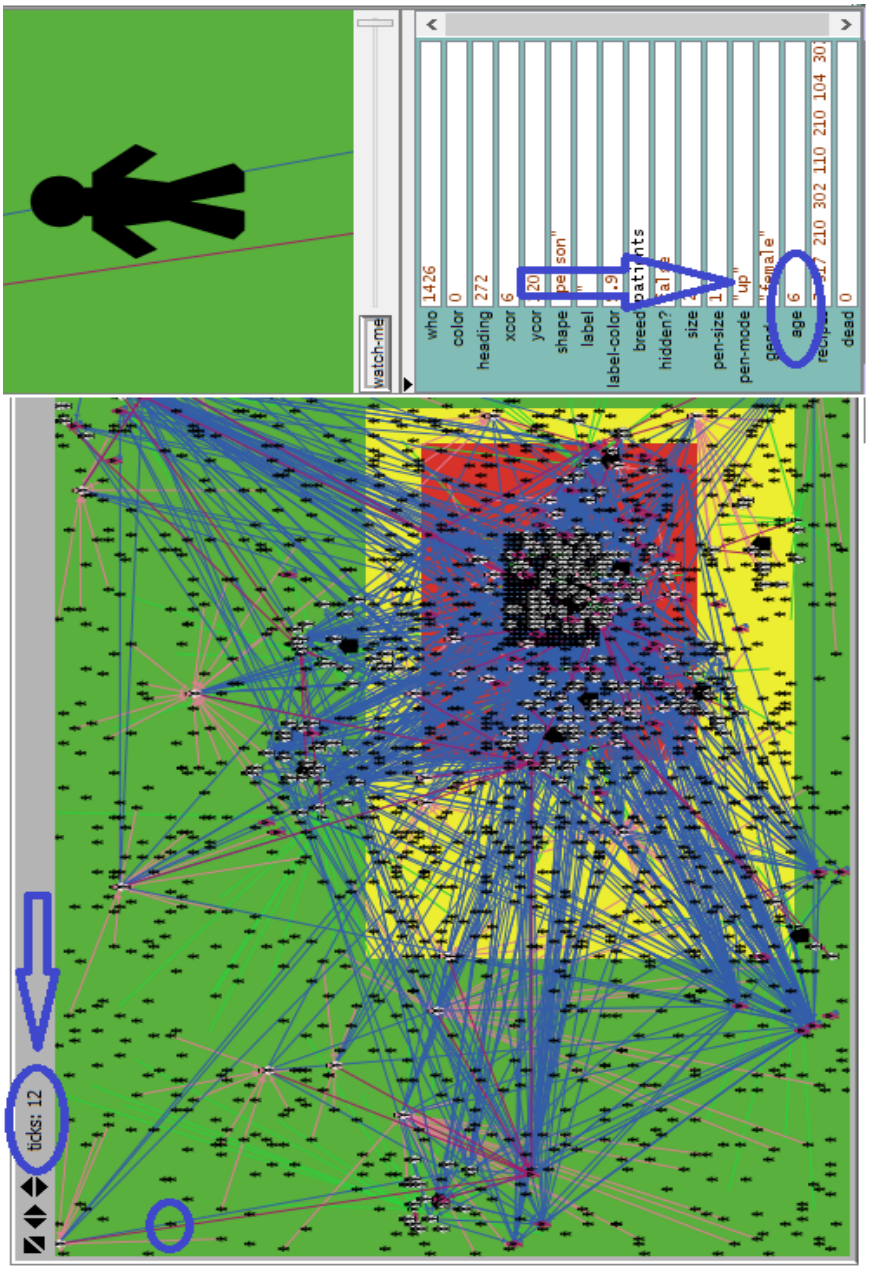


Figure 2.28: ..after twelve ticks are passed away, the patient is one year older.

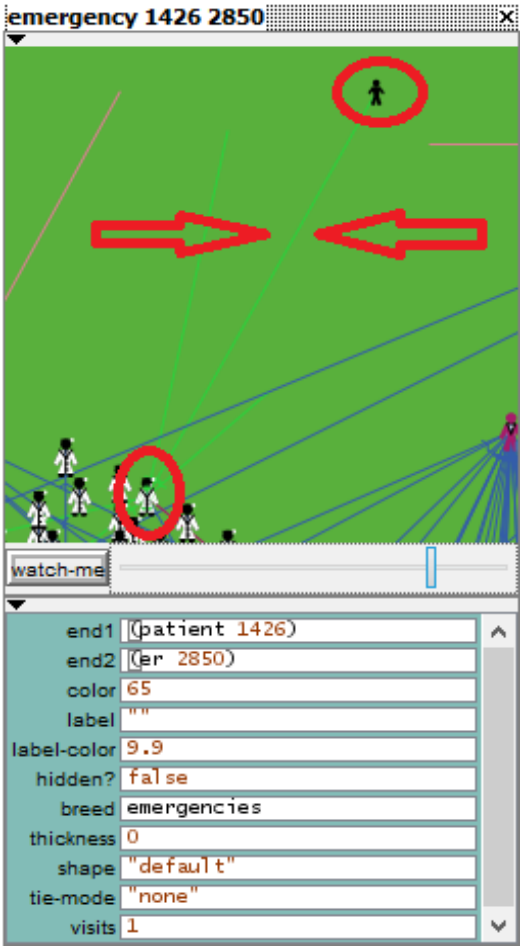


Figure 2.29: Generic “emergency” inspection.

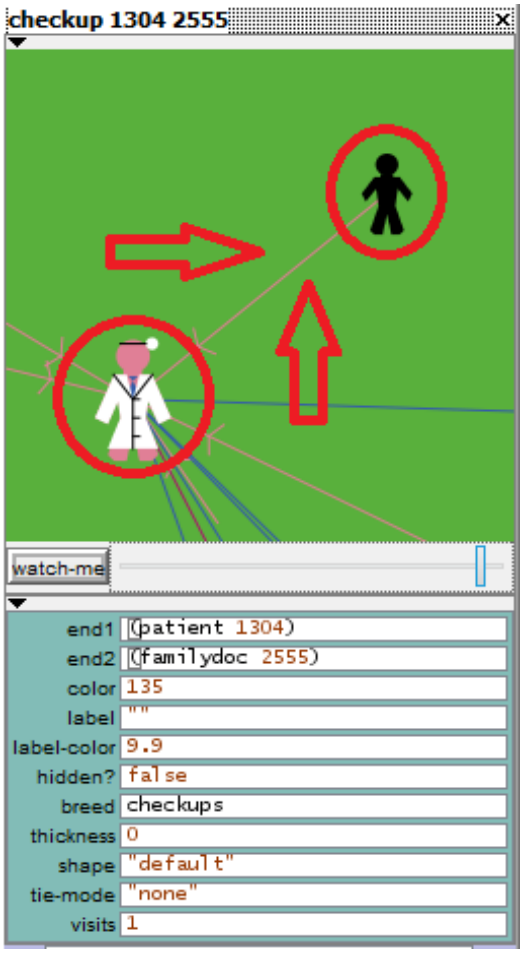


Figure 2.30: Generic “checkup” inspection.

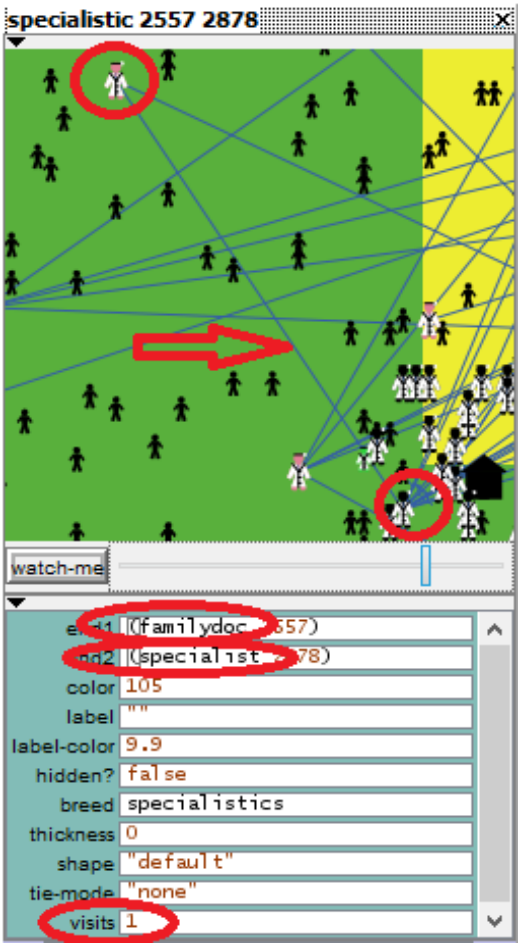


Figure 2.31: Generic “specialistic” inspection.

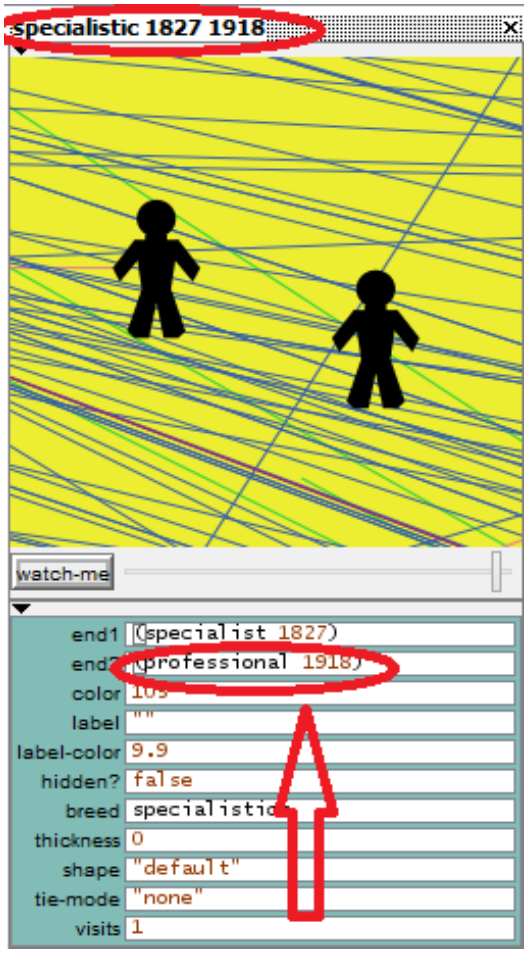


Figure 2.32: Generic “specialistic” inspection.

