*Year :* 2011

# Policy Diffusion: An Agent-Based Approach

Luyet Stéphane

Université de Lausanne

Faculté des sciences sociales et politiques (SSP)

Institut d'études politiques et internationales (IEPI)

Thèse en vue de l'obtention du titre de

docteur ès sciences politiques

de l'université de Lausanne

# Policy Diffusion: An Agent-Based Approach

Stéphane Luyet

Thèse présentée le 31.05.2011

Devant le jury constitué de

**Prof. Dietmar Braun (directeur), Université de Lausanne**

Prof. Fabrizio Gilardi, Université de Zurich

Prof. Henry Volken, Université de Lausanne

LAUSANNE 2011

UNIL | Université de Lausanne
Faculté des sciences
sociales et politiques

**IMPRIMATUR**

Le Conseil de la Faculté des sciences sociales et politiques de l'Université de Lausanne, sur proposition d'un jury formé des professeurs

- Dietmar Braun, directeur de thèse, professeur à l'Université de Lausanne,
- Fabrizio Gilardi, professeur à l'Université de Zürich
- Henri Volken, professeur à l'Université de Lausanne

autorise, sans se prononcer sur les opinions du candidat, l'impression de la thèse de Monsieur Stéphane Luyet, intitulée :

***« Policy Diffusion : An Agent-Based Approach» .***

Lausanne, le 27 avril 2011

***Le Doyen de la Faculté***

Professeur
René Knüsel

**Abstract**

Methods like Event History Analysis can show the existence of diffusion and part of its nature, but do not study the process itself. Nowadays, thanks to the increasing performance of computers, processes can be studied using computational modeling. This thesis presents an agent-based model of policy diffusion mainly inspired from the model developed by Braun and Gilardi (2006). I first start by developing a theoretical framework of policy diffusion that presents the main internal drivers of policy diffusion – such as the preference for the policy, the effectiveness of the policy, the institutional constraints, and the ideology – and its main mechanisms, namely learning, competition, emulation, and coercion. Therefore diffusion, expressed by these interdependencies, is a complex process that needs to be studied with computational agent-based modeling. In a second step, computational agent-based modeling is defined along with its most significant concepts: complexity and emergence. Using computational agent-based modeling implies the development of an algorithm and its programming. When this latter has been developed, we let the different agents interact. Consequently, a phenomenon of diffusion, derived from learning, emerges, meaning that the choice made by an agent is conditional to that made by its neighbors. As a result, learning follows an inverted S-curve, which leads to partial convergence – global divergence and local convergence – that triggers the emergence of political clusters; i.e. the creation of regions with the same policy. Furthermore, the average effectiveness in this computational world tends to follow a J-shaped curve, meaning that not only time is needed for a policy to deploy its effects, but that it also takes time for a country to find the best-suited policy. To conclude, diffusion *is* an emergent phenomenon from complex interactions and its outcomes as ensued from my model are in line with the theoretical expectations and the empirical evidence.

**Résumé**

Les méthodes d'analyse de biographie (event history analysis) permettent de mettre en évidence l'existence de phénomènes de diffusion et de les décrire, mais ne permettent pas d'en étudier le processus. Les simulations informatiques, grâce aux performances croissantes des ordinateurs, rendent possible l'étude des processus en tant que tels. Cette thèse, basée sur le modèle théorique développé par Braun and Gilardi (2006), présente une simulation centrée sur les agents des phénomènes de diffusion des politiques. Le point de départ de ce travail met en lumière, au niveau théorique, les principaux facteurs de changement internes à un pays : la préférence pour une politique donnée, l'efficacité de cette dernière, les contraintes institutionnelles, l'idéologie, et les principaux mécanismes de diffusion que sont l'apprentissage, la compétition, l'émulation et la coercition. La diffusion, définie par l'interdépendance des différents acteurs, est un système complexe dont l'étude est rendue possible par les simulations centrées sur les agents. Au niveau méthodologique, nous présenterons également les principaux concepts sous-jacents aux simulations, notamment la *complexité* et l'*émergence*. De plus, l'utilisation de simulations informatiques implique le développement d'un algorithme et sa programmation. Cette dernière réalisée, les agents peuvent interagir, avec comme résultat l'émergence d'un phénomène de diffusion, dérivé de l'apprentissage, où le choix d'un agent dépend en grande partie de ceux faits par ses voisins. De plus, ce phénomène suit une courbe en S caractéristique, poussant à la création de régions politiquement identiques, mais divergentes au niveau globale. Enfin, l'efficacité moyenne, dans ce monde simulé, suit une courbe en J, ce qui signifie qu'il faut du temps, non seulement pour que la politique montre ses effets, mais également pour qu'un pays introduise la politique la plus efficace. En conclusion, la diffusion *est* un phénomène émergent résultant d'interactions complexes dont les résultats du processus tel que développé dans ce modèle correspondent tant aux attentes théoriques qu'aux résultats pratiques.

# Remerciements

Un tel travail doit tellement à tellement de personnes que j'espère n'en oublier aucune (si ça devait arriver et ça va certainement arriver, j'espère également que les malheureux oubliés sauront me pardonner).
Ma toute première pensée va à Virginie qui m'a toujours poussé à questionner la nécessité de la voie, notamment méthodologique, suivie tout au long de ce travail.
Il faut maintenant entrer dans le vif du sujet. Ainsi je tiens à remercier plus que chaleureusement le Professeur Fabrizio Gilardi qui m'a, le premier, mis le pied à l'étrier des simulations informatiques (toutes ces couleurs qui bougent ne cesseront jamais de me fasciner) et qui par la suite m'a accompagné et soutenu tout au long de ce travail grâce à ses conseils et remarques plus que précieux. Le Professeur Henri Volken mérite aussi ma reconnaissance pour les cours dispensés, notamment sur la complexité (j'ai lu *La Proie* de Michael Crichton, ce qui a achevé de me convaincre de la pertinence des modèles centrés sur les agents) et la théorie des graphes et les discussions qui m'ont ouvert de nouveaux horizons, surtout pour le futur de ce travail. Le Professeur Dietmar Braun mérite mes remerciements les plus sincères pour l'ouverture d'esprit dont il a fait preuve face à une méthodologie un peu *ésotérique* dans la boîte à outil du politologue.
Dans un monde où l'altruisme pur est une denrée relativement rare, je crois que je ne pourrais jamais assez remercier Marek Blaszczyk qui, au prix d'une infinie patience et de beaucoup de temps, à réussi à m'inculquer quelques notions de programmations en Java et qui m'a plus qu'aidé dans les prémisses de ce travail. Outre la satisfaction d'apercevoir quelques éclairs de compréhension sur mon visage, je ne peux donc juger que des raisons purement altruistes qui l'ont poussé à m'aider et l'en remercier. Je garde jalousement un dé, même s'il en faudrait plusieurs....
Ce travail n'aurait certainement jamais vu le jour sous cette forme sans ma participation à différents cours d'introduction à la programmation orientée objet. Ces participations, en tant qu'auditeur libre, n'ont pu se faire sans l'accord des professeurs. Ainsi je tiens ici à remercier tout particulièrement le Professeur Bertrand Meyer de l'ETHZ qui s'est toujours montré encourageant face à mes progrès très hésitants. L'ETHZ compte également parmi son corps professoral le Professeur Lars-Erik Cederman qui dispense des cours d'introduction et avancé en modélisation centrée sur les agents appliqués aux sciences sociales. Durant ses cours, j'ai appris les bases et fourbi mes premières armes dans ce domaine.
Je me dois ici de remercier tous les collègues de l'IEPI. Au cours de ces trois années d'assistanat, j'ai beaucoup apprécier les conversations et les échanges fructueux sur des sujets divers et variés, dont certains ne peuvent (mal)heureusement pas être mentionnés ici.

L'année passée à Constance au bénéfice d'une bourse pour chercheur débutant du fonds national suisse de la recherche scientifique (fonds *n$^o$* PBLA11-119660), au sein de la formidable équipe dirigée par le professeur Christopher Knill, à programmer différents modèles a été plus que bénéfique puisque cette vie un peu ascétique m'a permis de développer mon modèle jusqu'à obtenir la version présentée dans cette thèse. Avant d'en arriver là, plusieurs modèles intermédiaires ont été présentés lors de conférences où, à ma grande surprise, ils ont suscité beaucoup d'intérêt. Ainsi les participants à l'*Annual Midwest Political Science Conference* de 2007 et 2009 (panel *Comparative Politics: Industrialized Countries*) à Chicago et au workshop *International Relations* de la Conférence annuelle de l'Association Suisse de Science Politique en 2006 et 2009, en particulier le Professeur Cederman pour ses conseils toujours pertinents.

A ma sortie du monde académique, j'ai été engagé en tant que chef de projet à l'observatoire suisse de la santé, où j'ai été accueilli à bras ouverts. Je veux remercier toute l'équipe (ancienne et nouvelle) et en particulier Paul et Hélène pour la confiance témoignée, Marcel pour les discussions toujours intéressantes et l'aide *cartographique* inestimable, Gaëlle (je n'ai pas souvent l'opportunité d'impressionner quelqu'un avec mon travail) et Sarah (j'ai presque tout débarrassé!). En ce qui concerne les collègues, une mention spéciale est décernée à Katharina qui m'a toujours beaucoup aidé. C'est toujours un plaisir de travailler avec toi.

Les organisateurs des marathons de Stockholm et de Berlin méritent aussi une petite pensée. 42 km à l'autre bout de l'Europe, ça laisse du temps pour penser à la thèse dans la mesure où il fallait bien remplacer la douleur physique, surtout quand Virginie n'apparaît pas au dernier tournant (satané mauvais temps... J'espère que c'est pardonné. De toute façon, je ne souriais pas... ou plus)...

***A mes parents: un tout grand merci et... ça a pu paraître long, mais, aussi étonnant que cela puisse paraître, ça a fini par arriver.... J'ai fini! Et Janique, merci pour ton indéfectible soutien (et les offres répétées d'aide logistique)!***

Enfin, last but never least, le peu de lignes que je peux leur accorder ici ne leur rendra pas l'hommage mérité, parce qu'en plus de tout, on a bien ri de (beaucoup) et avec (un peu) moi. Special dedicace à Virginie, Michel, Floriane, Séba, Vivi, Rim, Pierre-Antoine, Katharina, Nadja, Sandrine, David (encore désolé...), Hugues, Jean-Marc et Alida...

A Virginie. Merci pour tout... et qui mérite tellement plus

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

The idea that a policy diffuses is not new. The fact that political choice is not an independent, but rather an interdependent, process is now well-established in political science. Most methods used in the study of diffusion, like Event History Analysis (EHA), can show the existence of diffusion and help explain its nature, but not study the process itself. Moreover, the traditional methods can highlight either the country (internal) factors or the external (international) factors that explain diffusion. Thus, diffusion, as will be explained throughout this work, should be seen as an emerging process resulting from interactions between autonomous and heterogeneous countries. This way of studying diffusion is largely excluded from the "classical" political science and needs to be asserted.

Therefore, in this thesis, we will build a computational model, mainly inspired from the one developed at the theoretical level by Braun and Gilardi (2006), that will help explain diffusion as a global pattern emerging from microlevel interactions:

> *"How can the international policy diffusion be explained on the basis of the interdependencies that exists between countries, since it occurs between autonomous and heterogeneous countries, for the most part without any central authority?"*

In this introduction, to have a broader view of the impact of policy diffusion, we will first rely on some real-world examples. From that those examples, the main implications of the use of the concept of diffusion in political science research will be highlighted. Before briefly developing the structure of this thesis, the significance of the use of computational agent-based modeling will be tackled.

## 1.1 What will be learned?

Diffusion is an interdisciplinary concept, since it is studied in several scientific fields, both in natural sciences, such as physics, or social sciences, such as economics, sociology, or political science.

A look at daily newspapers can give numerous examples of diffusion of several sorts, such as the contagion of revolutions, the economic crisis or the propagation of viruses, whether they be computational or infectious. For instance, the early eighties have seen the emergence and the propagation of a very dangerous infection, AIDS, which has spread through different communities to the point of contaminating entire populations. At the political level, answers were needed in order to find ways to fight this plague. Consequently, each country had to develop a response. Therefore, the problem for this research is to highlight how the different countries have influenced each other in creating such a policy, according to the fact that spurious diffusion may be an option.

