Alma Mater Studiorum · Università di Bologna

SCUOLA DI INGEGNERIA E ARCHITETTURA
DIPARTIMENTO DI INFORMATICA - SCIENZA E INGEGNERIA
CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

TESI DI LAUREA

in

SISTEMI INTELLIGENTI

AGENT-BASED SIMULATION FOR RENEWABLE ENERGY INCENTIVE DESIGN

Candidato:
VALERIO IACHINI

Relatore:
Chiar.ma Prof.ssa
MICHELA MILANO

Correlatore:
Dott. Ing.
ANDREA BORGHESI

Sessione II
Anno Accademico 2013/2014

Abstract

Con questo lavoro s'intende proporre un nuovo approccio per modellare la diffusione di sorgenti rinnovabili di energia in ambito residenziale. A tal proposito, abbiamo deciso di adottare un modello basato ad agenti, dove gli agenti rappresentano le famiglie che risiedono nella regione in esame. Questo implica uno studio del territorio per determinare quali sono le caratteristiche delle famiglie che vi abitano. Il caso di studio è quello del Piano Energetico della Regione Emilia-Romagna che mira ad aumentare la produzione di energia soprattutto da fonti rinnovabili come biomasse e solare. Siamo partiti, quindi, dallo studio dei micro dati usati dalla banca d'Italia per ottenere le statistiche rilevanti sulle famiglie residenti in Emilia-Romagna. Questi dati ci hanno permesso di generare delle famiglie in modo artificiale e riprodurre virtualmente gli aspetti socio-economici della regione. Le famiglie generate per mezzo di un software sono collocate nel mondo virtuale associando ad ognuna di esse un'abitazione. Queste abitazioni sono acquisite analizzando i dati vettoriali degli edifici messi a disposizione dalla regione. Una volta predisposto il mondo virtuale, il modello ad agenti determina il livello diffusione simulando ogni anno la potenza installata dalle famiglie. La scelta di un agente d'installare un impianto è influenzata dalle relazioni sociali, dalla condizione economica, dai benefici ambientali derivanti dall'adozione e dal periodo di recupero dell'investimento.

In this thesis, we propose a novel approach to model the diffusion of residential PV systems. For this purpose, we use an agent-based model where agents are the families living in the area of interest. The case study is the Emilia-Romagna Regional Energy plan, which aims to increase the production of electricity from renewable energy. So, we study the microdata from the Survey on Household Income and Wealth (SHIW) provided by Bank of Italy in order to obtain the characteristics of families living in Emilia-Romagna. These data have allowed us to artificial generate families and reproduce the socio-economic aspects of the region. The families generated by means of a software are placed on the virtual world by associating them with the buildings. These buildings are acquired by analysing the vector data of regional buildings made available by the region. Each year, the model determines the level of diffusion by simulating the installed capacity. The adoption behaviour is influenced by social interactions, household's economic situation, the environmental benefits arising from the adoption and the payback period of the investment.

Contents

List of Figures

List of Tables

Introduction

Public policy making is a set of complicated process with the purpose of addressing public problems that involve progressive and interactive environment. Factors such as globalization make our society ever more complex, so the decision-making process must be adapted to the rapidly changing globalised world. The cities are becoming larger; therefore, the decisions taken by policy makers affect more and more individuals. This growth increases the chance that entities involved have conflicting interests that impact the achievement of goals. Policy makers must find a balance between the individual interests and the global objectives. It is not always simple because the amount of data to be examined, and the number of constraints to be considered can be very high. However, if they are assisted by tools that can provide predictive models, they could evaluate the consequences of their decisions.

The ePolicy project is a FP7 STREP project funded by the European Union, which is devoted to the development of Decision Support Systems (DDS) for assisting decision-makers to design socially accepted and sustainable policies from the point of view of the environment. A Decision Support System may employ several techniques from different fields such as artificial intelligence, operations research, sociology, economics, etc.
ePolicy aims to help decision makers to evaluate social, economic and environmental impacts during the policy making life-cycle. Of course, when the problem is complex, and there are several requirements to be met, a DDS can aid to get the most benefit from the available data to formulate a plan able to produce the desired effect on the environment.
The ePolicy case study is the Emilia-Romagna Regional Energy plan. In Emilia-Romagna, the regional government has set a target to increase the production of renewable energy from sources such as solar and biomass.
In this work, we focus on the solar energy and we propose a model for the residential PV system diffusion to evaluate public policies in this domain.
Many researchers have found that the diffusion of an innovation is strongly

influenced by social aspects. In recent years, agent-based modeling has generated significant attention as a tool for modelling social and individual behaviours. Consequently, we propose an agent-based model that simulates the micro-based behaviour of households in order to evaluate and explain macro-level phenomena. In this work, we tackled the challenge of reproducing the households behaviours when they decide to estimate the opportunity to utilize a PV system for their houses. Hence, an Agent Based Model is an intuitive approach to addressing the problem since we can concentrate on the factors that impact on the adoption of a PV system by analyzing the behaviour of individuals.

To test our model we recreate the Emilia-Romagna environment by analyzing data from the Survey on Household Income and Wealth (SHIW) provided by Bank of Italy [2012]. However, the process described is valid for any region or country.

The ultimate goal is to integrate our model in a DDS for the policy makers to evaluate alternative plans.

In the first chapter, we introduce the ePolicy project and the Emilia-Romagna Regional Energy plan case study. In Chapter 2, we provide the related works overview. In Chapter 3, we present in detail the proposed model. In Chapter 4, we describe the household generation. Finally, in Chapter 5 we present the model implementation, and we discuss the results obtained.

Chapter 1

Overview

Policy makers have to deal with extremely complex environments that rapidly change over time. Their decisions are transposed into a plan that is composed of several actions in order to archive the objectives. These plans may involve different entities and affect three pillars of sustainable development: economics, social iteration and environment. So, It is necessary to reach an appropriate balance between individual interests and the objectives of the plan.
The complexity of the environment makes it hard to assess the long-term effects of the plan. For this reason, the politicians must be able to get the most benefit from the available data to formulate a plan able to produce the desired effect. In addition, during the policy-making life cycle, policy makers can provide several alternative plans, so they need to find a way to evaluate different alternatives. A Decision support system (DSS) is often used to assist policy makers in under- standing the consequences of complex decisions.

DSS means a vast class of software tools that aim to help decision makers in case of complex problems by facilitating the analysis of large amounts of data and suggesting strategies and policies to be adopted. Over the past 30 years, there has been a growing interest in the DSS among AI researchers, which has led to incorporating artificial intelligence techniques to model problems and simulate decision impacts. DSS architecture contains three essential components:

- the database or knowledge base,
- the model,
- the user interface.

The model adopted in a DSS can be an agent-based model (ABM) when is necessary to simulate the actions and interactions of autonomous individuals. For example, political models cover more subjects who have different characteristics and interests. These subjects are represented by agents in an ABM model that interact with the environment and respond to changes that are made in accordance with the decisions taken by politicians.
An ABM allows us to model the problem by defining the behaviour of entities involved. Normally, the behaviour of individuals is very simple but the emergent behaviour from the interaction of many agents can be complex to be modelled directly. So, ABMs are useful to understand emergent phenomena by simulating the micro-based behaviour of agents.
The ePolicy project was created to demonstrate the contribution that the DSS can give to politicians to make decisions when the problem is complex, and there are several requirements have to be met. The purpose is to provide an open source tool, easy to use and that it can supply useful indications for the user.

1.1 ePolicy

The ePolicy project aims to support policy makers in their decision process and to evaluate of social, economic and environmental impacts during policy making. The project is coordinated by the University of Bologna, and it involves nine partners from academia, research institutions, regional governments and the private sector of the European Union.
Policy makers have to deal with complex problems that have a large number of variables and constraints which concern different environmental, social and economic aspects.
An important factor that can help policy makers in their decisions is the feedback from citizens. Through social networks, blogs and other means, citizens can judge the decisions and contribute to the creation of policies. So, decision makers have the opportunity to know the social impacts through opinion mining on e-participation data.
We can summarise the policy making life cycle as shown in Figure 1.1:

- the global level optimization produces plans and scenarios for policies taking into account the objectives, the financial aspects and the environmental and social impacts on a large scale.

- The individual level simulation reproduces the social behaviour based on personal opinions.

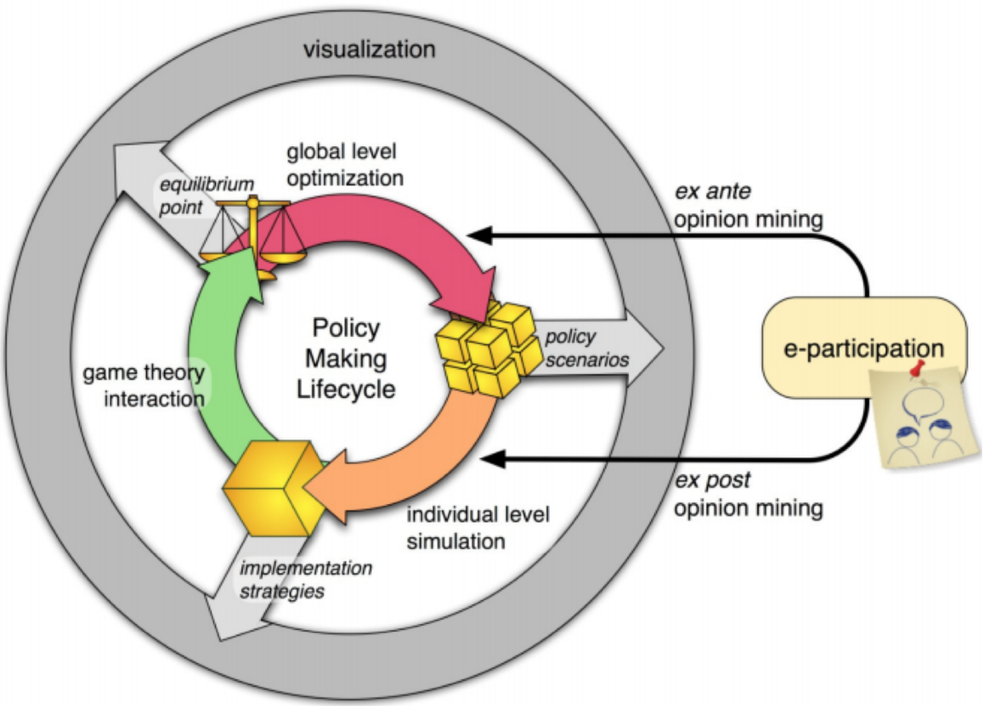


Figure 1.1: Policy making life cycle.

- The integration between the overall goal and personal goals is done using the techniques of game theory.
- The feedbacks and opinions of the entities involved are obtained using opinion mining techniques.
- Tools for visualization of the results can help decision makers.

Figure 1.2 shows the general scheme of the system. At the base, we have the involved entities in the decision whose opinions are used in the policy making life-cycle. Above the entities there is the individual level simulation. This level consists in an ABM that simulates the behaviour and interaction of the individual entities. At the top is the global level optimization that tries to find a solution by taking as input financial aspects, impact, constraints and objectives.

Thus, the ePolicy project aims to equip policy makers with integrated models, optimization, visualization, simulation and opinion mining techniques that improve the outcomes of complex global decision making. The ultimate goal of the project is to provide tools that are capable of evaluating

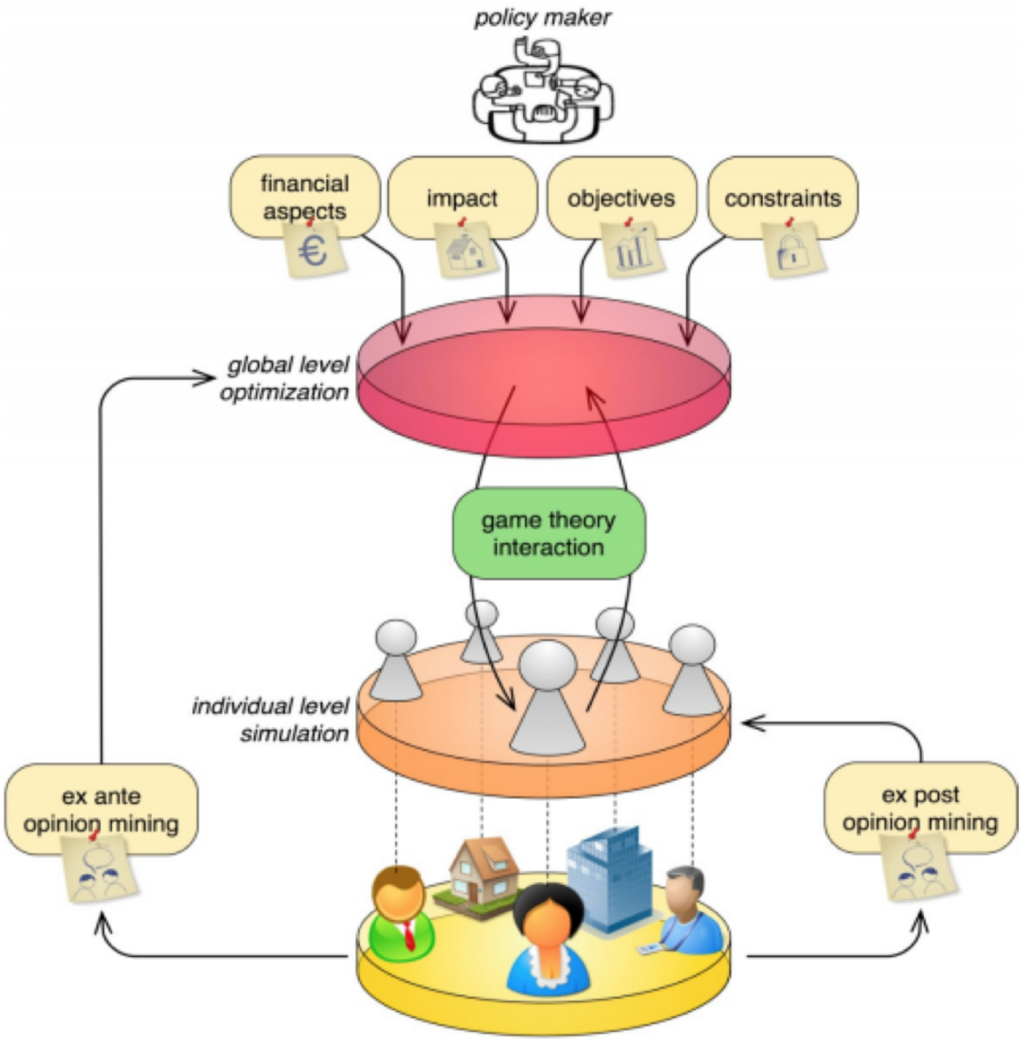


Figure 1.2: General scheme.

several alternative plans and to provide for each of them an analysis of costs and benefits. These tools make use of the most advanced techniques of artificial intelligence for solving constraint satisfaction, optimization, planning and other problems.

These ideas have been used to solve a particular problem: the energy planning in Emilia-Romagna. The regional government has set the target to increase the production of renewable energy from sources such as solar and biomass. Since 2009, these are also Italian and European goals with the Directive 2009/28/EC of the European Parliament and of the Council of 23 April 2009 on the promotion of the use of energy from renewable sources

(European Commission [2009]).
ePolicy seeks to develop a software system capable of supporting decision makers in the development of an incentive system to increase the installed number of photovoltaic systems with minimal effort for the region.
To illustrate the problem, in the next sections of the chapter, we are going to introduce the situation of photovoltaics in Italy.

1.2 Photovoltaic systems in Italy

After the introduction of national incentives, the Italian photovoltaics (PV) market has experienced a remarkable growth. The number of PV systems has more than doubled each year from 2008 to 2011. However, the growth rate in 2012 was lower than 2011, because in the 2012 the number of installed systems was 45% more than 2011. The installed power increased from 87 MW in 2007 to 16.420MW in 2012. The power has grown more than the number of installed systems because large plants came into operation, but the average size of the plants has decreased from 38.7 kW in 2011 to 34.3kW in 2012. The phenomenon is linked to a reduction in the installation of large systems determined by Legislative Decree 1/2012, which has limited the size of plants installed on the ground. Plants that came operation in 2012 have an average power equal to 24.6 kW that is lower than the plants installed in 2010 and 2011. During 2012, the operating power has increased to 3.646 MW.

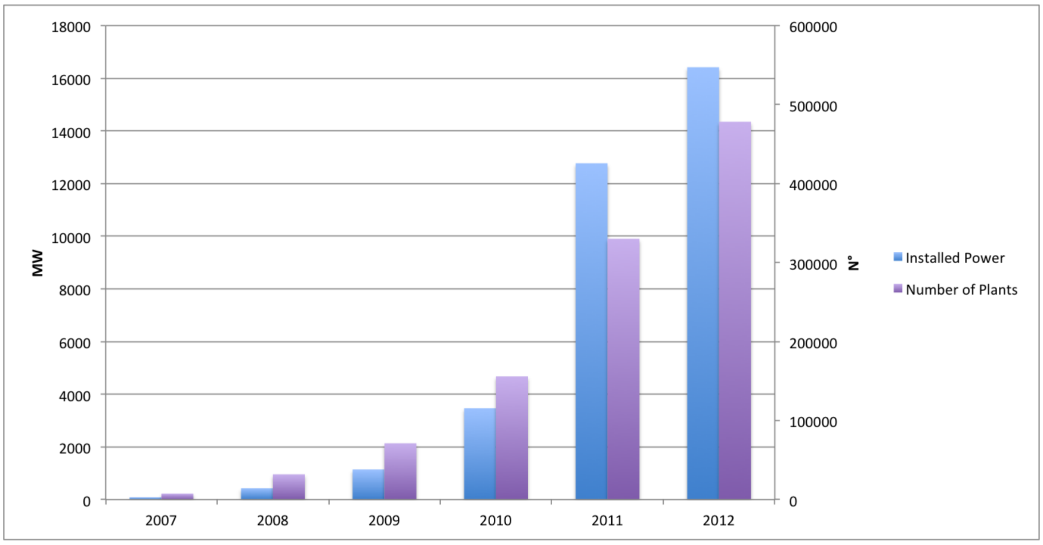


Figure 1.3: Evolution of the Italian PV market.

The distribution of power among the Italian regions is not homogeneous. According to data from the Gestore dei Servizi Elettrici S.p.A. (GSE), the highest number of plants is found in the North, especially in Lombardia and Veneto (Figure 1.4). This fact is a strange because the number of installed PV systems is much higher in the North, although the irradiation level is lower than other areas of the country. In addition, most of the installed systems in the North belong to households and are characterized by small capacity. In the southern regions of Italy, a very substantial part of the power is installed on the ground. This fact could suggest that more aspects other than pure geographical or economical one should be taken into account.

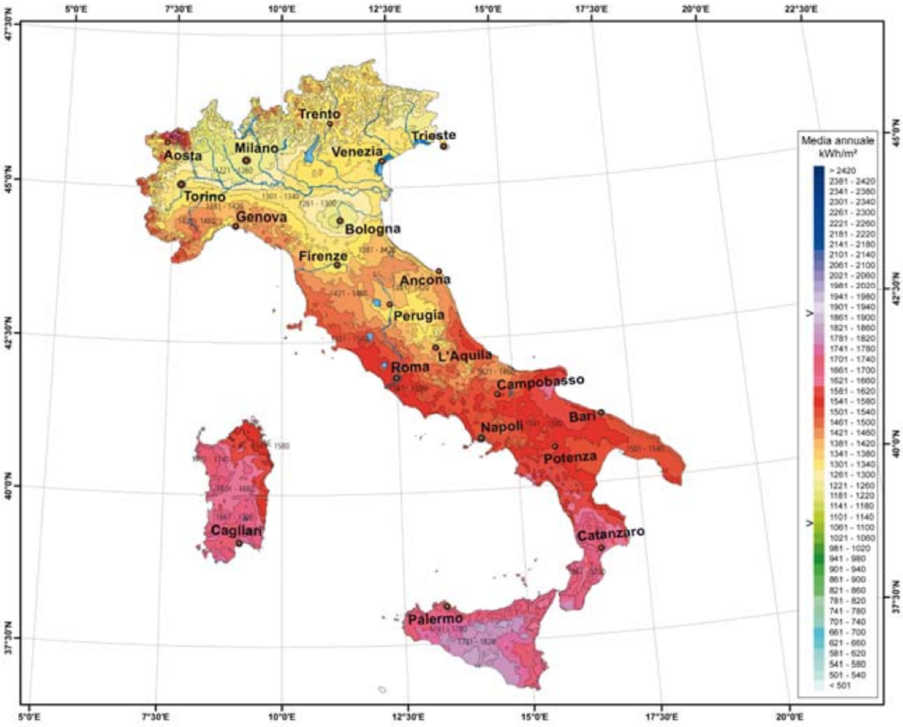


Figure 1.4: The Italian average solar radiation between 1981-2000 (GSE [2011]).

1.3 The Italian incentives for PV systems

The Italian mechanism to encourage the installation of solar systems is called “Conto Energia” (CE). This mechanism, that rewards with tariffs the energy produced by photovoltaic systems for a period of 20 years, became operational in 2005 (First CE). Since then, the incentive scheme has been renewed five times with a series of adjustments and changes. Unlike in the past, where the incentive for the production of energy from renewable sources was done by contributing non-repayable money, the CE introduces a funding system to increase earnings from energy production. A necessary condition to obtaining the tariffs is that the system must be connected to the grid and must have at least 1 kW of peak power. The CE does not provide an incentive for stand-alone systems.

The main changes introduced by the Second CE was the application of the incentive fee on all energy produced and not only on that produced and consumed in place, the simplification of bureaucratic procedures for obtaining the incentive and the differentiation of rates based on the type of architectural integration.

Until the fourth CE, the feed-in tariff was applied on all the energy produced by the plant. The fifth CE divides the tariff into two parts: the inclusive tariff applied to the energy fed into the grid and the self-consumption tariff applied to the energy consumed on site.

Figure 1.5 shows the CE results from 2007 to 2014. The CE considers two different support schemes. The first scheme is a net metering plan for plants with a capacity of less than 200 kW. In this schema, PV-generated electricity not consumed is fed into the grid. Then it can be retrieved by the household when needed. Besides the payment for each produced kWh of electricity, the GSE provides a contribution that guarantees the repayment of a portion of the expenses incurred by the family for getting electricity from the grid.

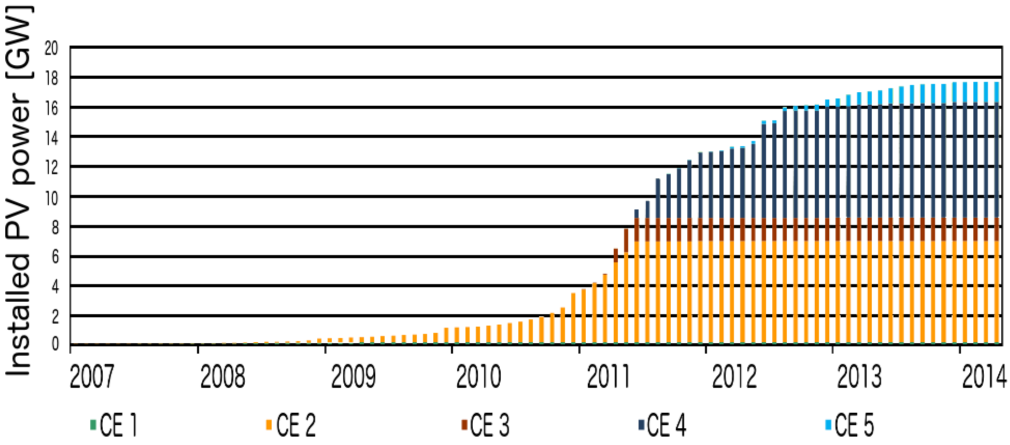


Figure 1.5: The Italian PV installed power GSE [2014].