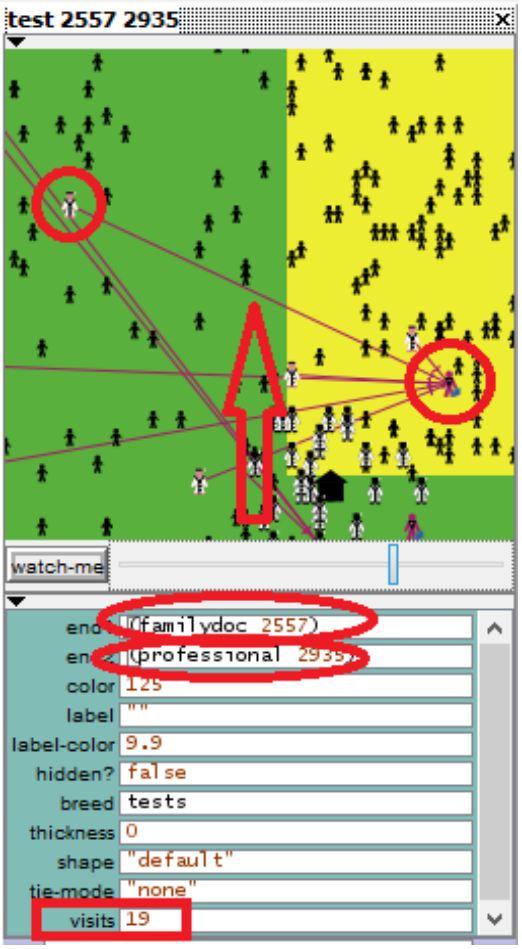


Figure 2.33: Generic “test” inspection.

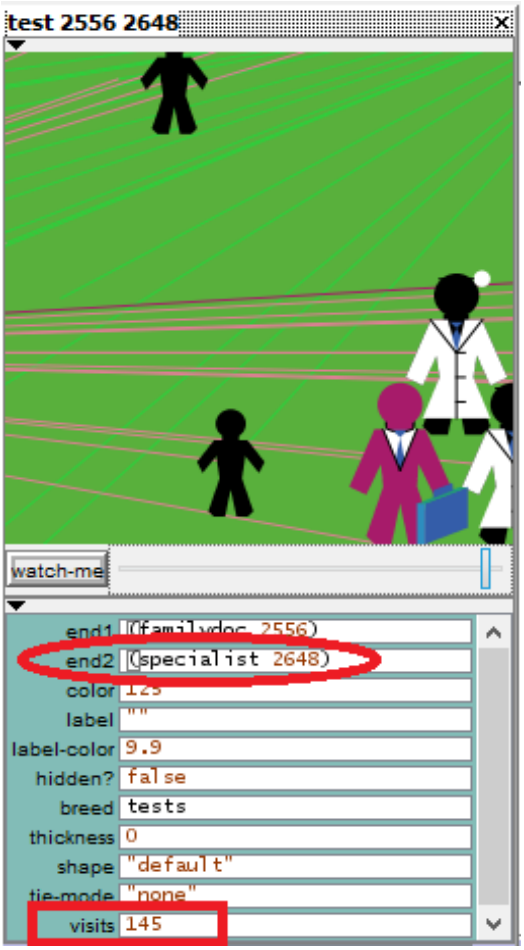


Figure 2.34: Generic "test" inspection.

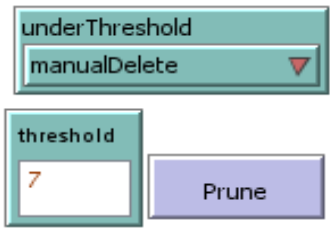


Figure 2.35: Anticlockwise, starting from the top, we can see the chooser "underThreshold", the input "Threshold" and the button "Prune".

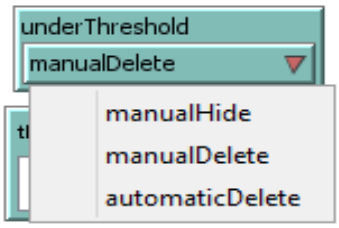


Figure 2.36: Different possibilities the user can choose between while managing such chooser.

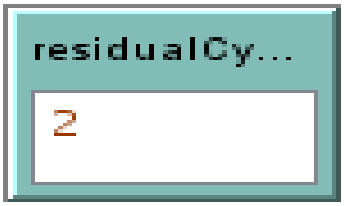


Figure 2.37: The input residualCycles shows the number of cycles the program is going to run, in addition to that in progress.

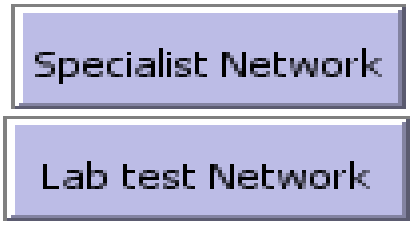


Figure 2.38: “Specialist Network” and “Lab test Network” Buttons, managing the networks creation.

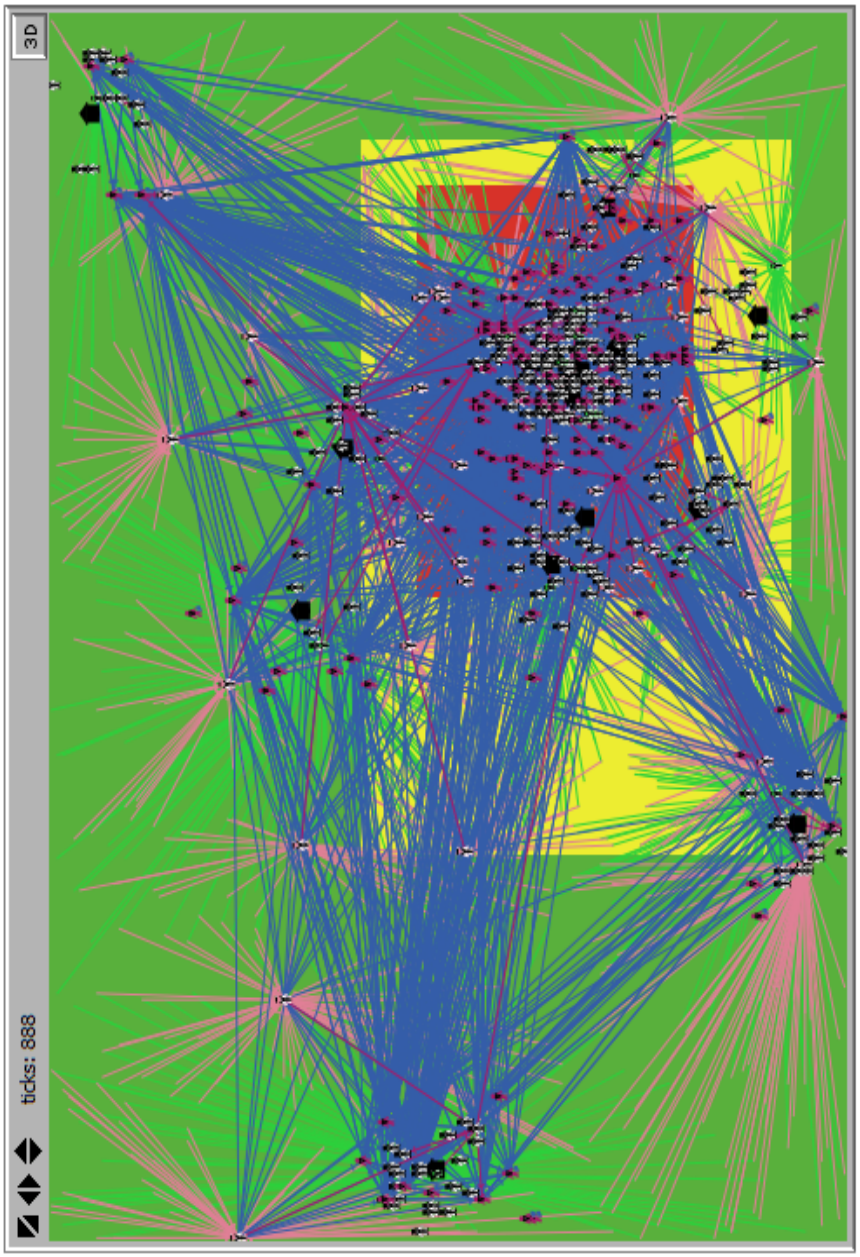


Figure 2.39: A generic simulation representation, after it has finished all the commands it has been asked for: patients are dead and all their movements are reported on the screen.