Another problem of public health gives a typical example that can be applied to the study of diffusion, namely anti-smoking laws. The spread of anti-smoking laws has been studied in the case of the United States of America. Shipan and Volden (2006) explained the reasons for this spread. They highlight the role of learning as the main mechanism at play and that diffusion driven by learning is taking place in states that are more populated. However, bans on smoking have also spread throughout the European countries, including Switzerland, which introduced a federal law effective May 1, 2010. However, between April 2007 and May 2010, half of the Swiss cantons had already introduced such a law[1]. In the particular case of Switzerland, the whole process of diffusion began in 2005 with the canton of Tessin, which discussed

[1] http://www.bag.admin.ch/themen/drogen/00041/03814/03815/index.html?lang=fr

an anti-smoking bill a few months after its southern neighbor, Italy, introduced a rather restrictive bill to ban smoking from public places. This has inspired the following comment from a journalist from the *Neue Zürcher Zeitung* (NZZ): "What was possible without much trouble for its big neighbor since the beginning of the year, should also be, after the wish of the government and a majority of the parliament, as quick a reality as possible in canton Tessin"[2].

The particular case of health care policies offers other interesting examples of diffusion. For instance, the introduction of diagnosis-related groups (DRGs) for the financing of hospitals in 2012 at the Swiss federal level illustrates the impact of diffusion for the shape of a particular policy. Switzerland, in this particular case, has been influenced by the DRG system introduced in Germany, which in turn has been affected by the system's Austria has developed. This chain of inspiration can be traced back to the development of the DRGs at Yale University at the late sixties (Gilardi, Füglister and Luyet, 2009).

These two examples perfectly illustrate learning as a mechanism of diffusion. However, other mechanisms can be at play when discussing diffusion, namely emulation and competition. To put some emphasis on the former mechanism, we can rely on the propaganda of the Swiss People's Party that has, for the launch of and the vote on an initiative on the deportation of the foreign lawbreakers, covered Switzerland with posters representing white sheep that are kicking a black (foreign criminals) sheep out of Switzerland. This poster campaign was emulated elsewhere in Europe, notably in Germany and in Portugal.

When talking about competition, Swank (2008) argues that it is the main driver of international tax policy diffusion. In addition, Switzerland gives a notable case with the fiscal competition that exists between the cantons. Gilardi and Wasserfallen (2010) show that the competition between Swiss cantons is an important driver of fiscal policy change. However, this race to the bottom is not only countered by political and institutional constraints, but also by the participation in intergovern-

[2] My own translation from `http://www.nzz.ch/2005/10/12/il/articled7ykk_1.176468.html`.

mental networks. Thus, empirical evidence on the existence of diffusion of policies are manifold and so are the political topics.

At the more general level, diffusion stretches from the global diffusion of market globalization (Simmons, Dobbin and Garrett, 2008) to democracy (O'Loughlin et al., 1998). Furthermore, empirical studies explore more specific topics, mainly in welfare state arrangements such as the spread of pension reforms (Brooks, 2007); health care reforms (Gilardi, Füglister and Luyet, 2009), as already mentioned above, or unemployment benefits retrenchment (Gilardi, 2010).

Most of these studies can show the existence of diffusion as the result of an interdependent process, but they cannot show the process itself. Moreover, a distinction is made between internal (domestic) factors of policy diffusion (see e.g. Swank and Steinmo, 2002; Swank, 2006) and external (international) factors as already mentioned (see e.g. Gilardi, Füglister and Luyet, 2009). What is generally missing is the link that exists between micro and macro features. For instance, Swank and Steinmo (2002) explained the diffusion of tax policies through capitalist democracies with the help of two sets of explanations. Firstly, domestic factors, mainly unemployment and public sector debts, influence tax policy, although not in the same direction and, at the international echelon, the significance of capital mobility and trade has been emphasized. Nevertheless, the link between these two sets of explanations is not made.

Therefore, an attempt will be made here to study the process of diffusion and the influence of the possible interactions with a more appropriate methodology that allows one to see this process in its global nature; that is, as a result of micro interactions. In other words, the purpose of this thesis is exactly to highlight the spread of policy at the macro level, as a consequence of interactions between countries. Moreover, we do not have the ambition to explain the spread of a particular policy, but we participate to the scientific on policy diffusion by emphasizing the theoretical underpinning of policy diffusion at the global context from local interactions. More precisely, we would like to underline how the process of diffusion evolve and apply

this to all policies. The difficulty here is to be general enough to develop a comprehensive model for the understanding of diffusion and restricted enough for that understanding to be useful for the explanation of the process of diffusion.
In other words, the aim of this thesis is to develop a theoretical framework for the explanation of policy diffusion and the potential clustering that grows from this process due to the influence of domestic factors of countries and the different interdependencies that exist between countries, which leads us to the question posed at the beginning of this introduction.

## 1.2 What are the implications of diffusion processes?

The study of the diffusion of innovation; that is, the adoption of a new concept, may it be technical or ideal, is of great help for our purposes. It can be traced back to the early forties with the Ryan and Gross' study of the diffusion of hybrid corn seeds in rural Iowa (Rogers, 2003). Even if it is a rather specific case, this study emphasized the most important results of policy diffusion, notably the famous S-shaped curve of policy diffusion, which leads to convergence. The fact that policies become more similar through time – the definition of convergence – is only one result of the process of diffusion (Braun and Gilardi, 2006). In other words, convergence is one possible *result* of the *process* of diffusion (Gilardi, 2011).
However, when one looks at the evolution of a map of the world considering the implementation of the different main type of welfare states throughout the world during the $20^{th}$ century, one can see whole groups of neighboring countries introducing the same type and developing their welfare state regimes according to the one of their neighbors, which has given rise to the well-known typology of welfare states developed by Esping-Andersen (1990) that is, Christian democratic, social democratic, and liberal. In the same manner, health care systems can be typologized as follows (Palier, 2004):

1. <u>National Health Service systems</u>: Such systems are characterized by free global

health care coverage and are in place mainly in northern Europe (Sweden, Denmark, United Kingdom) and southern Europe (Italy, Spain, Portugal and Greece);

2. Health Insurance systems: These systems are mainly developed around health insurance paid by social contributions and are in place in countries such as France, Germany, Luxembourg, Belgium and Austria;

3. Liberal systems: These systems are mainly built around the notion of individual responsibility. Under this type can be classified the United States, Switzerland, and some Latin American countries.

Clearly, this shows some extant divergence in the world. This regional clustering is highlighted in many other fields involving diffusion such as, for example, the dissemination of democracies (see e.g. Elkink, 2009; Gleditsch and Ward, 2006). The explanation of such regional patterns, in time and space, needs to be done not only on the basis of purely internal factors, but also with the help of external pressures. Consequently, the interdependencies that exist between countries must be studied as they are.

The problem is that, in comparative political studies, this nonlinearity is, too often, not taken into account, meaning that countries are treated as independent cases. Moreover, internal factors are mainly operationalized in studies that show spurious diffusion, i.e., 'the fact that a pattern may look like diffusion even though it is not driven by diffusion' (Braun and Gilardi, 2006, 299), or the nonexistence of diffusion as an important driver of policy change (Simmons and Elkins, 2004). In other words, the procedures that lead to a policy change are mostly internal to the countries. This means that the different countries are considered as independent. Therefore, this lack of independence between cases in comparative analyses, labeled as "Galton's problem," must be fought and researchers must pay attention to these interdependencies.

Even if the interdependent paradigm has gained in importance in political science thanks to the development and use of new methods (see e.g. Strang and Soule, 1998; Berry and Berry, 2006), with the results that the different mechanisms of diffusion are now well documented at the theoretical and empirical level, most of the empirical efforts are made at the sub-national level, leaving the cross-national level largely neglected (Gilardi, 2011).

Nevertheless, only a few attempts have been made to develop a comprehensive framework for the study of policy diffusion (Braun and Gilardi, 2006). Furthermore, except a few tries (see e.g. Axelrod, 1997*b*; Elkink, 2009), diffusion has to this point not been studied as a process. More precisely, without minimizing the influence of internal factors, most studies in the field of diffusion research focused on the external factors as represented by the different mechanisms (Gilardi, Füglister and Luyet, 2009). Thus, efforts should be made to integrate, in the same model, not only domestic and international factors as done, for instance, by Swank and Steinmo (2002) but also the different interactions that exist between the different countries that is the link between these two kinds of features.

For instance, looking at the diffusion of democracy, Cederman and Gleditsch (2004) investigate the spread of democracy by waves, which arise from a statistical analysis, by linking micro- and macro-level processes. They postulate that the more democratic states that surround a nondemocratic state, the higher the probability that the latter will become a democracy. Looking at the problem of the security of a democratic state in a nondemocratic environment, Cederman and Gleditsch's results show that the emergence of democratic clusters corresponds to a collective security mechanism.

Thus, the study of policy diffusion has practical significance, since it implies the study of the causes and consequences at the political level of the interactions that exist among the countries. In Section 1.1, we emphasized different real-world examples of diffusion processes. For instance, the study of the propagation of anti-smoking laws is a textbook case for the *diffusionists*, as it involves the main ingredients of policy diffusion theory. Once again, even if lots of studies have highlighted the existence of diffusion, as we will see in the next chapter, less is done to theorize and understand the process as a whole.

## 1.3 A first contact with agent-based modeling

From Sections 1.1 and 1.2 appears the problem of the operationalization, in the same model, of the internal and external factors, and of the link that exists between them. That is why we will develop our model using a particular methodology, which can take into account the interdependencies between the different agents. Such a methodology, defined as a ‘third way of doing science’ (Axelrod, 2003, 5), is called computational agent-based modeling.

The idea of studying diffusion in the sense of a theory-building development means that we do not need to develop hypotheses. For instance, the results of the model of segregation developed by Schelling (1978) are interesting because of its theoretical counter-intuitive results based on very simple idealized rules (Epstein, 2005). Such a model does not need to be a perfect representation of reality, since too many variables hugely diminished the explaining power of the model, as the phenomenon one wants to study is drowned in the details.

Hence, only main factors that drive the process need to be operationalized (or programmed, since our model is computationally developed). This means that we do not know the results of the interactions in such a model. In other words, the different initial conditions and the degree to which the model can explain the process are the red line of this work.

In terms of diffusion, two main concepts in the literature try to capture the behavior of the different countries/agents. First of all, we face the concept of threshold (Granovetter, 1978). The idea is that different agents facing the same phenomenon will be differently affected by the behavior of their neighbors *vis-à-vis* this phenomenon. The second notion that helped develop a comprehensive theory of policy diffusion comes from the aggregation of the different thresholds, namely bandwagon pressures (Abrahamson and Rosenkopf, 1997; Rosenkopf and Abrahamson, 1999). For instance, Abrahamson and Rosenkopf (1997) developed a computational model that helps understand the diffusion of innovations between organizations at the economical level. Moreover, their model stressed that organizations, before adopting the innovation, fix their threshold by assessing their potential profits (losses) from the innovation. Since they are uncertain about the future, the former adopters influence the latter. Thus, the aggregation at the global level of the different thresholds give rise to bandwagon pressures, defined as the more adopters of an innovation, the greater the pressure is for adoption.

Methodologically, we will develop our comprehension of the process of diffusion building and running a computational agent-based model. Such a tool is relatively new in the political scientist's toolbox, even if its use is not really new, and can be traced back to the late seventies and the segregation model (Schelling, 1978) and early eighties with the example of the evolution of cooperation (Axelrod, 1984).

Nowadays, thanks to the development of the power and accessibility of personal computers, such a methodology can be used by more and more researchers. In other words, following Moore's law stressing that the power of computer chips increases twice every 18 months and sharply decreases computer price, the development of agent-based toolkits has been important. Nevertheless, the computer is not the point, since Schelling has developed his model without the use of computers.

What is central to the development of agent-based models is whether or not one can generate macro level structures from micro level interactions (Epstein, 2006). The problem here is that without computers it would be impossible to develop complex models, since it is nearly impossible to calculate the exponential complex interactions between an increasing number of agents without the help of computers (Holland, 1998).

A computational agent-based model is thus a model that help study the results at the global level that ensue from the local interactions, based on a few simple assumptions and develop as a computer program. Moreover, such models can deal with the nonlinearity that characterizes the interrelations between the agents, which characterize the process of policy diffusion, and the interdependencies that may exist between the independent variables in more "traditional" methods. Put differently, the majority of studies run quantitative analyses that, despite their sophistication, are biased towards correlational accounts of diffusion that, in the end, have little to say about the process by which policies diffuse. Therefore, the researcher highlights the weight of the mechanism of diffusion under study, but cannot study the process of diffusion as a whole. For instance, the classical quantitative method to study diffusion is event-history analysis (EHA). Such a method, by controlling domestic and international influences, can show the existence of diffusion and part of its results, but not the process itself (Gilardi, 2011).

As noticed above, the process of policy diffusion can be considered nonlinear. Consequently, the development of a computational agent-based model for the study of the theoretical model seems to be the most interesting methodology to use, because it is a powerful tool for theory development (Gilbert and Terna, 2000; Adner, 2002; Repenning, 2002; Davis, Eisenhardt and Bingham, 2007) and a well-suited tool for the study of processes (Sastry, 1997; Gilbert, 1998; Bonabeau, 2002; Rudolph and Repenning, 2002; Axelrod, 2003).