In the second scheme, the electricity produced in excess is sold to the GSE, which guarantees a minimum purchase price. In this case, the GSE operates as an intermediary between the producer and the market.

The following chart that was obtained by averaging the tariffs for all power classes shows the trend of national incentives in Euro / kWh for the period 2006-2012. As you can see, from 2007 to 2010 the national feed-in tariffs have declined gradually with the further reduction in the second half of 2011 and the first half of 2012 (Fourth CE) and again in the second half of 2012, with the Fifth Conto Energia.

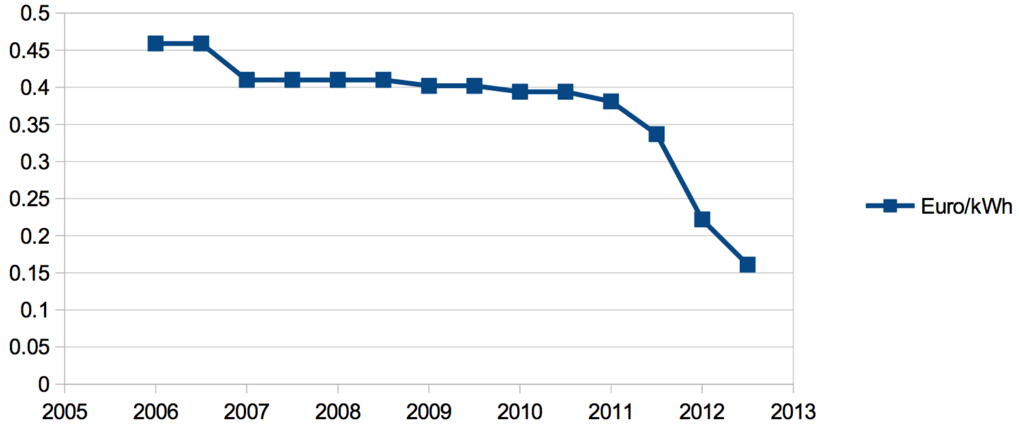


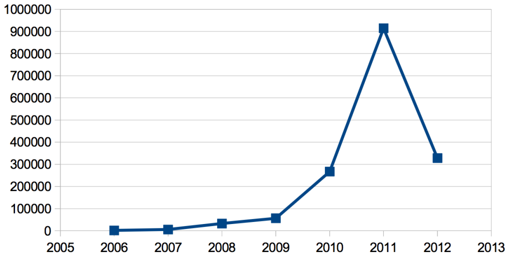
Figure 1.6: Trend of national incentives in Euro / kWh.

1.4 The Emilia-Romagna incentives for PV systems

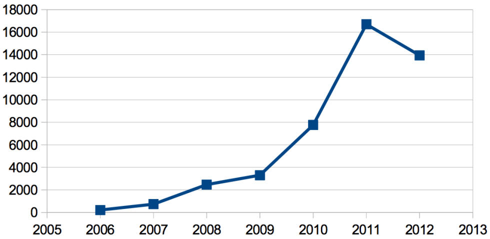
The Emilia-Romagna region has been chosen as a case study by the ePolicy project. The trend of solar installations in the area is shown in the following charts. In Emilia-Romagna, there was a reduction in the percentage of installations between 2011 and 2012. This trend is the same as found in other Italian regions.

Although the incentive rate has been decreasing over the years, the number of plants instead has increased until 2011. So, the reduction of plants in the 2012 is not only due to the economic factor but also by other factors. An explanation can be found to the limitation that the fourth CE imposed on ground installations that are larger than those on the roof.

The ePolicy project evaluates the application of four incentive mechanisms



(a) KW of installed PV power in Emilia-Romagna [KW].



(b) Number of installed PV in Emilia-Romagna.

that the region can put in place to provide a further incentive for the installation of PV (Borghesi et al. [2013]):

- Investment Grants - incentives are given as a grant, and no money is returned to the Region. The grants that are provided represent a proportion of the total plant cost. The financial requirement for the Region would be front-loaded as funds would need to be provided in advance of equipment installation.

- Fiscal Incentives - incentives are given as soft loans, including longer repayment periods or interest holidays. Again the financial requirement on the Region would be front-loaded as funds would need to be provided in advance of equipment installation. In this case the loan would, eventually be paid back to the Region.

- Interest funds -incentives are given to pay all or part of the interests on bank loans taken in order to purchase PV equipment. Again no money is returned to the Region. In this case the financial burden on the Region would be spread over the lifetime of the loans that are likely to be a number of years.

- Guarantee fund - the Region provides a guarantee to the bank providing the loan to an investor who is purchasing PV equipment that the loan will be repaid. This fund provides security to the bank that is, therefore, more likely to approve the loan request and to charge a lower interest rate than would otherwise be the case.

The goal is to find the best incentive that allows the greatest increase in installations with less effort. This intention requires a simulator able to reproduce the behaviour and interaction of families. This simulator must be integrated into a system capable of providing the right support for policy-makers to find the best solution to achieve the objectives. One important thing is that the simulator must provide a budget, for each type of incentive considered, that the region must allocate in order to obtain the desired increase of PV power.

The version of the software, which makes use of the model developed by Borghesi [2013] for the diffusion of PV systems, compares these incentives. The results are shown in the article Borghesi et al. [2013]. The purpose of Borghesi et al. [2013] is to identify the relationship between the capacity of PV systems installed and the budget allocated for regional incentives.

Each regional incentive was individually simulated multiple times for each value of the regional budget from zero to €40 million, in steps of €1 million. To learn the functions that govern the relationship between the available budget and the installed capacity, Borghesi et al. [2013] have used machine learning techniques. Figure 1.8 shows the simulation results. As you can see, the regional incentive that provides the greatest increase in capacity is the interest fund. In fact, the curve that relates the installed capacity and the available budget is almost always above the other.

The Interest Fund incentives are the ones that require the least amount of money for each installation. So, with this incentive mechanism the region can satisfy the vast majority of requests from citizens. Furthermore, the possibility of paying in installments allows the family to deal with the initial price of the system.

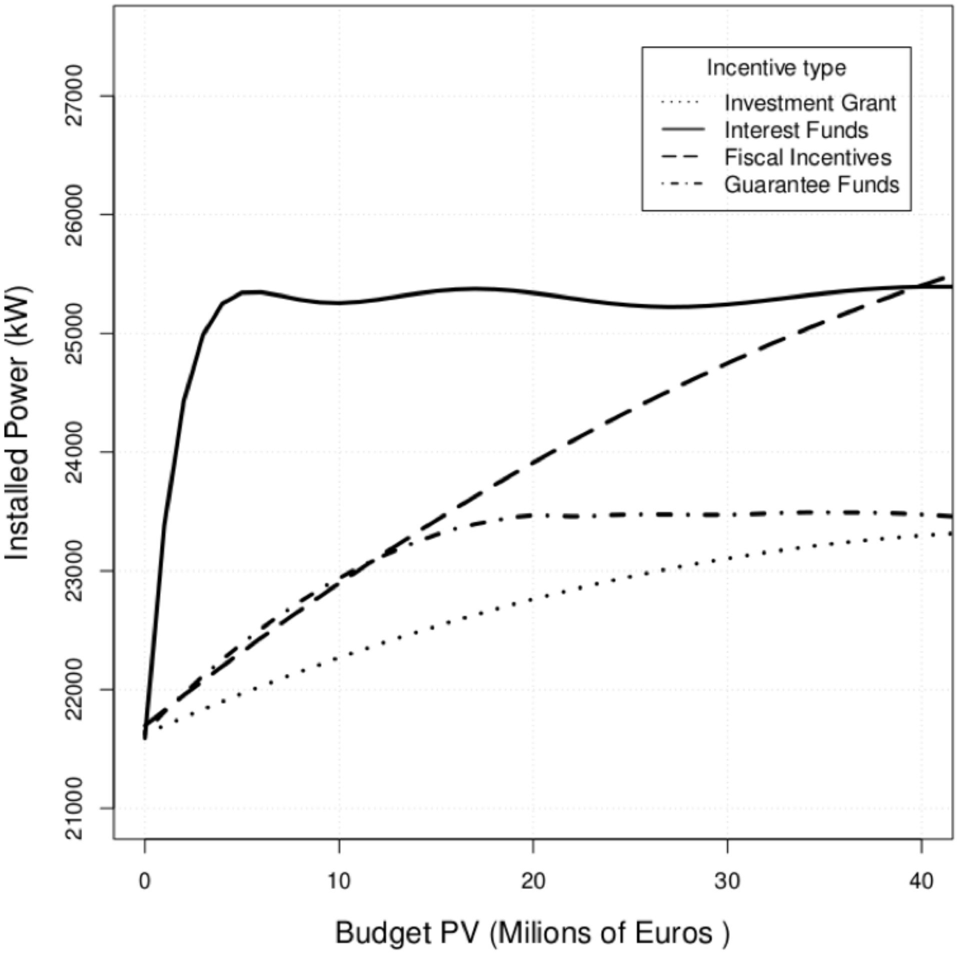


Figure 1.8: Comparison regional incentives Borghesi et al. [2013].

Chapter 2

Background

In recent years, Agent-based model has become increasingly popular as a modelling approach because it provides a systematical way to model the environment. This kind of model is mostly used in social science research to study how the environment evolves over time. In particular, ABMs are adopted to study the diffusion of an innovation that is strongly influenced by social aspects: people exchange information on the new idea (Rogers [2003]). This information reduces the uncertainty about the innovation and influences people in the decision whether or not adopt the innovation. An ABM can be used to model this information exchange between potential adopters to evaluate how the innovation spreads between them in different situations. For instance, a company can use it to determine the amount of advertising budget that should be allocated to reach the desired rate of spread.

For the reasons mentioned above, in this work we adopt Agent-based simulations for modelling the entities involved in a plan. In this context, entities can be individuals, companies, government agencies, etc., and then our model tries to reproduce their behaviour and how it is affected by economical, social and environmental factors.

Recently, ABMs were applied to model the diffusion of residential PV systems and evaluate the effectiveness of incentives. In this case, agents are households whose behaviour is represented by the decision to buy or not a photovoltaic system. An agent who has installed a plant increases the possibility that agents in the same neighbourhood decide to make the same choice.

This interest in the ABMs has led to the emergence of specific programming languages. These languages include constructs that facilitate the definition of the kinds of agents involved, their behaviour and their interaction. For our simulator, we used the development environment NetLogo (Wilensky [1999]). NetLogo provides a multi-agent programmable modelling environment that make easy to implement the model equipping it with GUI. The GUI allows the user to interact with the model parameter to explore the effects on the virtual world.

This chapter provides the reasons that led us to use an agent-based model for modelling the environment and an introduction to the related works.

2.1 Why an Agent-based model?

An agent-base model (ABM) is a computational model used to simulate the actions and interactions between agents. Using an ABM you can model the individual entities that populate the virtual world independently. Generally, entities are very simple, if taken individually, but their actions and interactions can reproduce complex phenomena.

ABMs provide a systematic approach for the development of a model. In fact, it starts with the identification of entities that are part of the model. Then development proceeds by determining the actions that an agent can perform to interact with and manipulate the environment around them. Once the environment where the agents operate has been defined the model is complete. The global behaviour is not specified in the model but arises from the behaviour of mere agents.

Of course, there are techniques that model the system at the macro level. These models are called macroscale models while the ABMs are a kind of microscale model. ABMs are easier to implement because simple behavioural rules leads each agent. In addition, the whole is greater than the sum of the parts (Bonabeau [2002]).

A field of application of ABMs is the simulation of natural systems. An example of natural systems is the ant colonies (Colorni et al. [1991]). The ants are the agents and follow simple rules. They randomly search for food, and upon finding it they return to the hive, dropping a pheromone trace which marks their trail. If another ant finds a pheromone trail, it will likely follow it. Ants that find the food source reinforce the pheromone trace in the track and as time passes the pheromone traces evaporate.

Figure 2.1 shows the NetLogo virtual world where the ants carry food back to the nest along the established route.

The emergent behaviour of the system is that the ants can find the shortest path to reach the food. The evaporation of the pheromone encourages the formation of a short path because in the long ones the pheromone has more time to evaporate.

In the case study of ePolicy, an ABM is used to model the spread of solar systems in the region Emilia-Romagna. As agents we considered the families inhabiting in the Emilia-Romagna Region and which could be interested in

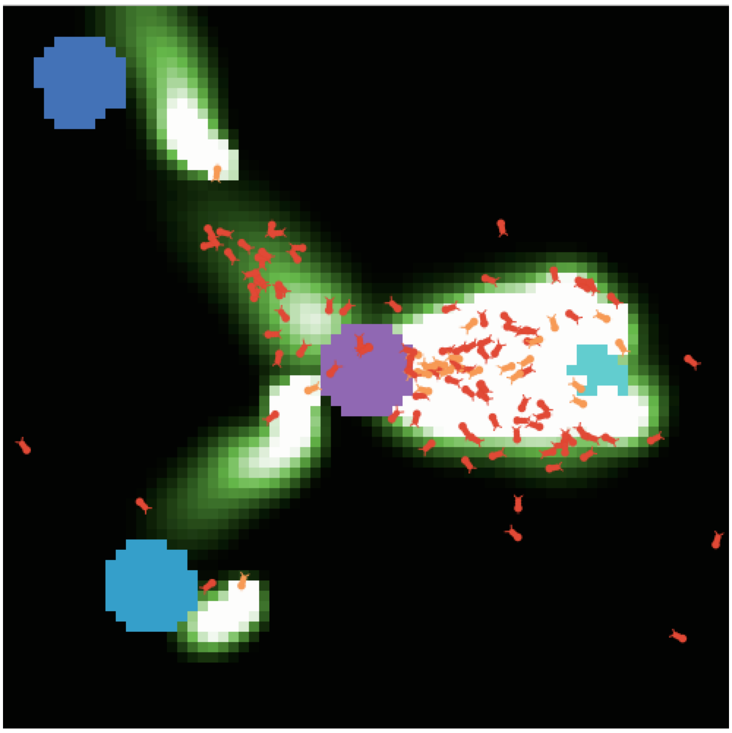


Figure 2.1: Ant colony simulation - NetLogo world representation Wilensky [1997].

the adoption of a PV system. The philosophy of the ABMs is K.I.S.S. (”Keep it simple, stupid”), and then the families are modelled in an easy way. In the proposed model, the agents calculate a utility function that determines the level of desire to adopt a PV system.
The researchers showed that the diffusion of innovation is strongly influenced by social aspects. People who have adopted an innovation spread their experiences, strengths and weaknesses, related to the innovation (Abrahamson and Rosenkopf [1997]; Chatterjee and Eliashberg [1990]). This communication reduces uncertainty about the innovative product and determines the degree of penetration among potential adopters. Thus, the ABMs can be used effectively to model this social aspect which is one of the main elements that drives the diffusion of photovoltaic systems.

2.2 Literature overview

Many scholars have tried to model the diffusion of innovations. Rogers M. claims that the diffusion of innovation is related with the communication between individuals, and the adoption is affected by the exchange of information. So, the innovation diffusion is a social process, and the communications between people play an important role in the decision to adopt or not the innovation. In this direction, Abrahamson and Rosenkopf [1997] have implemented a threshold model based on “bandwagon effect”. In this model, the increase of the adopters generates new information on innovation that produces a greater pressure on people who have not yet adopted the innovation. The potential adopters make an estimate on the profitability of innovation. However, they are unsure about the correctness of the assessment, so other people who have already adopted the innovation influence their decision. Abrahamson and Rosenkopf [1997] express this relationship with the following equation:

$$B_{i,k} = I_i + (A_i \cdot P_{k-1}) \tag{2.1}$$

Where $B_{i,k}$ is the bandwagon assessment of innovation at cycle k of the potential adopter i, I_i is the assessment of profitability of innovation and $A_i \cdot P_{k-1}$ is the bandwagon pressure. $P_{k-1}$is the amount of information received that create the bandwagon pressure after $k-1$ cycles and A_i denotes how much the potential adopter i weights this information. Also in the model proposed by Chatterjee and Eliashberg [1990], people influence each other in their decisions. The decision is based on two attributes: price and performance. The price is known, but the performance is uncertain and based on the perception that the potential adopter has of the innovation. This uncertainty is reduced over time because the potential adopter receives a stream of information about the performance by word-of-mouth from adopters.

Many of these models are agent-based model (ABM) where the agents are connected to form a small-world network. The small-world model was proposed by Duncan J. Watts and Steven Strogatz in their joint 1998 Nature paper. It consists of a random graph algorithm that produces graphs with the small-world properties that have high clustering coefficient and low mean-shortest path length. To prove the validity of this model, Stanley Milgram and other researchers conducted the small-world experiment to examine the average path length for social networks of people in the United StatesTravers and Milgram [1969]. This research has shown that human society is a small world network.

In a small-world network, most of the nodes are not neighbours, but most of them can be reached from every other by a small number of hops. In particular, the average distance between two nodes grows proportionally to the logarithm of the number of nodes in the network (Watts and Strogatz [1998]). This characteristic is obtained from a ring lattice where each node is directly connected to k immediate neighbours by random rewiring of some links.

Recently, some researchers have proposed specific models to describe the adoption of solar panels for domestic use. Zhao et al. [2011] have proposed a two level threshold ABM where agents are households. The low level is devoted to simulating each agent electric consumption and to provide and estimated payback time. Instead, the high level is related to model the customers' behaviour on adopting PV systems for 20 years. The adoption is based on four factors: payback period, household income, neighbourhood and advertisement. These factors are combined to define the desire level of a certain household for adopting a PV system. The model uses the following linear equation:

$$D_i = w_{pp} f_{pp} + w_{inc} f_{inc} + w_{nei} f_{nei} + w_{adv} f_{adv} \tag{2.2}$$

Where D_i is the desire level for the household i and w_{pp},w_{inc}, w_{nei} and w_{adv} are the weights associated with factors f_{pp},f_{inc},f_{nei} and f_{adv}. Each factor is represented by a value between 0 and 1. In order to have a desire level between 0 and 1, the next constraint is added:

$$W = w_{pp} + w_{inc} + w_{nei} + w_{adv} = 1 \tag{2.3}$$

If the desire level of the household exceeds the threshold, the household installs a PV system.

Palmer et al. [2013] proposed an ABM to estimate the PV system diffusion among households living in Italy. In particular, each agent represents a household characterised by eight attributes. These attributes are used to assign a cluster to each family. The clustering is based on Sinus Milieu ® groups formed by people that share similar characteristics. Moreover, agents are linked to form a small world network in such a way those who are in the same cluster are more likely to be linked together. The decision to invest on PV system is based on desire level proposed by Zhao et al. [2011]. The difference is that the weights used for each factor depend on the cluster of the family. Thus, people in the same conditions weight the various factors in the same way. The desire level (or utility function) $U(j)$ is calculated as:

$$U(j) = w_{pp}(sm_j)u_{pp}(j) + w_{env}(sm_j)u_{env}(j) + w_{inc}(sm_j)u_{inc}(j) + w_{com}(sm_j)u_{com}(j) \tag{2.4}$$

Where sm_j is the Sinus Milieu® group. As before, $u_{pp}(j)$ is the payback period factor, $u_{inc}(j)$ is the household's income and $u_{com}(j)$ represents the influence of neighbourhood and advertisement factors. Finally, $u_{env}(j)$ is added to take into account the environmental benefit of investing in a PV system.

Robinson et al. [2013] has proposed a model that uses a geographic information system (GIS) along with an ABM to study the diffusion of solar systems in order to take into account the real topology of the area of interest. In this case, an agent is mostly influenced by agents who have a similar opinion on technology. Each agent i has the variable x_i that represents its opinion and the variable u_j that represents its uncertainty. If an agent i has in its social network agents who have installed a photovoltaic system, the agent i randomly selects one of them, agent j, with a probability proportional to the similarity of opinions on technology. The relative agreement is calculated as follows:

$$\frac{h_{i,j}}{u_i} - 1 \tag{2.5}$$

where $h_{i,j}$ is the overlap of views between i and j and it is equal to:

$$h_{i,j} = \min((x_i + u_i), (x_j + u_j)) - \max((x_i - u_i), (x_j - u_j)) \tag{2.6}$$

The agent i opinion increases/decreases according to $h_{i,j}$. The opinion and the uncertainty of an agent j are updated as follows:

$$x_j = x_j + \mu((h_i j/u_i) - 1)(x_i - x_j) \tag{2.7}$$

and

$$u_j = u_j + \mu((h_{i,j}/u_i) - 1)(u_i - u_j) \tag{2.8}$$

where μ is the constant that controls the speed of convergence of opinions. Next, if the intention of the agent i is greater than a threshold, the system is compatible with its roof, the payback period is below the threshold and its budget can cover the expense, then the agent i installs the PV system.
In the next session, we introduce the previous work that is the basis of the proposed model.

2.3 Previous work

The original ABM, proposed by Borghesi [2013], simulates the diffusion of PV systems in Emilia-Romagna to understand the impact of regional incentives for a period between the first half of 2007 and the second half of 2036. During the simulation, new PV systems are installed by households until the second half of 2016 and the simulation proceeds until 2036 to cover the average life of a PV system that is estimated to be 20 years. On each step until the second half of 2016, new agents are added to the environment in a random position across the virtual world. The number of agents created each year is a parameter of the model. Each agent is characterized by:

- ID - an integer value used to distinguish agents;
- Roof area - the surface available for installing a PV system;
- Budget - the amount available for purchase a PV system;
- Average annual consumption - the average electricity consumption per year;
- The percentage of consumption that the agent wants to cover;
- Obstinacy - the agent's desire level to purchase a PV system.