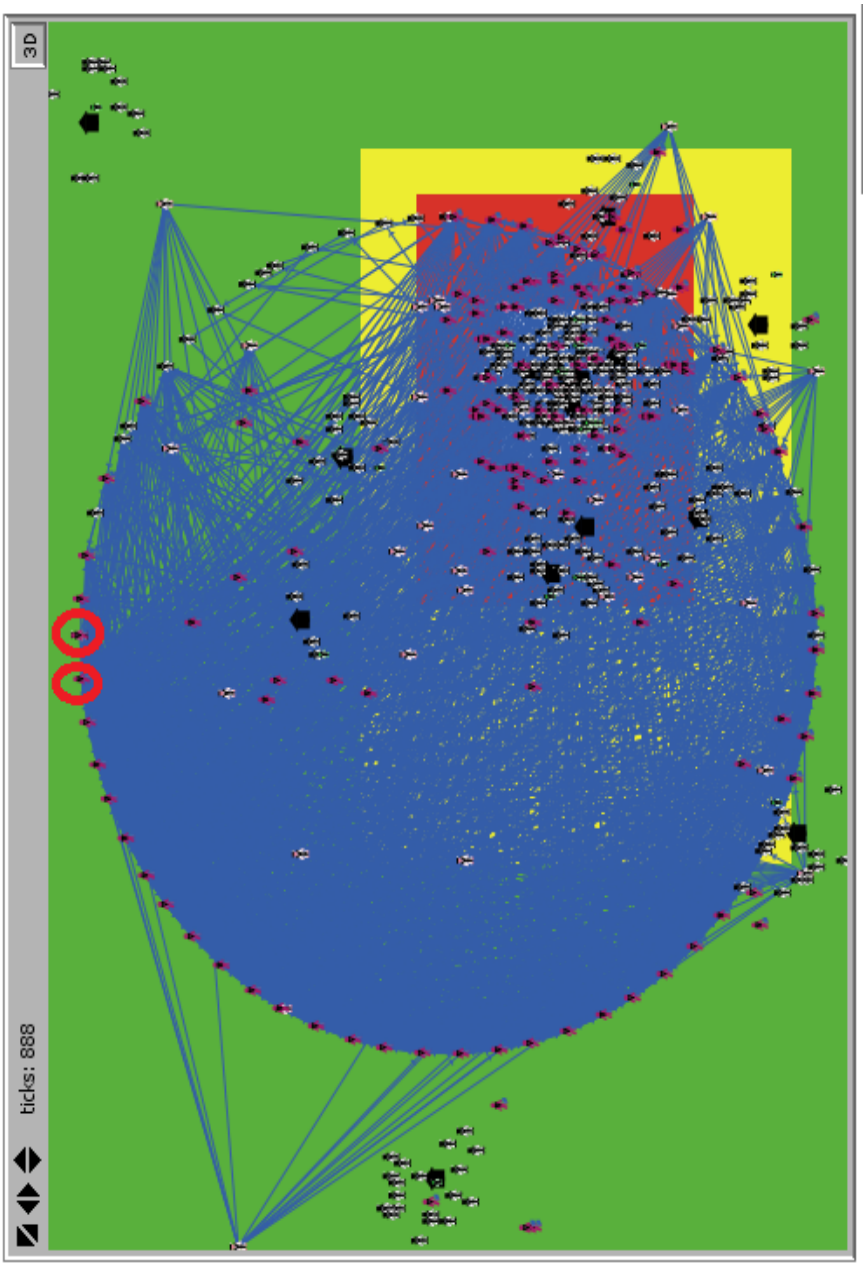


Figure 2.40: An alleged possible output we can obtain moving from Figure2.39 by pushing the button “Specialistic Network”.

(a) lower-level betweenness Agent (b) higher-level betweenness Agent

Figure 2.41: Inspection of the level of betweenness Agents are characterized by.

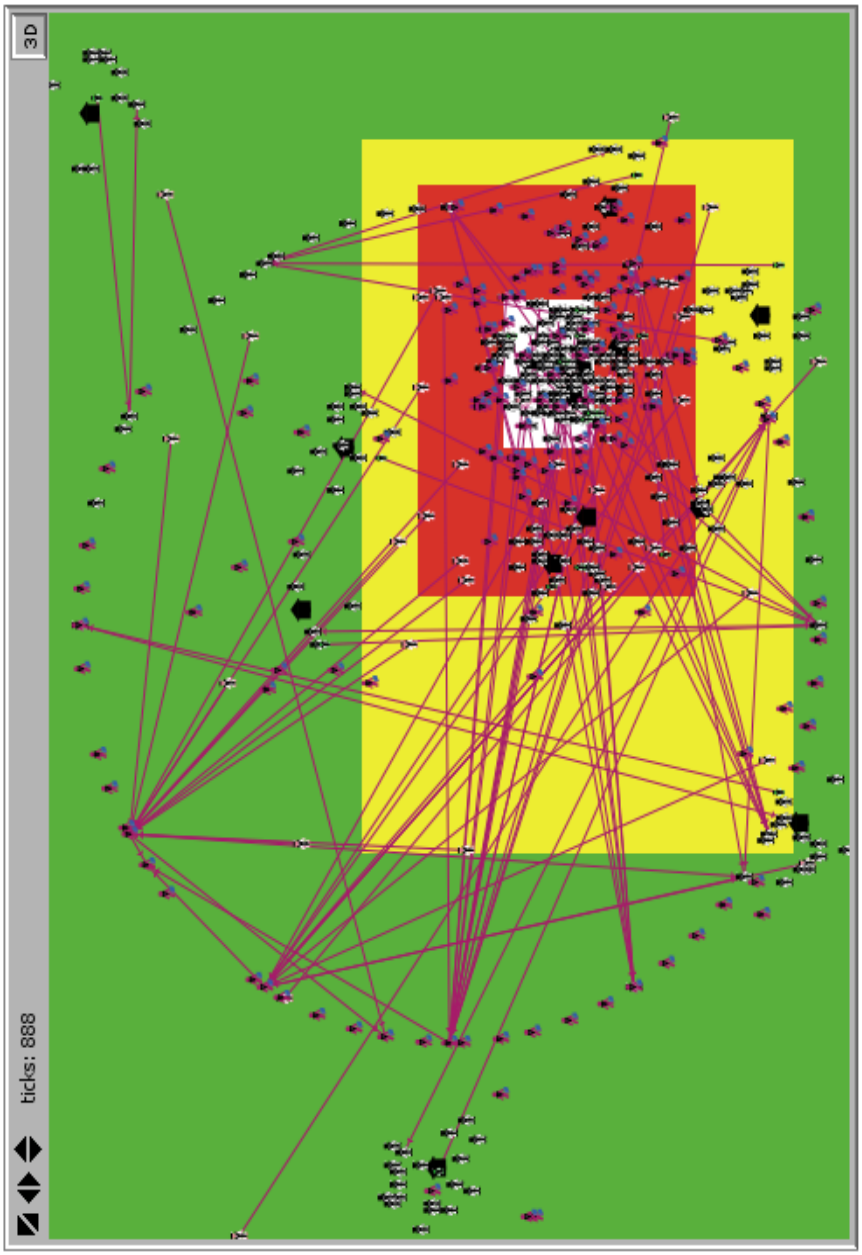


Figure 2.42: An alleged possible output we can obtain moving from Figure2.39 by pushing the button “Lab Test Network”.

2.6 Experiments

This section is devoted to a set of experiments we can test by means of our "reasoning tool". The way we will proceed is the following: we first of all will explain the type of situation we want to analyse. We then consider a set of assumptions we need to undertake, graphically observable in specific values we have given to the sliders at issue each time. Finally we will consider the graphical output and trying to explain the emergent phenomena, if there are.

We think our experiments turn out to be useful only if they are reproducible, i.e if anyone can try them themselves and get the same result that we got (moving from the same starting point). We should now open a brief parenthesis. In particular, we should refer to the process standing behind the so called "random numbers" NetLogo is able to generate. Using a computer programming name, we can call them "pseudo-random" number: they appear random, but they are de facto generated by a deterministic process. "Deterministic" means that you get the same results every time, if you start with the same random "seed". In particular, in the NetLogo dictionary we can read the following:

> NetLogo's random number generator can be started with a certain seed value, which must be an integer in the range -2147483648 to 2147483647. Once the generator has been "seeded" with the random-seed command, it always generates the same sequence of random numbers from then on.

Referred to this point, we have inserted in the interface a switcher who allows the user to activate this possibility. Figure2.43 shows it.

Since we want to focus on the networks among all the agents providing healthcare services, we will consider three different measures of Centrality provided by NetLogo, explained below. To each of them, we have attached the definition provided by NetLogo.

1) Betweenness-centrality: in NetLogo Network Extension website we can read the following explanation about such measure.

Figure 2.43: Seed switcher: when the on-mode is activated the user will get the same results by moving from the same starting points.

To calculate the betweenness centrality of a turtle (in our case Hospitals, Specialists, Laboratories), you take every other possible pairs of turtles and, for each pair, you calculate the proportion of shortest paths between members of the pair that passes through the current turtle. The betweeness centrality of a turtle is the sum of these. As of now, link weights are not taken into account

2) Eigenvector-centrality: as before, in (https://github.com/NetLogo/NW-Extension)

The Eigenvector centrality of a node can be thought of as the amount of influence a node has on a network. In practice, turtles that are connected to a lot of other turtles that are themselves well-connected (and so on) get a higher Eigenvector centrality score.

Eigenvector centrality is only defined for connected networks, and the primitive will report false for disconnected graphs. (Just like distance-to does when there is no path to the target turtle.)

In this implementation, the eigenvector centrality is normalized such that the highest eigenvector centrality a node can have is 1... As of now, link weights are not taken into account.

3) Closeness-centrality: still in (https://github.com/NetLogo/NW-Extension)

The closeness centrality of a turtle is defined as the inverse of the average of its distances to all other turtles

2.6.1 Healthcare prevention instruments: the specialistic case

This experiment will examine the effects of the so called "Preventive healthcare". It consists in all those measures taken for disease prevention, as opposed to disease treatment. This practice of prevention in the healthcare domain consists in a set of anticipatory actions which can be undertaken at different levels: local, regional, or national. So, it clearly consists in a policy affair. There exist several illnesses prevention schemes: *individual (both adults and children) should indeed visit their doctor for regular check-ups, even if they fell healthy, to*

i) perform disease screening

ii) keep up to date with immunizations and boosters

iii) be familiar with risk factors for illnesses and with what instead is able to guarantee an healthy life.

Common examples of disease screenings are those for colon cancers, cervical cancer (the so called "pap test") and breast cancer ("mammography"). All prevention strategies can be categorised in *primary, secondary, and tertiary prevention levels.* The concept of primary prevention has been firstly used by Hugh R. Leavell and E. Gurney Clark in the 1940s. Later on they expanded the levels to include the secondary and tertiary too. They all are still in use today. We now specify the main features all of them are endowed with.

1) Primary prevention. Schemes belonging to this category can be divided into two further subcategories.

i) **health promotion:** it includes non-clinical life choices, ranging from maintaining a balanced and complete diet to a an habitual and constant training. Even if such "activities" do not perform any cure to specific illnesses, they are able to promote a general level of health.

ii) **specific protection:** they complement the goal of health promotion (a general well-being) by means of specific actions.

2) **Secondary prevention.** It can be further divided into:

i) **Early diagnosis and prompt treatment:** aimed at containing the disease and preventing its spread to other individuals.

ii) **disability limitation:** oriented at preventing potential future complications and disabilities from the disease.

Differently said, such prevention type consists in all those methods which can be used in order to detect a disease prior its symptoms to appear.