Two main concepts need to be introduced in order to fully understand the *beauty* of computational agent-based models, namely complexity and emergence. The complexity characterizes a system where the whole is more than just the sum of its parts and is represented by nonlinear interactions between its components.

In the context of policy diffusion, the nonlinearity can be exemplified as follows: If the costs for acquiring relevant information about new policies are divided by two, as a consequence, the number of countries interested in these new policies does not increase by two, if we assume that the diminishing costs increase the interest in the new policy. Thus the interactions between the different countries can lead to unexpected results at the macro level. Consequently, emergence can be defined as unexpected global patterns that arise from local interactions. The fact that the global patterns are unexpected means that they are not directly programmed, only the rules of interactions are. For instance, Schelling's segregation model is programmed with a simple condition stressing that, within a world populated with two kinds of agents, say rich people and poor people, an agent tolerates less than 50% of dissimilar neighbors. If this percent –this threshold– is exceeded, the agent randomly searches for a better place that suits this criterion best. After a few iterations, we assist in a full segregation of the two kinds of agents. Thereby, this segregation *emerges* only from the fact that an agent tolerates a certain number of poor (or rich) people. Furthermore, the expected result should be more regional than global clustering.

When studying diffusion, researchers usually embrace the problem either on the local, micro, level or on the global, macro, level. At the micro level, the interest is on variables that capture the domestic political, institutional, and economical contexts (Simmons and Elkins, 2004); and, at the global level, the emphasis is made on the different mechanisms of diffusion (Braun and Gilardi, 2006), the most-studied one being learning. However, as already mentioned, the link between the micro and macro level is still missing. Moreover, we have to rely on two different sets of theories – policy diffusion and computational agent-based model – to fully understand the building and use of a computational agent-based model.

Chapter 2 will provide the basic concepts needed to go through this particular study of policy diffusion. In this chapter, based on a well-accepted definition of policy diffusion, its main implication for our thesis will be emphasized, and the choice of the different internal and external factors, namely policy effectiveness, policy preference, the neighborhood, proximity, as well as institutional constraints and learning, competition, and emulation as mechanisms of diffusion will be theoretically based. Moreover, the theoretical evolution of the process of diffusion is synthesized.

In Chapter 3, the main theoretical features of our methodology will be tackled. In order to understand the necessity to study diffusion using computational agent-based modeling, we first explain the particular concept of complex adaptive systems. Secondly the implications of such system are developed; i.e., the concepts of complexity and emergence need in depth clarification, as well as its advantages and weaknesses. Furthermore, examples of computational agent-based models and their main conclusions will be briefly described. A special emphasis will be placed on Axelrod's model of dissemination of culture.

The next chapter, Chapter 4, corresponds to the description of our model of policy diffusion both at the theoretical and the implementation level. More precisely, how the different internal and external factors are operationalized and implemented in the code of the program will be developed. The same will be done for the different conditions necessary for a policy change to occur.

Chapter 5 will present the results from the various simulation runs and their theoretical implications for the diffusion research field. The results of this computational model are in line with the theoretical expectations and empirical evidence; that is, policies diffuse in the shape of an S-curve. As a consequence, the countries are partially converging and the world is clustering. Moreover, average effectiveness follows a J-curve. Furthermore, the test of the internal validity and its necessity for such computational agent-based models will be described in more detail. More precisely, we will highlight the importance of the random implementation of the agents for the development of the model.

A general conclusion will close our way through our first model that emphasizes the process of policy diffusion. Although the model offers some interesting proofs on the theory of policy diffusion, it also gives some fascinating conclusions on the results of the behaviors of countries when they change policy. In other words, not only are spatial and temporal clustering emphasized, but also the particular development of the average effectiveness in the model is highlighted. However, our model provides only a partial answer for the understanding of policy diffusion, which opens doors for future research.

# Chapter 2

# A conceptual framework of policy diffusion

## 2.1 Introduction

Many phenomena that are of some interest for social scientists involve diffusion; as, for example, welfare state policies or health care policies. Furthermore, diffusion is an interdisciplinary concept and can be based on such diverse scientific fields as economics, sociology, political science, physics or biology. As each research field has its own approach and terminology, it is difficult to have a comprehensive analytical framework. Except for a few studies (for example Braun and Gilardi, 2006), the concepts used in the study of diffusion are based mainly on their own terminology. One reason for this situation resides in historical development of this concept in different fields.

According to Rogers (2003), political science has a rather weak tradition in diffusion research. In the past two decades, though, diffusion has become a key topic and a growing research field in political science. Thanks to the fall of the Berlin Wall and the wave of democratization that spread through the former communist republics of Eastern Europe as well as the increase in the spread of liberal policies, best known as globalization, impressive theoretical and methodological developments have been made to better understand this concept (see e.g. Simmons and Elkins, 2004; Simmons, Dobbin and Garrett, 2006; Elkins, Guzman and Simmons, 2006; Swank, 2006; Lee and Strang, 2006; Meseguer, 2006*a*; Levi-Faur, 2005; Gilardi, Füglister and Luyet, 2009; Gilardi, 2010; Gilardi and Füglister, 2008). These works have emphasized the interdependent character of policy change. In other words they show that policies do diffuse, but still less is known and understood about the mechanisms that cause the governments' interactions, except for a few studies (see e.g. Volden, Ting and Carpenter, 2008; Shipan and Volden, 2008; Gilardi, Füglister and Luyet, 2009; Füglister, 2009).

The aim of this chapter is to make a contribution to the building of a comprehensive framework in the policy diffusion field. Thus, the idea is to be general enough that this model can be adapted to different social sciences and can attempt to link the most important concepts in order to build a model that can be used in several social fields.

To truly understand diffusion, we need to imagine a world without it. In an independent world, a country that is facing policy problems has, in order to resolve these problems, no other choice than to experiment with policy changes on its own. When a problem occurs, an independent country tries to improve its policy by using its own resources (social, economical and political). Facing the scarcity of resources, this country must often make difficult choices; for example, if it is facing scarcity of resources, a country may have to choose between investing in basic infrastructure, such as roads, vs. the social sector, such as education.

This way of finding new, sometimes original, solutions has its limits, at least economical ones. Indeed, the search for experiments that have been conducted to find a completely independent solution induce heavy costs. An important way to overcome these limitations resides in a government's capacity to look at what others do and to be influenced by them. One of the aims of this chapter is to put a theoretical emphasis on this capacity; that is, to focus on the different determinants of diffusion. For instance, a country facing increase in its hospital financing public expenditures may want to change this situation, as a great bulk of public expenditures concern hospital financing (OECD, 2006). One way to achieve this aim is to find new and innovative solutions elsewhere and to try them out. As a results, during the last decades, more and more countries have introduced patient classification financing systems; more precisely, some form of *Diagnosis Related Groups* (DRGs[1]) (Gilardi, Füglister and Luyet, 2009).

Since the industrial revolution, undoubtedly, the world has become more and more interdependent. Even if some eras are characterized by more-protectionist policies; as, for example, the period between the two World Wars – more specifically, the Great Depression in the 1930s – relationships between countries have never ceased. Williamson (1996) puts emphasis on 3 main periods. Until WWI, a period of globalization is observed that is characterized by trade openness. Between 1914 and 1950, the period is characterized by deglobalization; that is, the return of some protectionist policies, and after 1950 we assisted in the development of a new period of globalization, especially after the two 1970s-era oil crises. Since then, the world has become more and more interrelated. These crises have accelerated the development of a new period of globalization and of liberalization, which represents one of the main political and economical features of the last decades of the twentieth century. Simmons and Elkins (2004) have studied this phenomena, defined as the spread of neoliberal policies (and ideology) – that is, policies that seek the free movement of merchandise, capital and people and that are characterized by extremely strong interactions between countries – not by putting emphasis on domestic factors, but by highlighting the role of international politics. Their study stresses that the decisions made in one country influenced those made in other countries. In other words, the countries are interdependent in their policy decisions; that is, the study of diffusion implies that a policy change depends on what the others have done. More precisely, a country, before modifying its current policy, looks at the changes that have been introduced in its neighborhood. This conclusion is consistent with policy diffusion

[1]DRG is a system to classify patients according to their diagnostics (the same diagnostic should involve the same treatment) expected to have the same cost.

as defined in Section 2.3 and is used as the starting point of this thesis.

However, these interdependencies – the interrelations between the countries – are not fixed once and for all. They are subject to change, because the dynamics of the process depend on the links that exist between the internal factors and the different mechanisms of diffusion. Plus, it is necessary to consider the evolution of the process of diffusion through time and space (Simmons and Elkins, 2004; Elkink, 2009; Polillo and Guillén, 2005), as developed in Sections 2.3.2 and 2.3.3. Furthermore, the micro features will be explained in Section 2.4, whose evolution depends on the political, economical and/or cultural system of the country under study and that will be defined and described in Section 2.5.

Firstly, diffusion is put in a historical frame (Section 2.2) that is of great importance to highlight the evolution of this concept, not only at the theoretical level but also at the methodological level. Seeing diffusion in its historical perspective allows us to highlight the different faces of diffusion in general and in political science in particular. Secondly, a largely accepted definition of the concept of diffusion will be explained (Section 2.3). Furthermore, some results of the diffusion process need to be emphasized, because there may be some misunderstandings, as diffusion includes a wider range of phenomena. However, a theoretical model of policy diffusion that depends not only on internal factors (Section 2.4) of a country will be developped but, also on interdependencies that exist among countries, and expressed by three main mechanisms of diffusion (*horizontal* diffusion), namely learning, competition, emulation. A fourth mechanism will be explained, i.e. coercion. It has a particular place in the process of diffusion as we will see Section 2.5. Each of these mechanisms will be developed and put into perspective in their respective contexts in Section 2.6. A conclusion will sum up the main arguments in the broader view of the development of the process of diffusion.

## 2.2 A brief history of policy diffusion research

Diffusion as a research field is not really new. However, it is a disunited field and, as a result, several traditions were born and have produced their own theoretical approaches, not only in the natural sciences, but also in the social and human science fields. This dissimilarity can be seen as a great obstacle to overcome, because of the different terminology and concepts used. The following lines will highlight some of the main traditions that have led to the development of diffusion studies in political science.

### 2.2.1 The foundations of diffusion studies

In the social science, the tradition of diffusion research started with some basic questioning of sociologists.

- *The early sociologists*: The foundations of research on diffusion can be traced back to the end of the $19^{st}$ century and the early $20^{th}$ century, when early sociologists, such as Simmel or Tarbe, started to emphasize the nature of the social interactions and, thus, the individual behavioral changes.

  Tarbe was interested in the diffusion of innovations; more precisely, in the reasons why some innovations will spread while some others will remain unknown. Tarbe's view was very accurate. Even though the words used were different, the embedded concepts are the same as the ones still investigated today. His reflections on the nature of the spread of ideas have led him to the discovery of some fundamental outcomes of diffusion processes. For instance, he had already emphasized the main outcomes of diffusion, such as the S-shaped curve of the process, imitation as a crucial mechanism or the foremost influence of networks (Rogers, 2003; Greenhalgh et al., 2005).

  In the early $20^{th}$ century, another influential sociologist, George Simmel, published his own reflections on social interactions. In the Simmelian tradition, social interactions exist not between given and fixed agents, but between evolving agents. Consequently, theses interactions transform the agents in time and space. Therefore, social interactions exist in a spatiotemporal space. As Cederman (2005, 866) stressed it, the social reality as envisaged by Simmel corresponds to a continuous process of interacting agents, resulting in the emergence of social forms in a "spatiotemporal continuum." Hence, Simmel puts a very interesting insight on the social relations that allowed the rise of basic concepts used in the study of diffusion, in particular on the significance of networks as a tool that allows one to study social interactions and, more precisely, the diffusion of innovations (Rogers, 2003). At this point, they lack the methodological tools to analyze their hypotheses, although they have found the key results of diffusion processes.

The Simmelian approach calls for a final remark. The idea of social reality as a result of spatiotemporal of social interactions is central not only to the theoretical foundation of diffusion as a research field, but also for the development of agent-based models, because Simmel saw the rise of social products as an emergent phenomenon resulting from individual interactions (Cederman, 2005). This is central for the understanding of the use of agent-based models, as will be explained in Chapter 3.

- *The rural sociologists*: As Rogers (2003) noticed, about 40 years later than these first reflections on the spread of new ideas and on social interactions, the rural sociologists, whose aim was to study rural societies, were the first scholars to study diffusion *per se*. In the diffusion context, their field of research concentrated on how and why innovations are spread among farmers.