Only agents that know PV technology perform an assessment, the others do not become part of the system. The knowledge diffusion is defined by the initial percentage of agents who are aware of PV technology and the yearly increase of this rate. The increase could vary following a linear relationship, a quadratic one or a cubic one. In the model proposed by Borghesi [2013], the impact of knowledge diffusion is very high: the annual installed power varies significantly by changing this parameter. Another factor that considerably impacts the simulated results is the annual increase of the percentage of agents who knows about PV panels. Using the different models for the growth of knowledge change how fast the knowledge increase each year.
Thus, when the agent is generated, the simulator determines, using a simple probabilistic model, if he knows or not the technology of solar panels, and if he knows, he makes an assessment. First, the agent establishes the annual kW that PV system must generate with the following equation:

$$annualkW = (Average\ annual\ consumption \cdot percentage\ of\ consumption) \tag{2.9}$$

Hence, the size of the system:

$$dimension = \frac{annual\ kW}{Annual\ average\ solar\ radiation} \cdot m2kWp \tag{2.10}$$

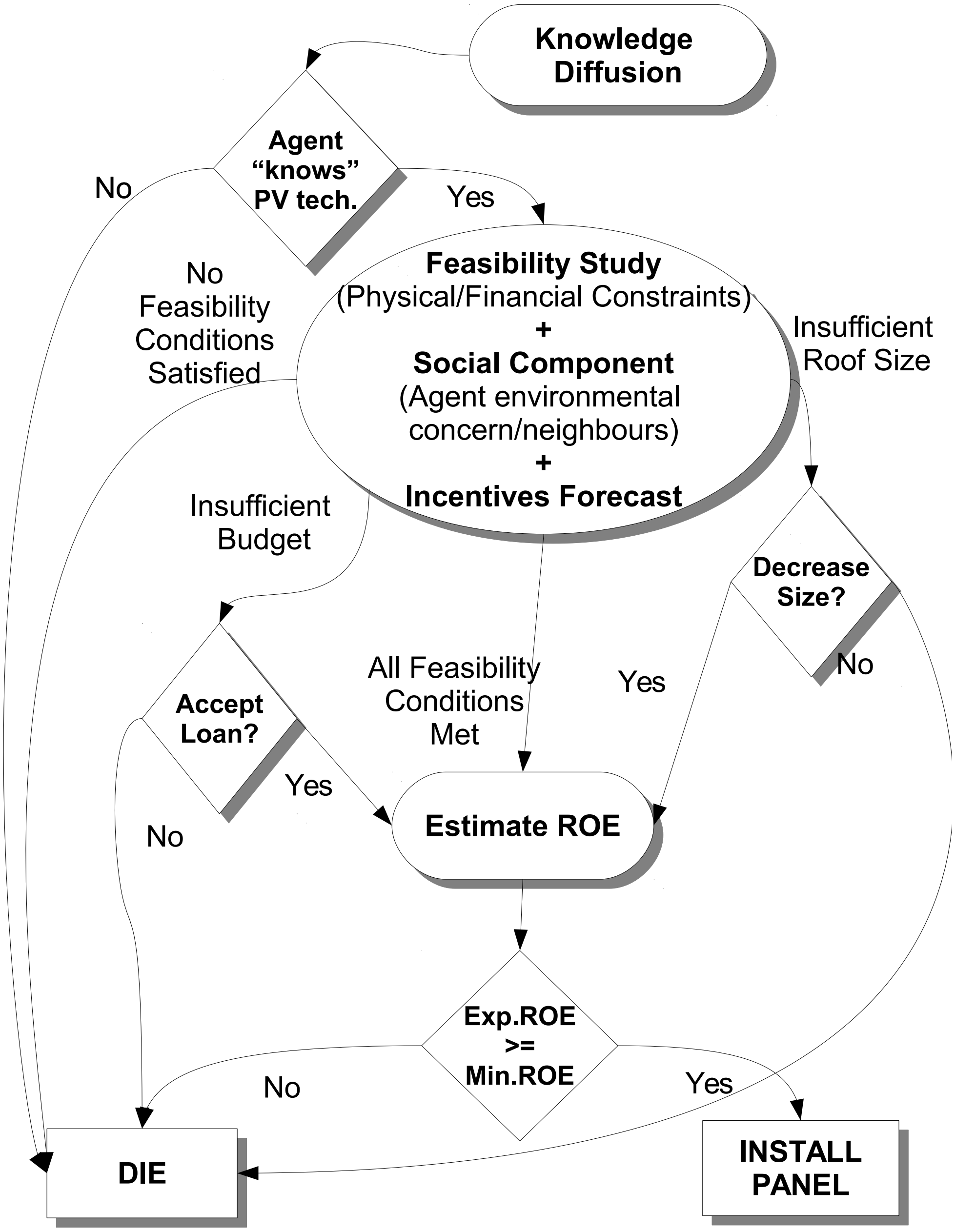


Figure 2.2: Decision algorithmBorghesi [2013].

Where $m2Kwp$ is the constant that relates the square meters with kWp of PV system. Once the size is determined, the price is calculated as follows:

$$price = kWp\ PV\ system \cdot average\ price \tag{2.11}$$

The simulation is subdivided in steps that represent every single semester. In each semester, the simulator creates a user-defined number of agents, which are spread in the virtual world. After the creation of an agent, the system proceeds to estimate its intention to buy or not a PV system. As shown in Figure 2.2, the steps that lead to the decision are:

1. If the PV system is smaller than the size of the roof and its cost is lower than the budget, the agent evaluates the possibility of increasing the dimension of PV system;

2. If the system is bigger than the size of the roof and its cost is greater than the budget, the agent leaves the system;

3. If the size of the plant is greater than the surface, but the budget is sufficient the agent evaluates to scale down the power of PV system;

4. If the size of the system is less than the available surface, but the budget is low, the agent evaluates the possibility to take out a loan.

In all cases, except in step two, the obstinacy of agent comes into play. The obstinacy of an agent increases with the growth of the number of neighbours who have installed a PV system. Agent's neighbourhood consists of agents whose distance is less than the radius specified as a parameter.The social interaction between agents is modelled as the sensitivity of an agent to the influence of neighborhood. The sensitivity is a value that can vary and a higher values correspond stronger influence from the neighbourhood. Thus, we expect higher probability to install PV panels.

In summary, each agent has a value (a component of obstinacy seen above) that represents how significantly his behaviour is influenced by friends, trying to reflect the human tendency to follow the group choice. In particular, the decisions of each agent are modified by his sensibility to neighbour's behaviours and the size of the area of influence, which is the radius that determines the circular area within the choices made by an agent may affect the actions of others. Figure 2.3 shows the NetLogo virtual world where the areas of influence are denoted by almost circular shapes centered on the houses that represent the agents.

In step 1, if the obstinacy is greater than 50%, the agent evaluates the possibility of increasing the PV system dimension. The dimension is set to the roof area, and if the budget is greater than the new price, the system is installed. Otherwise, the PV system is realized with the dimension calculated by the equation 2.10. In step 3, the agent accepts to scale down the power if the following constraint is verified:

$$(PV\ system\ size - roof\ area) \leq (roof\ area \cdot obstinacy) \tag{2.12}$$

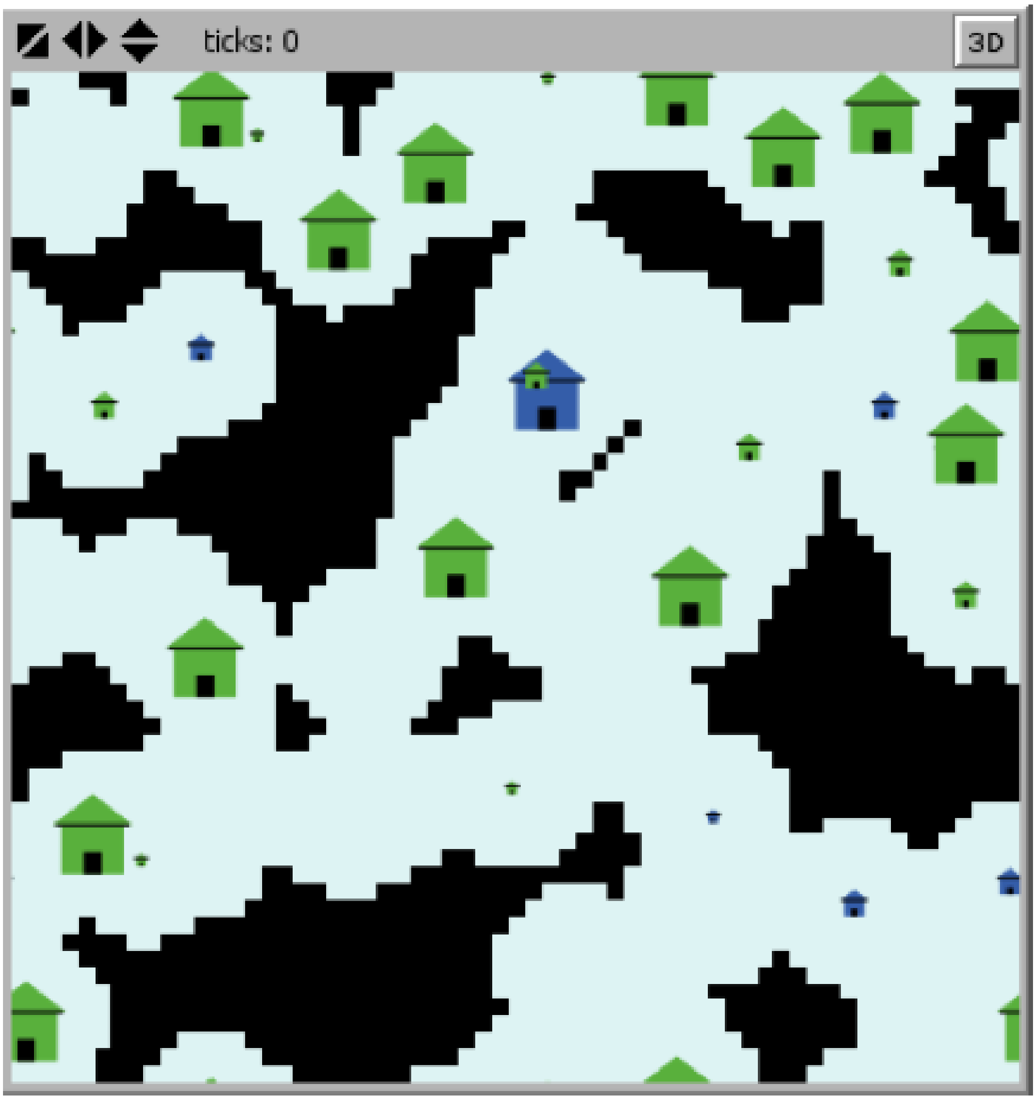


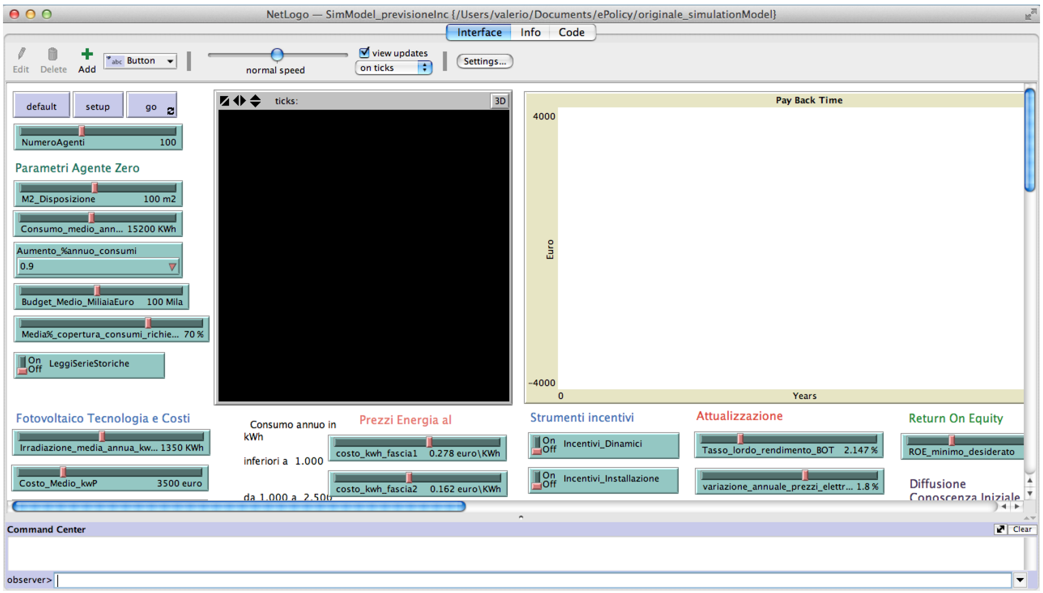
Figure 2.3: Social interactions shown in the virtual world of NetLogo.

Instead, in step 4, the agent takes out a loan to cover the price if the next constraint is verified:

$$(PV\ system\ price - budget) \leq (budget \cdot obstinacy) \tag{2.13}$$

In both step 2 and 3, if the condition is not met the agent leaves the system. Otherwise, if all the above conditions are met, the agent estimates the ROE of investment. The assessment takes into account the PV system cost, the national incentive (GSE price), the regional incentive, the energy price and eventually the mortgage payment. The gains are calculated as the sum of

the bill savings and the sale of energy. Thus, an agent installs a photovoltaic system if the estimated ROE is greater than a threshold.
Besides parameters with significant influence such as knowledge diffusion and agents interaction, the model has parameters that affect the results in a less evident way, such as almost all the pure economic factors. As an example of an economic parameter with a lesser influence than the previous ones is the annual reduction of the cost of PV panels. This parameter represents the reduction of prices due to technological advancements and in the model is a variable given as a percentage that tell how much each year the cost of a panel decreases in comparison to the last year price.
The plant comes into operation in the same year and semester in which the agent makes a decision. Each PV system is characterized by year, semester, type, technology, power band and dimension. The energy that can produce a PV system is linked to the geographical location and orientation. As a simplification, for all PV system the orientation is assumed to south with tilt to 30 °.



NetLogo — SimModel_previsioneInc {/Users/valerio/Documents/ePolicy/originale_simulationModel}

☑ view updates
on ticks

NumeroAgenti 100

M2_Disposizione 100 m2
Consumo_medio_ann... 15200 KWh
Aumento_%annuo_consumi 0.9
Budget_Medio_MiliaiaEuro 100 Mila
Media%_copertura_consumi_richie... 70 %
LeggiSerieStoriche

Irradiazione_media_annua_kw... 1350 KWh
Costo_Medio_kwP 3500 euro

Consumo annuo in kWh
inferiori a 1.000
da 1.000 a 2.500
costo_kwh_fascia1 0.278 euro\KWh
costo_kwh_fascia2 0.162 euro\KWh

Incentivi_Dinamici
Incentivi_Installazione

Tasso_lordo_rendimento_BOT 2.147 %
variazione_annuale_prezzi_elettr... 1.8 %

ROE_minimo_desiderato

observer>

Figure 2.4: The simulator GUI.

Once the execution of the simulation is finished, the GUI (Figure 2.4) shows a variety of information, many of them are attractive to investors, such as PBT and the average ROE for different semesters, others are useful to determine the characteristic of the simulated environment, such as the overall installed power, the total expenditure for the installation and the percentage of plants built.

2.4 Development tools

NetLogo (Wilensky [1999]) is a programmable modelling environment for simulating natural and social phenomena. The development environment is written in Java to be as independent as possible from the execution platform. The NetLogo language inherits and extends the features of multi-paradigm programming language Logo, made in the 60s at the Massachusetts Institute of Technology and characterised by its derivation from Lisp and the numerous applications in the field of education. The code that defines agent's behaviours is interpreted, so the model is not previously compiled into machine-language instructions.

The development environment consists of a graphical interface organised into three tabs: interface, info and code. The interface tab allows you to interact intuitively with the parameters that govern the model or perform actions through the use of buttons, sliders, or other items. This interface has the important function of showing the movements of the agents inside the virtual world and present during and after the simulation information in the form of charts, tables, etc. The info tab provides information on the model. Besides the info tab there is the section that relates to the code, which defines the behaviour of the entities that act in the virtual world. The code of the simulation resides all within a single list, and it is divided into several procedures. The agents, entities that can execute instructions, can be of four

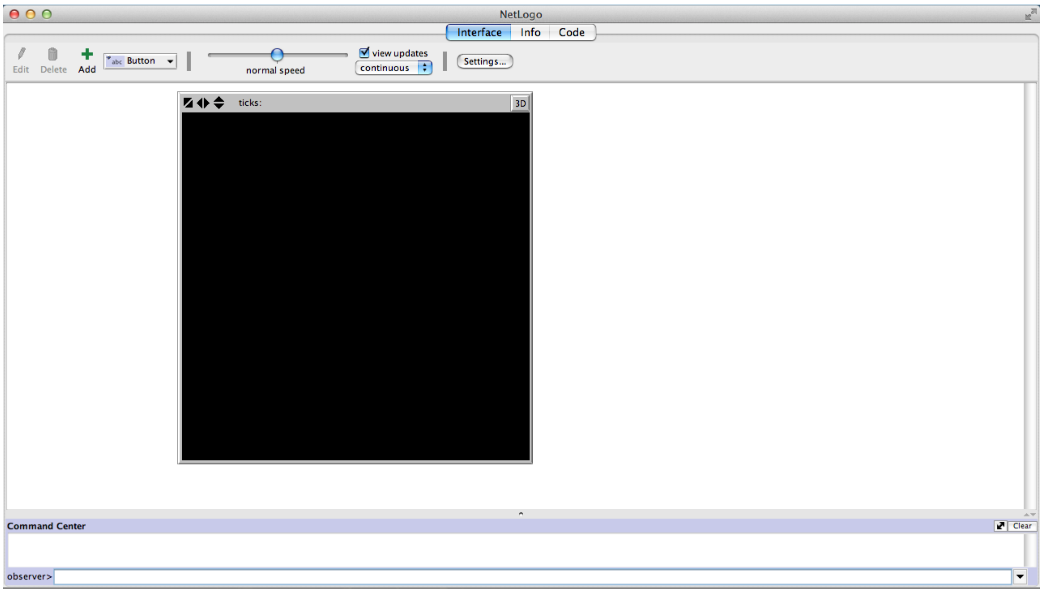


Figure 2.5: NetLogo interface.

types (Figure 2.6):

- Patches are organized in the grid to form the two-dimensional world;
- Turtles move on this grid;
- Links are agents that connect two turtles;
- The observer oversees everything that happens, and it does everything that the turtles, links and patches cannot do.

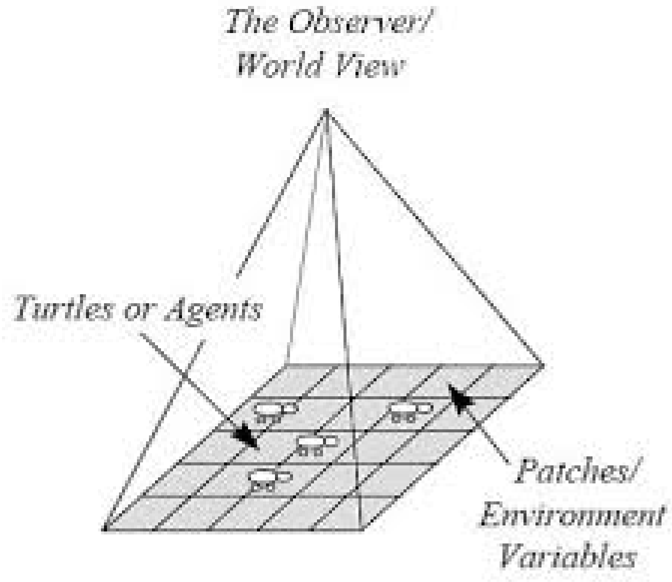


Figure 2.6: The virtual world of NetLogo.

All agents can issue commands and procedures. A Command is a Logo instruction that an agent can perform to interact with others or to change its state. Instead, the procedures combine a series of commands in a new command.
In NetLogo, you can define breeds of turtles or links. Breeds allow you to divide agents and then define specific behaviours. For each type of agents, NetLogo provides an agentset that is a set of agents. An agentset allows executing a series of commands to all or part of the agents in it. The agentset makes the code cleaner and more readable and facilitates the implementation of the model.

NetLogo provides two ways to update the simulated time: continuous or tick-based. The continuous mode updates the view a user-specified number of times per second. In tick-based models, an update occurs when the instruction *tick* is executed. The continuous mode is more expensive because the updates are more frequent than the tick-based, resulting in greater use of resources. Also, since we are not able to manage when the model needs to be updated, the system may be in an inconsistent state when the simulation is stopped. Usually, the continuous mode is used for debugging because it allows checking in detail how the system evolves.
NetLogo makes it possible to perform many times a simulation and try different configurations of parameters to study the results. These results help to understand the emergent behavior due to the interaction of multiple agents.

Chapter 3

Proposed Model

This chapter describes the model proposed to simulate the diffusion of photovoltaic systems. First, we proceed to investigate the related works, and then we are going to explain our solution to address the problem. In Italy has been put in place a national feed-in-tariff to stimulate the installation of PV systems. The region may apply a greater incentive to reach the 2020 target. The 2020 climate and energy package (European Commission [2009]) is an ambitious goal that aims to arise the share of EU energy consumption produced from renewable resources to 20% before 2020. We consider four types of incentives that a region can implement for the adoption of PV systems:

- Investment Grants - money given by the region to a household so that it can invest in PV system;
- Fiscal Incentives - the region provides loans with low-interest rates;
- Interest funds - the region pay part or all of the interest that the citizen owes the bank;
- Guarantee fund - the region guarantees for those who want to take a bank loan. In this way, it is easier to get a loan.

In this work, we propose a tool for decision makers to evaluate these regional incentives. This tool requires a model that can predict the photovoltaic system diffusion among households. Therefore, we propose an agent-based model that simulates the micro-based behaviour of households in order to evaluate and explain macro-level phenomena. We focused mainly on families living in the region of Emilia-Romagna, but the process described below is valid for any region or country.

The goal of proposed simulator is to recreate the phenomena of PV system distribution, attempting to create a relationship among families to simulate

the diffusion of knowledge about the advantages of PV systems. These families are placed through an actual density distribution on the virtual world. The simulation consists of two major phases: the configuration phase, where the simulator creates a virtual word that has features similar as possible to the real and the running phase where the system simulates the degree of adoption among the agents from the first half of 2007 to the second half of 2016.
In the configuration phase, households are generated. Each family is characterized by attributes such as income, age and education level of main income earner. The distributions of these attributes are obtained from the Survey on Household Income and Wealth (SHIW) provided by Bank of Italy [2012]. After that, each family is assigned to a social group composed of families who share similar conditions in order to have a similar behaviour on families that have similar wealth. Then, each of them is assigned to a building on the simulated world. These buildings were obtained by processing the shapefiles that the region provides on the website Emilia-Romagna [2014].
In the running phase, the simulator simulates the behaviour of households for a period from the first half of 2007 to the second half of 2036. Annually, each household proceeds to evaluate the adoption of PV system. In the model, the desire level for adoption of a PV system is estimated by means of a utility function that an agent calculates according to its characteristics. In our system every agent makes the best choice that is the PV system that maximise its reward, in terms of the production and saving, because we want a generic simulator that can simulate different incentive. Thus, we do not kwon what is the best size of PV system a priori. Normally, families ask for advice from installers and consultants, so it is fair to assume that in making the decision a family makes the best choice. So, an agent estimates the optimal size of the PV system that guarantees the best return on equity (ROE).
If the value of the utility function exceeds the threshold, the agent installs the system. The utility function takes into account the income of the family, the payback period, the environmental benefits and the relationships with other families. These factors are weighted differently depending on the social group of the family. The weights for each group are determined by calibrating the model on real data over the 2007-2012 period.

3.1 Model description

In the proposed model, the agents represent the families living in the region of Emilia-Romagna. As already mentioned, the simulation is divided into

two phases. In the first phase, the households are generated. In the second phase is simulated the behaviour of the agents.
The generation begins by establishing for each household age class, education level, income, family size and budget. The distribution of each attribute is obtained from the SHIW data. In addition, each agent is assigned to a class that represents a group of people who share the same characteristics. This class influences the value of the utility function.
The budget of a household for the purchase of a PV system is derived from its income by the equation:

$$budget = e^{inc_{norm}} ap \tag{3.1}$$

where:

- ap is the average price of a PV system;
- inc_{norm} is normalized income obtained from $\log \frac{income - m}{v}$;
- $v = \sqrt{2}\Phi^{-1}(\frac{Gini + 1}{2})$ is the lognormal distribution variance;
- $m = \log m_{inc} - \frac{1}{2}v^2$is the lognormal distribution mean.

The equation states that if a family has the income around the mean, the family expects to pay the average PV system price. Otherwise, if the family income is lower or higher than the average, the family will aim to spend less or more for a PV system.