Let us take as an example the case of a syphilis patient: the early diagnosis and prompt treatment phases would include the usage of antibiotics to destroy the pathogen and screening and treatment of any infants born to syphilitic mothers. The disability limitation phase would instead been constituted of continuous check-ups on the heart, and central nervous system of patients to limit any damaging effects such as blindness or paralysis.

3) **Tertiary prevention.** It is aimed at reducing the damage caused by symptomatic disease by focusing on mental, physical, and social rehabilitation. So, differently from secondary prevention, which try to avoid disability to occur, tertiary prevention tries to maximize the remaining capabilities and functions of an already disabled patient.

The case we are going to analyse now focuses on the consequences of a policy action (so, for example, a set a measures adopted by the state,

through the board of health) aimed at reducing the probability for a patient to contract an illness requiring specialistic services to get cured. The main assumptions we are undertaking are the following:

i) *Policy measures do have "positive effects" upon all the age groups we have divided the patients in.*

ii) *"Positive effects" could be interpreted in terms of lower probabilities of getting ill (so we will lower all those sliders labelled as "p12, p22, p32, p42 and p52"), in comparison with the case in which the policy action does not take place, and so characterized by higher probabilities in all the five age groups.*

Given such assumptions, if we compare such situation with one in which probabilities of contracting an ill requiring specialistic cures are higher, we expect a lower number of total "specialistics". In order to count such links within different simulation, we have inserted a "monitor", a specific element on the interface able to count the exact number of the agents we are interested in as the ticks are going by. Figure2.44 and Figure2.45 show such monitors respectively at the beginning and at the end of a particular simulation.

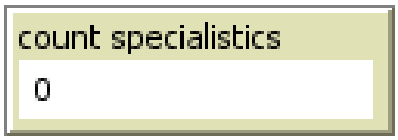


Figure 2.44: Monitor counting the number of specialistics at the beginning of a simulation.

In the following subsection we will consider in detail the consequences an ongoing prevention scheme brings in the healthcare specialistic network.

2.6.2 Prevention-scheme experiment

In this section we will adopt the following logic: we move from a starting situation, explaining the reasons behind the choice of particular values we

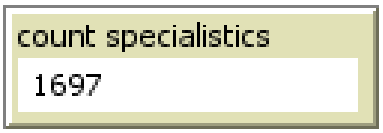


Figure 2.45: Monitor counting the number of specialistics at the end of a simulation.

have attributed to the variables, observing the output. We then analyse the case in which a preventive scheme has been undertaken, focusing on possible different emerging scenarios within the networks and on the amount of specialistic cures have taken place. Figure2.46 shows our starting point. We now just consider the set of variables located on the right-hand of the simulated world, in particular:

i) We have attached increasing probabilities of getting ill as the age categories are higher. Respectively: prob1 (0,15), prob2 (0,20), prob3 (0,25), prob4 (0,30) and prob5 (0,35).

ii) We have attached increasing recipes-length to each age category too, meaning that patients of each age category can be affected by a greater number of illnesses then those belonging to the previous ones.

iii) Focusing only on the probability that each patient is affected by type-two illnesses (those that need a specialist-based cure), so p12, p22, p32, p42 and p52, we have assumed they are increasing as the generic patient enters the following age category (respectively, 27, 40, 45, 47 and 52).

Figure2.47 shows the simulation once all the patients are died and so, once the simulation is over. If we consider the "count specialistic" monitor it displays "3304", which is the number of specialistic visits that have taken place within the simulation. The logic behind the introduction of preventive measures by means of the board of health is the following: we have assumed, reasonably, that such introduction will lower both

i) the probability of contracting a ill of the second type (reduced by 10%)

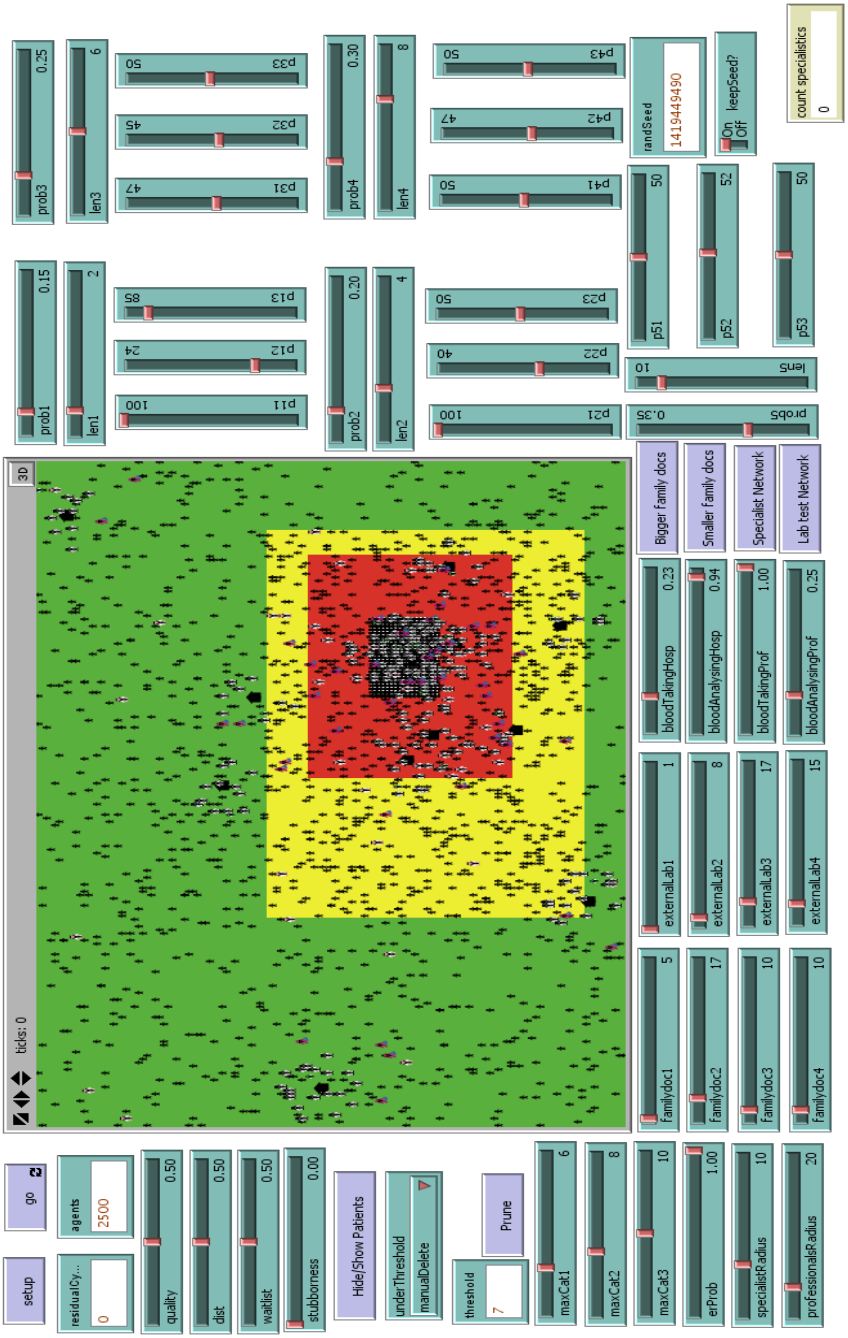


Figure 2.46: Prevention-schemes experiment: starting condition.

ii) the maximum number of illnesses that patients belonging to each age category can be affected by (again, around 10%).

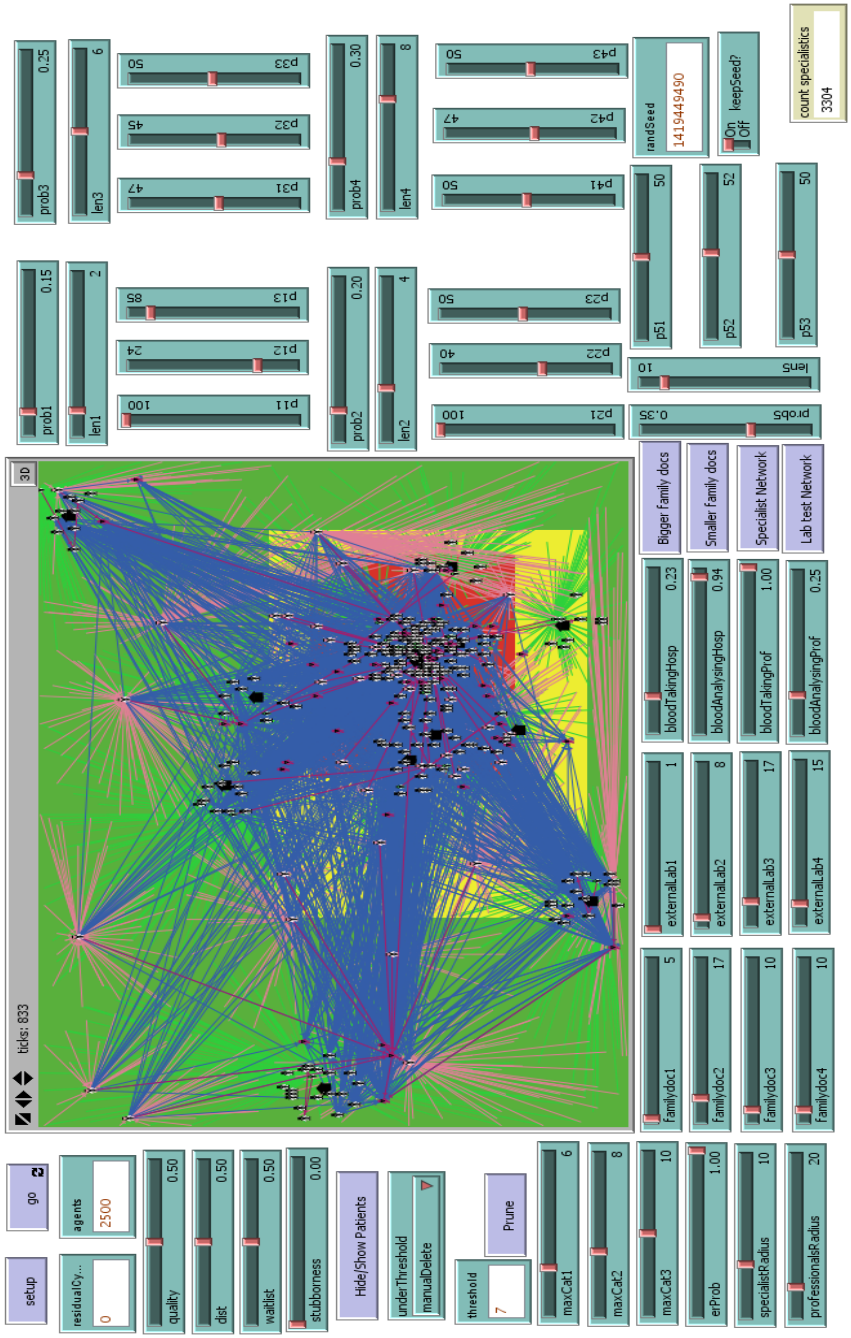


Figure 2.47: Prevention-schemes experiment: starting condition. The set of links among agents once the simulation is finished.