  In 1943, Ryan and Gross published their seminal work on the diffusion of hybrid seed corn. With this research, they could empirically show the results expressed theoretically by the early sociologists. Based on qualitative data from survey interviews of farmers in a chosen farming region of Iowa, Professor Bryce Ryan, with the help of his assistant Neal Gross, tried to highlight how new corn seeds are adopted in a typical community and why these new, more productive, seeds have taken time to be adopted (up to twenty years later for the late adopters). Their main findings were that the process of diffusion takes time because the relative mistrust of the potential adopters, even if the innovation allows great success and follows a S-shaped curve. Ryan and Gross emphasized the ideal of the different types of agents (innovators, late adopters, etc.) and their sociocultural characteristics – among others, the innovator is better educated and richer – and the impact of networks; or, in other words, interpersonal relations (Rogers, 2003; Greenhalgh et al., 2005).

- *The medical sociologists*: In the late 1950s at Columbia University's Bureau of Applied Social Research, and in a totally independent manner – without following a diffusion process; Columbia's medical sociologists happened to have the same research idea on the diffusion of new prescribed drugs – the primary aim of the medical sociologists' studies was to highlight doctors' adoption of antibiotics; and, at the end, when comparing their results to those of Ryan and Gross, they noticed that their results were the same. More precisely, their results were that diffusion follows an S-shaped curve, networks – interpersonal relations – are important and a better education and wealth are the main characteristics of the innovator (Greenhalgh et al., 2005, 54).

  The classic study of this tradition was conducted at Columbia University, with a team of sociologists under the lead of John Coleman, Elihu Katz and Herbert Menzel. This team interviewed 125 physicians on their use of tetracycline, an antibiotic developed by Pfizer, the pharmaceutical firm that financed the research. Moreover, these doctors had to designate other practitioners as members of their network that were interviewed too. Hence, they already put some emphasis on the influence of networks. The drugstore prescriptions gave the researchers "an objective measure of each doctor's time of adoption" of tetracycline (Rogers, 2003, 66). As already mentioned, and at their own surprise (Rogers, 2003, 66), they came to the same conclusion as Ryan and Gross, notwithstanding the fact that they have faced the same "social, historical and ideological context" (Greenhalgh, Robert, Bate, Macfarlane and Kyriakidou, 2005, 54).

- *The limitations of these early studies*: The theory and methodology of these early studies, even if they found the future of the discipline, can be questioned. Some of their limitations still are noteworthy:

  1. The S-curve describes the cumulative proportion of adopters, as explained Section 2.3. It is a purely descriptive tool and gives no insight on how and why the adoption of an innovation occurs. It has no predictive power. Nevertheless, it still is of great help for our purpose since its main interest concerns the positions of the different agents as the process of diffusion unfolds (Berry and Berry, 2006, 229).

  2. These seminal works on diffusion (rural and medical sociology) take place in a particular political, economical, and social context, in the era of the Glorious Thirties after the Second World War, where the benefits of innovation were not questioned.

     Ryan and Gross's research took place in an era when the Iowa Agricultural Extension Service and seed corn companies pushed for adoption of new agricultural technologies with better returns. This diffusionist tradition increased after World War II, during the era called the Glorious Thirties 'that celebrated innovation and change for its own sake" (Greenhalgh et al., 2005, 58) and was characterized by extensive economical and demographic growth. However, the diffusion paradigms developed at this time could not be applied in all countries, but only in developed ones, since the developing countries were confronted by other issues (a rural society, poverty, lack of infrastructure, etc.). This leads us to the third main criticism, the pro-innovation bias.

  3. The pro-innovation bias still is a major criticism of and concern for scholars who study diffusion. This bias comes from the fact that innovations

that spread are easier to study than those that are rejected. Moreover, successful policies are more likely to be copied or imitated (Simmons and Elkins, 2004; Volden, 2006; Shipan and Volden, 2006). However, to challenge diffusion is different from questioning the origin of the innovations and their first adopters. Why some innovations spread out while others do not is still an open question. Nevertheless, this research field needs something that spreads in order to have a subject to study.

4. At the methodological level, the Galton's problem was (and still partly is) pertinent. This methodological problem was highlighted by Sir Francis Galton at the end of the $19^{st}$ century. In brief, this problem appears when units of analysis are taken as independent even when they are not. In other words, the different countries in our case are treated as independent even if they were not, and, as a result, the nonlinearity that exists among them is not taken into account methodologically. Thus, the different relations were considered as linear. Nowadays, computational agent-based modeling can be used to overcome these problems, as will be explained in the chapter 3.

This sociological tradition has posed the foundations for the study diffusion in political science. Even if McVoy (1940) had already started to study the different patterns of policy diffusion in the United States, his work was considered sociological; therefore we must turn to the late 1960s and early 1970s to see the start of diffusion research in the specific field of political science.

### 2.2.2 Diffusion in political science

In the specific field of political science, the tradition of diffusion research is rather new compared to sociology. Inspired by other social sciences, mainly sociology, political science has now caught up most of its theoretical and methodological delay.

**Diffusion in the political science at the theoretical level**

The formative works of Walker (1969) and Gray (1973) found the conceptual frameworks for later research in the political science field. As already mentioned, these studies have searched for inspiration in the sociologists' works on diffusion of innovation, notably the Ryan and Gross study and the Coleman et al. study.

Walker (1969) has highlighted the theoretical underpinnings in his study of diffusion research. Moreover, his theoretical essay could be seen as a response to the limitations of the early studies, as he has tried to focus on how and why the adoption of an innovation occurs. In so doing, and contrary to the early diffusionists who concentrated their study on the interpersonal level, Walker has developed his thoughts at the state level. He has made a fundamental theoretical point, since he defined diffusion as the interplay between internal and external factors. Interestingly, he found the same relevant results on the characteristics an innovator must have as did Ryan and Gross; i.e., "the larger, wealthier, more industrialized states adopt new programs somewhat more rapidly than their smaller, less well-developed neighbors" (Walker, 1969, 884). He also has identified the importance of horizontal communication channels for the adoption of novelty, especially the interactions in a policy network (Füglister, 2009). As a result of the process of diffusion between American states, Walker could highlight the existence of regional clustering.

During the 1980s, the diffusion paradigm was put aside for a time. There are no major pieces of literature during this decade. One has to wait the beginning of the 1990s and the path-breaking article of Berry and Berry (1990), which uses event-history analysis to study the diffusion of state lotteries. However, since the mid-1990s, the theoretical debate has increased and produced an ever-growing literature.

More precisely, the diffusion paradigm has regained interest among scholars because of the wave of democratization that has characterized the period after the fall of the Berlin Wall, notably in the former eastern European communist states and of the globalization of liberalism – the spread of liberal policies – that primarily shapes world economic relations (Simmons, Dobbin and Garrett, 2008). A great bulk of research has put emphasis on the different mechanisms of diffusion, especially learning, which is one of the most-studied mechanisms (see e.g. Meseguer, 2003, 2004, 2005; Volden, Ting and Carpenter, 2008; Gilardi, Füglister and Luyet, 2009; Gilardi, 2010). Recently, a step further has been made with studies that try to disentangle the different mechanisms of diffusion (see e.g. Volden, Ting and Carpenter, 2008; Gilardi, 2010; Füglister, 2009).

#### Diffusion in political science at the methodological level

The early *diffusionists* used structured questionnaires for interviewing individuals engaged in the process of change. When it comes to studying diffusion at the state level, a methodological tool to study the spread of policy did not exist. Therefore, Walker (1969) tried to overcome this lack by developing an innovation score. To do so, he analyzed eighty-eight different pieces of legislation in different sectors, such as welfare, health, and administrative organization. For each piece of legislation, he found the date of adoption, and then calculated the difference between the first and last introduction. Then, the score of each program corresponds to 'the percentage of time which elapsed between the first adoption and its own acceptance of the program' (Walker, 1969, 882), and the innovation score for each state is calculated as 1 minus the average of the sum of the scores on all pieces of legislation. By providing this innovation score, Walker made the first attempt to answer the question of how policies diffuse. One of the significant conclusions of his work was that the likelihood of a state introducing a new policy is higher if its neighbors have already introduced the policy (Walker, 1969, 897), in other words, he put emphasis on bandwagon pressures that will be discussed more in depth in Section 3.4.6.

His attempt to show why policies diffuse was done by calculating correlations between the innovation score and several identified determinants, such as socioeconomic or political factors. The calculus of correlations shows the degree of interdependence that exists between the innovation score and the identified patterns of diffusion. In other words, this methodology shows if the adoption of an innovation is dependent upon income level, for example. His results have already been summarized above.

Gray (1973), remarking that the index developed by Walker did not separate between independent and interdependent adoption and in order to introduce some more methodological rigor in the study of diffusion, used a linear model to answer the same questions; i.e., how does a policy innovation spread – the shape of the curve representing the cumulative proportion of adopters – and why is a state an early adopter.

Moreover, she used Spearman rank-ordered correlations[2] to estimate the strength of the relations between the dates of adoption and the innovative laws under study. As a result of her study, she found that diffusion tends to follow an S-shaped curve. Moreover, these two authors highlighted the characteristics of the innovators – wealthier and more industrialized.

A methodological breakthrough has been made by Collier and Messick (1975), who put some emphasis on the "Galton's problem," briefly explained in Section 2.2.1. In this seminal work which, surprisingly, was ignored at the time of its publication, they showed how and why Social Security has diffused across the United States by regressing the year of the first adoption of Social Security against the percentage of workforce. The result is that the higher the percentage of workforce, the later the introduction of Social Security.

In the early 1990s, a second important improvement, at the methodological point, was made with the use of event history analysis (Berry and Berry, 1990, 2006) in the study of policy diffusion, which has allowed for great progress in the quality (and the quantity!) of research as well as ease of comparability between the different studies (Karch, 2007). The basic idea is to estimate the odds of the occurrence of an event. For instance, using such a method to study diffusion, we can attempt to find the chance that a country will adopt a policy change if others have already done so (Henisz, Zelner and F., 2005; Shipan and Volden, 2006; Gilardi, Füglister

[2]This coefficient shows how well two variables are related.

and Luyet, 2009). With event-history methods, we try to estimate the timing of the first adoption (Boehmke and Witmer, 2004).

The first studies of diffusion in federal states have questioned the adoption of an innovation (Walker, 1969; Gray, 1973). Once these questions were answered, and with the development of the use of event history analysis, the greatest bulk of policy diffusion studies have questioned the process of diffusion itself. In other words, the research examined *how* a policy diffuses.

Nowadays, with the development of new methodologies or the deepening of existing ones, as, for example, computational agent-based modeling (Elkink, 2009; Macy and Willer, 2002) or the dyadic event-history approach (Volden, 2006; Gilardi and Füglister, 2008) or even the mixing of existing ones, as Gilardi (2010) who explains the diffusion of unemployment benefit retrenchment and its interpretations by the different actors using dyadic approach and multi-level analysis, the main purpose of recent research is to disentangle the different mechanisms of diffusion, since they are widely accepted and documented, as we will explain in Section 2.5.

The country, as a nation state, does not decide political changes; rather, its government does. It is clear that the latter makes the critical political decisions; as, for example, the introduction of a new policy. This change involves a lot of different actors; for example, governments, lobbies, citizens, bureaucrats and so on, that play the political game. For our purpose, four main internal factors are defined and their role in the diffusion process explained. Even within a country – between states or cantons – processes of diffusion can play a central role in the policy change (see e.g. Volden, 2006; Shipan and Volden, 2006; Gilardi and Füglister, 2008). Therefore, to study diffusion in an international context, it is crucial to fix the level of analysis once and for all. In this work, we concentrate on the study of diffusion between countries. Of course, one should be aware of the different forms of government and the number of individuals; but, as is common in the diffusion literature, the most commonly used assumption is that a country changes its policy for a more effective one, no matter the consideration for this change – electoral or ideological. Thus, in the context of this study, the country will be the level of analysis; that is, the actors, in my model. In other words, *country* becomes here a synonym of *government*.

## 2.3 A definition of the process of diffusion

Now that we have gone through the historical, theoretical and methodological development of the concept of diffusion, we will define what we mean by diffusion and explain the main implications of the definition we choose.

### 2.3.1 The definition of diffusion

We can define international policy diffusion with the following largely accepted definition:

> "International policy diffusion occurs when government policy decisions in a given country are systematically conditioned by prior policy choices made in other countries" (Simmons, Dobbin and Garrett, 2006, 787).

Before explaining this definition in depth, a brief remark needs to be made. As we have already noted, prior to the existence of a process of diffusion a policy must exist; then policy diffusion processes can occur. Therefore, the causes of first adoption will not be studied here.