Then each family is associated with a building on the territory of the region taking into account the family size and income. Buildings are sorted by their roof size and families with high income and high number of members are assigned to the bigger ones. The buildings are obtained from Ersi shapefiles that Emilia-Romagna region has released on the website Emilia-Romagna [2014].
The Shapefile or simply shapefile is a geospatial vector data format for geographic information systems software. This format was developed and regulated by ERSI, in order to improve interoperability between GIS systems. The shapefile describes points, polylines and polygons, which represent objects placed on the map. Normally, "shapefile" refers to a set of files with the extension Shp, Dbf, Shx and others that share the same name. As shown in Figure 3.1, the shapefiles provided by region contain a polygon for each building detected. Since the simulation requires only the positions and the

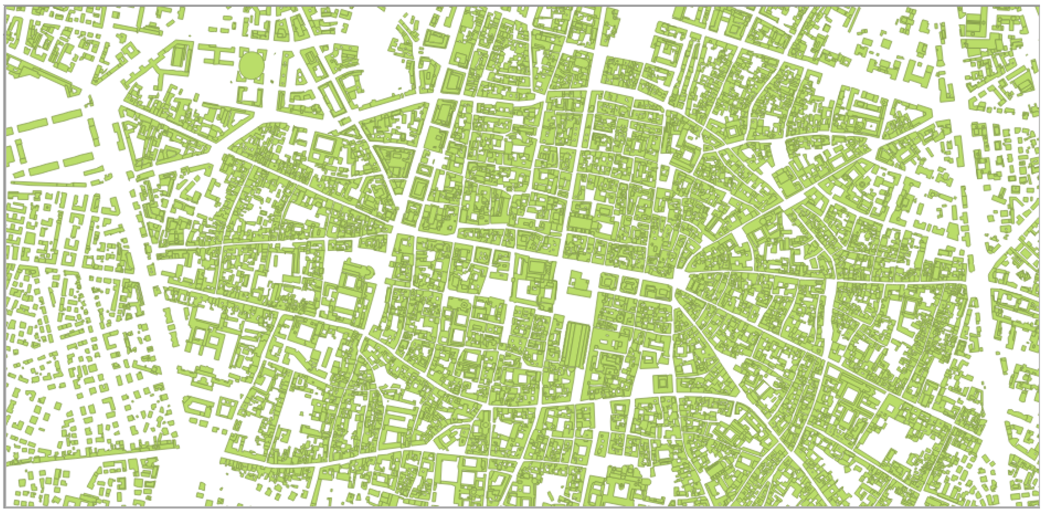


Figure 3.1: Polygons of buildings contained in shapefiles.

areas of the roofs, it was necessary to preprocess these files. Using QGIS (QGIS Development Team [2009]), a free and open source Geographic Information System (GIS), it was possible to manipulate these shapefiles. QGIS is a very powerful tool, which allows to capture, store, manipulate, analyze, manage, and present all types of geographical data. It was possible to calculate the area and the centroid of the vertices for each polygon.

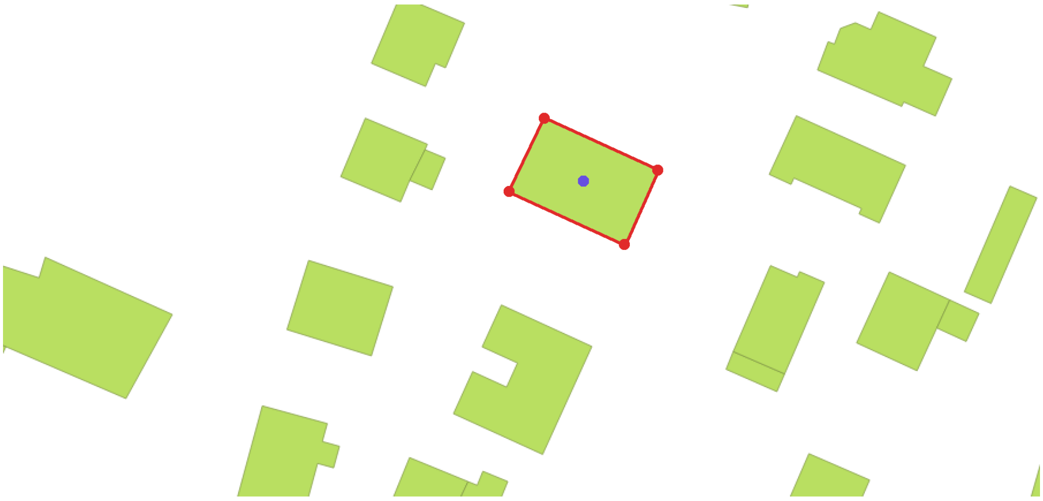


Figure 3.2: Buildings preprocessing.

Figure 3.2 shows the preprocessing performed with QGIS. For each polygon, we calculate the area, and we keep only the centroid (the point in purple) of vertices (the points in red). In the model, the polygon area is assumed as roof area, and the centroid is assumed as the location of the house on the map.

As said before, shapefiles alone are not sufficient to describe the buildings because they contain only the geometries. So, the region also provides the dbf files that describe each building with several attributes, including the TY_ EDI that specifies the type of the building and STAT_ E, which indicates the state of the building.

TY_EDI	D_TY_EDI
1	Generic
4	Bell tower
6	Church / basilica
7	Industrial building
9	Rural building
12	Mill
13	Observatory
14	Palace tower / skyscraper
15	Sport hall
18	Palace tower / skyscraper
19	Villa
20	Townhouse
97	Not known
98	Not assigned
99	More
701	Shed
702	Hangar

STAT_E	D_STAT_E
1	Operational
2	Under construction
3	Abandoned / ruined

Table 3.1: The possible values for TY_ EDI and STAT_ E

Thus, through QGIS it was possible to obtain only buildings that are mostly houses using the following query:

TY_EDI = 1 or TY_EDI = 19 or TY_EDI = 20 and (STAT_E = 1)

Buildings with a roof surface too small were discarded because a one kWp rooftop solar plant requires at least 8 square meters with the technologies considered in the model.

The use of actual buildings allows us to reproduce the characteristics of the area in the virtual world. The characteristics of the territory are the arrangement and size of the buildings. Generate the buildings arranged as the real ones and whose dimension reflects the real ones is not simple. In addition, the actual building allows us to spread the agents in the virtual world assigning each agent to the home. In this way, we can build a social network taking into account the position of the agents on the virtual world. A small-world network is obtained from a regular lattice by adding some random links. However, we have agents arranged in an almost "random" manner on the map. So, in the next section we explain how we create a social network in order to get the small-world properties in our simulated social network.

3.2 Network generation

Families are connected together to form an extensive social network. The network of relationships plays an important role in the model because it specifies how information is transmitted between the families. As mentioned earlier, it is important that the generated social network has the small-world properties because researchers have shown that the small-world network maps well the real network of relationships that exists between people. Since the families are geographically distributed on the region, we need to find a system to generate a network in such a way that it gets high clustering and short paths properties.

A technique for achieving this is the rank-based model proposed by Liben-Nowell et al. [2005]. They analyzed roughly 500.000 users of the blogging site LiveJournal, who provided a U.S. zip code for their home address and links to their friends on the system. In this way, Liben-Nowell et al. [2005] were able to discover that the probability that a node u is connected with another w is related to the physical distance. Figure 3.3 shows the population density in the LiveJournal data.

However, the population density is non-uniform so, if we define the probability that a node u is connected with a node v as $1/d^2$, an agent who lives in a sparsely populated area is less likely to have links with other people. Liben-Nowell et al. [2005] claimed that two people living 500 meters away in a sparsely populated area are more likely to know each other than two people who live at the same distance in a densely populated area. Therefore, Liben-Nowell et al. [2005] defines the agent's proximity as rank:

$$rank_u(v) =\mid \{w : d(u,w) < d(u,v)\} \mid \tag{3.2}$$

A node u ranks a node w as the number of other nodes that are closer to v than w is. Now, the probability that the node u creates a link with the node w is proportional to $rank_u(w)$. However, we have more information

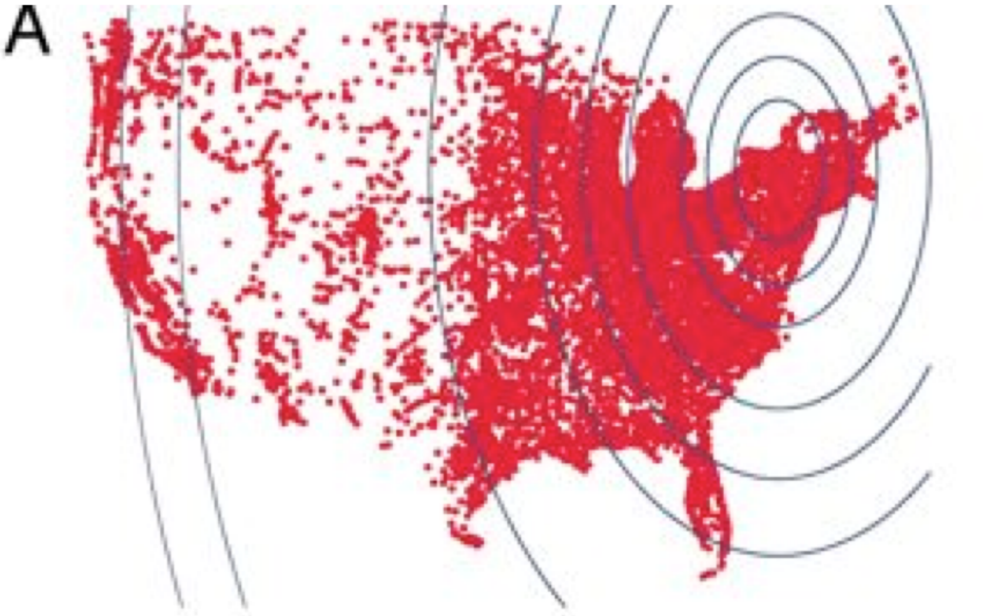


Figure 3.3: The population density of the LiveJournal Liben-Nowell et al. [2005] (Image from Liben-Nowell et al. [2005].)

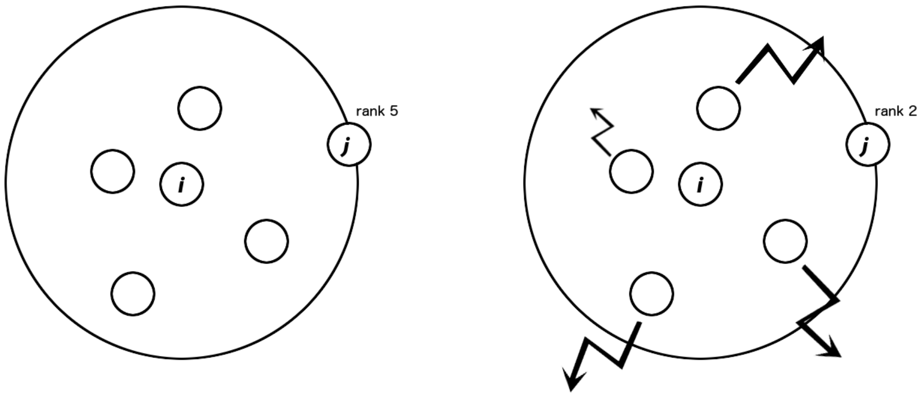
about nodes. As previously mentioned, nodes are heterogeneous, and they are characterized by age class, education level, and income. Thus, we can use this information to extend the ranked based model, so the rank that a node attaches to another node does not depend only on the physical distance but also on the attributes proximity of nodes. Hence, we can define the rank as:

$$rank_u(v) =\mid \{w : p(u,w)d(u,w) < p(u,v)d(u,v)\} \mid \tag{3.3}$$

where, $p(u,w)$ is a proximity measure between the attributes. We use a dissimilarity function defined as:

$$p(u,w) = 1 + \frac{\frac{\mid u_{age} - w_{age} \mid}{4} + \frac{\mid u_{edu} - w_{edu} \mid}{7} + (1 - e^{-\mid u_{inc} - w_{inc} \mid})}{3} \tag{3.4}$$

In this way, as shown in figure3.4b, nodes that are different from the node u are rejected and thus have a lower probability of having a link with u. The rationale behind is that people who have similar age, a similar level of education and similar economic opportunities are more likely to know each other because they have more opportunity to meet. In Figure 3.4, the circles represent the households, and the arrows represent the repulsion expressed



(a) Original rank-based friendship. (b) Extended rank-based friendship.

Figure 3.4: Comparison of rank-based model with homogeneous and heterogeneous nodes.

by the equation 3.3 when two nodes do not have exactly the same attributes. Using the equation 3.2 we do not consider the differences between the nodes. The result of this method is shown in Figure 3.4a. But, if we use the equation 3.3 for calculating the rank, the result is what we can see in Figure 3.4b, where the different nodes are moved away in proportion to $p(u, w)$. This shift causes a different ordering of the nodes, so the nodes that were closer to i than j are now farther than j.

In a small-world network, there are two types of links: the homophilous links and the weak ties (Easley and Kleinberg [2010]). The homophilous links connect node that are similar. Instead, the weak ties connect in a random way two nodes. The homophilous links are created by using the extended rank-based friendship. Two nodes located close together, and similar are more likely to share a link. In this way, we manage to get a high global cluster coefficient that is a characteristic of small-world network. However, this do not allow us to get a short average path length. So, we decide to randomize the network after we have built a "regular" network (shown in Figure 3.5) that is the network made by only homophilous links.
The network randomization adds to the network the weak ties that are long range links. These links reduce drastically the average path length because they connect distant parts of the network. The randomization process takes every edge and rewires it with probability p.

Different values for p produce different results. Figure 3.6 shows the networks obtained from varying p. As the rewiring probability increases, the

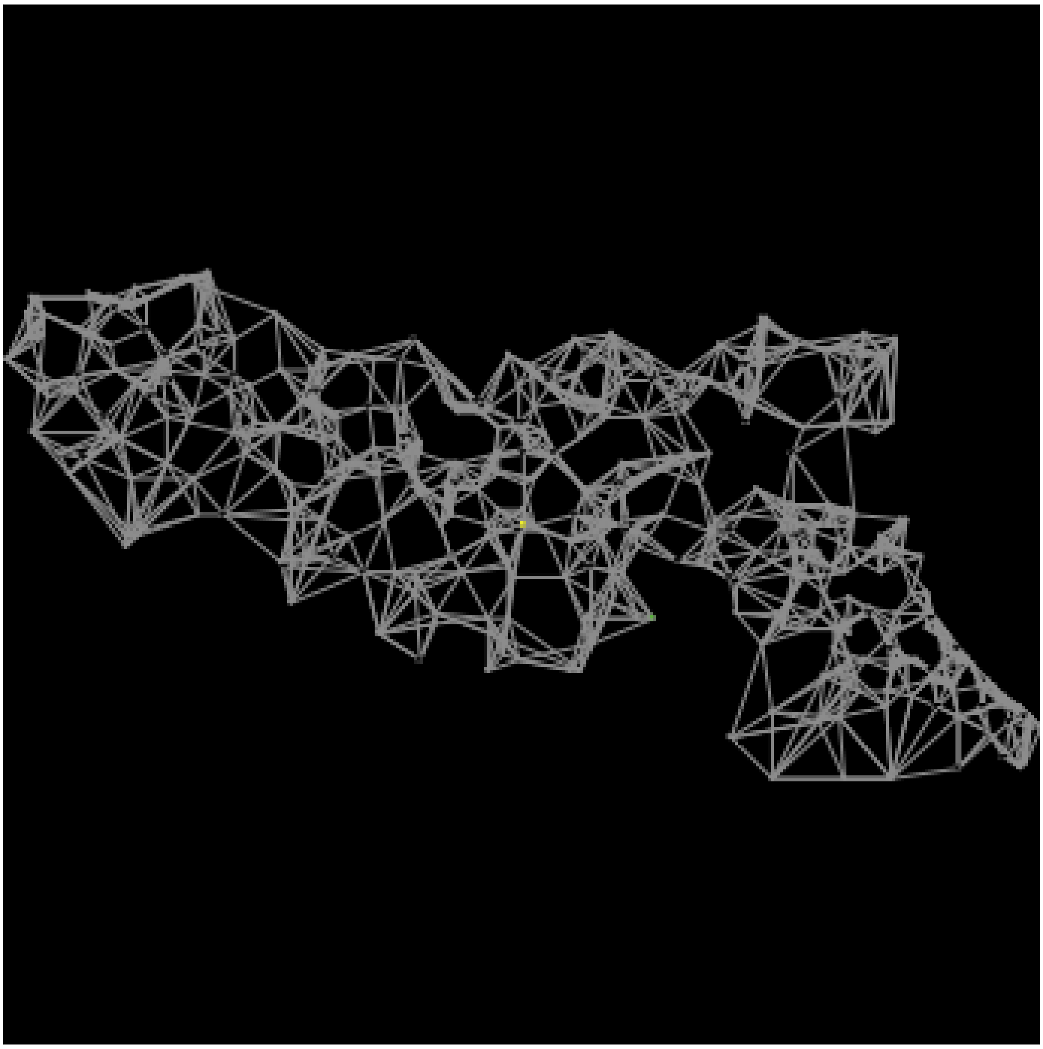
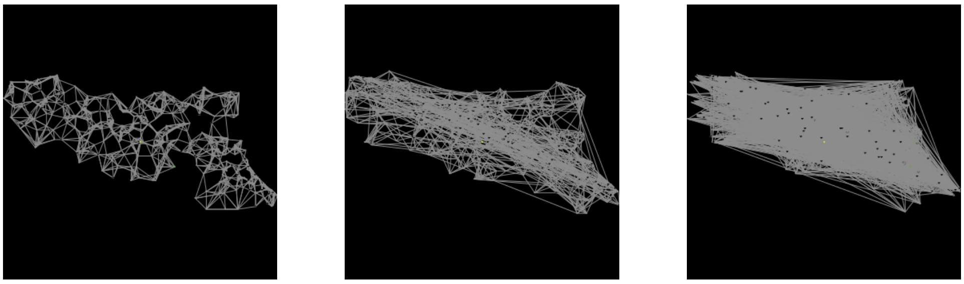


Figure 3.5: The resulting network made only with homophilous links

network becomes more irregular and then the clustering coefficient increases, but the average path length decrease. In summary, high p values produce low clustering coefficient and short average path length. Instead, low p values produce high clustering coefficient and long average path length.
In the model, the clustering coefficient and the average path length affect the speed of information spread and how information flow. In fact, a high clustering coefficient value implies that the information is more easily exchanged between nodes that are closer. On the other hand, a short average path length allows information to reach a remote area of the network faster. This results in a diffusion of information not limited to a portion of the network



(a) $p = 0$ (b) $p = 0.15$ (c) $p = 1.$

Figure 3.6: The resulting networks from varying p.

but, the information can get out of a cluster of people and reach remote areas.
As we will explain later, the adoption behaviour of an agent is affected by choices made by its neighbours. In particular, as the number of agent's neighbours that have adopted a PV system increases, the agent's pressure to install a PV system increases as well. Since, the neighbourhood of an agent i is defined as the agents that share links with i, another important factor that affect the PV system diffusion is the maximum node degree, namely the number of links that an agent has. The maximum degree determines the maximum number of neighbours of an agent. So, if an agent has many neighbours, it is less influenced by the choice made by a single neighbour. Otherwise, if an agent has few neighbours, the adoption of the agent is most affected by the choice made by a neighbour.
In addition, the network degree influences the clustering coefficient and the average path length because, along with the number of agents, it also determines the number of links. A high number of agents worsens the clustering coefficient and the average path length because many agents implies a large network with multiple paths. Instead, a high node degree means that there are more ways to reach a node. So, if we rewire a link of a node bringing it in another zone of the network, in order to reduce the average distance with nodes that are located there, the node may still have other links with nearby nodes. Therefore, as the node degree increases, the clustering coefficient and the average path length decreases.

In summary, we need to find a compromise between the number of agents, node degree and rewiring probability in such a way to get high clustering coefficient e short average path. These small-world characteristics were also found in the real social network by researchers, therefore, is important that our generated networks have the same characteristics. We start the network generation by placing the households into the virtual world as the real one. After that, we wire them by adding links in such a way that similar and nearby families are more likely to share a link. Next, we randomize the network by rewiring links with probability p. The rewiring allows us to add some long range connections in order to reduce the average path length with the cost of increasing the clustering coefficient. The result is a network that allow the stream of information to reach all its parts quickly.

In order to create an element of uncertainty, an agent can break up a link with a certain probability and randomly reconnect to another agent. In this way, the network is not static but is dynamically changed during the simulation to allow a flexible exchange of information between agents. Since the interactions impact the agent adoption, the rewiring modifies the social interactions thus it is a decisive factor in the final result. The rewiring probability is identified by the model calibration.

3.3 The household behaviour

In the proposed model, the behaviour of the agent has been completely revised. First, each agent determines the right size for a PV system that allows him to get the maximum gains under the constraint on the size of the roof. To measure this gain, we decided to use the return on common equity (ROE), a measure of profitability that calculates the ratio between the net income and equity. To determine if the ROE is good or bad, it is compared to the performance of alternative investments such as BOT, CCT, bank deposits,etc. The estimation of the ROE takes into account costs and gains for a period of 20 years which is the estimated lifetime of a PV system. The procedure for estimating the ROE calculates the cash flow for each year. The cash flow is calculated as the difference between total earnings and total expenditure related to the PV system for a period of one year. The expenses that are taken into consideration are:

- The cost of the system is calculated by the equation 2.11;
- Maintenance costs;
- Interest to the bank / region.

The sources of income are:

- Electricity bill savings due to the self-consumption;
- Sales to the grid operator.

Some of these costs and earnings are based on global model parameters. The global variables that we use for the assessment of feasibility are:

- The electricity prices charged to final consumers (divided into five bands of consumption);
- The annual change in electricity prices that is a dynamically modifiable parameter;
- The average cost of PV panels per kWp;
- The incentives for the installation that are the principal mechanism by which the region can influence the choices of the agents;
- The minimum prices guaranteed by the GSE for the dedicated withdrawal that is an instrument for the sale of electricity on the market. It consists in transferring the electricity to the GSE that recompenses the producers by paying a price for each produced kWh.