Now we are going to test if a reduction of such probabilities has an effect on the number of emerging "specialistics" (we reasonably expect that such changes will lower such number). Since the "keepSeed?" button is on, we can compare the previous simulation with this one. We first of all have lowered

i) *all the probabilities previously mentioned by approximatively 10%*

ii) *number of illnesses each patient can be at most affected by (more or less 10%).*

The results of such simulation are shown in Figure2.48. Contrary to our prediction, the number of specialistics created is not lower, but even greater (3305 against 3304).

We now consider the case in which such measures have a greater impact (suppose a 20%). Results are shown in Figure2.49.

In this case, the reduction taken into consideration has a significant impact on the number of specialistics, which is sensitively decreased from 3305 to 3209, confirming our initial hypothesis.

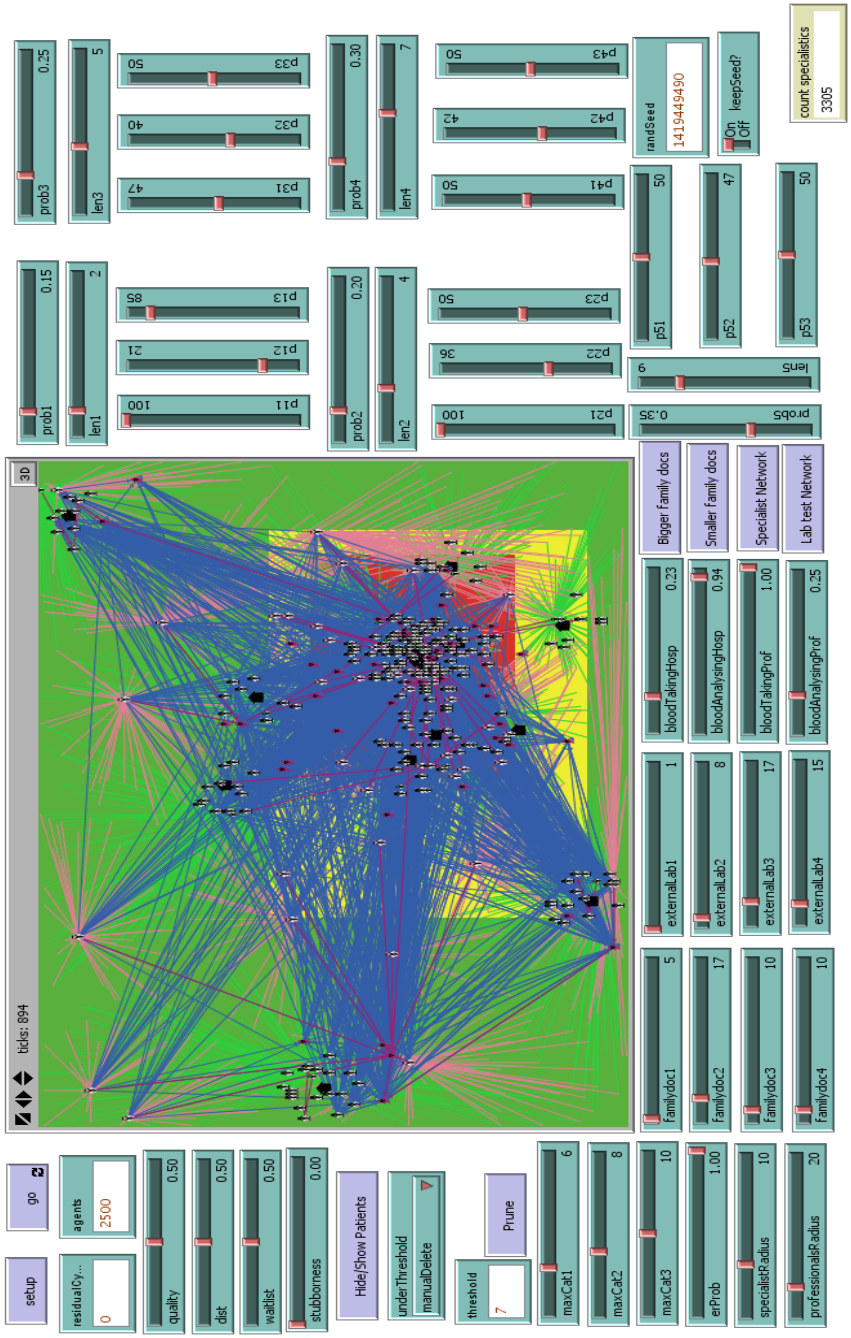


Figure 2.48: Prevention-schemes experiment: output with modified (reduced by approximatively 10%) probabilities.

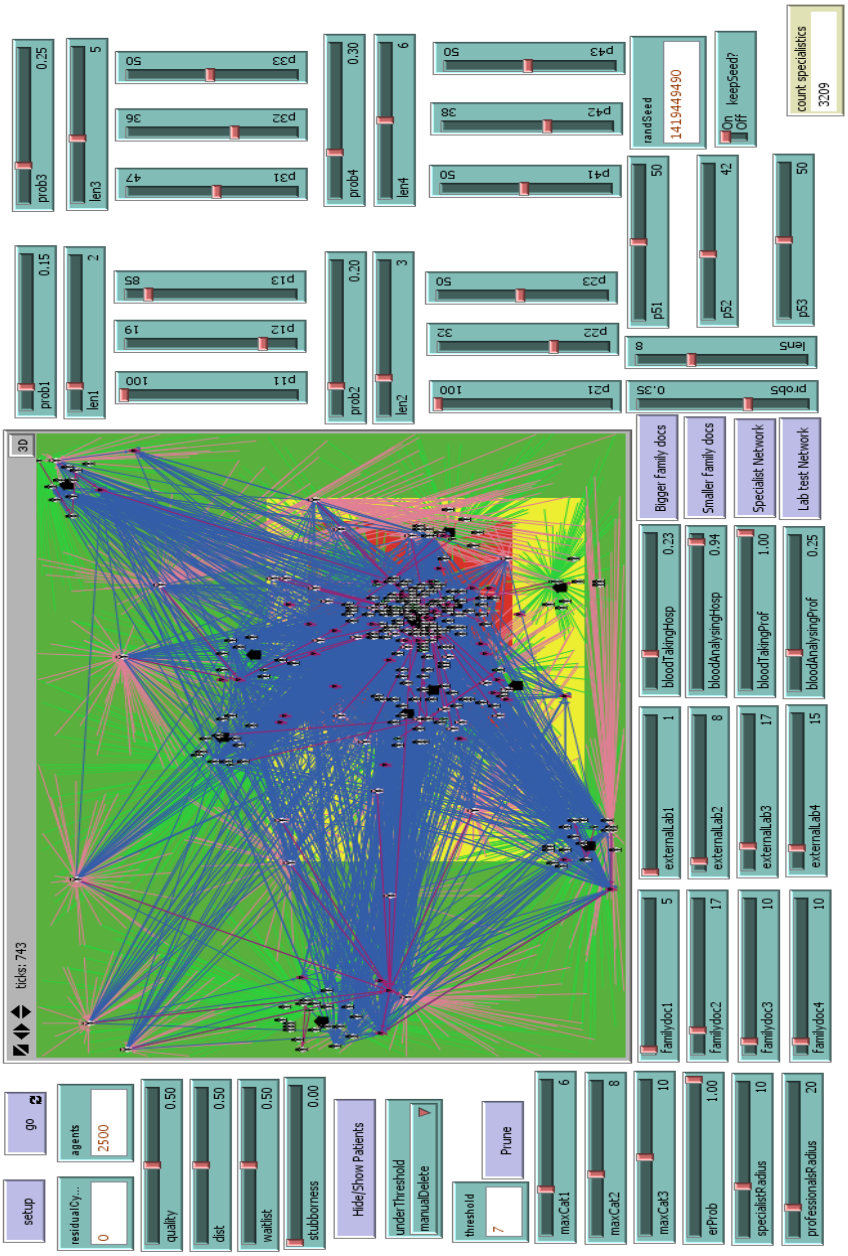


Figure 2.49: Prevention-schemes experiment: output with modified (reduced by approximatively 20%) probabilities.

2.6.3 Decision-making process experiment

As we have already said in the previous pages, one of the most attractive feature of our work consists in the possibility for patients to get cured according to three different parameters, or personal preferences. Respectively: quality of the healthcare system providing a cure, the waiting list characterizing it and the distance from the agent at issue. If the stubborness slider shows a null value, then the patient will get cured in those healthcare structures that have been considered as those fitting best with its own preferences. However, if the stubborness slider shows a positive value equal to one, then such movement will be completely random. We now describe such situation more in detail. First of all, in the code section we have modified the proportions of patients spread around the four districts of our world as follows:

```
to make-patients
set-default-shape patients "person"
let t1 round (0.05 * agents)
let t2 round (0.75 * agents)
let t3 round (0.1 * agents)
let t4 round (0.1 * agents)
create-patients t1 [ set size 2 set color black move-to one-of
patches with [pcolor = white] ]
create-patients t2 [ set size 2 set color black move-to one-of
patches with [pcolor = green]]
create-patients t3 [ set size 2 set color black move-to one-of
patches with [pcolor = yellow]]
create-patients t4 [ set size 2 set color black move-to one-of
patches with [pcolor = red]]
end
```

The reason behind such change in the population proportions per area is the following: we want to show the impact of the “distance” preference on the agents movements (still focusing on the “specialistic side” of the cures).