According to this definition, diffusion implies interdependencies between agents, because a country looks at what the others do before deciding whether or not it should change its policy. Yet, two countries may introduce the same reform without looking at each other, only because they are facing the same political problems. In some studies this is expressed as the *null* hypothesis; i.e., the hypothesis that stresses the independence between the cases (see e.g. Simmons and Elkins, 2004; Simmons, Dobbin and Garrett, 2006; Elkins, Guzman and Simmons, 2006). However, it is best expressed by the notion of spurious diffusion, which "captures the fact that a pattern may look like diffusion even though it is not driven by diffusion" (Braun and Gilardi, 2006, 299). In such cases, the change is independent. More precisely, the policy change is driven only by internal factors (Berry and Berry, 1990, 2006). Thus, this kind of change is often known as the "umbrella causation" (Hennessy, 2009), that is, the fact that people open their umbrellas independently during a rainstorm, as each of the people has a different threshold towards the rain. Of course, some people do respond independently to some external conditions – may they be structural or conjonctural (here the rain), but it is necessary to consider that more easily influenced people may act interdependently. For example, they may ask a neighbor about the different advantages of the umbrella and then buy one for their own use. Thus, we cannot put aside the fact that some followers open their umbrella only because the majority has done so. If the first part corresponds to spurious diffusion, the followers are engaged in a diffusion process (according to different mechanisms of diffusion that will be developed in Section 2.5) (Levi-Faur, 2005, 22). The problem with differentiating between spurious diffusion and diffusion is linked at the methodological level with the difficulty of treating cases in a interdependent manner, best known under the above mentioned label of "Galton's problem," or in other words the problem of disentangling independent cases from interdependent cases.

Since diffusion implies interdependences between countries, and as the process unfolds, it can lead to several equilibriums. One of the most studied equilibriums is convergence (Braun and Gilardi, 2006); that is, all potential adopters have introduced the same policy. Interestingly, the computational model developed by Axelrod (1997*b*)[3] to study the dissemination of culture leads to the conclusion that divergence still exists in a convergent world.

Therefore, according to the chosen definition, policy diffusion is more than simply convergence, as the different interactions between the countries create the process of diffusion that leads to several different results, including convergence. This result can, thus, be defined as "the growing similarity of policies over time" (Holzinger and Knill, 2005, 776). Policy convergence observed between countries is in part explained

[3]This model will be more deeply explained Section 3.4.7

by the same mechanisms of diffusion described in Section 2.5. When convergence occurs, policies as well as countries never become totally identical; that is even if countries converge at the micro level, divergence still remains at the global level (see e.g. Axelrod, 1997*b*).

In the next subsection, we will deepen our exploration of this definition by explaining the temporal side of the above definition and its implication for the study of policy diffusion.

### 2.3.2 Diffusion and the temporality of the process

The process of diffusion is characterized by a strong temporal dimension. It is a backward-looking process, because countries look at what have been done in other countries before deciding whether or not to introduce a change in the policy. In other words, countries at time $t$ look at what others have done at time $t - 1$.

Therefore, the process of diffusion is path dependent, meaning that it corresponds to a "temporal process in which early choices create self-reinforcing effects that are inherently difficult to reverse" (Hacker, 2004, 697). This means that time has an influence on the evolution of the process, with the consequence that the percentage of adopters tends to follow an S-shaped curve (Greenhalgh et al., 2005; Berry and Berry, 2006; Rogers, 2003; Gray, 1973). For instance, the introduction of a prospective payment system for hospital financing in OECD countries is influenced by the prior experience of such system in other countries, and resulting in a S-curve (Gilardi, Füglister and Luyet, 2009).

Figure 2.1 shows us that, at the beginning of the process, there are only a few adopters and so the curve stays near 0, until a point where the number of adopters is sufficiently high, so that the slope of the curve increases sharply. In other words, at this point, the number of adopters is sufficiently high to induce countries that hesitated or were not really interested changing their policy to start taking into account the eventuality of changing.

This temporal heterogeneity; that is, the fact that not all countries have the same

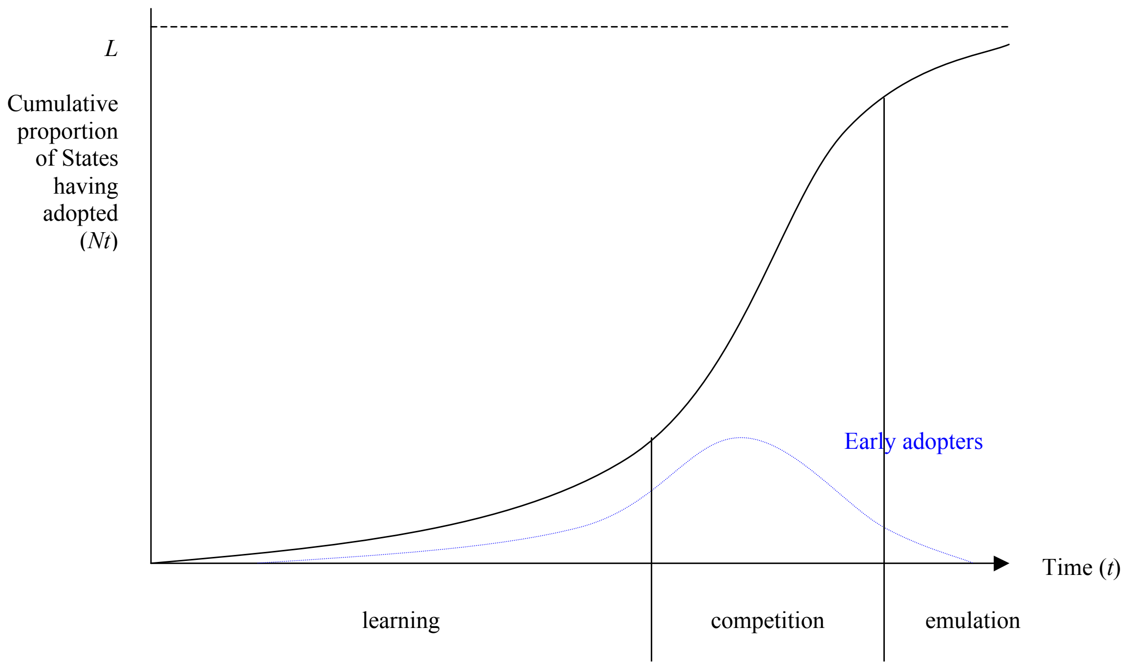



Figure 2.1: The cumulative proportion of adopters. adapted from Berry and Berry (2006, 227)

horizon of change has been highlighted by Strang and Tuma (1993). In their study they reanalyze the data collected by Coleman, Katz, and Menzel on medical innovation (the first prescription of tetracycline in four US cities[4]). Using an event history model, Strang and Tuma (1993) have emphasized the temporality of diffusion processes. More precisely, they show that new adopters are more prone to publicize innovation and that the adoption of the innovation is influenced by prior events; more precisely, prior adoption.

Moreover, each mechanism of diffusion has its own temporality. For instance, imitation has a shorter life than the other mechanisms (Shipan and Volden, 2008). More importantly, the process of diffusion as a whole (the cumulative proportion of adopters) integrates the fact that each mechanism has its time of play. Assume that the new policy is more effective. At first, the early adopters[5] learn from each other. As other countries realize that the introduction of a new policy allows early adopters to be more effective, they start a competition because they do not want

[4]The results of the Coleman et al study has been briefly explained Section 2.2

[5]Because of their characteristics, there are only few innovators and they are more prone to learn from each others.

to lose their market share (see e.g. Dobbin, Simmons and Garrett, 2007; Simmons and Elkins, 2004). When enough countries have changed their policy, this latter becomes a common norm and emulation is at work. Therefore, the global process of diffusion corresponds to the conjunction of each mechanism at play and the potential numbers of adopters.

What is really important is that the duration of each mechanism is different. Shipan and Volden (2008) show that imitation for example as a shorter *existence*. When studying the process as a whole, we can see it as the addition of the different durations of the effects of each mechanism. This can be shown on the S-shaped curve of policy diffusion, with a longer path at the beginning of the process (learning) than in the end (imitation).

It may now be clear that the process of diffusion occurs through time. However, as it involves several countries, it therefore takes place in a defined space.

### 2.3.3 Diffusion and the spatiality of the process

The process of diffusion occurs through a defined space between neighbors. This neighborhood need not be only geographical/physical , but can also be cultural (Meseguer, 2005; Levi-Faur, 2005), economical, specifically trade (Martin, 2009), ideology (Grossback, Nicholson-Crotty and Peterson, 2004) or demographical (Volden, 2006).

As geographical boundaries may sometimes be difficult to overcome (natural barriers), the proximity that exists among countries involved in a process of diffusion can be best expressed by the cultural and/or the economical "borders." In the analysis of diffusion, the proximity that exists between the agents involved in the process must be defined in a larger way. Too often, the neighborhood is operationalized as purely geographic. Thus, the proximity that defines the neighborhood in the context of diffusion must contain other dimensions, such as social, political and economical ones (see e.g. Boschma, 2005; Beck, Gleditsch and Beardsley, 2006).

As we have just seen, the rate of adopters increases sharply, up to the point where almost all potential adopters have adopted the policy change. In Figure 2.1, in a specific point of time can a drastic change in the slope of the cumulative curve be observed. This change is driven by bandwagon pressure[6] (Abrahamson and Rosenkopf, 1993, 1997; Rosenkopf and Abrahamson, 1999), which can be defined as follows: the more countries that have changed their policy, the higher the incentive (the pressure) to change. The result of such pressures suggests that spatial dimensions do characterize policy diffusion, as shown by the clustering in Figure 2.2.

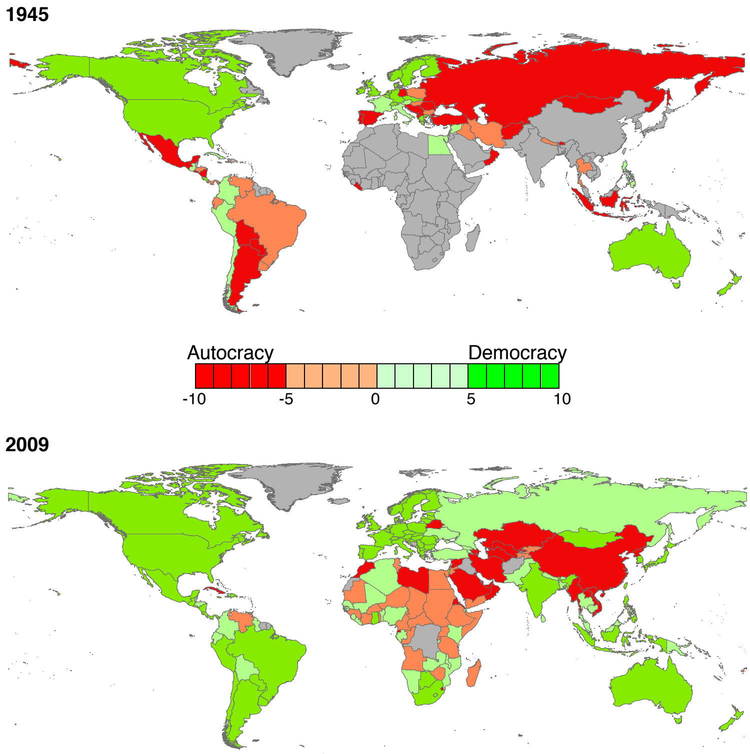



Figure 2.2: Autocracies and democracies in the World, 1945 and 2009 (adapted from Gleditsch and Ward, 2006, 915)

In their study on diffusion of democracies, Gleditsch and Ward (2006) highlight the impact of neighbors on the adoption of democracy. More precisely, they stress out that the more neighbors that are democracies, the higher the chance to become one. They similarly emphasize the spatial clustering related to the diffusion of democracies. This notion of clustering finds its basis in the interactions between the different neighbors. Thereby, the difficulty here is to define the neighbors because,

[6]Bandwagon pressures will be explained more in details in the next chapter, when the conceptual framework of agent-based models will be tackled.

once again, several dimensions – which are interrelated – characterize the concept of neighborhood.

As already explained above, a country can be considered as a neighbor, even if it has no geographical borders and the neighborhood can be based on cultural, economical, and/or political similarities (Boschma, 2005; Amin and Wilkinson, 1999). For example, despite the difference in their political systems and the fact that they do not share any borders, the United States of America and Great Britain can be considered neighbors as they share, for example, the same language and the same economical "ideology." On the contrary, even though they share a common border, North and South Korea cannot be considered as neighbors, in the *diffusionist* sense. The fact that diffusion leads to the existence of *convergence in divergence*[7] is just another way of expressing the development of clusters as a result of the diffusion process.

The way a cluster develops depends not only on the mechanisms of diffusion at play, but also on their influence on the different determinants of change, labeled as the conditionality of the process of diffusion. This is the subject of the next section.

### 2.3.4 Diffusion and the conditionality of the process

Gray (1973) and Walker (1969) already emphasized the fact that some internal factors play a key role in the process of change, because they are influenced by what happened elsewhere. However they did not highlight the fact that policymakers might be dissimilarly influenced.