The amount of energy that is sold to the operator and the amount of energy self-consumed depend on the energy produced by the plant and the consumption of the household. In particular, the energy consumed $e_{consumed}$ by an household i is assumed to remain constant over the period. Instead, the energy produced $e_{produced}$ by the plant p decreases over the years and it is calculated as :

$$e_{produced_{p,y}} = e_{produced_{p,y-1}} + (e_{produced_{p,y-1}} * \text{efficiency}_{loss}) \tag{3.5}$$

where efficiency$_{loss}$ is the solar panel degradation rate. According to research carried out by independent institutions in the field, the performance of a new photovoltaic decreases by 1% per year, so after 20 years makes 80% of what was initially. Then, using the equation 3.5, we can calculate the amount of energy sold e_{sold} during the year y by the household i as follows:

$$es_{sold_{i,y}} = \begin{cases} e_{produced_{p,y}} - e_{consumed_i} & \text{if} \quad e_{produced_{i,y}} > e_{consumed_i} \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

where $e_{produced_{p,y}} - e_{consumed_i}$ represents the difference between production and consumption. If the production of energy exceeds the household consumption, then the household sells the surplus to the grid operator. Similarly, the energy self-consumed$e_{self-consumed}$ by the household i in the year y is calculated as:

$$e_{self-consumed_{i,y}} = \begin{cases} e_{consumed_{i,y}} & \text{if} \quad e_{produced_{i,y}} > e_{consumed_i} \\ e_{produced_{p,y}} & \text{otherwise} \end{cases} \tag{3.7}$$

The amount of earnings depends on the year and semester of entry into operation because it determines the Conto Energia applied. From the second to the fourth CE, the incentive fee is applied to all the energy produced by the plant. So, in this case, earnings are calculated as follows:

$$revenue_{i,y} = e_{produced_{p,y'}} * incentive_{y',s'} + es_{sold_{i,y'}} * GSE_{minprice} + e_{self-consumed} * e_{price_y} \tag{3.8}$$

where y' and s' are, respectively, the year and the semester of installation of the system, e_{price_y} is the price of electricity for the year y' and $GSE_{minprice}$ is the minimum price guaranteed by the GSE in the year y' . The GSE minimum price also depends on the power band of the system.
The Fifth Conto Energia redefines the incentives given for the production of electricity from photovoltaic sources. In this case, for systems with nominal power up to 1 MW is provided an all-inclusive tariff determined on the basis of power. So the tariff payable is the sum of the all-inclusive tariff on the share of production fed into the grid and the premium rate on the share of production consumed.

$$revenue_{i,y} = e_{self-consumed} * incentive_{self-consumed_{y',s'}} + es_{sold_{i,y'}} * incentive_{all-inclusive_{y',s'}} \tag{3.9}$$

Now we have the elements to calculate the cash flow for the year y:

$$F_{i,y} = revenue_{i,y} - expenses_{i,y} \tag{3.10}$$

We can calculate the cumulative discounted cash flow (CDCF) as follows:

$$CDCF = \sum_{t=1}^{N} \frac{F_{i,y}}{(1+r)^t} \tag{3.11}$$

where r is the discount rate. The main reasons for which the series of future cash flows are discounted to present value is related to the fact that earnings close in time to the initial investment can be reused to obtain new profits. So with the discount you give more weight to earnings closer in time.
A household solves the optimization problem 3.12 to find the size of the system that provides the highest ROE.

$$\begin{aligned} \text{max} \quad & \text{ROE} \\ \text{subject to} \quad & size \leq roof_{area} \end{aligned} \tag{3.12}$$

So, the size of the PV system is the one that maximizes the ROE. The budget constraint is relaxed because the agent can take out a loan if it is not enough. In order to solve the problem 3.12, we decided to use the simulated annealing algorithm because it is able to find a good solution in a short time. Simulated Annealing is a metaheuristic paradigm that was proposed by Kirkpatrick et al. [1983] in 1983 to solve optimization problems. This paradigm aims at finding a global minimum when there are multiple local minima. The name and inspiration come from annealing processes in metallurgy. According to the laws of statics, a system where:

- s is a state
- $f(s)$ is the energy of the state's
- T is the temperature of the system

fluctuates from one state to another with probability given by :

$$e^{\frac{-f(x)}{kT}} \tag{3.13}$$

where k is Boltzmann constant. This process is simulated starting from an initial solution that represents the initial state of the system. Then the algorithm generates a new solution starting from the current state and explores the neighbourhood of the current solution and selects one. It then goes on to calculate the value (energy state) of the new solution. The new solution is accepted with probability:

$$e^{\frac{-(f(s_{new})-f(s_{old}))}{T}} \tag{3.14}$$

where $f(s_{new})$ and $f(s_{old})$ are respectively the energy of the new solution and the old solution. At each cycle, the temperature is decreased, and the process ends when it is lower than the threshold. It returns the best result that has been found. The pseudo-code is:

```
s = s0; e = E(s)
sbest = s; ebest = e
k = 0
while k < kmax and e > emax
  T = temperature(k/kmax)
  snew = neighbour(s)
  enew = E(snew)
  if P(e, enew, T) > random() then
    s = snew; e = enew
  if enew < ebest then
    sbest = snew; ebest = enew
  k = k + 1
return sbest
```

Listing 3.1: Simulated anealing peseudo code Wikipedia [2004]

Note that the probability of accepting a worse solution is smaller than that of accepting a better solution, however, is not zero. This aspect allows the algorithm to circumvent the local minima partially.

We use the simulated annealing algorithm to find the best value for the ROE function. The implementation of the algorithm applies the ROE function to a new plant size for estimating its ROE. The ROE function takes into account any regional incentives and any mortgage payments.

When the size of the system has been established, the agent calculates the utility function which is based on the one proposed by Palmer et al. [2013]. In particular, the utility function used is the following:

$$U(v) = w_{pp}(cls_v)u_{pp}(v)+w_{budget}(cls_v)u_{budget}(v)+w_{env}(cls_v)u_{env}(v)+w_{com}(cls_v)u_{com}(v) \tag{3.15}$$

where, $w_{pp}(cls_v)$,$w_{budget}(cls_v)$,$w_{env}(cls_v)$ and $w_{com}(cls_v)$ are the weights associated with each partial utility for each household class.

3.3.1 Economic utility

The partial utility $u_{pp}(v)$ is called by as economic utility. This function estimates the expected payback period pp of a particular PV system for agent j. The function value range is between 0 and 1, so we map the payback period range [0,20] into the range [0,1]. The simplest method to do this is to subtract the min(pp) considered, namely one year, and then divide the value obtained by $\max(pp) - \min(pp)$, where $\max(pp)$ is the maximum life of the investment, which is 21 years because 20 years is the expected useful life of the PV system. Thus, as Palmer et al. [2013], we calculate the $u_{pp}(v)$ as follows:

$$u_{pp}(v) = \frac{21 - pp(v)}{20} \tag{3.16}$$

where, $pp(v)$ is the payback period for the PV system that an agent v wants to install. The payback period is defined as the number of years required to recover the initial investment in a photovoltaic system.To assess this period, it is necessary to calculate the net present value (NPV) of the PV system.

In fact, when the NPV value turns from negative to positive, a household recovers from its initial investment. We calculate the NPV by subtracting the initial investment $I(v)$ to the CDCF calculated by the equation 3.11 as follows:

$$NPV(v) = I(v) - CDCF(v) \tag{3.17}$$

The regional and national incentives act on this factor because they reduce the payback period.

3.3.2 Budget utility

The household's budget factor is determined as follows:

$$u_{budget}(v) = \frac{1}{\frac{v_{equity}}{e^{v_{budget}}}} \tag{3.18}$$

where, w_{equity} is the initial investment obtained by subtracting any incentives that act on the initial outlay at the PV system price.
The $u_{budget}(v)$ is based on the agent's budget, which in turn is determined by the agent's income by the equation 3.1. As mentioned before, we suppose that households with an income around the mean buy an ordinary PV system.

3.3.3 Environmental utility

This partial utility captures an agent's attitude toward the ecological benefits linked with the adoption of a PV system. The environmental factor (u_{env}) is calculated as the oil not consumed, which is correlated with the amount of CO_2 emissions saved. For this reason, in equation 3.21 is used the conversion factor from MWh of energy to TOE (tonne of oil equivalent). A TOE is defined as the amount of energy released by burning one tonne of oil, or 0.187 TOE for each MWh produced (Autorità per l'energia elettrica e il gas [2008]).

$$u_{env}(v) = \frac{1}{1 + e^{oil_{notconsumed} - oil_{consumption}}} \tag{3.19}$$

where,

$$oil_{consumption} = \frac{averange\ annual\ energy\ consumption \cdot 20}{1000} \cdot 0.187TOE \tag{3.20}$$

and

$$oil_{notconsumed} = \frac{kWp\ PV\ system \cdot 20}{1000} \cdot 0.187TOE \tag{3.21}$$

3.3.4 Communication utility

Finally, the impact of the social interaction on the adoption decision is described by the partial utility $u_{com}(v)$. The neighbourhood of an agent is defined by agents who share a communication link with it. The communication factor is calculated as follows:

$$u_{com}(v) = \frac{1}{1 + e^{\frac{1}{2}^{L_{v,tot} - L_{v,adopter}}}} \tag{3.22}$$

where, $L_{v,tot}$ is the total number of links of the agent v and $L_{v,adopter}$ is the number of links with actual adopters.

At the beginning of the simulation, there are few adopters in the model, so

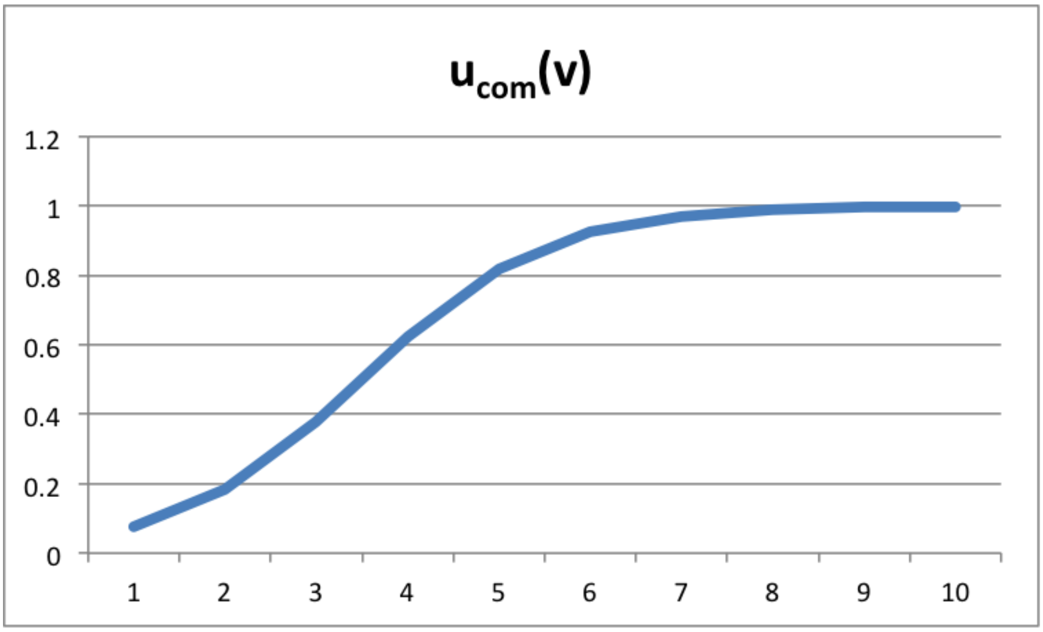


Figure 3.7: The communication utility values.

the interactions do not impact the initial adoption decision. In fact, as you can see in Figure 3.7, the value of $u_{com}(v)$ starts with a value close to zero. Then, when the diffusion takes place, its value increase until neighbours install plants.

If $U(v)$ exceeds the threshold, the agent w installs the PV system. The threshold and the weights are obtained by means of the parameter tuning explained later.

In summary, the utility function takes into account the payback period, the environmental benefit of the investment, the household's budget, and the influence of communication with other agents. The utility function takes into account the relationships between agents whit the partial utility $u_{com}(v)$. The neighbourhood of an agent is represented by agents with which it shares a link. If an agent installs a PV system, all agents in its neighbourhood are affected creating a pressure to build a plant. The economic aspect is described by the partial utility $u_{pp}(v)$. An incentive has an impact on this function because its value depends on the PBT of the plant. The agent's attitude toward the environmental/ecological advantages associated with the adoption of a PV system is captured by the partial utility $u_{env}(v)$. Finally, the partial utility $u_{budget}(v)$ defines the economic possibilities of an agent.
At each time step, the model computes the utility function for all the agents who have not adopted a plant yet. To do this, it compute the best size of the plant that guarantee the best ROE. The ROE estimation considers the national and regional incentives, the loan with the bank and maintenance costs. Furthermore, the simulator updates all statistics regarding the plants installed, such as the year of life, energy produced and the yield.

Chapter 4

Households Generation

In this chapter, we describe the first phase of the simulation, namely the generation of the families. This phase is crucial because it affects the outcome of the simulation. Specifically, the characteristics of the families in the area influence the diffusion of innovations.

Usually, an innovation has a very high price at the beginning due to the high costs of production. However, the cost is reduced over time because the technological improvements, especially those in the production phase, make manufacturing more efficient. Indeed, many technologies follow an S-shape curve that relates the investments made by the company with the performance of the technology (Schilling and Izzo [2013]). In the first stage, the performance improvement is slow because the technology has yet to be fully understood. Later, when researchers have a better knowledge of the technology, the improvement begins to be rapider. However, when the technology reaches its natural limit of performance, the improvements slow down (Figure 4.1). Similarly, the diffusion of innovations follows an S-curve. In the initial stage, the adoption is slow because the technology is poorly understood when it is introduced to the market. When the knowledge about the technology has spread, the innovation enters the mass market and the rate of adoption increases. Finally, the adoption rate will begin to decrease when the market has been fully saturated. Figure 4.2 shows the relationship between the S-curve and the market segments. A possible classification of these segments has been proposed by Rogers [2003]. Rogers has identified five categories of adopters:

- **Innovators** are those who have the highest social status, have a high level of education and economic availability. Their wealth allows them to take risks of buying the innovation, even when it is not widespread. The high level of education allows them to be up to date on front-end technologies and interact with other innovators.

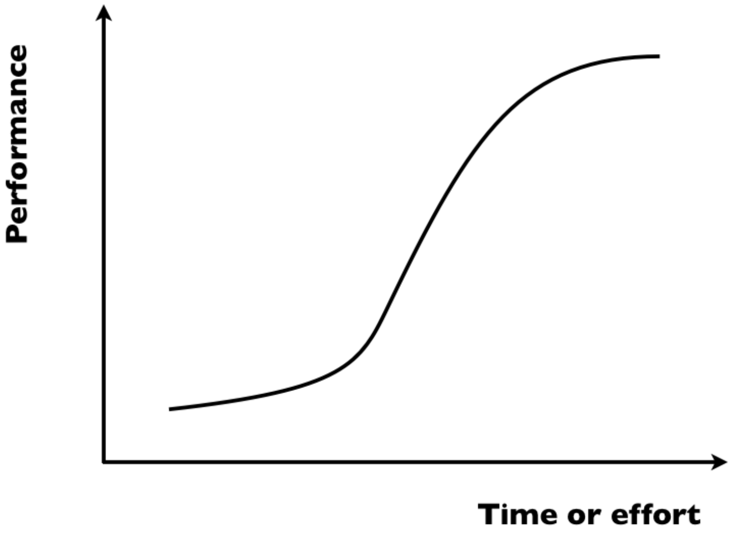


Figure 4.1: Generic S-curve.

- **Early adopters** represent the second group. Who belongs to this category is well-integrated into the social system and has a great potential to influence the behaviour of others. Early adopters have a higher social status, financial liquidity and high level of education.

- **Early Majority** represent the central part of the adoption curve. They are slower and more cautious in the adoption process and anticipate little the average consumer in the market. Early Majority have above average social status. They play an important role in the diffusion process.

- **Late Majority** have a skeptical attitude for innovations. They do not adopt a new product until they do not feel social pressure from their peers. They have below average social status and little financial liquidity.

- **Laggards** represent the remaining share of the market. They base their decision mainly on experience rather than on the influence of social network. Who belongs to this class has the greatest degree of skepticism for innovations.

The diffusion of photovoltaic plants in Italy has followed the S-curve shown in Figure 1.3. The main reasons that discourage the purchase of a PV system are the high initial investment and the long period of payback. These

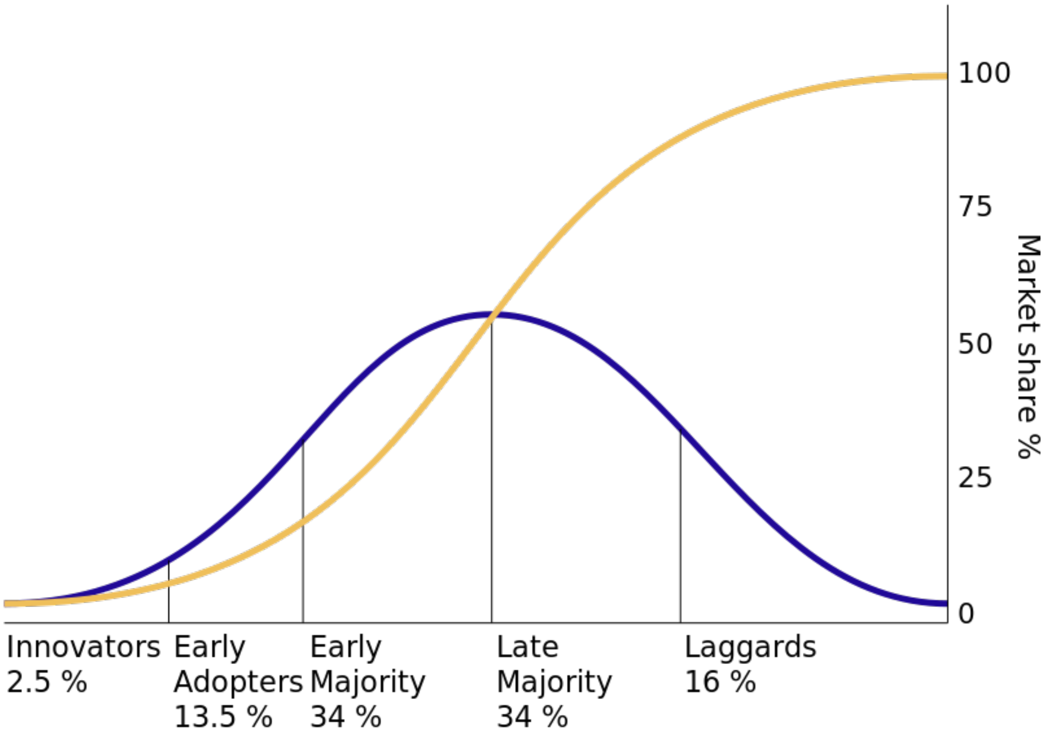


Figure 4.2: The diffusion of innovations according to RogersWikipedia [2014].

factors have implied that only families with financial liquidity and risk tolerance have purchased a plant in the early years. The introduction of national incentives and the reducing of plant costs have increased the annual growth rate, which has reached its peak in 2011. After 2011, there was a steady decline in adoptions due in part to the continuous reduction of the incentive fee.

The diffusion of photovoltaic systems in Emilia-Romagna has followed the trend as the other regions. To model this curve is necessary to sample households whose characteristics reflect those of the entire population. Thus, it is necessary to get families that fall into the classes identified by Rogers in order to obtain the S-shape of the distribution. To classify these families, we decided to describe each of them with the attributes: age class, education level, income, family size and budget. In order to obtain the distributions of these attributes, we used data from the Survey on Household Income and Wealth (SHIW)Bank of Italy [2012]. The SHIW is a statistical survey conducted by Bank of Italy. This survey has begun in the 1960s with the goal of studying the economic behaviour of Italian households. The survey uses a sample of 8000 households distributed over 300 Italian municipalities. The results of the investigations are given every two years through the Statistical Bulletin Supplements. In addition, the Bank of Italy also provides microdata. Through microdata, we derived a model for the generation of the households. The result of generation is households that will be used to simulate the spread of solar systems.

4.1 The microdata

The data are available in three different formats: ASCII, SAS and STATA. The format of the data that we have decided to use is the ASCII. This data are compressed into one single zip archive that contains a series of comma-separated value files (csv). Each file represents a dataset of those shown in Table 4.1. The primary key to merge household level information is NQUEST (household ID). NQUEST must be considered together with NORD (ID of each household member) to merge individual level information. For our purpose, we are interested in the characteristics of the main earner and the economic conditions of his family. For this reason, we use the datasets CARCOM12 and RFAM12. The dataset CARCOM12 contains all the social-demographic characteristics of each household member and other relevant information:

- NQUEST - household ID;
- NORD - component ID;
- CFRED - head of household, defined as the major income earner;
- ETA - age (years);
- CLETA5 - age class (Up to 34 years, 35-44, 45-54, 55-64, more than 64 years);
- NCOMP - Number of household members
- NPERC - Number of household income earners
- PERC - Income earner;
- Q - working status (1=employee, 2=self-employed, 3=not-employed)
- QUAL - employment status (1= blue-collar worker, 2= office worker or school teacher, 3=cadre or manager, 4= sole proprietor/member of

Dataset	**Content**	**Primary key**
Q10A	Households' composition	NQUEST
LAVORO	Employment	NQUEST NORD
Q10C1	Financial Assets and financial information	NQUEST
Q10C2	Financial Assets and financial information	NQUEST
Q10D	Properties and debts	NQUEST
Q10E	Expenditures	NQUEST
Q10F	Insurance	NQUEST
Q10G	Information provided by interviewer	NQUEST
CARCOM12	Characteristics of the individuals	NQUEST NORD
USCITI	Individuals that left the panel household	NQUEST NORDP
ALLB1	Payroll employees	NQUEST NORD
ALLB2	Self-employed worker	NQUEST
ALLB3	Family business	NQUEST
ALLB4	Working shareholder/parter	NQUEST NORD
ALLB5	Pensions	NQUEST NORD
ALLB6	Other income sources	NQUEST NORD
ALLD1	Property, other than principal residence	NQUEST
ALLD2_RES	Loans for main residence	NQUEST
ALLD2_AIMM	Loans for properties other than principal residence	NQUEST
ALLD2_FAM	Loans for consumer credit	NQUEST
ALLD2_PROF1	Loans for business purposes of family businesses	NQUEST
ALLD2_PROF2	Loans for business purposes of self-employed	NQUEST NORD
	Derived datasets	
RFAM12	Household Incomes	NQUEST
RISFAM12	Household Expenditure and Savings	NQUEST
RICFAM12	Household Wealth	NQUEST
RPER12	Individual Incomes	NQUEST NORD
PESIJACK12	Replication weights	NQUEST

Table 4.1: Datasets available in the 2012 annual database

the arts or professions, 5=otherself-employed, 6=pensioner, 7=other not-employed)

- AREA3 - geographical area (1=North, 2= Centre, 3=South and Islands)
- AREA5 - geographical area (1=North-east, 2= North-west, 3=Centre, 4=South, 5=Islands)
- IREG - Istat code for region of residence (1=Piemonte, 2=Valle d'Aosta,

3=Lombardia, 4=Trentino,5=Veneto, 6=Friuli, 7=Liguria, 8=Emilia-Romagna, 9=Toscana, 10=Umbria, 11=Marche, 12=Lazio, 13=Abruzzo, 14=Molise, 15=Campania, 16=Puglia, 17=Basilicata, 18=Calabria, 19=Sicilia, 20=Sardegna)

- NASCREG - region of birth (Istat code)
- NASCAREA - geographical area of birth (1=North, 2= Centre, 3=South and Islands)
- ACOM4C - town size (0-20.000 inhabitants, 20.000-40.000, 40.000-500.000, more than 500.000 inhabitants).
- STUDIO - 1 none, 2 primary school, 3 lower secondary school, 4 vocational secondary school (3 years of study), 5 upper secondary school, 6 3-year university degree/higher education diploma, 7 5-year university degree, 8 postgraduate qualification.

To avoid the curse of dimensionality, we do not use all the attributes to describe a household. The curse of dimensionality refers to the phenomenon that many data analysis become significantly harder as the dimensionality (the number of attributes) of data increases. Since, the data are more scattered in the space, it is more difficult to create a model that assigns the right class to each object. In order to overcome this problem we characterize the families with four attributes: education level, age class, number of members and income. The RFAM12 contains for each household the sources of income. The attributes of this datasets are shown in Table 4.2. These data are linked to the main earner through a join with the key attribute NQUEST (family ID). It was possible to derive the probability distribution for age, education level and income of the primary earner by analysing CARCOM12 and RFAM12 datasets. The attributes that we used for this purpose are CFRED, CLETA5, STUDIO, NCOMP, NPERC, PERC, QUAL, AREA3.
We use CFRED to get the family members that are head of household. The heads of families are defined as primary income earners. So we use their attributes to determine the income of a family. In particular, we use the attribute CLETA5 to get the age class of the head of household and estimate its school level. From the data, we found that the salary of an individual is correlated to its age. Another evidence is that the income of a family is linked to the education level of the major income earner. Thus, we use the attribute STUDIO to calculate its income.
It is obvious that the number of household income earners (NPERC) affect the income of a family because a larger number of earners means a higher

Variable name	**Description**
Y	Net disposable income
YL	Payroll income
YL1	Net wages and salaries
YL2	Fringe benefits
YT	Pensions and net transfers
YTP	Pensions and arrears
YTP1	Pensions
YTP2	Arrears
YTA	Other transfers
YTA1	Financial assistance (wage suppl. etc.)
YTA2	Scholarships
YTA3	Alimony and gifts
YTA31	Received
YTA31	paid(-)
YM	Net self-employment income
YMA1	Self-employment income
YMA2	Entrepreneurial income
YC	Property income
YCA	Income from real-estate
YCA1	Actual rents
YCA2	Imputed rents
YCAF	Income from financial assets
YCF1	Interest on deposits
YCF2	Interest on government securities
YCF3	Income from other securities
YCF4	Interest payments
$Y = YL + YT + YM + YC$	

Table 4.2: Variables of RFAM12

family income. Moreover, the NPERC is related to the number of members of the family. It is clear that a family composed by a single element can have at most one income earner. The employment status (QUAL) it is another parameter that changes the person's income. Finally, the attribute AREA is used to get only the people who reside in the area of interest.
We start by selecting the elements that are heads of families (CFRED = 1), live in the north (AREA3=1) and are workers or pensioners (QUAL<7). We did not select only families that live in Emilia-Romagna because the sample was too small to obtain meaningful data. Then, to generate the families, we

determine the age class of the principal earner.
In our analysis, we used Weka (Hall et al. [2009]): free software written in Java that provides a collection of visualization tools and algorithms for data analysis.