So we expect that, if the majority of patients is located in the more external belt, then a greater role will be played by those agents providing healthcare cures set in the more external side of our world.

Figure2.50 shows the movements of 2000 patients within our world, when their three preferences have more or less the same weights and the stubborness slider has a zero value. In order to understand more clearly such distribution, Figure2.51 shows the same scenario with those links that have more than seven visits (we have applied the prune button after having set a seven visits threshold). As it clearly emerges, there is a strong concentration around those healthcare structures located in the two inner districts.

We now want to test if, considering the distance criterion as the only one guiding the patients movements, a different scenario will emerge, given the fact that the majority of the patients is now located in the last belt, and so far from the healthcare institutions located in the inner districts. To test it, we will change the "preferences slider" as follows: quality (0), distance (1), waitlist (0) and stubborness (0). Figure2.52 shows the related graphical result. It is immediate to notice the greater role now played by those healthcare services providers located far from the inner areas, and so close to the majority of patients. To get a clearer picture of such situation, again we apply the prune button to consider only the links endowed with more than seven visits. Figure2.53 depicts it so explicitly. Other alleged experiments can be applied to test those situation in which patients attribute a greater value to the quality or the waiting lists characterizing healthcare structures.

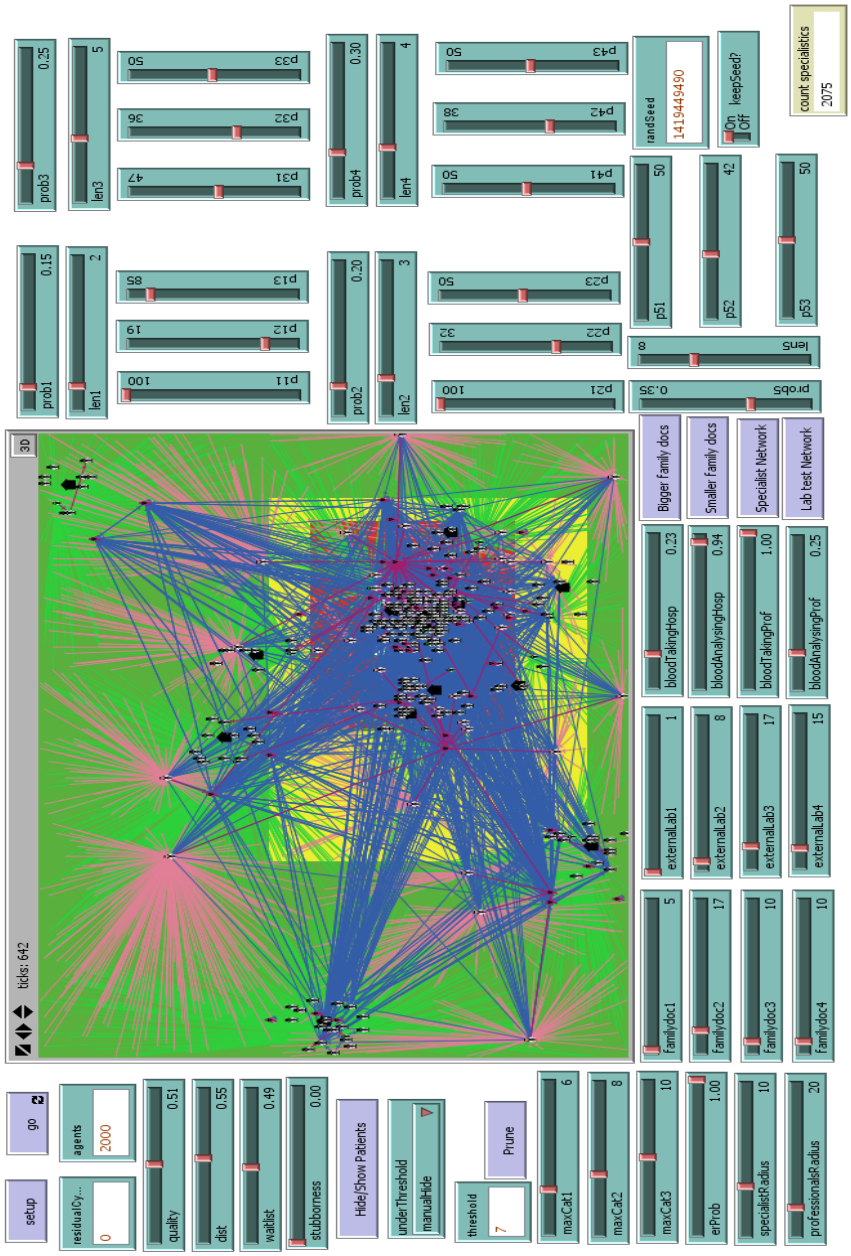


Figure 2.50: Decision-making schemes experiment: same weights attributed to patients preferences.

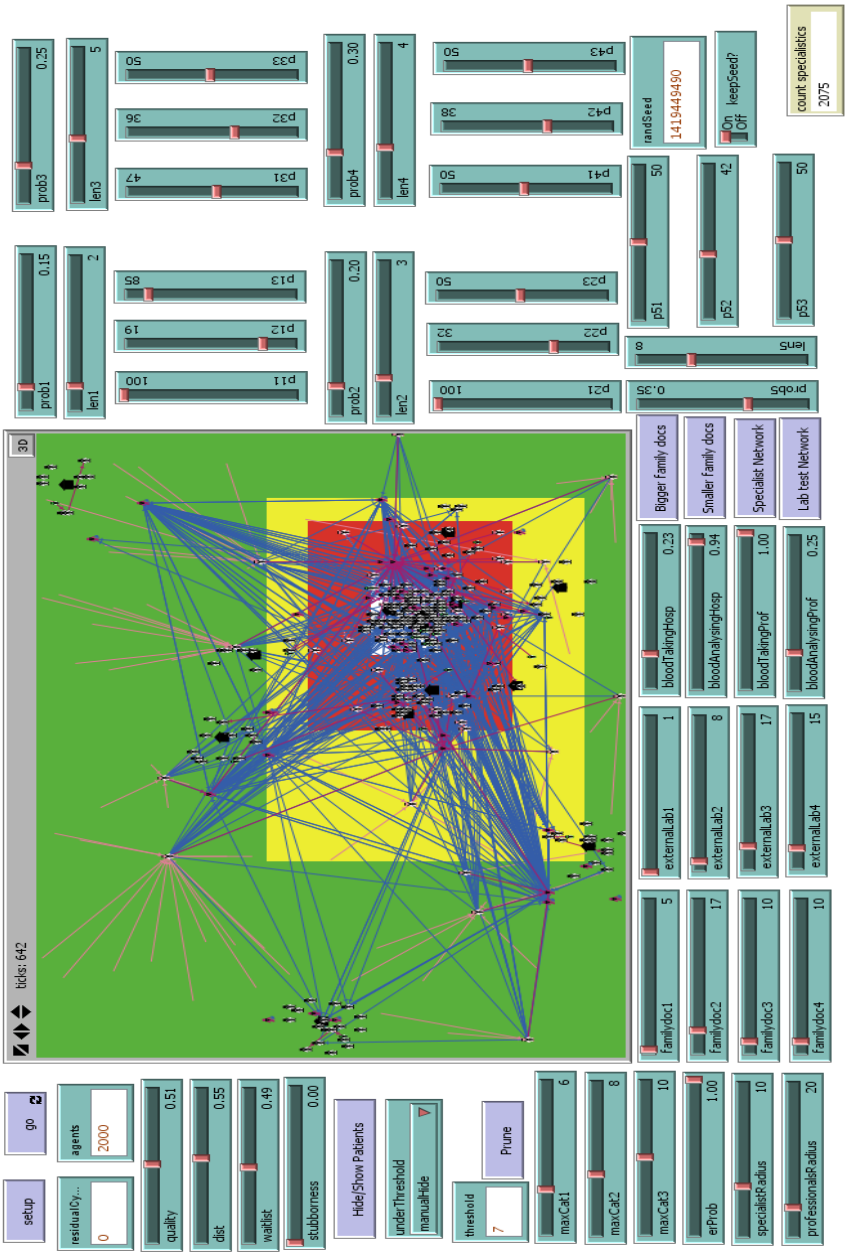


Figure 2.51: Decision-making schemes experiment: same weights attributed to patients preferences. Application of the prune button.

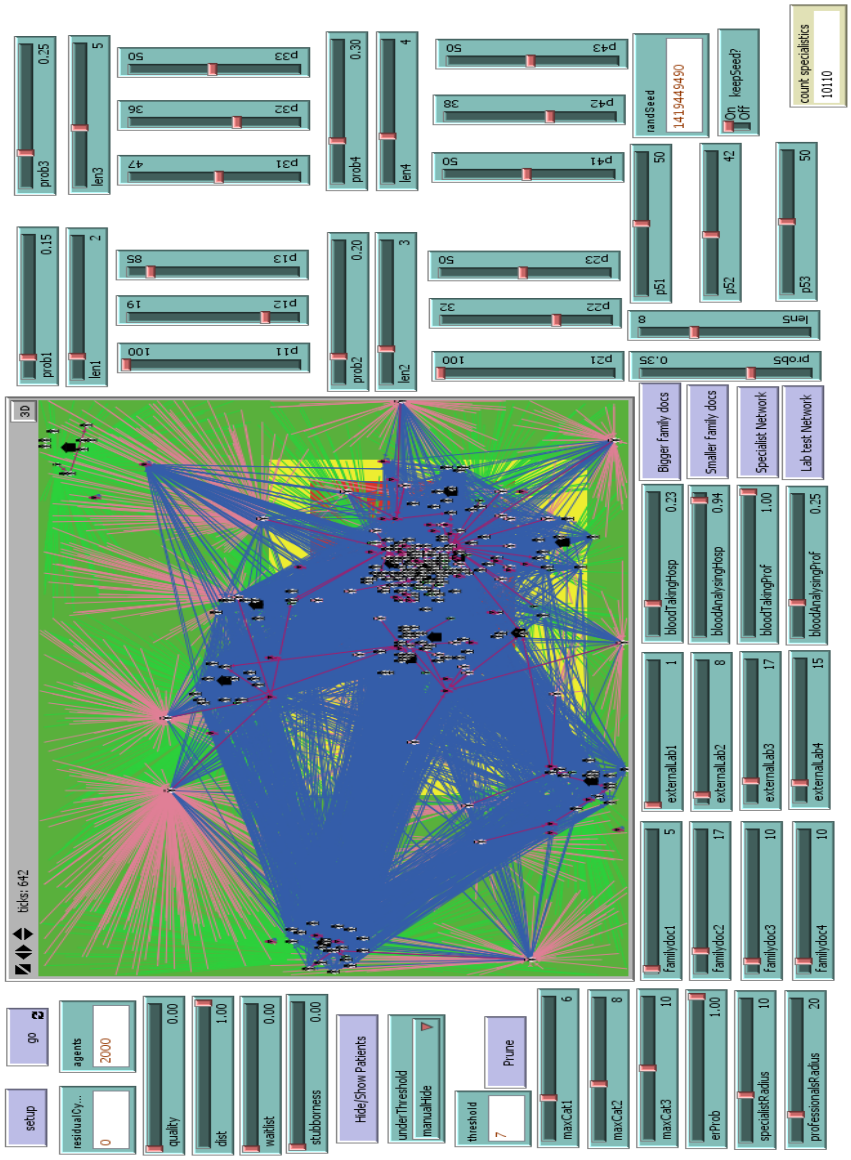


Figure 2.52: Decision-making schemes experiment: only distance counts as a patient preference.