This idea that policymakers react in a different manner to the same influence of the neighborhood, as expressed by the different mechanisms of diffusion, is now known under the label "conditional diffusion" (Martin, 2009; Shipan and Volden, 2008), which can be defined as follow:

> "Units[8] $i$ and $j$ may share the same degree of interdependence as units

[7] See Axelrod (1997*b*) and the explanation of his model Section 3.4.7.

[8] Countries in our case

> $h$ and $j$.Yet, $i$ and $h$ may be differently affected by $j$'s policies because the different circumstances of $h$ and $j$ vary" (Martin, 2009, 2).

In other words, ensuing from the spatial interdependence as explained in Section 2.3.3, the influence of a mechanism of diffusion, say learning, can have different effect on $i$ and $h$, because the countries are intrinsically different–their internal factors, that is at the political, economical, cultural, and/or institutional levels. Therefore, the conditional nature of policy diffusion highlights the sensitivity of the different countries to the mechanisms of diffusion. In other words, when facing the same problem and making the same decision, two different countries may have different results.

We can imagine that less powerful countries in economical term and/or political terms are less susceptible to learning, but more susceptible to emulating or being coerced[9]. For example, in the case of the introduction of antismoking policies in the different states of the USA, Shipan and Volden (2008) use the population of a state as a proxy of its strength. In other words, the more populated a state is, the less susceptible to emulation and the more susceptible to learning it is.

Thus, as a consequence of this concept of conditional diffusion, one of the biggest problems faced by diffusion scholars is to disentangle the effect of the different mechanisms of diffusion, as domestic conditions are affected by external decisions. Martin (2009) shows that the ideology matters in the case of the diffusion of tobacco tax policies. In other words, the more liberal a government is, the more influenced it is by tax policies developed in neighboring states. Furthermore, Gilardi (2010) shows that policy makers are differentially influenced by their ideology. Right-leaning governments tend to be vote seekers while leftist governments typically seek policy effectiveness. These studies highlight the interplay that exists between domestic factors and the external influence introduced in the country by the different mechanisms of diffusion and, consequently, the conditionality of the process of diffusion. In sum, conditionality means that all policymakers do not react in the same manner

[9]For a deeper explanation of these mechanisms, see Section 2.5

to the pressures exerted by the neighbors (as expressed by the different mechanisms of diffusion) (Gilardi, 2010). Following Radaelli (2005), it seems that the divergence that persist between the countries is explained by the internal factors, that is the political environment of a country characterized with the types of administration, government (and the strength of the government), the political *game* and the different lobbies. Moreover, Botcheva, Martin. and Martin (2001) highlight the fact that divergence is caused by the presence of heterogeneity in domestic polices.
In other words, the remaining divergence results from the *resistance* of the internal factors, i.e., external pressures do not have the same impact on the internal determinants of change, as we will explain in the next section.

## 2.4 The internal determinants of change

Let us recall that, from Section 2.3.1, each country has its own threshold for entering the process of change and, from Section 2.3.4, each country has a different domestic *sensitivity* toward the influence of external factors, with the consequence that the process of policy change can be either *slowed down* or *speeded up*.
In this section, we will explore the country and describe the main political, social, and economical factors that play a role in the process of policy change. These factors are the ideology; that is, the preference for a policy, the political insecurity; i.e., the fear of losing power, the effectiveness of the policy, and the institutional constraints, approximated by the veto players.

### 2.4.1 Ideology

The preference for a policy corresponds to the ideology. For instance, leftist parties are supposed to introduce policies that are more state oriented and rightist parties are in favor of more market-oriented policies. In other words, the different governments are ideologically oriented. For example, Gilardi (2010) shows that, in the context of the diffusion of unemployment benefits retrenchment, rightist governments are more prone to cut unemployment benefits, even if it is a bad solution. In other words, as argued by Volden, Ting and Carpenter (2008), ideological position may be an important factor for driving diffusion.

The main assumption behind this factor is that each government not only knows its ideal position on a left/right continuum, but also the position of the different policies on the same axis. Consequently, a government may want to introduce the policy with the closest ideal point to its ideal ideological position (Grossback, Nicholson-Crotty and Peterson, 2004).

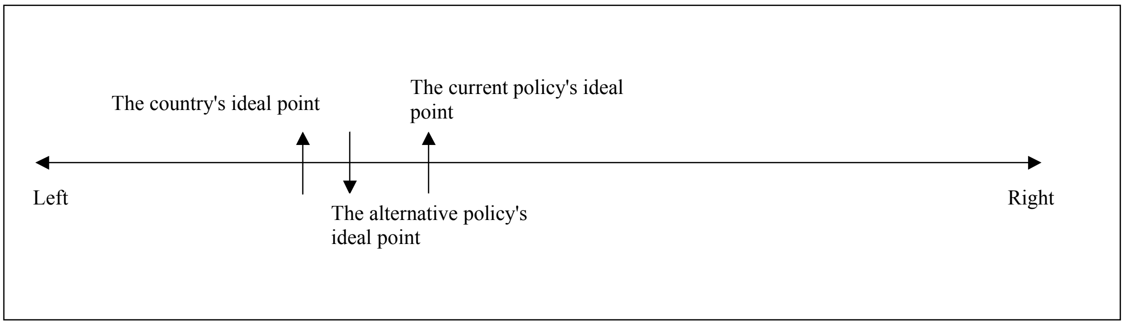



Figure 2.3: The ideological dimension

Figure 2.3 schematically expresses the idea of this assumption. In this figure, the ideological threshold of the current policy is far from the one of the government. Therefore, following Grossback, Nicholson-Crotty and Peterson (2004), a country may want to change its policy to get closer to its ideal ideological point. And the way to overcome this dissatisfaction is to get involved in a process of change.

As it is difficult to know where to place this ideological point of a policy on the left/right axis, they assume that a country, let us call it Country A, knows its best placement on this continuum by looking at its neighbors (countries B, C, D and so on). The assumption is that, when a neighbor changes its policy, it gets closer to that placement, so that Country A can "infer where the policy lies on the liberal/conservative[10] issue space" (Grossback, Nicholson-Crotty and Peterson, 2004, 525). Suppose that Figure 2.3 represents the situation of Country A; it would want to change to a more leftist policy, which could be more in line with its current

[10]It corresponds to the US equivalent of the left / right axis on the figure 2.3.

preference. Hence, when ideologically close neighbors have changed their policy, they create an incentive for Country A to equally change.

Nevertheless, this ideological point is not fixed once and for all. It is subject to change when elections and voting are taking place, which is the subject of the next section.

### 2.4.2 Political insecurity

The level of competition among the elites is one of the possible dimensions that characterize democracy (Elkink, 2009, 23). This competition leads the winner(s) to power; that is, the possibility to govern the country and thus to *impose* one's ideas or ideology. As a consequence of this competition, to keep the power has a cost because the electors need to be convinced (Besley and Case, 1995). If not, there is a high risk to loose power.

Furthermore, in democracy, when elections are near, in order to keep the reins of power, existing governments are more prone to accept policy change supported by the majority of the population even if it is not in line with the dominant ideology; that is, in their search for votes, parties adapt their electoral platform in order to satisfy most of the citizens (Kollman, Miller and Page, 1998, 1992).

The political insecurity factor, then, competes with ideology for the introduction of a policy change. As noticed by Braun and Gilardi (2006), the fact that the policy-seeking and vote-seeking factions of governing elites compete may induce a bigger weight to voting when elections are near and a bigger weight to policy when they are further away. In other words, political insecurity increases when elections are approaching and the government in place may want to flatter voters by introducing a policy that is ideologically close to their preferences. This factor evolves following waves, and these waves correspond to the time between elections, as shown in Figure 2.4.

Way (2005), in the case of the diffusion of financial market regulation, puts emphasis on the fact that governments that fear to loose power are more prone to reform

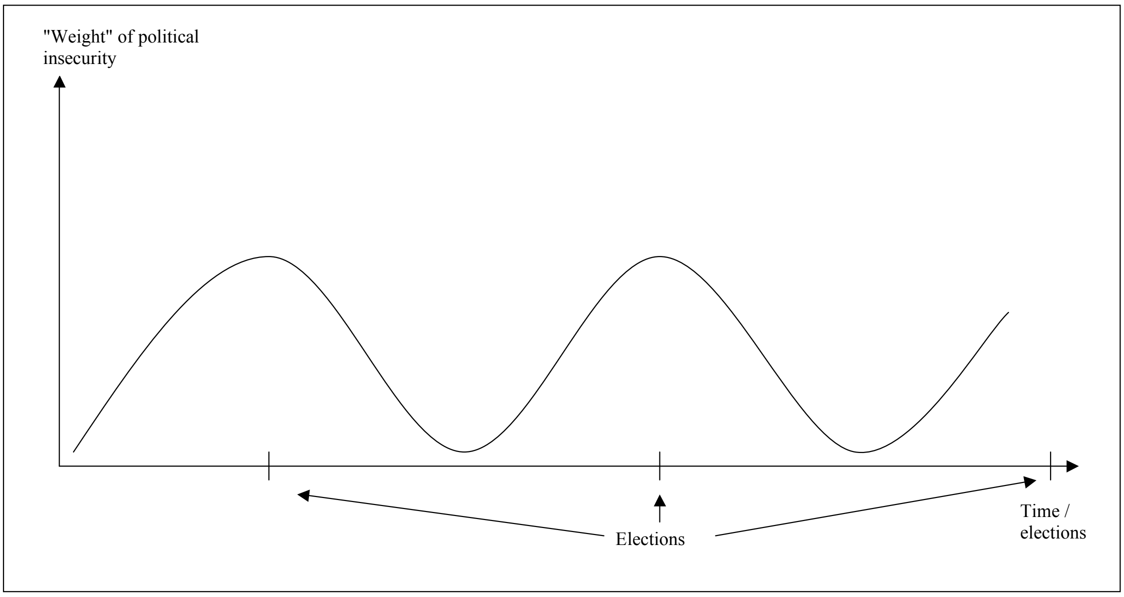



Figure 2.4: The ideological dimension

their financial market. In the same vein, Gilardi (2010) stresses, in the context of the diffusion of unemployment benefits retrenchment, that the government in place is more focused on the consequences of the policy change if retrenchments do not convince the electorate and, thus, would decrease its chance for reelection. Thus the fear of losing power may be an important internal driver for policy change. As elections do not occur every year, there are of course other internal factors that play a role all along the process of diffusion.

### 2.4.3 Policy effectiveness

A policy is designed in order to attain a certain objective and, thus, a policy is effective if it achieves this desired outcome (Braun and Gilardi, 2006). Welfare states' policies provide good examples in order to highlight not only the policy effectiveness, but also the change in the effectiveness that calls for a policy change.

For instance, unemployment policies aim at providing a replacement rate in case of job loss and at helping unemployed workers find a new job. Such a policy is deemed effective if the unemployment rate decreases after its introduction. In a period of economic crisis characterized by increasing unemployment rates, the current policy may be unable to face these new challenges

Another example we can relate to is the aging policies that most developed countries introduced after World War II and during the baby boom that aimed at replacing part of one's wage after one's active life, in an era characterized by economic and demographic growth, as well as a shorter life expectancy and the expansion of welfare state policies.

However, these policies are no more effective in today's era: they are challenged by what have been called "post-industrial pressures" (Pierson, 2001). More precisely, the welfare state's expansion stopped about thirty years ago. Three causes for this change have been highlighted: globalization, that is trade openness; deindustrialization, i.e. the shift toward a service economy; and sociostructural change, with the aging of society as its most remarkable change (Häusermann and Palier, 2008).

Consequently, most governments have to find a more effective policy in order to face these new challenges. Thus, in the case of the aging policies, the creation of an individual savings account has been a widely accepted tool as a solution to solve these problems. As shown by Brooks (2007, 2005), reforms of pension systems have spread among countries.

Hence, when a country changes its current policy, it usually introduces a policy that is supposed to be more effective (see e.g. Volden, 2006; Dobbin, Simmons and Garrett, 2007; Shipan and Volden, 2008; Gilardi, 2010). These researches show the importance of policy effectiveness in the context of policy diffusion, because they emphasize the fact that countries seek the most effective policy, according to the pieces of information they have.

### 2.4.4 Institutional constraints

In the context of policy change, institutional constraints may be a force in favor of or against the introduction of an alternative policy. By the end of the 1960s, the veto players were identified as a critical determinant for policy change. In his seminal work, Walker (1969) underlined the impact of groups with "veto power" on the speed of adoption of an innovation.