4.1.1 The age class

The age class is a discrete ordinal attribute, which can take five values: up to 34 years, 35-44, 45-54, 55-64, more than 64 years. In the dataset, these classes are represented as integers from 1 to 5 in such a way that the order is preserved.

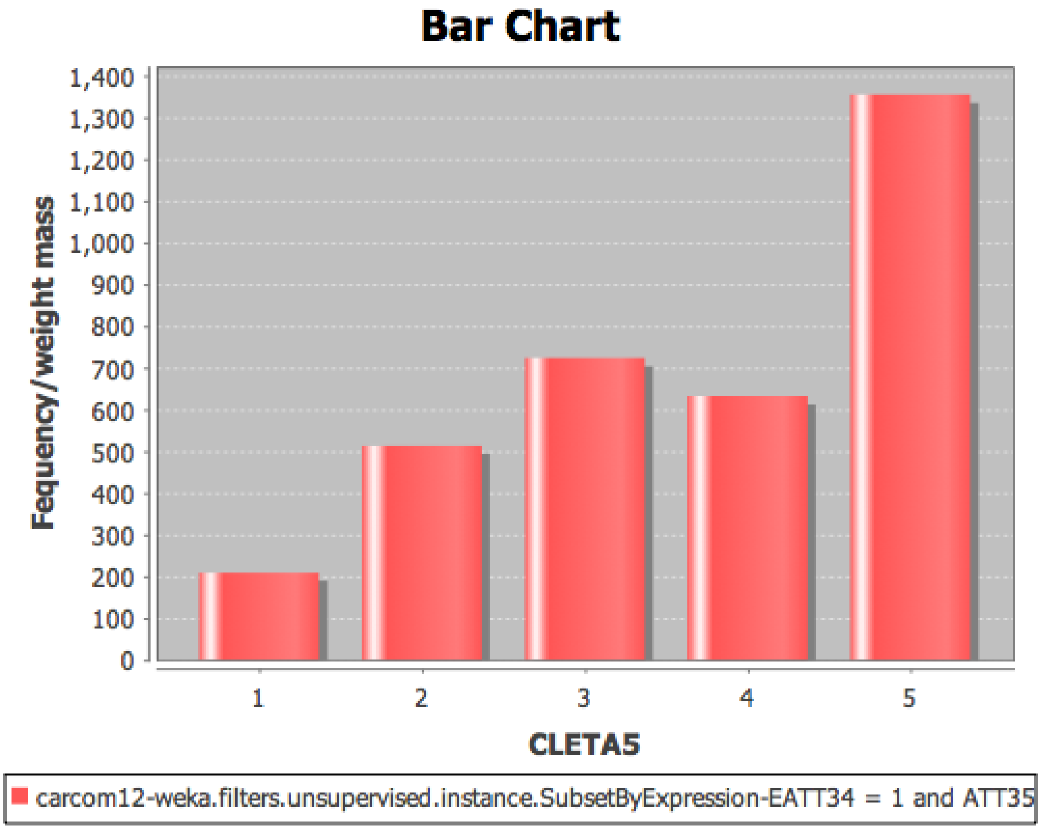


Figure 4.3: Age class frequencies.

Figure 4.3 shows the frequencies of these values. We estimate the probability of occurrence for age class value by dividing each frequency by the total number of family's heads. Table 4.3 shows the probability distribution in which each outcome for each age class is linked with its probability. Once the age class has been assigned, we proceed to assign an education level.

Age Class	Probability
1	0.061
2	0.149
3	0.211
4	0.184
5	0.395

Table 4.3: Age Class: probability distribution

4.1.2 The education level

The education level is represented by mapping the level of study with an integer value. Levels of education that we consider are: (1) none, (2) primary school, (3) lower secondary school, (4) vocational secondary school (3 years of study), (5) upper secondary school, (6) 3-year university degree/higher education diploma, (7) 5-year university degree, (8) postgraduate qualification. The education level is affected by the age class attribute. This relationship can be observed in the following table that shows the distribution of probability of the education level attribute. For example, if we look at the level of education 5, we can observe that the age affect the probability that a person belong to this level. Indeed, the probability $P(EDU = 5|AGE = 3)$ is about 0.40, but $P(EDU = 5|AGE = 5)$ is about 0.20. The conditional probability $P(EDU = 5|AGE)$ is not equal to the a priori probability $P(EDU = 5)$, so there is a dependency between the age class and education level attributes. This aspect is made even clearer by the scatter plot 4.4.

4.1.3 The number of members and annual energy consumption

We need to establish the number of family members for defining the energy consumption. The probability distribution is shown in Table 4.5. These probabilities were derived from the frequencies shown in the bar chart 4.5. The average number of members in the dataset is 2.3. The average annual consumption of energy is estimated according to the number of family members. Enel computes consumption as shown in Table 4.6. Then, we use these estimates to determine the average annual consumption of a family. For instance, the consumption is estimated equal to 6000 kWh/year for a family of four.

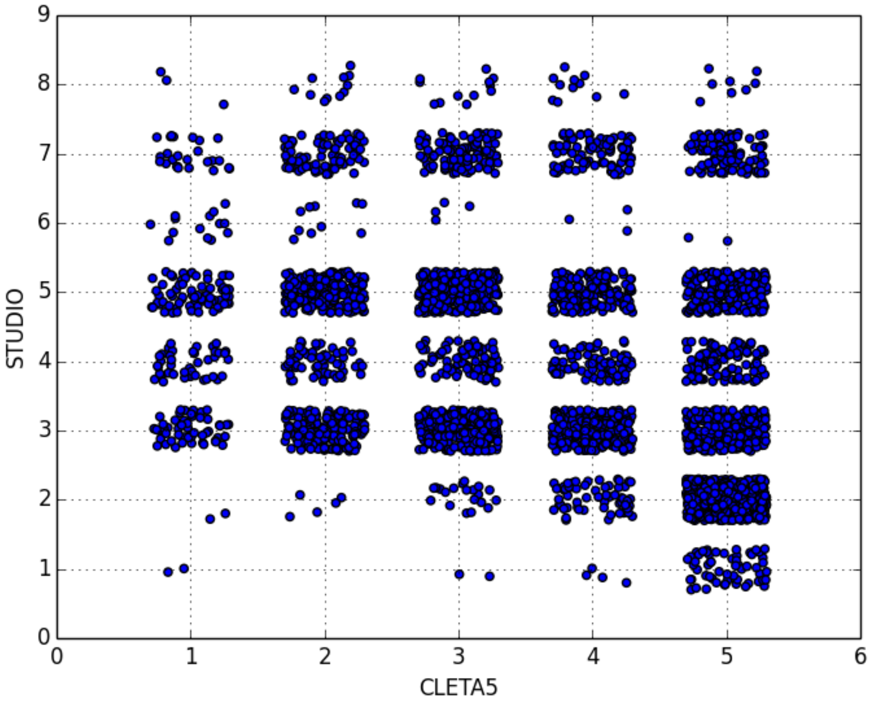


Figure 4.4: Age class frequencies.

4.1.4 The income of a family

We decided to use the regression to assign an income to the family. Tan et al. [2007] defines the regression as a predictive modelling technique where the target variable to be estimated is continuous. Let D denote a data set that contains N observations,

$$D = \{(x_i, y_i)|i = 1, \ldots, N\} \tag{4.1}$$

Each x_i corresponds to the set of attributes of the i-th observation (also known as the explanatory variables) and y_i corresponds to the target (or response) variable. Regression is the task of learning a target function f that maps each attribute x into a continuous valued output y. We seek a function f that can predict the family income from the characteristics of the major income earner. To find this function we used a linear regression. Thus, the function f is a linear combination of a set of coefficients and explanatory variables, whose value is used to predict the outcome of the

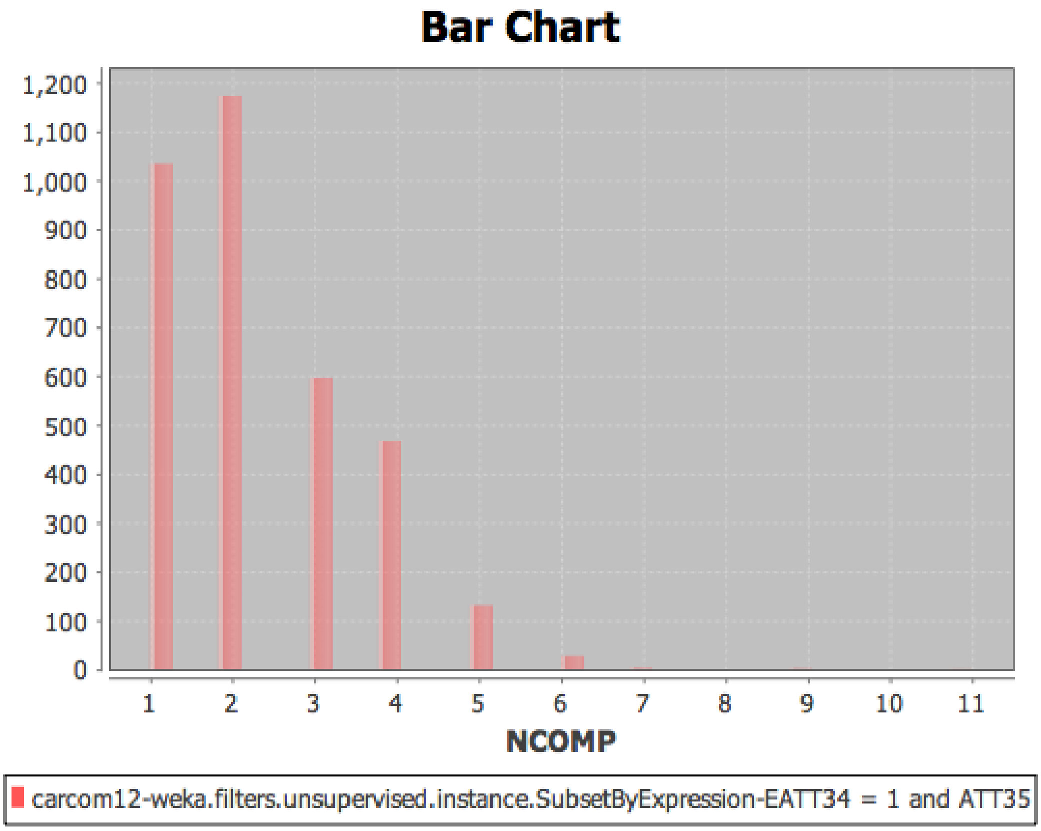


Figure 4.5: Number of members frequencies.

dependent variable.

$$f(x) = \sum_{i=1}^{N} w_i x_i \tag{4.2}$$

Moreover, to the value of the function is also added an error term ϵ.

$$y = f(x) + \epsilon \tag{4.3}$$

This random noise ϵ is used to capture measurement errors of the attributes or factors that were not included in the model. The target variable y is treated as random variable and it may assume different values even when considering the same attribute. We use the Ordinary Least Squares (OLS) method for estimating error term ϵ.
As explanatory variables we decided to use STUDIO, CLETA5, NCOMP, NPERC to predict the income, where NPERC is the number of recipients of the family. The value of NPERC is related to NCOMP and it is obtained

from the probability distribution 4.7. The linear regression model is derived from data contained in CARCOM12 and RFAM12. To evaluate the goodness of the fit we use the R-squared measure. R-squared measure is defined as:

$$R^2 = \frac{SSM}{SST} = \frac{\sum_i [f(x_i) - \bar{y}]^2}{\sum_i [y_i - \bar{y}]} \tag{4.4}$$

The R-squared value is between 0 and 1. If the value of R^2 is close to 1 there is a linear relationship between the attribute set and the the target variable. Through the explanatory variables STUDIO, CLETA5, NCOMP, NPERC the R-squared value obtained is 0.415. This value is not very high, but for our purpose it is more than enough because our goal is to provide an estimate of the income.

4.2 The subdivision of families in social classes

As mentioned in the chapter, households are divided into social classes to assign a similar behaviour to the families that share similar characteristics. In our model, the social class of a family is reflected in the different weights used to calculate the utility function. In fact, the weights are selected according to the social class of the family. These weights are obtained by means of parameter tuning, but households should be subdivided well to fit the curve of diffusion of PV system in Italy. In this section, we will explain how we got the division into social classes of families.

The objective is to identify the types of families according to the classification of Rogers so that we can obtain the S-curve. We start to simulate from 2007. In that year, the adoption rate was very low: only who had financial liquidity and attitude to explore decided to purchase a photovoltaic system. Rogers classifies these individuals as innovators and they are only 2.5% of the market. The decrease in the cost of the plants and the introduction of national incentives contributed to increase the rate of diffusion.

Since we do not have a classification model for households who purchased a solar panel, it is difficult to make the exact subdivisions identified by Rogers. However, we do not want to fit the curve by grouping the families, but we want to help the parameter tuning to identify the weights that allow us to obtain the same curve of diffusion in the virtual world. Specifically, the combination of the clustering, which attempts to group similar families, and the different weights found by parameter tuning leading the simulation towards our goal. Thus, the clustering is a method that allows us finer control for mapping the diffusion curve.

Each family can be considered as a point in three-dimensional space: age class, education level and income. We used the K-means clustering technique to subdivide these points in 5 groups. The K-means is a prototype-based technique that attempts to find a user-specified number of cluster (k). Each cluster C_i, with $i = 1, \ldots, k$, is represented by its prototype c_i , defined as the centroid of the group of points. The K-means algorithm attempts to minimise the total intra-cluster variance repositioning the centroid at every step until the centroids do not change. It starts with a random set of centroids $c_1, \ldots, c_k$ and then assigns each point x to the nearest centroid. After that, it calculates the new centroids by averaging the points in a cluster as follow:

$$c_i = \frac{1}{m_i} \sum_{x \in C_i} x \tag{4.5}$$

where m_i is the number of points in the cluster C_i. The result obtained by applying K-means clustering to generated households is shown in Figure 4.6. It is is difficult to evaluate the goodness of a clustering because we do

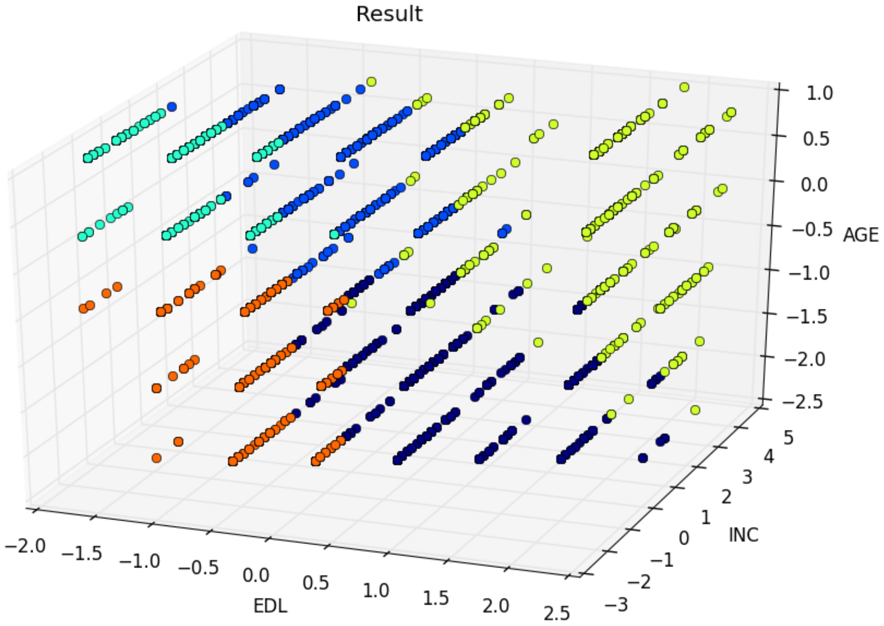


Figure 4.6: Clustering of households.

not have the ground truth class labels to be used as a reference. When the ground truth is unknown, unsupervised techniques can be used to evaluate the clustering. Unsupervised techniques measure the goodness of a clustering structure without using external information. A common unsupervised method is the silhouette coefficient that relates the cohesion of a cluster with the separation between clusters (Tan et al. [2007]). The silhouette coefficient is defined for each sample i and it is composed of two scores:

- The cohesion $a(i)$ - the mean distance between the sample i and all other points in the same cluster.
- The separation $b(i)$ - the mean distance between the sample i and all other points in the next nearest cluster.

for the sample i, the coefficient is:

$$s(i) = \frac{(b(i) - a(i))}{\max(a(i), b(i))} \tag{4.6}$$

The value of the silhouette coefficient can vary between -1 and 1. If the value is negative that means that $a(i)$ is smaller than $b(i)$, so the sample i is closer to the objects of another cluster than other objects of the cluster to which i belongs. Samples with a large $s(i)$ (almost 1) are very well clustered. An overall measure of the goodness of a cluster can be obtained by computing the average silhouette coefficient of all samples.
Using k means clustering we got a silhouette coefficient of 0.35.

4.3 Implementation

For the generation of the necessary data to the simulator, we have implemented a set of tools in Python. We use Python because it is a dynamic object-oriented language, easy to learn and with a large community of users. Many modules have been developed for Python and this has made the development of tools really simple. We developed several tools to analyse microdata, sample households respecting the probability distributions, perform the clustering, sample buildings and organize the social network. The process that leads to an instance for the simulator is shown in Figure 4.7. The set of tools reflecting the process 4.7. A program called *Learner* has the task to analyze microdata and provide the probability distributions and the regression model as output. The dataset CARCOM12 and RFAM12 are loaded by the Python module Pandas (pandas community [2012]). In particular, the data are loaded in memory into DataFrames objects: a two-dimensional size-mutable, heterogeneous potentially tabular data structure with labeled axis (rows and columns). It provides many features, including the ability to make the join between two tables, to select the data and do operations on them. These features have greatly simplified the implementation of the *Learner*. Indeed, it was easy to make the join between the CARCOM12 and

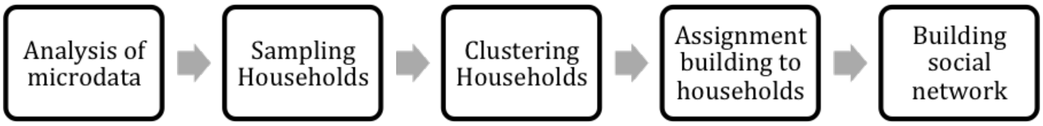


Figure 4.7: Households generation steps.

RFAM12 datasets using the key NQUEST, select families with the required characteristics and calculate the probability distributions of attributes.
The regression model was obtained using the implementation of OSL contained in the Statsmodels module.
The Learner program is run only once because the probability distributions are saved in pickle files for efficiency. The pickle file is used for serializing and de-serializing a Python object structure. Thus, using the pickle file is possible to save the DataFrame objects structures that contain probability distributions, save the income model and load them at a later time.
The Python module *Generator* loads pickle files generated by the *Learner.* The *Generator* module uses the probability distributions and the income model to sample households. This module is imported by the *Clustering* module, which provides the procedure *get_households.* This procedure takes as input the number of households to be sampled and returns families with the labels of their clusters. As mentioned in the Section 4.2, the clustering technique used is K-means. We decided to use the implementation of K-means provided by the Python module sklearn. This module provides many data mining techniques, so it was possible to try different clustering methods and find the one that provided the highest silhouette coefficient.
The main program is *instance_generator.py.* This program reads a configuration file which contains the following information:

- *sample_size* - the number of families to be sampled for each instance;

- *inputfile* - The shapefile that contains the buildings;
- *outputpath* - The output folder. This folder will contain the shapefile and dbf files generated;
- *num_instances* - The number of instances that the program will create;
- *degree* - The maximum number of links for each family in the social network.

As a first step, *instance_generator* reads the shapefile and loads buildings in memory. Then, For each instance:

1. it samples a number of buildings equal to *sample_size*;
2. it generates *sample_size* households using the cluster module;
3. it assigns each family to a building;
4. it invokes the *setFriends* method of *socialnetwork* module to build the social network.

The *socialnetwork* module implements the extended rank-based friendship explained in section 3.2.
The attributes of households are stored in dbf file that, along with the shapefile, is written in the output folder at the end of execution. Figure 4.8 shows the toolchain described above. The DBF files and SHP files are loaded by the NetLogo model.

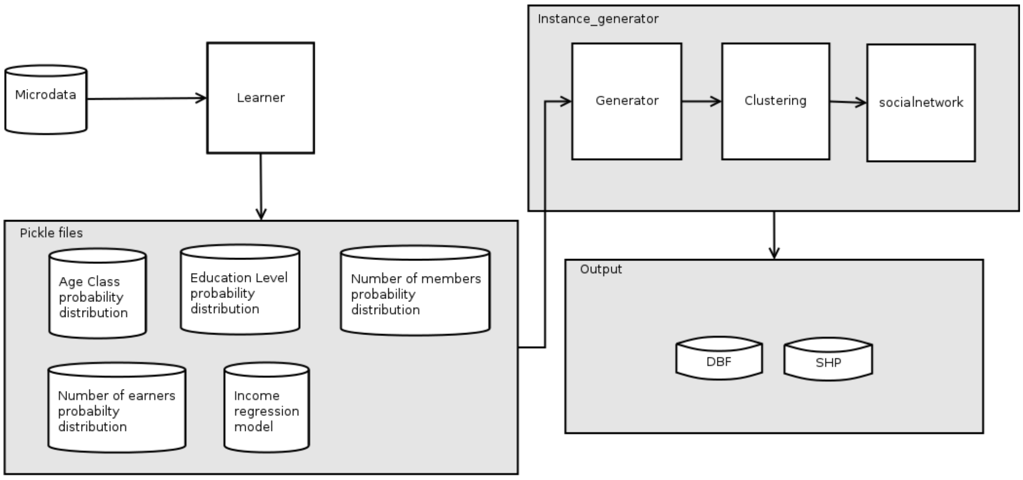

Microdata
Learner
Instance_generator
Generator
Clustering
socialnetwork
Pickle files
Age Class probability distribution
Education Level probability distribution
Number of members probability distribution
Number of earners probabilty distribution
Income regression model
Output
DBF
SHP

Figure 4.8: Tool chain

In summary, these Python scripts allow us to generate one or more scenarios for our model. A scenario consists of an SHP file that contains the buildings position and a DBF file that contains the household's characteristics and the building's characteristics. These files are loaded in the model to spread the agents in the virtual world and reproduce the area of interest. In order to get an accurate parameter tuning, we create more scenarios for the same number of agents. These scenarios were used to fine-tune the parameters in our model as we are going to see in the following chapter.