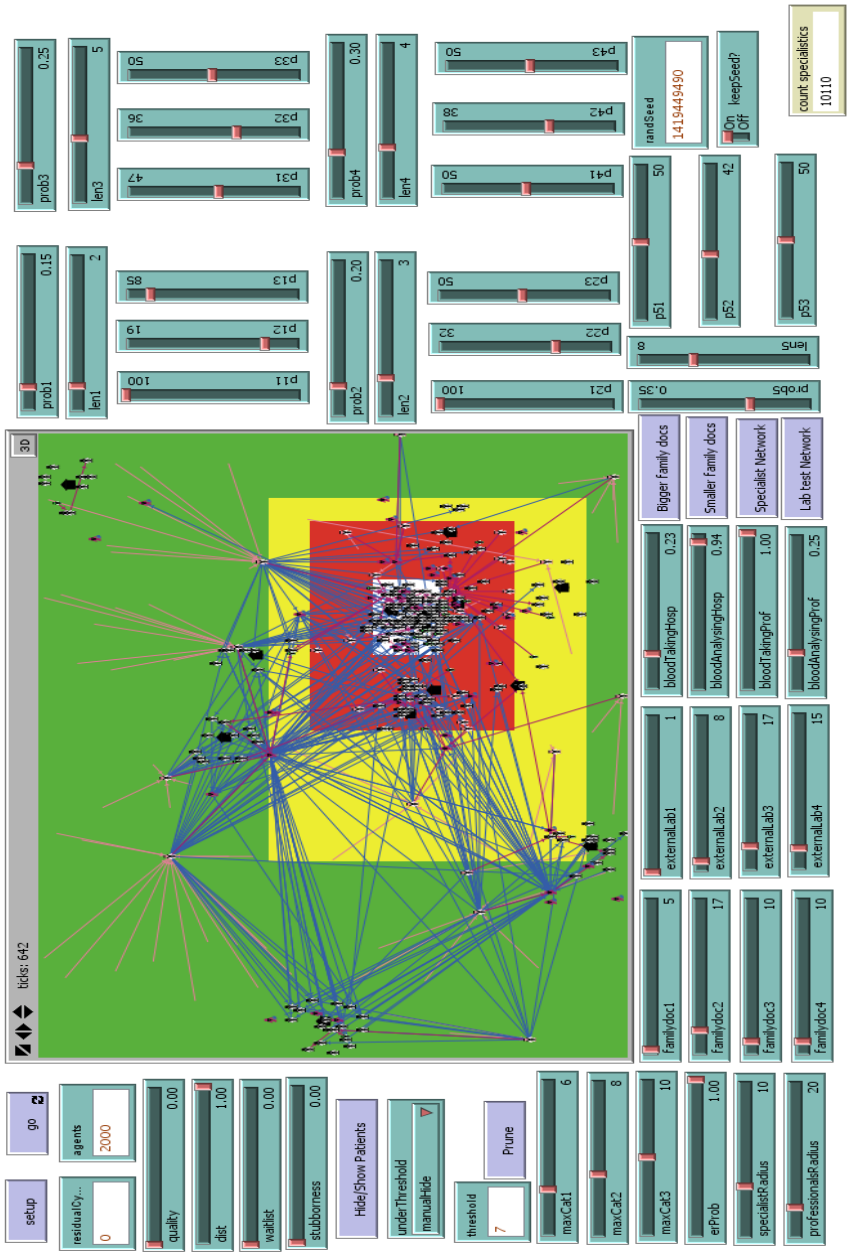


Figure 2.53: Decision-making schemes experiment: only distance counts as a patient preference. Application of the prune button.

2.6.4 Emergency Room experiment

In this subsection we are going to analyse another feature of our model: the emergency room management. In particular, we have allowed the model for the possibility of choosing the probability of hospitals providing er services. Such feature is run by the slider called "erProb". Its value ranges between 0 and 1, with the former representing the limit case in which no hospitals provide er services and the ladder case in which all of them offer such services. We are going to test the effect of such changes. We are going to compare the case with the highest probability of hospitals providing er services with the one in which at most the thirty percent of the hospitals do that. We expect a lower number of hospitals "hit" by green links. Figure2.54 represents the first case. As we can see, there are several healthcare institutions providing such cures. As a consequence, the distance between patients and the place where they get cured (which can be considered as the length of the green links) is quite short, in particular in those areas where there are several hospitals. To get this clear results we have graphically eliminated all the other typologies of links.

Figure 2.55 instead represents the same case but with a maximum thirty percent of institutions providing such service. The result is dramatic. Just one hospital provides such service. As a consequence the links are now very long, since the offer is really poor all arounf the district. It turns out to be really interesting to apply those thoughts to the real world, taking into account all the consequences that such last scenario would have on the quality of the service (and, of course, on the number of the dead).

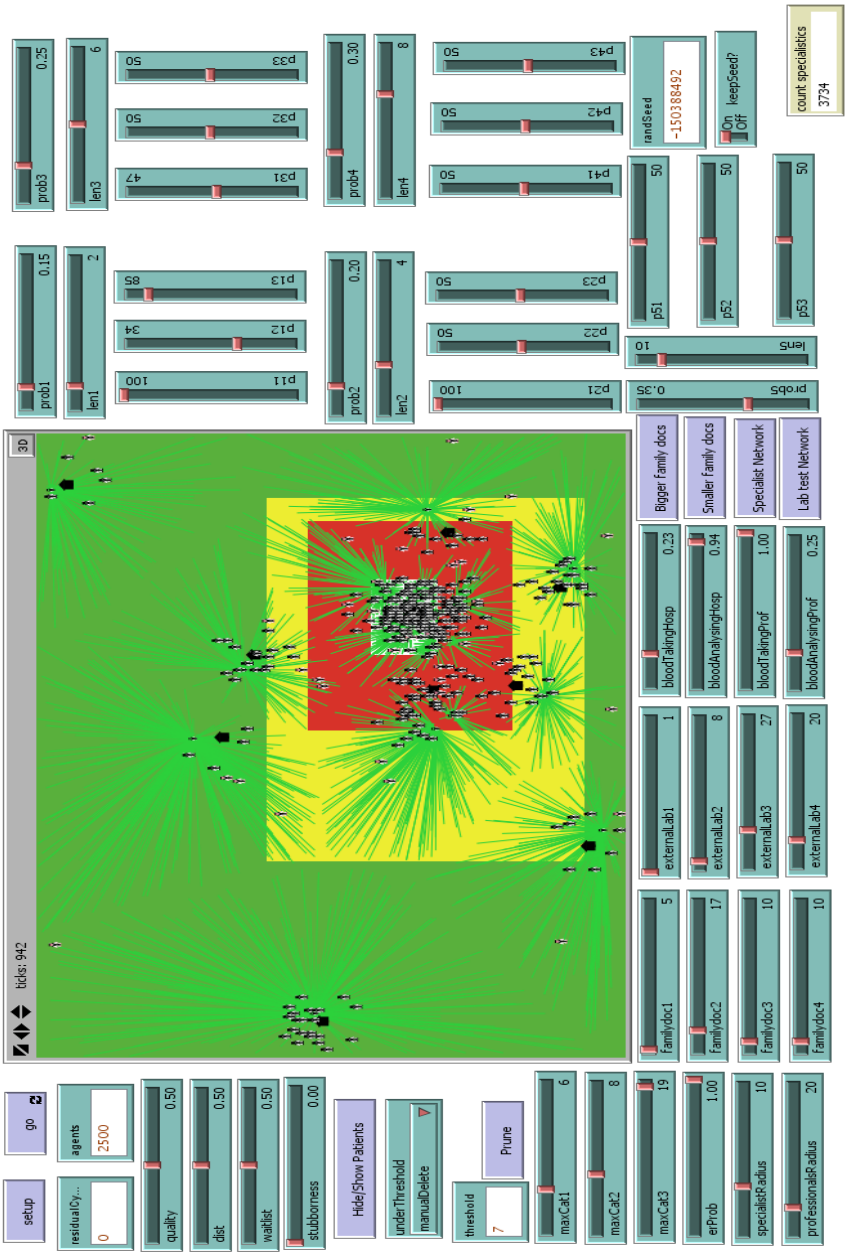


Figure 2.54: Er best scenario. The highest probability of hospitals providing er services.