The expression of *institutional constraints* is here approximated with the notion of veto players. Therefore, the idea of institutional constraints is strongly linked with the veto players' approach (Bonoli, 2001). The idea is as follows: if actors have some veto power, they will use it in the political context to block decisions that go against their preferences (Ganghof, 2003). In other words, political actors, due to a different preference or ideology toward a policy, will introduce some rigidities (or constraints) into the process of change. Therefore, the veto players take into account the environment differently and, thus, shape the institutional system differently (Bonoli, 2003). More precisely, the more veto players, the more institutional constraints are implemented and the lower the probability for a policy change to be voted into law.

The assumed role of veto players is consistent with empirical research on the role and importance of a veto player as, for example, in the context of capital control policies (Kastener and Rector, 2003) or in the context of the spread of income tax policies (Hallerberg and Basinger, 1998).

These two studies stress the influence of external factors on veto players and the role of veto players in policy change. Further, they highlight the fact that the different political systems react differently to change, according to the number and importance of veto players. In other words, their studies show that the more veto players, the lower the probability for a policy change to be voted into law (or the greater the time it takes for a policy change to be introduced). Hence, the relative strength of veto players can be an approximation of the sensitivity a country has toward the different mechanisms of diffusion, as explained in Section 2.3.4.

In sum, policy change can be based on internal factors. To simplify, we can imagine that a country calculates an internal policy change score, such as the innovative one developed by Walker (1969). In other words, a country calculates a weighted average sum that gives its incentive level for starting a change. This corresponds more or less to the definition of a threshold, as developed by Granovetter (1978). However, a threshold model works because of the interactions – the interdependence that exists between the agents: Suppose 100 people, each with a different rioting *index* (which predicts each actor's threshold for entering a riot) have thresholds ranging from 0 to 99. For instance, the actor with the threshold of 0 triggers the riot. Then, the actor with the threshold of 1 engages in the riot, and that activates the third actor and so on, up to the point where every person is engaged in the rioting. In such model, each agent has a different threshold.

In the context of policy diffusion, the different internal factors are combined to estimate the ideal point at which a country considers a policy change. Consequently, at the individual level, each country has its own incentive toward a change. In other words, each country has its own threshold upon which it bases its decision to join the process of diffusion

With these results in mind, we can make the assumption that veto players are the entry points for the information on new policies that are introduced in neighboring countries. In other words, they are influenced by the different mechanisms of policy diffusion.

## 2.5 The mechanisms of policy diffusion

In Section 2.3.1, diffusion has been defined as an interdependent process that occurs between countries that influence policy decisions. In modern democracies, policy decisions are mainly internal to the countries. In Section 2.4, we highlight some of the most important internal factors that influence a policy change.

However, participation in the process of diffusion not only depends on internal factors, but also on the different interdependencies that exist among countries. These interdependencies are expressed by four largely accepted mechanisms, namely learning, competition, emulation, and coercion. The remainder of this section is dedicated to the explanation of these mechanisms.

### 2.5.1 Learning

Learning is defined as a process whereby the experience of policy makers in other countries supplies relevant information about the results of a policy and permits the update of policy makers' prior beliefs on the consequences of this policy (Meseguer, 2004, 2005, 2006*a*; Simmons, Dobbin and Garrett, 2006; Braun et al., 2007; Gilardi, Füglister and Luyet, 2009). Consequently, the experience of others is fundamental for learning to occur. If no country has experienced a change, no learning can take place (Shipan and Volden, 2008). Moreover, to learn, policy makers must update their beliefs on the effects of the alternative policy (Dobbin, Simmons and Garrett, 2007).

To take account of the updating of the beliefs, one may focus on the process of Bayesian updating (Simmons, Dobbin and Garrett, 2008; Meseguer, 2003, 2004, 2005, 2006*a*). The idea is that at each time step, the country changes its beliefs on its current policy according to new and perhaps more consistent data. Figure 2.5 gives an example of the possible evolution of the beliefs of two agents, the pessimistic and the optimistic. A simple simulation of agents choosing between two policies is used to build this figure. At each time step, they are facing new data on the current and alternative policy effectiveness of their neighbors and, as a response, must update their preferences.

To explain Figure 2.5 more accurately, we need to make several assumptions: The agents are purely rational and the same information is identically available for each agent. A retrenchment policy for unemployment benefits is introduced and one agent (the light-gray one) is very optimistic that this policy will cut the unemployment rate – he has a high preference towards this policy – and the other (the dark-gray one) is rather pessimistic, with a low preference towards this policy. We can imagine that at each time step, new information on the consequences of this policy is available. Unemployment rate is decreasing rather slowly for the optimistic and more than expected for the pessimistic so that they update their beliefs, with the consequence that the optimistic agent becomes more and more skeptical about the effectiveness of the policy, which decreases its preference, and the opposite takes place with the pessimistic agent.

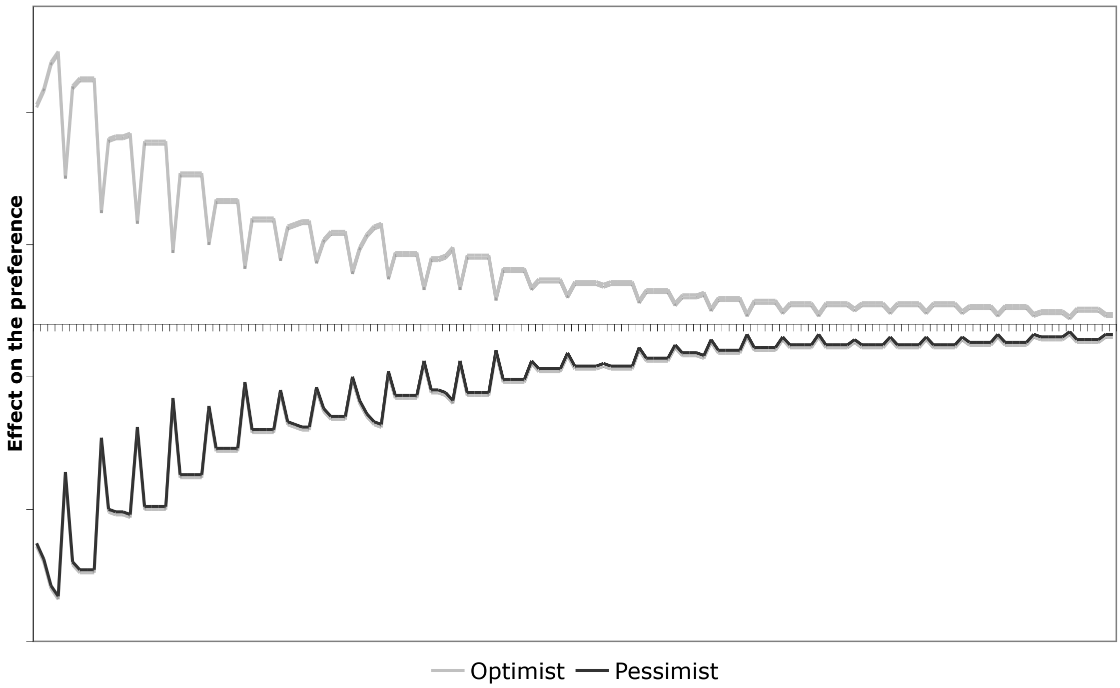



Figure 2.5: Bayesian updating

In the theory of policy diffusion, scholars usually distinguish between *purely* rational learning and *bounded* rational learning. Purely rational learning corresponds to the idea that governments scan all the available information before deciding a policy change. Rational learning assumes zero-cost information. Hence, in this version of learning, a country assumes that the information is not only free but also symmetric. In other words, every country has the same free access to the same information. This clearly poses a problem of uniformity or homogeneity between the countries. Assuming that countries are homogeneous while learning is a rather strong assumption. As a consequence, purely rational learners, while facing the same information, even if they are intrinsically different, will use this information in the same way and, with this assumption, should obtain the same results. Hence, this is a unrealistic situation. As a consequence, we cannot assume this homogeneity.

However, governments scan all the *relevant* information (Meseguer, 2005, 72). Countries try to accumulate information on the alternative policy from their neighbors using *cognitive short-cuts*–they only look at successful policies and/or successful countries (Simmons, Dobbin and Garrett, 2008, 29)–that facilitate the learning (see e.g. Simmons and Elkins, 2004; Weyland, 2002*b*,*a*). In more realistic cases, countries are using a bounded version of learning. Bounded learning involves, following cognitive psychologists, generalization problems, and overestimation problems (Weyland, 2002*a*). The former implies that, based on a narrow set of observations, people generalize their conclusions. In such a case, information on the consequences of a policy supplied by the innovations used elsewhere may be more or less relevant (Simmons and Elkins, 2004), and the latter is characterized by the lack of analysis of the alternative policy. In other words, a country introduces policy innovation without the necessary adjustments to the national context, because its government lacks the critical information needed to understand the consequences of the alternative policy (Simmons and Elkins, 2004). This haste is, as shown by Strang and Meyer (1993), a consequence of the *proselytism* of the new adopters.

Even though Meseguer (2006*b*) stresses that these two versions of learning are not necessarily incompatible, it seems nevertheless that the bounded version suits diffusion best. The fact that the closest neighbors have more weight; that is, bandwagon pressure[11], speaks for the bounded version of learning. Moreover, countries involved in a process of diffusion tend to interact more with similar neighbors (see e.g. Case, Rosen and Hines Jr., 1993; Abrahamson and Rosenkopf, 1997; Shipan and Volden, 2008).

However, learning does not necessarily imply the introduction of the best policy. The fact that rationality is bounded leads to the adoption of a version of the policy that seems the best one, according to the current choices of the neighbors.

The experience of *others* influences how beliefs are updated and, thus, the willingness to introduce a new policy. It seems logical to hypothesize that, if the neighborhood is larger, so is the possibility to get information. Logically, the more information, the higher the probability to introduce the best-suited policy.

[11]See section 3.4.6 in the chapter 3 for an explanation of this concept

Nevertheless, countries that change their policy should not introduce a carbon copy[12] of the alternative policy (the policy of a chosen neighbor), but only accommodate the alternative policy to their needs. Moreover, as we already stressed, the process of diffusion is likely to be conditional: the policy makers are not all equally responsive to the influence of the neighbors and, thus, do not learn in the same manner (Gilardi, 2010).

Consequently, conditional learning is the solution of the above mentioned problems of generalization and underestimation of the learning process. Volden, Ting and Carpenter (2008) have created a game-theory model that shows the existence of conditional learning. For instance, a state that wants to introduce an antismoking law will firstly introduce it, and then abandon it later on if ineffective, or will wait until the effectiveness of this policy has been proven. However, the conditionality of the learning process works both ways; that is, if the policy seems successful as well as whether the policy seems ineffective (Volden, 2010). If the policy change in one or more neighbors induces an electoral setback, the country will be less likely, in turn, to change its policy, in order to avoid the bad consequences seen in the neighboring countries and, conversely, tend to learn more from successful examples (see e.g. Gilardi, 2010; Gilardi, Füglister and Luyet, 2009; Volden, 2006; Shipan and Volden, 2006). The take-home message here is that countries tend to learn from their successful neighbors, which corresponds to a bounded rational version of learning.

In the next subsection, the role of competition for the process of diffusion is explained.

### 2.5.2 Competition

This mechanism is mainly an economically driven mechanism. By economically driven we mean that "governments act strategically in order to attract economic activity" (Simmons and Elkins, 2004, 173). The introduction of a policy change can

[12] Sharman (2010) shows that policy diffusion may be the result of a simple "copy-paste" of legislation, which leads to strange consequences. Venezuela, while defining its tax blacklist of countries, just copied and pasted Mexico's list and "ended up blacklisting itself" (Sharman, 2010, 625).

give a country a gain in competitiveness (Simmons, Dobbin and Garrett, 2006). As a result of this hypothesis, this kind of mechanism has been widely developed in the study of the diffusion of globalization (see e.g. Dobbin, Simmons and Garrett, 2007; Simmons and Elkins, 2004; Simmons, Dobbin and Garrett, 2006; Elkins, Guzman and Simmons, 2006).

For example, the different countries compete to acquire economic advantage by attracting capital flow or by reducing the fiscal burden. If a country has, for instance, introduced fiscal advantages for attracting new enterprises, one of its neighbors (or all of them) will do the same in order not to lose the country's economic attractiveness. As stressed by Dobbin, Simmons and Garrett (2007), if one of the neighbors of a country, by ameliorating a policy, increases its attractiveness, it exerts some pressure over a change. Thus, competition in the context of liberalization is a significant driver of policy diffusion. In other words, governments, in the case of market liberalization, compete with their neighbors (Simmons and Elkins, 2004).

Furthermore, in the example of the reduction of fiscal burden, a government may cut taxes to be more in conformity with its neighbors, even if this policy change takes place during a policy-seeking period[13]. Based on this view, Besley and Case (1995) developed a model of yardstick competition in the case of tax setting that shows that citizens benchmark their government with that of one of their neighbor states and may punish the government electorally if the policy seems unjustified. Governments, in order not to be punished, compete with their neighbors to stay in conformity. Again, competitive pressures are significant when analyzing the diffusion of tax policies (Swank, 2006).