Age Class	**Education level**	**Probability**
1	1	0.009
1	2	0.009
1	3	0.244
1	4	0.201
1	5	0.335
1	6	0.067
1	7	0.120
1	8	0.014
2	2	0.010
2	3	0.279
2	4	0.113
2	5	0.398
2	6	0.019
2	7	0.160
2	8	0.021
3	1	0.003
3	2	0.028
3	3	0.342
3	4	0.108
3	5	0.372
3	6	0.005
3	7	0.126
3	8	0.016
4	1	0.006
4	2	0.096
4	3	0.335
4	4	0.119
4	5	0.277
4	6	0.005
4	7	0.144
4	8	0.017
5	1	0.042
5	2	0.428
5	3	0.204
5	4	0.074
5	5	0.172
5	6	0.001
5	7	0.073
5	8	0.006

Table 4.4: Age Class: probability distribution

Number of members	**Probability**
1	0.301
2	0.341
3	0.173
4	0.136
5	0.038
6	0.008
>6	0.002

Table 4.5: Number of members: probability distribution

Number of members	**Average annual consumption (kW/year)**
1	2700
2	3500
3	4500
4	6000
>5	7500

Table 4.6: Number of members: probability distribution

NCOMP	**NPERC**	**Probability**
1	1	1
2	1	0.287
2	2	0.713
3	1	0.218
3	2	0.524
3	3	0.264
4	1	0.212
4	2	0.559
4	3	0.139
4	4	0.090
5	1	0.277
5	2	0.423
5	3	0.192
5	4	0.069
5	5	0.038
6	1	0.296
6	2	0.370
6	3	0.148
6	4	0.148
6	5	0.037
7	2	0.667
7	3	0.333
9	2	0.5
9	3	0.5
11	4	1

Table 4.7: Number of earners: probability distribution

Chapter 5

Model Simulation and Calibration

In this chapter, we explain the model simulation and calibration. The simulation has been designed with the aim to analyze the requirements related to the development of a new photovoltaic project in the planning stage and test the feasibility of the idea. Therefore, the simulator provides a useful assessment tool for individual investors, but at the same time allows to obtain global information such as the total amount of energy produced by photovoltaic technologies or the costs incurred from the region. Then, it is possible to integrate the simulation results into the optimization problem which has the goal to create the regional energy plan.

To correctly model the dynamics of the complex system studied was necessary to introduce a number of parameters that govern various aspects of the simulation. For example, the electrical energy that each system can generate is closely linked to the geographic position, and orientation of the photovoltaic panels that make it up (to simplify the model, the orientation and the angle of inclination of the panels were considered optimal, i.e. to the south and 30 inclination). In order to create a model that simulates the installation of new PV panels in any area the annual average solar irradiation is a global parameter of the model whose value can be changed through the graphical interface. Again, the user can control the average cost of a PV system via the GUI so, it can be dynamically changed during the simulation. Other variables related to the plant are the loss of efficiency of the photovoltaic panels and the annual maintenance cost.

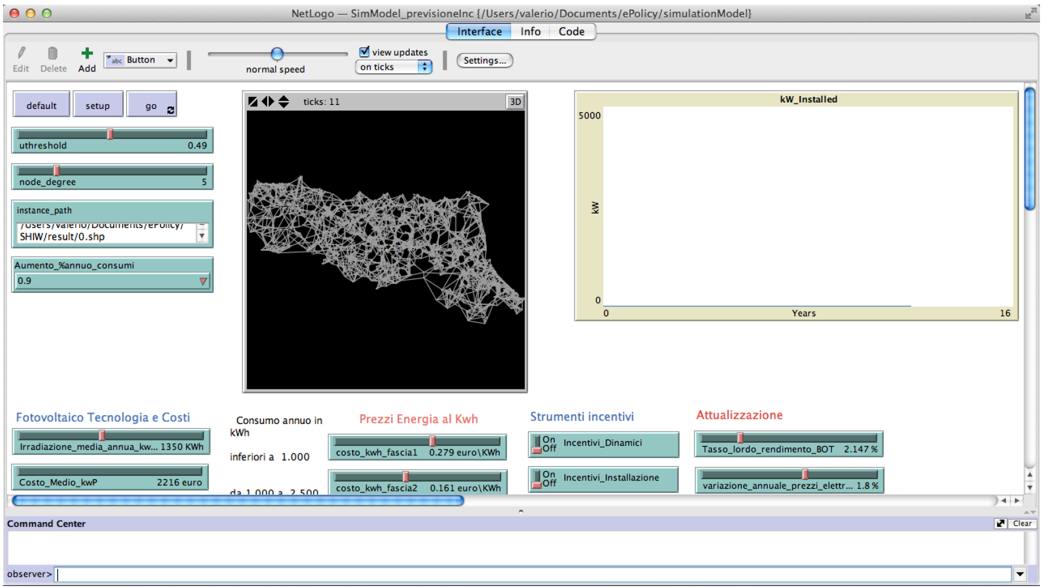
Once you have established basic parameters, every year the simulator performs the economic evaluation that is linked to the performance of PV systems installed. This phase takes into consideration factors closely related to the cost of electricity as well as the incentive tariff recognized for the

energy produced by the plant. In fact, the revenues derive from either the self-consumption or the sales to the GSE.
In the direction of simplify and speed up the model calibration, all previously mentioned parameters are kept constant during the parameter tuning. We set these parameters to reflect the condition that we can found in Emilia-Romagna; during the model calibration, we focus only on the weights of the utility function.
As mentioned previously, we use NetLogo to provide an implementation of our model. We started by loading the household data generated from the developed tools in the NetLogo model. Loading data from outside allow us to reproduce any area of interest without change the NetLogo model.

Once the data has been loaded, the model proceeds to evaluate, for each semester, the dissemination of photovoltaic systems in the area.
In order to obtain the same PV system diffusion occurred in the Emilia-Romagna we calibrate the model on real data computed using the historical PV power installation trends provided by the GSE using the **irace** package (López-Ibáñez et al. [2011] for R language R Development Core Team [2008]). **irace** generates many configuration of weights for the utility function 2.4 that are tested on instances. An instance consists in a DBF file and in an SHP file that are generated according to the process described in the Chapter 4. At the end, **irace** returns the best configurations found during the "race". In this way, we obtained the weights for the utility function 2.4 that guarantee a similar diffusion of residential PV systems in the virtual world.
This calibration process took a long time since each simulation involves a large number of agents. However, once we had the weights, performing experiments on the virtual world was quite simple.

5.1 Model simulation

In our NetLogo model, we have a breed (a kind of agents) called household that contains all the attributes relative to the family. Households are loaded from the shapefile when a user presses the setup button. The shapefile and the associated DBF are read using the GIS extension. For each building, the simulator creates a household with the values for the attributes contained in a row of the DBF.
The households are placed in the virtual world mapping WGS84 coordinates with the coordinates of patches and are connected together by means of links. These links are read from the DBF file that contains for each family a list of



☑ view updates
on ticks
uthreshold 0.49
node_degree 5
instance_path
/Users/valerio/Documents/ePolicy/SHIW/result/0.shp
Aumento_%annuo_consumi
0.9

Fotovoltaico Tecnologia e Costi
Irradiazione_media_annua_kw... 1350 KWh
Costo_Medio_kwP 2216 euro

Consumo annuo in kWh
inferiori a 1.000
da 1.000 a 2.500

Prezzi Energia al Kwh
costo_kwh_fascia1 0.279 euro\KWh
costo_kwh_fascia2 0.161 euro\KWh

Strumenti incentivi
On Off Incentivi_Dinamici
On Off Incentivi_Installazione

Attualizzazione
Tasso_lordo_rendimento_BOT 2.147 %
variazione_annuale_prezzi_elettr... 1.8 %

Figure 5.1: The NetLogo simulator screenshot.

IDs of friends.

So, the social network is generated a priori and loaded into the model. In this way, the setup of the virtual world is faster. Figure 5.2 shows the result.

After households are loaded, the simulation is performed for the first half of 2007. We use tick-based mode to update the simulated time. Each tick represents a semester in which all the agents that have not installed a solar system decide whether to install a PV system or not. In addition, for each step and for each adopter the simulator updates the energy produced by the plant taking into account the annual reduction of efficiency and the earnings from energy production.

In the code, there is the report called *utility-function* 5.1 that calculates the desire level to install a photovoltaic plant of the family who invokes it.

```
to-report utility-function
  let winc 0
  let wpbt 0
  let wenv 0
  let wcom 0

  if class = 0 [
    set winc a0
    set wpbt (1 - winc) * b0
    set wenv (1 - (winc + wpbt)) * c0
```

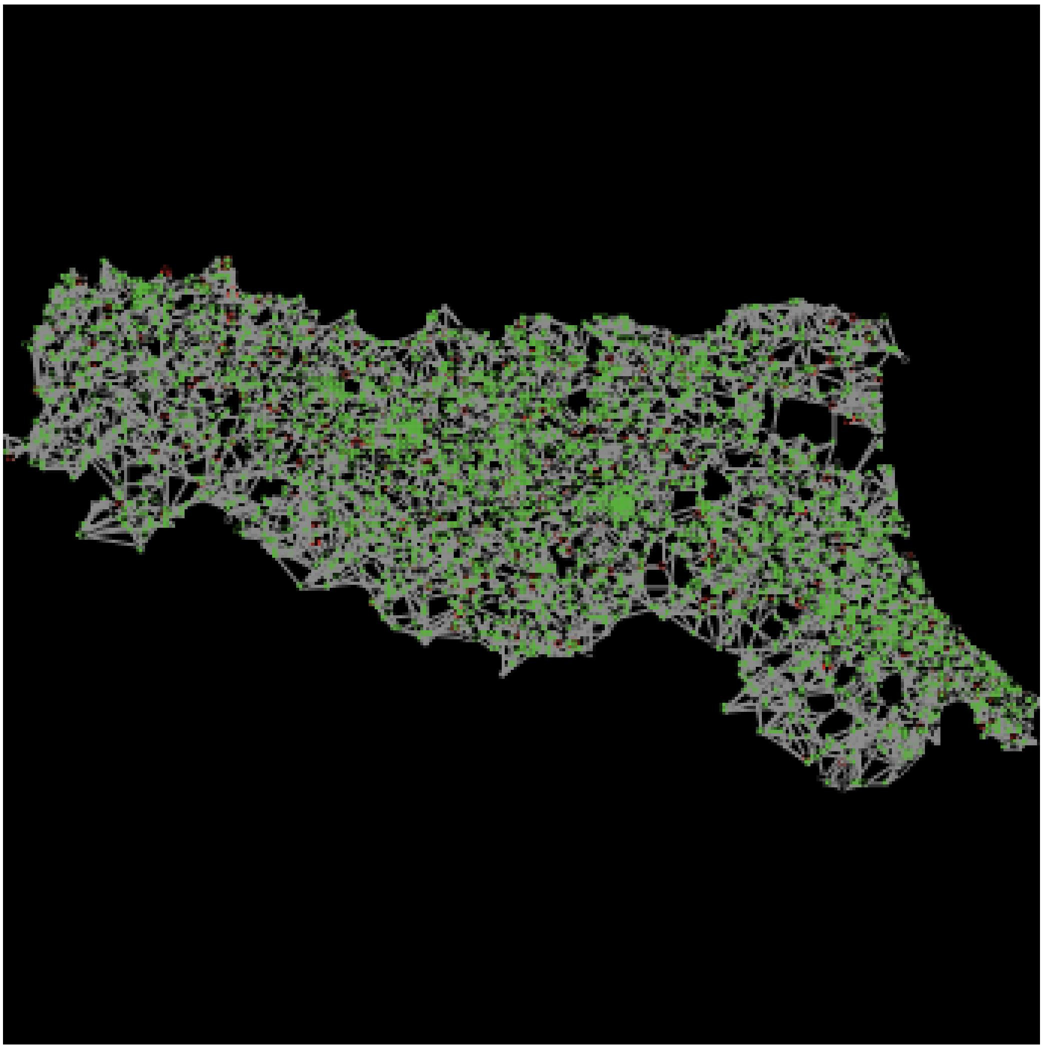


Figure 5.2: World view after household are loaded.

```
]
if class = 1 [
  set winc a1
  set wpbt (1 − winc) ∗ b1
  set wenv (1 − (winc + wpbt)) ∗ c1
]
if class = 2 [
  set winc a2
  set wpbt (1 − winc) ∗ b2
  set wenv (1 − (winc + wpbt)) ∗ c2
  ]
  if class = 3 [
    set winc a3
    set wpbt (1 − winc) ∗ b3
    set wenv (1 − (winc + wpbt)) ∗ c3

  ]
  if class = 4 [
    set winc a4
    set wpbt (1 − winc) ∗ b4
    set wenv (1 − (winc + wpbt)) ∗ c4
  ]

  set wcom (1 − (winc + wpbt + wenv))

  let uf wpbt ∗ economic−utility + wenv ∗ enviromental−utility
  + winc ∗ income−utility + wcom ∗ communication−utility

  report uf
end
```

Listing 5.1: NetLogo report that calculates the utility function

The report *utility-function* extracts the weights for the utility function from the **irace** parameter a_{cls}, b_{cls}, c_{cls} where *cls* is the agent class. We use these parameters because the utility function value is between 0 and 1. So, with this trick we can specify parameters whose values are dependent on **irace**. After that, the function 5.1 call all the partial utilities to compute the desire level of a household.

As mentioned before, the agents calculate the size of the system that ensures the best investment. The simulated annealing algorithm solves this optimization problem. To make the code more readable and efficiently, we decided to implement this algorithm as an extension of NetLogo using the API that developers make available. So, the algorithm was written in Java and then loaded by the primitive *extension*.

The model has many global parameters that are used to estimate the ROE of the investment and other statistics. Global parameters are used to simplify the model and, at the same time, to make it general. In fact, these parameters are the cost of PV panels, PV technology, electricity prices for household consumers, the GSE minimal price, BOT yield, maintenance costs, etc. Many of these parameters can be set by the user via the graphical inter-

face before and during the execution. Others, such as guaranteed minimum prices by the GSE and electricity prices are read from external files.
Different values for the global parameters can produce different results. For instance, high PV system installation cost reduce the value of ROE and so discourage a household to adopt a PV system.
Usually, we use the same values for the global parameters to simulate the behaviour of the agents in order to study the phenomenon of diffusion of residential PV system having the same boundary conditions. Indeed, we focus our attention on the weights for the utility function and the number of agents.

In addition to the household breed, we have define the region breed that is devoted to managing the incentive system. We add one region agent during the setup phase. The region agent allocates a budget for an incentive system. The various types of incentives are exclusive, and it is necessary to choose which one to apply before the simulation starts. Therefore, it is not possible to study through the simulator the interactions between different incentives. An agent can not take advantage of regional incentives, for example, it may not be aware of these incentives, or may have no intention to take out a loan. To simulate this aspect, the agent has a probability to know about the regional incentive. The region also has the task to update the incentive statistics, such as the budget used, during the simulation.
Once the setup phase is finished, and the model has been populated with the agents, the user can press the button Go to start the simulation. The model simulates the creation of new photovoltaic systems from the first half of 2012 to the second half of 2016. Since the incentive applied to energy produced by plants is guaranteed for a period of 20 years, the duration of the simulations is extended until the second half of 2036. During the simulation, the model updates the plants'statistics such as years of life, yield and energy produced. Moreover, the model calculates the agent's statistics such as revenues, ROE and PBT.
The graphical user interface show useful information to understand how the virtual world changes over time. Each semester, the simulator updates data on the installed kW, the average payback time and regional incentive. This information is very important for politicians or companies to assess what would be the response of investors to changes in model parameters.

5.1.1 Model calibration

Most of the algorithms for optimisation problems need the setting of many parameters. Usually, there isn't an optimal setting of these parameters for every problem that the algorithm can handle, but the optimal configuration depends on the problem under consideration.

Normally, the calibration process runs the model many times and, at the end of each execution, the output generated by the model is compared with the real-world data. The model is adjusted automatically based on the difference before the next execution. The objective is to find the optimal parameters that provide a good fit. For simple mathematical functions with limited number of variable, the calibration process is based on the assumption that the function includes one or more error terms, and then to fit the function to the data the calibration process minimizes the error term. Generally, when we are dealing with agent-based models, the number of parameter involved and the computational resources needed do not allow us to try all parameter combination. Moreover, the solution spaces involved are complicated and the variables are often interdependent in non-linear manner, rendering the mathematical optimization inappropriate. In these cases, we can use IA techniques to calibrate the model. Evolutionary algorithms (EA) and swarm algorithms may provide a sufficiently good solution without exploring the entire search space. For instance, genetic algorithms (GA) that belong to the larger class of EA try to refine the initial population of random individuals by applying the mutation and the crossover operators (two basic operators of GA). At the end of each step, a new generation of individuals is created from the best individuals of the previous generation by modifying and combining their chromosomes. The new generations also contain the previous best individuals and their sizes remain constant; thus the number of model evaluations remains constant or decrease over time.

In our model, it is necessary to determine the weights of the utility function to map the actual diffusion curve of photovoltaic plants in Emilia-Romagna over 2007-2012 period. For this reason we decided to use the **irace** package (López-Ibáñez et al. [2011] for R language R Development Core Team [2008]) that implements the iterated racing procedures proposed by Balaprakash et al. [2007] and further developed by Birattari et al. [2010].

Assume that we have an algorithm with N parameters, x_d with $d = 1, \cdots, n$. The parameters tuning problem consists in finding a configuration $\theta = \{x1, \cdots, x_{N^{param}}\}$ of these parameters that minimizes the measure cost $c(\theta, i)$, where i is an instance of the problem López-Ibáñez et al. [2011].

The iterated racing method consists in three steps: in the first step the new configurations are sampled from a particular distribution, in the second step these configurations are tested on instances of the problem, and in the last step the distribution is adjusted with the best configurations. At the end, it returns the best configurations found during the race.

In our case, the configuration is a set of weights for the utility function

2.4, the instance is a sample of households and the measure cost is the distance between simulated diffusion curve and actual curve.
Since, each test requires a long time and the model has many parameters, it was necessary to use a cluster of computers to cover the parameter space quickly. The architecture for parameters tuning is the one used by Belletti [2014] shown in figure 5.3. In the architecture, there is a central server for coordinating the test and a set of clients for performing the NetLogo simulations. The server communicates with clients via ssh to setup the tests and retrieve the results. To be more precise, the central server executes **irace** and chooses the clients to perform tests on instances. Each client performs a NetLogo simulation with the parameters provided by the server. When the simulation ends on a client, the result is returned to the server that is responsible for selecting the best configurations. On startup, all machines load the same disk image through the process of network booting. This image contains the Ubuntu 12.04 operating system in which NetLogo 5.0.5 and Java Runtime Environment 7 are installed. The file system is mounted read-only, so it is not possible to make changes to any file. However, for temporary file is provided a RAM disk with a capacity of 100 MB. The Ubuntu image is configured to mount the shared NFS directory from the server. Technical specifications for the client machines: Intel® Core™i5-2400@3.1 GHz and 4 GB of RAM.

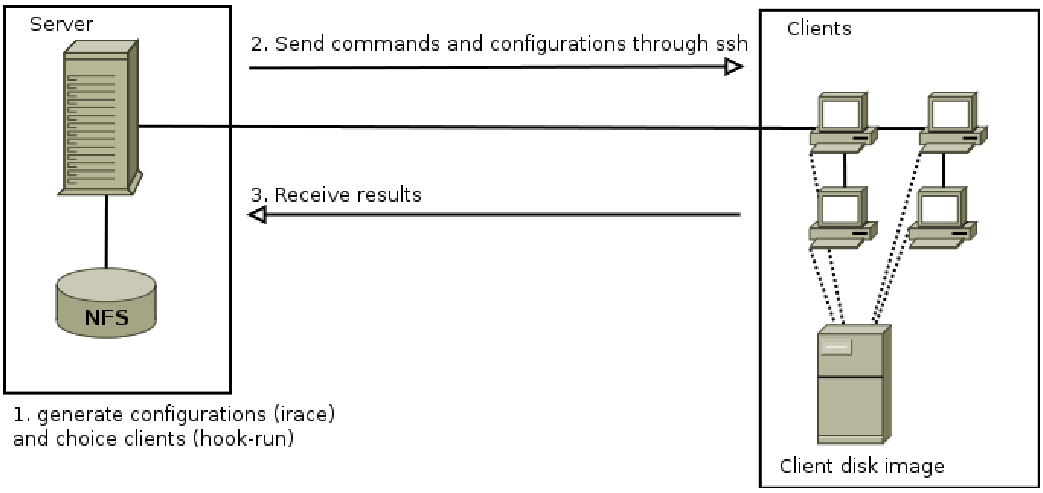

Server
2. Send commands and configurations through ssh
Clients
3. Receive results
NFS
1. generate configurations (irace)
and choice clients (hook-run)
Client disk image

Figure 5.3: Parameters tuning architecture.

The server side is a Linux machine where is installed **irace** and the NFS sever. Technical specifications for the server machine: Intel® Xeon® X5570@2.93 GHz and 48GB of RAM.

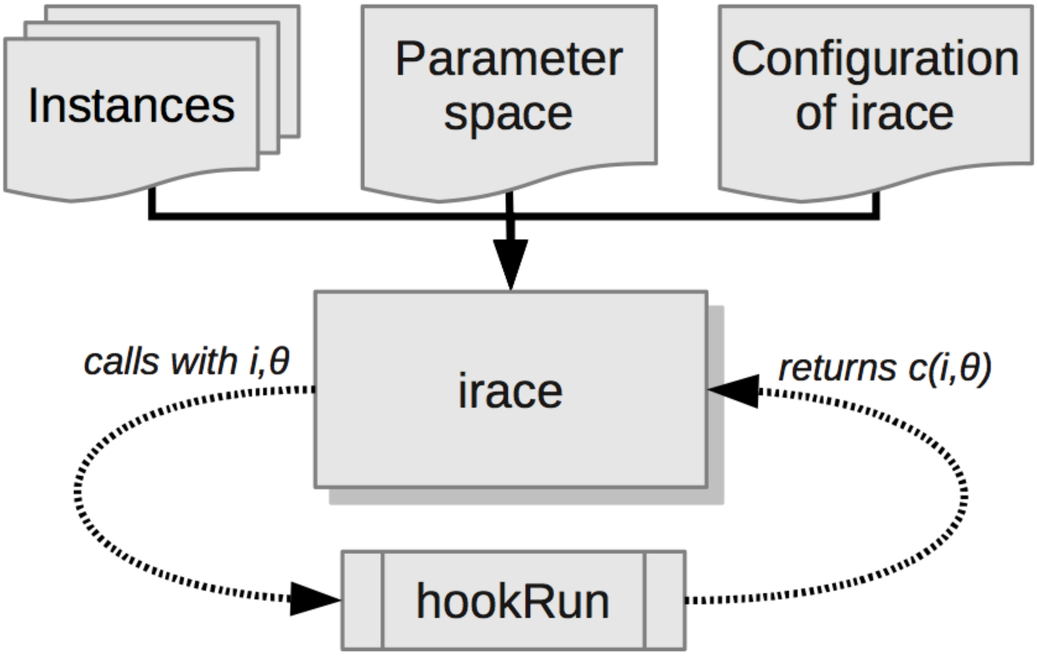


Figure 5.4: Scheme of **irace** flow of information.

The program **irace** requires four main inputs:

- A description of the parameter space X: the parameters to configure, their types, ranges and constraints;
- The set of tuning instances;
- The configuration of **irace** itself;
- A function or an auxiliary program called hookRun.