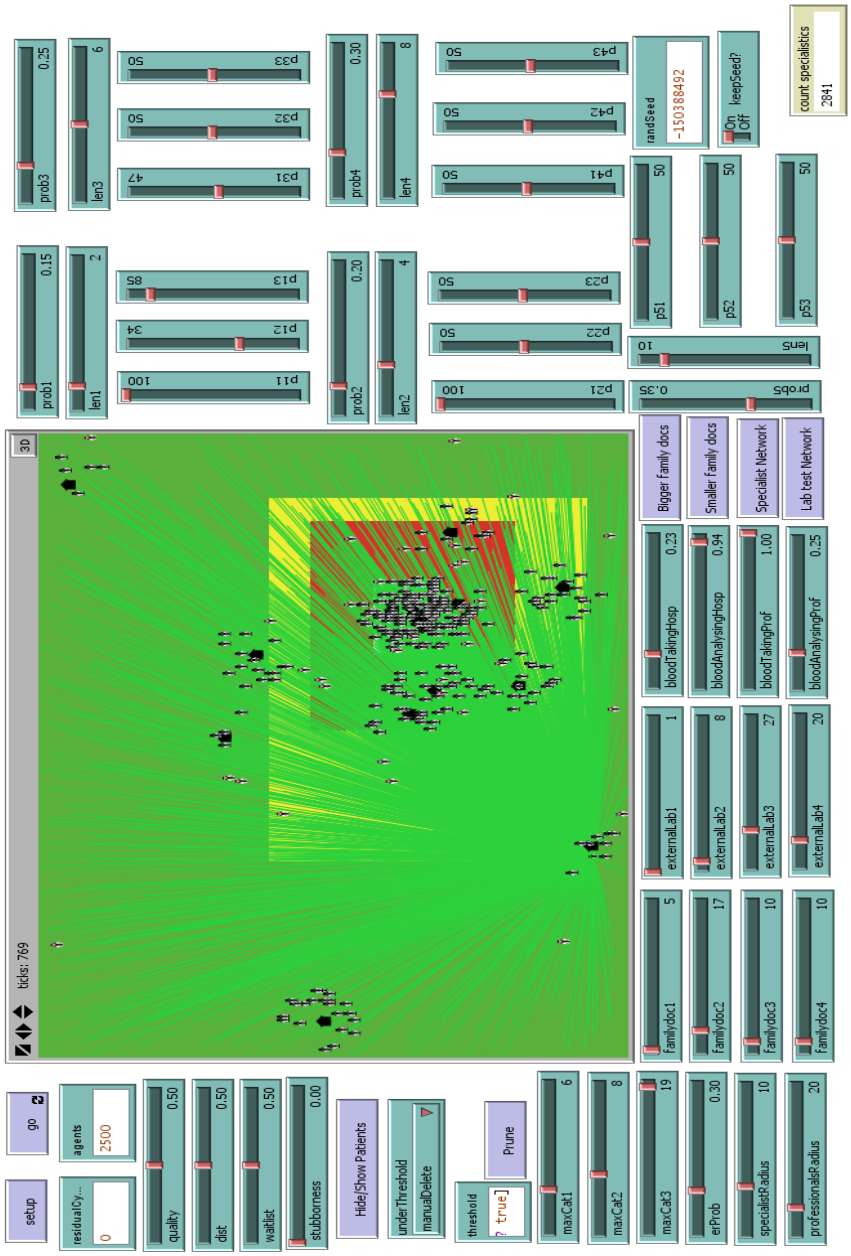


Figure 2.55: Er worst scenario. Really low probability of hospitals providing er services.

Conclusions

This thesis lays its foundation in giving an insight to the healthcare domain. In particular, it has been recently registered an increasing appealing towards this domain as a consequence of the need for restructuring those segments that are grasping too many resources within a national economy. Given this situation, our goal has become the creation of a computer programme allowing any user to get a better understanding within this domain. Once we had clear in mind what we were aimed at, we have identified in NetLogo the optimal software to create our framework. Indeed, it has allowed us to reproduce in a controlled environment what can most likely happen in the real world. It is so turned out to be an essential tool in reaching our goal. During our work, we have adopted a rigidly precise structure in order to create the best flow of concepts to get any potential reader in the position of a good understanding of the topic. For this reason, we have developed our work as follows:

Literature Review: this part has been devoted to cover the theoretical aspects of our work. We have reported and analysed the ideas and, sometimes, the empirical findings of the most influential researchers in each topic we have developed. In particular, we have coped with a series of topics we have considered essential to create the appropriate mindset to deal with the programme developed. Such topics are linked each others and all together represents what we have considered the best way to follow to enter the topic of simulating an healthcare environment. In detail: we started by defining the general purpose of the

simulation, highlighting its points of strength and at the same time the limits it encounters while trying to be applied in real works. We then analysed the methods for validation of population-based disease simulation models, considering subsequently the particular case of "Agent based Social Simulation", which is the one we have applied in our work. We then concluded this part analysing both the topic of Complexity and its implications in several aspects of our everyday life, together with the concept of Network, with particular emphasis for the social networks and their main features.

The Model: this part represents the operative section of our work. Indeed, by means of the simulation software NetLogo, we have developed our own programme. It has been constructed in order to allow first of all us and, ultimately, any user to carry on a set of experiments in a controlled environment representing the real world. We have represented a complete set of health-care Agents, such as patients, hospitals and doctors. We have allowed patients to get cured in different institutions providing treatment both according to their will or to the family doctor decision. We have then reproduced a likely real situation that can be then analysed from different points of view according to the user aim.

Those two main points stand for the thesis structure. A detailed analysis of the literature concerning the field we wanted to developed and then the model creation followed by a set of experiments aimed at testing the consistency of this model.

Even if we are strongly convinced of the structure we have applied to the work and of the results we have achieved, we are at the same time aware that other ways could be explored to get better insights. This is mainly due to the type of data we have been working with. Only a part of them can be categorized as "real", while the other part has been developed as rational hypothesis but, still, they are hypothesis. This is why we consider our work a starting point that can be improved in future works relying upon a greater amount of real data.

Acknowledgements

I would first like to thank my colleague Enrico Carlo Girotti, for the well managed teamwork which led to the NetLogo model. I am very thankful to my supervisor, Professor Pietro Terna, for his continuous effort in order to make my work original and consistent. He always provided me with interesting and useful ideas that helped me all over this project. I am very thankful to my assistant supervisor as well, Professor Sergio Margarita, for the time spent inspecting my work. I would like then to thank my family for the support they provided me with during my entire life. They always helped me in taking the right choices and decisions. Finally, I am really grateful to my closest friends, Luigi Scavino, Giacomo Raimondo, Simone Martinetto, and my basketball team-mates who also contributed in building up the person I am today.

Bibliography

Allen, W. (2013). *Complicated or complex? Knowing the difference is important*.

Brailsford, S., Harper, P., Patel, B. and Pitt, M. (2009). *An analysis of the academic literature on simulation and modelling in health care*. In «Journal of Simulation», vol. 3(3), pp. 130–140.

Bransburg-Zabary, S., Kenett, D. Y., Dar, G., Madi, A., Merbl, Y., Quintana, F. J., Tauber, A. I., Cohen, I. R. and Ben-Jacob, E. (2013). *Individual and meta-immune networks*. In «Physical biology», vol. 10(2), p. 025003.

Cioffi-Revilla (2014). *Introduction to Computational Social Science*.

Davidsson, P. (2002). *Agent based social simulation: A computer science view*. In «Journal of artificial societies and social simulation», vol. 5(1).

Dror Y. Kenett, S. H. (no year reported). *Introduction to network science*. URL `http://images.iop.org/journals_icons/Info/0295-5075/J%20EPL%20LF%200513%20Networks%20Intro-2.pdf`

Eldabi, T. (2009). *Implementation issues of modeling healthcare problems: misconceptions and lessons*. In *Winter Simulation Conference*. Winter Simulation Conference, pp. 1831–1839.

Epstein, J. M. (1996). *Growing artificial societies: social science from the bottom up*. Brookings Institution Press.

Escudero-Marin, P. and Pidd, M. (2011). *Using ABMS to simulate emergency departments*. In *Proceedings of the Winter Simulation Conference*. Winter Simulation Conference, pp. 1239–1250.

Fackler, J., Hankin, J. and Young, T. (2012). *Why healthcare professionals are slow to adopt modeling and simulation*. In *Proceedings of the Winter Simulation Conference*. Winter Simulation Conference, p. 97.

Fackler, J. and Spaeder, M. (2011). *Why doesn't healthcare embrace simulation and modeling? What would it take?*. In *Simulation Conference (WSC), Proceedings of the 2011 Winter*. IEEE, pp. 1137–1142.

Fone, D., Hollinghurst, S., Temple, M., Round, A., Lester, N., Weightman, A., Roberts, K., Coyle, E., Bevan, G. and Palmer, S. (2003). *Systematic review of the use and value of computer simulation modelling in population health and health care delivery*. In «Journal of Public Health», vol. 25(4), pp. 325–335.

Gell-Mann, M. (1995). *What is complexity? Remarks on simplicity and complexity by the Nobel Prize-winning author of The Quark and the Jaguar*. In «Complexity», vol. 1(1), pp. 16–19.

Gilbert, N. (2004). *Agent-based social simulation: dealing with complexity*. In «The Complex Systems Network of Excellence», vol. 9(25), pp. 1–14.

Glouberman, S. and Zimmerman, B. (2002). *Complicated and complex systems: what would successful reform of Medicare look like?*. Commission on the Future of Health Care in Canada Toronto.

Johnson, N. (2009). *Simply Complexity: A clear guide to complexity theory*. Oneworld Publications.

Kirchhof, P. and Meseth, N. (2012). *A survey on the use of simulation in German healthcare*. In *Simulation Conference (WSC), Proceedings of the 2012 Winter*. IEEE, pp. 1–10.

Kopec, J. A., Finès, P., Manuel, D. G., Buckeridge, D. L., Flanagan, W. M., Oderkirk, J., Abrahamowicz, M., Harper, S., Sharif, B., Okhmatovskaia, A. *et al.* (2010). *Validation of population-based disease simulation models: a review of concepts and methods*. In «BMC Public Health», vol. 10(1), pp. 710–.

Li, X., Mao, W., Zeng, D. and Wang, F.-Y. (2008). *Agent-based social simulation and modeling in social computing*. In *Intelligence and Security Informatics*. Springer, pp. 401–412.

Shoham, Y. and Leyton-Brown, K. (2008). *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press.

Wikipedia (2014). *Betweenness centrality — Wikipedia, The Free Encyclopedia*. [Online; accessed 31-August-2014].
URL `http://en.wikipedia.org/w/index.php?title=Betweenness_centrality&oldid=623270916`