These inputs can be passed as either R data structures or as files. The package **irace** provides a command-line wrapper for Unix environments, called *irace*, which invokes R and executes **irace**. By default, **irace** searches the required files in the current working directory.
We first define the parameter file, *parameter.txt*, that describes the parameter space as follow:

```
# rewiring probability
rp "--rewiring-probability " r (0.0, 1.0)

# cluster utility function weights
# cluster 0
```

```
a0 ”−−a0 ” r (0.0, 1.0)
b0 ”−−b0 ” r (0.0, 1.0)
c0 ”−−c0 ” r (0.0, 1.0)

# cluster 1
a1 ”−−a1 ” r (0.0, 1.0)
b1 ”−−b1 ” r (0.0, 1.0)
c1 ”−−c1 ” r (0.0, 1.0)

# cluster 2
a2 ”−−a2 ” r (0.0, 1.0)
b2 ”−−b2 ” r (0.0, 1.0)
c2 ”−−c2 ” r (0.0, 1.0)

# cluster 3
a3 ”−−a3 ” r (0.0, 1.0)
b3 ”−−b3 ” r (0.0, 1.0)
c3 ”−−c3 ” r (0.0, 1.0)

# cluster 4
a4 ”−−a4 ” r (0.0, 1.0)
b4 ”−−b4 ” r (0.0, 1.0)
c4 ”−−c4 ” r (0.0, 1.0)
```

Listing 5.2: Parameter space

As you can see in the listing 5.2, we define five groups of parameter that are used to obtain the utility function weights. Each parameter is a real number whose range is between 0 and 1. Since the sum of the weights must be 1, we calculate the weights with the equations:

$$w_{budget}(cls) = a_{cls} \tag{5.1}$$

$$w_{pp}(cls) = (1 - w_{budget}(cls))b_{cls} \tag{5.2}$$

$$w_{env}(cls) = (1 - (w_{budget}(cls) + w_{pp}(cls))c_{cls} \tag{5.3}$$

$$w_{com}(cls) = 1 - (w_{budget}(cls) + w_{pp}(cls) + w_{env}(cls)) \tag{5.4}$$

where *cls* is the cluster of a household. We also create a configuration file, *tune-conf*, to overwrite some default options of **irace**. In particular, we set the number of digits to 3, the execution directory as *netlogo-arena* and the number of calls to the hook-run script equal to the number of available clients. We also set the instance directory to a folder in the NFS share directory; in this way, the instances are available to the clients.
The evaluation of the candidate is done by means of the auxiliary program *hook-run*. This program gets as input the instance path, the candidate index and the configuration of parameters. As output, it returns a numeric value that represents the cost of the candidate for a given instance. In addition, the *hook-run* has the task of selecting a client from those available on which to run a simulation. To avoid that two hook-run processes select the same client a lock is used. Each client available is represented by a file with its IP address as name. A new process attempts to acquire the lock on clients and if it is available, selects a client by deleting the corresponding file. At the end of the execution, it recreates the client file to make it available for future simulations.
The hook-run establishes a ssh channel with a client to launch an execution of *netlogo-exp-sim.py*. This python script takes as input the parameters configuration and the instance path. Then, it creates a xml file that contains the parameters for the NetLogo model that is used to launch a simulation. When the NetLogo simulation is finished, *netlogo-exp-sim.py* script reads the file created by the NetLogo code that contains the results of simulation. Then, it calculates the sum of squared errors (SSE) between the simulated values and actual values. The SSE is returned using the same channel opened by the hook-run. This value is used by **irace** as cost measure.

5.2 Results

In this section, we are going to summarize the results obtained with the proposed model for the adoption of PV systems. As mentioned before, our goal is to obtain a simulated trend as close as possible to the real PV power installation rate which took place between 2007 and 2012. Thus, to verify the correctness of the model, we decided to provide a NetLogo implementation of the model and then to perform various tests. These tests were carried out in an automatic way by means of the **irace** package for the R language. As described in the section, **irace** needs instances to evaluate each configuration of parameters produced during the race. An instance consists of a Shapefile and a DBF file generated by the process described in Chapter 4. For each execution of **irace**, we decided to use instances with the same number of

agents, which differ in the sample of households. Each simulation, which uses the same instance and the same parameters for the model, can produce different results. During parameters tuning, we used the same seed for the pseudo-random generator in such a way that the different candidates have the same situation. If we do not set up a seed, each simulation can produce a slightly different result, and then each candidate should be evaluated multiple times on the same instance in order to obtain a meaningful measure.
We started with instances of small size to find out the relationship between the diffusion and the number of agents. The number of agents affects the execution time of a single simulation. Figure 5.5 shows the relationship between the number of agents and the execution time.

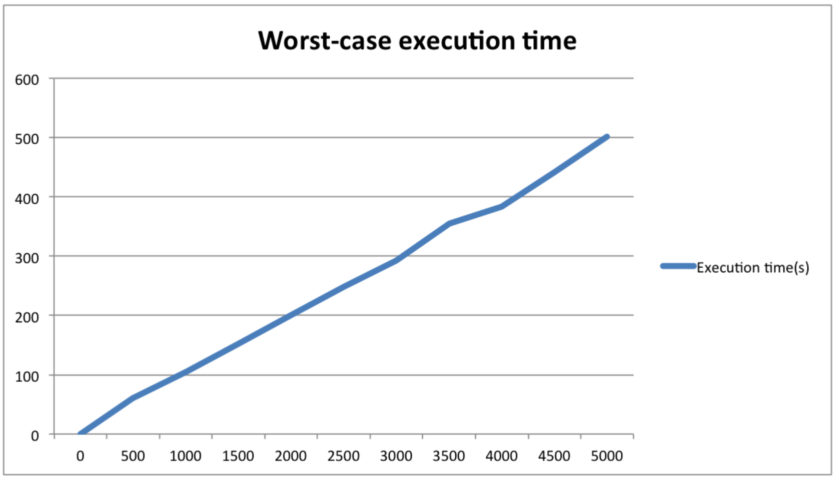


Figure 5.5: The worst-case execution time increasing the number of agents.

The Chart 5.5 was obtained by computing, for each semester and year, the best PV capacity for each agent in the social network. The PC with an intel® Core$^{\mathrm{TM}}$i5-2400 with clock frequency of 3.1 GHz and 4 GB of RAM was used to calculate these execution times. As you can see, the time required grows linearly as the number of agents increases. At first glance, these times do not seem excessive. For example, instances of 100 families require a simulation time of 20-30 seconds of seconds. But, if we decide to use instances that consist of 5,000 agents the running time increases to 8 minutes. This execution time means that the race to find the best configuration with 10000 experiments requires about 13 hours to be carried out with 100 PCs.
At the end of each run of **irace**, we have the configurations of parameters that produced the best results.
In the remainder of this chapter, we show the results obtained with different configurations and different number of agents. These results are interesting

because, as explained before, the number of agents affects the execution time and therefore it is necessary to know the relationship between the number of agents and the results. If we use a small number of agents, the simulation is faster but also more susceptible to environmental changes. Using a large number of agents the simulation is slower but the results are more significant. The following result refers the period between 2007 and 2012 where the installed capacity in Emilia-Romagna is shown in Figure 5.6. We have to manipulate the values shown in Figure 5.6 to compare the simulated installed capacity with the actual installed capacity over the period. Because the capacity installed depends on the size of roofs and the number of agents, we normalize the installed PV power. Therefore, we divide the installed capacity of each year by the installed capacity of 2007. Figure 5.7 show the growth rate for each semester under consideration.

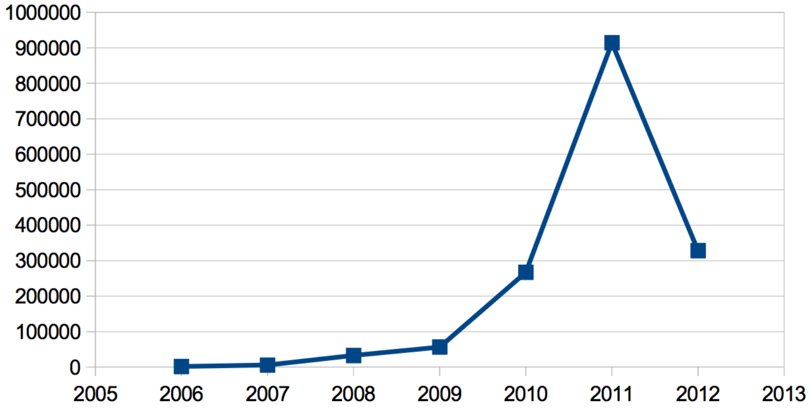


Figure 5.6: The installed capacity in Emilia-Romagna during the period from the first half of 2007 to the second half of 2012 .

Thus, once the simulation is finished we can compute the simulated growth rate for each year and compare this values to the real ones. The measure that we use to evaluate the fit is the sum of squared error (SSE) that is computed as follow:

$$SEE = \sum_{i=1}^{6} (ac_i - sc_i)^2 \quad (5.5)$$

where ac_i is the actual installed capacity in the year i and sc_i is the simulated installed capacity in the same year. It is important to obtain the smallest possible value of SSE. SSE value is affected by the weights chosen and by

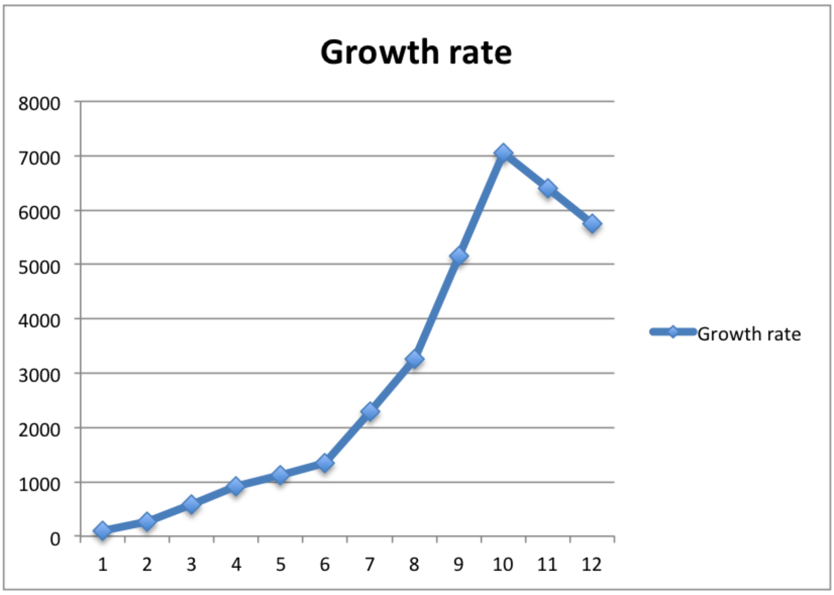
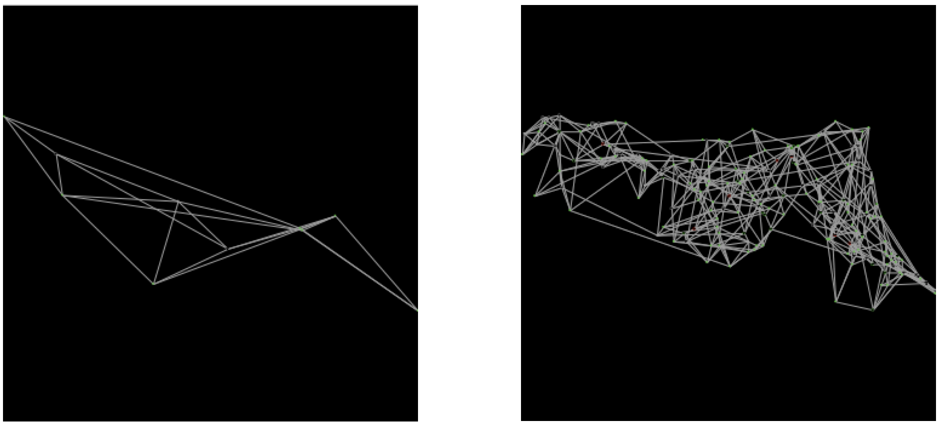


Figure 5.7: The capacity growth rate in Emilia-Romagna.

the sample of households. Further, we show the result obtained with the parameters tuning by changing the sample size.
From the simulations conducted we found that results are highly variable when the sample sizes are too small. This variability is because the sample is not statistically significant. The sample is representative when it reflects the characteristics of the population. If we create too few agents with the developed tools, we are not able to get the same distribution for attributes that are observed in SHIW data. So in this section we focus on samples with a size at least 100 families.
Figure 5.8a shows the arrangement of the families in the virtual world and the network of relationships when the sample size is 20. If we increase the sample size to 200, the result is that shown in Figure 5.8b. In this case, the shape of the region is clearer.

An important parameter for **irace** is the number of experiments, namely the maximum number of invocations of hookRun that will be performed by **irace**. The number of experiments determines how profoundly **irace** explores the parameter space. A low number, especially when we have a large number of parameters, doesn't allow **irace** to explore adequately the search space with the result that the parameter configurations found are not the best. Obviously, we tested various values for this parameter until we have obtained satisfactory results. In theory, high value for this parameter allows to get best results, but the time to complete a race increases and becomes



(a) The virtual world with 20 agents. (b) The virtual world with 200 agents.

Figure 5.8: Comparison of virtual world with 20 and 200 agents.

unsustainable.

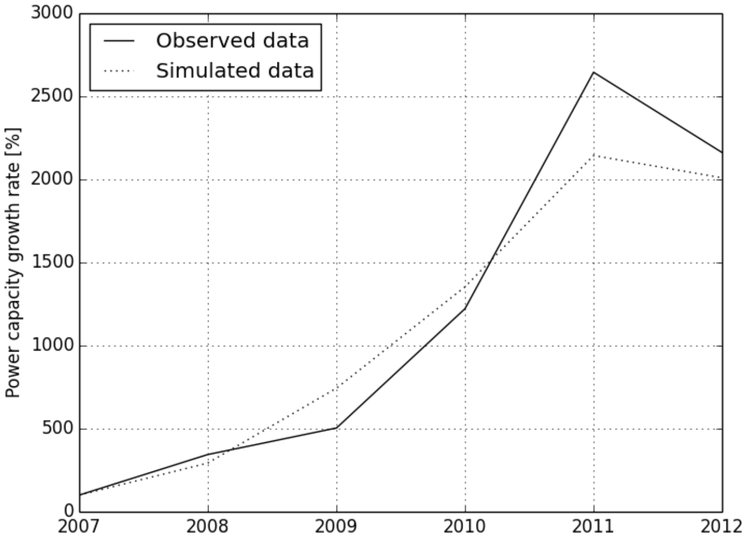


Figure 5.9: Calibration result.

Figure 5.9 shows the calibration results for 2000 agents. The weights found are shown in Table 5.2. These weights are calculated using the equations 5.1, 5.2, 5.3 and 5.4. The utility function threshold is set to 0.5 for all classes.

Agent class	Weight type	Value
Class 0	w_{com}	0.5409
	w_{pp}	0.0746
	w_{env}	0.0062
	w_{budget}	0.5399
Class 1	w_{com}	0.9301
	w_{pp}	0.0625
	w_{env}	0.1173
	w_{budget}	0.2497
Class 2	w_{com}	0.9421
	w_{pp}	0.0535
	w_{env}	0.0349
	w_{budget}	0.1463
Class 3	w_{com}	0.8630
	w_{pp}	0.0078
	w_{env}	0.0040
	w_{budget}	0.1488
Class 4	w_{com}	0.8533
	w_{pp}	0.1176
	w_{env}	0.0100
	w_{budget}	0.2743
	rewiring probability	0.1846

Table 5.1: Model parameters.

Year	Percentage error
2007	0.0
2008	0.513
2009	-2.399
2010	-1.317
2011	5.011
2012	1.522

Table 5.2: Annual percentage errors.

As previously mentioned, the execution time is reasonable if we execute the model one time, but the calibration model requires more than one execution. Thus, we decide to calibrate the model on 2000 agents that allow us to obtain an SSE of 35.18. The annual percentage errors are shown in Table 5.2. The growth rate errors are relative small, and this allow us to fit well the curve in Figure 5.7. Even if the rate error in the 2011 is about 5%, the simulated curve has a similar trend as the real curve. As shown in Figure 5.4, the proposed model allow us to replicate in the virtual world the actual diffusion of PV systems in Emilia-Romagna. The results are very encouraging because the system that is populated by autonomous and interacting decision-making entities was able to recreate the conditions of diffusion. A surprising result is the reduction of PV power capacity installed after 2011 that reflect the actual pattern. In fact, the installed capacity in the virtual world between the 2007 and 2011 has a positive trend, but in 2012, we reported a brusque reduction. This characteristic of the curve is difficult to reproduce with pure mathematical methods. For this reason, the described process has proven to be valuable for modeling the diffusion of residential PV systems. However, the followed process can be used to model many of diffusion patterns that involve social interactions.

Conclusions

In this work, we have set ourselves the aim of studying the diffusion of residential PV systems in Emilia-Romagna. To this end, we have developed an agent-based model to replicate the interactions and behaviours of households, in order to reproduce and understand the dynamics that lead these families to invest in a PV panel. We started by analysing data on the incomes, savings, wealth and other aspects of Italian households. These data has allowed us to get the statistics regarding the family living in Emilia-Romagna and develop tools that can generate representative samples of families. In addition, these tools arrange the families in the territory by placing them in buildings obtained by preprocessing the vector data file provided by the region. Next, these families are linked together to form a small-world network. To accomplish this, we have proposed a ranked based technique that consider the distance and the attribute proximity to establish a link between two nodes. After that, devised a simulation in order to reproduce and evaluate the residential PV systems diffusion in Emilia-Romagna over the 2007-2036 period. Each year and semester, the model implemented in our simulator estimates the PV power capacity and others important statistics, evaluating the desire to purchase a PV system for each family considered in the simulation.The desire level takes into account the household's economic condition, the payback period of investment, the influence of neighbours and the environmental benefits deriving from the adoption. After an accurate and automated fine-tuning of the parameters used in our model, we were able to achieve good results in terms on PV installed power, i.e. the adoption rate obtained through our simulator closely resembles the historical Emilia-Romagna PV plants installation trend.

Further improvements can be made on the model. For example, we can improve the estimation of household's consumption. Now, the consumption of a family is estimated simply through mapping the number of members to average annual consumption provided by Enel. This assessment is static and does not consider many aspects such as the dimension of the house, the type of job, the number of children, etc. In addition, it would be interesting being able to analyse the real data on installed systems and their owners because it can help to better understand the relationships between the characteristics of the system and the attributes of the owner. Also, studying the reasons that led an adopter to purchase a PV system could help us to get more insights and better understand the underlying social phenomena. Others factors can be found and therefore they can be taken into account to evaluate the desire level of purchasing a PV system. However, it is necessary to keep the model simple because it must be able to respond to changes of the environment. If we introduce many factors, the model may become too complex and difficult to verify.

This project have helped me to realise that the public politics are complex. Decision makers have to deal with various constraints and have to analyse a large amount of data. I came to the conclusion that the DDS are valuable tools that can help policy makers to design plans able to meet the requirements and achieve the objectives. Moreover, this work has allowed me to understand the power and flexibility of ABMs as tools to model social and other aspects. The entities and behaviours that are contemplated by ABMs are almost always simple, but the resulting systems can produce surprising outcomes, thanks to the emergence of unpredicted behaviours. In addition, this thesis has permitted me to know the R package **irace** for the parameter tuning. Many optimisation algorithms have various parameters that we need to set before apply them on an instance of the problem. **irace** allows to find a good configuration for these parameters in order to have better results. A configuration may resolve the problem faster than another one. Therefore, learning about parameter tuning tools is crucial when we are dealing with optimisation problems often.

Bibliography

Abrahamson, E. and Rosenkopf, L. (1997). Social network effects on the extent of innovation diffusion: A computer simulation. *Organization science*, 8(3):289–309.

Autorità per l'energia elettrica e il gas (2008). Delibera een 3/08 - aggiornamento del fattore di conversione dei kwh in tonnellate equivalenti di petrolio connesso al meccanismo dei titoli di efficienza energetica. Technical report, Autorità per l'energia elettrica e il gas.

Balaprakash, P., Birattari, M., and Stützle, T. (2007). Improvement strategies for the f-race algorithm: Sampling design and iterative refinement. In *Hybrid Metaheuristics*, pages 108–122. Springer.

Bank of Italy (2012). Survey on household income and wealth. Technical report, Bank of Italy.

Belletti, R. (2014). Tuning automatico di parametri in un sistema di controllo semaforico auto-organizzante. Master's thesis, Universitá di Bologna, Corso di Studio in Ingegneria informatica [LM-DM270].

Birattari, M., Yuan, Z., Balaprakash, P., and Stützle, T. (2010). F-race and iterated f-race: An overview. In *Experimental methods for the analysis of optimization algorithms*, pages 311–336. Springer.

Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences of the United States of America*, 99(Suppl 3):7280–7287.

Borghesi, A. (2013). Integrazione di ottimizzazione e simulazioni per il piano energetico regionale dell'emilia-romagna. Master's thesis, Universitá di Bologna, Corso di Studio in Ingegneria informatica [LM-DM270].

Borghesi, A., Milano, M., Gavanelli, M., and Woods, T. (2013). Simulation of incentive mechanisms for renewable energy policies. In *ECMS*, pages 32–38.

Chatterjee, R. A. and Eliashberg, J. (1990). The innovation diffusion process in a heterogeneous population: A micromodeling approach. *Management Science*, 36(9):1057–1079.

Colorni, A., Dorigo, M., Maniezzo, V., et al. (1991). Distributed optimization by ant colonies. In *Proceedings of the first European conference on artificial life*, volume 142, pages 134–142. Paris, France.

Easley, D. and Kleinberg, J. (2010). *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press.

Emilia-Romagna (2014). Dati emilia-romagna.

European Commission (2009).

GSE (2011). Rapporto statistico 2011: Solare fotovoltaico. Technical report.

GSE (2014). Totale dei risultati del conto energia (primo, secondo, terzo, quarto e quinto conto energia) - ripartizione per regione e classe di potenza degli impianti in esercizio. aggiornamento al 30 aprile 2014. Technical report, Gestore Servizi Energetici (GSE), Rome, Italy.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. *SCIENCE*, 220(4598):671–680.

Liben-Nowell, D., Novak, J., Kumar, R., Raghavan, P., and Tomkins, A. (2005). Geographic routing in social networks. *Proceedings of the National Academy of Sciences of the United States of America*, 102(33):11623–11628.

López-Ibáñez, M., Dubois-Lacoste, J., Stützle, T., and Birattari, M. (2011). The irace package, iterated race for automatic algorithm configuration. Technical Report TR/IRIDIA/2011-004, IRIDIA, Université Libre de Bruxelles, Belgium.

Palmer, J., Sorda, G., and Madlener, R. (2013). Modeling the diffusion of residential photovoltaic systems in italy: An agent-based simulation. Technical report, E. ON Energy Research Center, Future Energy Consumer Needs and Behavior (FCN).

pandas community (2012). pandas: Python Data Analysis Library. Online.

QGIS Development Team (2009). *QGIS Geographic Information System.* Open Source Geospatial Foundation.

R Development Core Team (2008). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Robinson, S. A., Stringer, M., Rai, V., and Tondon, A. (2013). Gis-integrated agent-based model of residential solar pv diffusion.

Rogers, E. (2003). *Diffusion of Innovations, 5th Edition.* Free Press.

Schilling, M. and Izzo, F. (2013). *Gestione dell'innovazione.* Collana di istruzione scientifica. Serie di discipline aziendali. McGraw-Hill Companies.

Tan, P.-N., Steinbach, M., and Kumar, V. (2007). *Introduction To Data Mining.* Pearson Education.

Travers, J. and Milgram, S. (1969). An experimental study of the small world problem. *Sociometry*, pages 425–443.

Watts, D. J. and Strogatz, S. H. (1998). Collective dynamics of 'small-world'networks. *nature*, 393(6684):440–442.

Wikipedia (2004). Simulated annealing — Wikipedia, the free encyclopedia. [Online; accessed 22-July-2004].

Wikipedia (2014). Diffusion of innovations — wikipedia, the free encyclopedia. [Online; accessed 12-September-2014].

Wilensky, U. (1997). Netlogo ants model. Technical report, Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

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

Zhao, J., Mazhari, E., Celik, N., and Son, Y.-J. (2011). Hybrid agent-based simulation for policy evaluation of solar power generation systems. *Simulation Modelling Practice and Theory*, 19(10):2189–2205.