A third example can be found In the context of welfare state reforms, where such a mechanism can also play a substantial role. A country that has introduced a better health policy, for instance, will have decreasing health care costs on GDP. Therefore, it can use this gain to increase its investment in infrastructure, creating jobs and inducing a virtuous circle. Moreover, this wealthier country attracts more foreign

[13]See Section 2.4.2 for a reminder

investment, resulting in increasing economic growth. Thus its neighbors will be interested in this new health policy, so that they also hope to gain in competitiveness. Swank (2005) argues that welfare state retrenchments are the results of the diffusion of neoliberal policies due to competitive pressures. In other words, states tend to cut welfare benefits as a result of the competitive pressures that exist among their competitors. Hence, his hypothesis is that a country tends to adopt the political changes of its close competitors. To illustrate this, Swank (2005) uses the example of Denmark, which tends to be engaged in a race to the bottom in the case of welfare reforms with its closest competitors, Sweden and Britain. His study clearly shows that prior adoption of a welfare policy by the competitors has a significant effect on the current welfare state's policy changes.

Thus, this kind of interdependence exists because countries compete not only to attract scarce economic resources and to stay competitive, but also because governments of these countries want to keep the reins of power.

To summarize, economic competition forces a country to change its regulation in order to adapt to international competitive pressure, if the government in place wants to stay.

### 2.5.3 Emulation

Emulation can be defined as a process through which countries adopt a policy change because it is an accepted norm (Simmons, Dobbin and Garrett, 2006). Hence, emulation is a mechanism that includes different scopes of *taking-for-grantedness* (Braun and Gilardi, 2006), such as imitation and norms, for example.

In other words, policy change is accepted as "a legitimate state responsibility, something which all states ought to have" (Walker, 1969, 890). More precisely, emulation, as a mechanism of diffusion, is mainly driven by social constructivism; that is, 'the social construction of appropriate behavior" (Lee and Strang, 2006, 889). In other words, the introduction of similar policies is shaped by shared internal factors build upon a common development.

For example, Gilardi (2005) has shown that Independent Regulatory Agencies (IRAs) "have progressively become a normal way of organizing regulatory policy." The same conclusion applies for the spread of liberalism in general (Simmons and Elkins, 2004; Simmons, Dobbin and Garrett, 2006), in the particular case of public-sector downsizing (Lee and Strang, 2006) or in the case of tax policy. In this sense, countries who want to change their policy may imitate peer countries "simply because they are peers" (Meseguer, 2005, 73).

Norms are defined as common beliefs that are shared by a large extent of a social system (Rogers, 2003; Elkins and Simmons, 2005) as in our western societies, for example, where the "market" is the dominant economical "ideology." Moreover, as Simmons, Dobbin and Garrett (2006) stressed, the introduction of these common accepted norms may be purely symbolic, especially in the case of welfare state policies or human rights policies, since, sometimes, the introduction of these policies seems impractical.

Therefore, what diffuses is not the policy itself, but the representation, the social construction a country makes on the beliefs of the policy. Therefore, the symbolic properties of the policy mean more in the decision to change than objective characteristics, such as the effectiveness of the policy, for instance. Nevertheless, a government introduces a policy, not the representation of the policy.

Emulation and norms are, here, considered as equivalent mechanisms, because when they are following norms to induce a policy change, governments seek the symbolic characteristics of the policy. Thus, diffusion is driven by the *prestige* of that policy or even by the fact that this policy is taken as granted.

At a certain point of the process of diffusion; that is, at a point where the number of adopters is sufficiently high, new adopters only imitate what seems to be the best practices. More precisely, newcomers adopt a policy of peer governments with which they share some common features. In sum, norms, ideas or appropriate behaviors may cause a policy change and are labeled under the appellation of emulation (Gilardi and Wasserfallen, 2009).

Now that we have explained the three horizontal mechanisms of diffusion, namely learning, competition and emulation, we will describe a vertical – top-down – mechanism.

### 2.5.4 Coercion

Coercion is a process whereby pressures from powerful actors (international organizations or powerful countries) make nonconformist policies costly. Strictly speaking, coercion is not a mechanism of diffusion because it is not a horizontal mechanism but rather a top-down channel of diffusion (Meseguer, 2005). In other words, coercion involves a lack of choice in the countries with which to collaborate. For instance, the structural adjustment programs developed by the International Monetary Fund (IMF) are imposed upon governments and are strongly oriented toward liberalization. Consequently, coercion is, here, a mechanism that drives the diffusion of liberalization (Dobbin, Simmons and Garrett, 2007).
In this sense, coercion is more a top-down pressure, or an example of interactions between powerful and powerless countries, and can be defined as "the imposition of policies on national governments by powerful international organizations or powerful countries" (Braun and Gilardi, 2006, 309). Nevertheless, it is not always considered as a mechanism of diffusion per se and we will not use it throughout this work for at least two reasons:

1. the country level is the level of analysis and we assume that there is no supranational organization, so that only the horizontal mechanisms of diffusion are relevant;

2. we assume that the countries are equally powerful. In other words, no country has enough power to *impose* its political view.

It seems, here, important to briefly explain coercion because it involves policy change, even if this change is imposed rather than chosen.
Further, coercion can be stated as strong, where a policy is imposed "by govern-

ments, international organizations, and nongovernmental actors through physical force" (Dobbin, Simmons and Garrett, 2007, 454) or as soft; i.e., through "the manipulation of economic costs and benefits, and even the monopolization of information or expertise" (Dobbin, Simmons and Garrett, 2007, 454). The strong type of coercion is not really interesting for this study of diffusion, because it does not imply any choice. For instance, in the case of structural adjustment programs, the country that is helped by the IMF cannot choose the range of policy it may introduce in order to improve its current situation. However, the soft version of coercion is more in line with our view of diffusion, because it operates more through persuasion; as, for instance, the influence the European Union (EU) has on domestic policies (Braun and Gilardi, 2006). Therefore, soft coercion, while pushing toward international harmonization, corresponds to the sacrifice of some part of independence in order to respect the obligations that impose the membership of international institutions (Holzinger and Knill, 2005).

Although several authors have found no evidence of international organizations influencing domestic welfare state reforms (for the OECD Armingeon et al., 2004), (for the World Bank and the Interamerican Bank Weyland, 2004), (for the World Bank Brooks, 2005), it may be argued that, if such organizations strongly advocate privatization, this may change the policy preferences of politicians associated to this reform, thus making policy change and diffusion more likely. Moreover, these organizations may play a crucial role in lowering the transaction costs of searching for policy alternatives. For example, the OECD routinely issues prescriptions for reform in many welfare state domains; such as, for example, labor market policy (OECD, 2006).

By now, we have described the main internal determinants of change and the different mechanisms of diffusion that should help this policy change. The next section aims at putting all these fragments together in order to have a coherent framework of policy diffusion.

## 2.6 Synthesis

In this section, all the theoretical concepts developed previously will be synthesized and formalized into one framework that should serve as a basis for the construction of a computational model for studying policy diffusion. More precisely, we will explain how the different parameters interact and what the expectations from that are.

First of all, it is meaningful to say that most of the studies on diffusion highlight the impact of internal or external factors, even if they raise the importance of the other set of factors. For instance, in comparative studies, the study of diffusion implies placement of emphasis upon external factors. Without minimizing the effects of internal factors, they usually show the greater importance of external factors on the adoption of a new policy (see e.g. Elkins and Simmons, 2005). When internal factors are highlighted, it is more to underline the fact that interdependencies are of no impact.

Even if our model is not utility based, the theoretical model of policy diffusion developed by Braun and Gilardi (2006) strongly influenced our model, since it is also based on policy change. However, we add an explicit intermediate step: countries must choose an alternative policy before changing it. Plus, the evolution of and the interplay between the different parameters are different. Braun and Gilardi (2006) based the change on the comparison of policy makers' utility of the current policy (the status quo) and the expected one of the alternative policy. If the expected utility of the alternative policy is greater that the one of the status quo, then the country changes its policy for the alternative. The main parameter to be calculated here is the different utilities that depend upon payoffs associated with votes and policy. More precisely, policy makers are seen as vote seekers and/or policy seekers, according to a weight that takes into account the periodicity between the elections, meaning that policy makers become vote seekers when elections are near. Additionally, the expected utility of a policy depends on its effectiveness. Diffusion enters the model by influencing the different parameters, such as the effectiveness of the current and alternative policy, the different size of the payoffs, and so forth. Thus a change can occur if the expected utility of the alternative policy is greater than that of the status quo. Diffusion enters the model by influencing the different parameters –effectiveness of the current and alternative policy, the different size of the payoffs and so forth.

However, even if we use the same sets of parameters, we do not make the same use of the different parameters, as it will be explained below. We base our model on a simplification of this model. We assume that the different veto players, when searching for an alternative policy, implicitly integrate the costs for changing the current policy and that the different parameters are already aggregated at the country level. Moreover, countries are effectiveness seekers; that is, the expected utility is only expressed by the effectiveness and political insecurity and institutional constraints can only speed up or slow down the processes of choice and change. Therefore, they have no direct influence upon effectiveness. In other words, they only influence the ability to seek relevant information and the time of choice and change.

Nevertheless, we base our model on the decision of change for an alternative policy; i.e., a country that, when facing an ineffective policy, decides to change it. This implies that, even if the country level is the level of analysis, it is necessary to investigate the influence of inner factors on policy change. Hence, this decision to change is based on the evolution of two parameters; that is, the effectiveness of the current policy and the ideology or the preference a country has in favor of (or against) this policy. Formally, this can be stated as follows:

$$E < P \qquad \begin{array}{r} \text{where } E \text{ means the current effectiveness} \\ \text{and } P \text{ the preference for the current policy} \end{array} \tag{2.1}$$

More precisely, at each time step, these two parameters are compared. If the effectiveness level is lower than the preference, then the process for a change to occur is launched. In other words, such a process is started when the policy is so ineffective that it exceeds the preference level for the policy, meaning that such a policy has become so ineffective that a change is needed despite the ideological preference for that policy (see Sections 2.4.1 and 2.4.2).

Before changing the ineffective policy, the country has to look for an alternative policy. The fact that diffusion has been defined as an interdependent process that takes place between countries and influences policy decisions,[14] thus implying that the choice of an alternative involves interactions with neighbors. The way in which a country and its neighbors are connected is expressed by the different mechanisms of diffusion – namely, learning, competition, or emulation – as defined and explained in Section 2.5. In other words, the choice of an alternative policy is the opening gate for diffusion.

This means that, when the country is ready for a change, the different political players are seeking information in their neighborhood[15] and they are analyzing it in order to find the option that suits their preference best. In other words, governments (and other actors) are looking for pieces of information on the effectiveness of an alternative policy in countries that have already introduced it. Thus, each country seeks the best solution. At this step, we can say that each of the different actors in the political game[16] furbish their arms in order to ease (block) the introduction of an alternative policy that goes towards (against) their preference. In other words, policy makers, on aggregate, assume that the alternative policy will have at least the same effect as in the neighboring country(ies), meaning that they assume to gain the same benefits of changing as their neighbor(s) had.

Thus, when a country has chosen its alternative policy, the political/institutional constraints must be overcome in order to allow the policy to change. Furthermore, the policy change in a country depends on internal factors. Except when the mechanism at play is coercion, policy change, at least in democracies, is a mostly internal mechanism. More precisely, after furbishing their arms, the different actors fight against each other – they play the political game that leads to a change. These

[14] See section 2.3 for a reminder of the definition of diffusion and its implications.
[15] For a definition of the neighborhood, see section 2.3.3.
[16] See section 2.4.4

actors, defined as veto players; i.e., "a certain number of individuals or collective actors [that][17] have to agree to the proposed change" (Tsebelis, 2002, 2), as mentioned above, based on their ideology, the effectiveness of the current policy and the expected effectiveness of the alternative policy. As a result of this game, the country changes its current ineffective policy for a supposedly more effective policy.

This process may take some time depending on how many players there are – and their relative strength in the political landscape – and on the political insecurity the players are facing introduce some uncertainty about this process of change. Consequently, the veto players introduce some unpredictability in the process of policy change.

Furthermore, this political game ends up with the definition of a global threshold against which a change is possible. Remember that in Section 2.4.4, the role of the threshold in the process of diffusion has been highlighted with the example of a riot (Granovetter, 1978). When entering a riot, an agent includes it closest neighbor; i.e., the neighbor with the slightly greater threshold. Thus, the more actors that riot, the greater the incentive for entering the riot. Of course, diffusion processes are a bit more complicated than this rioting example. However, the more countries that have changed their policy, the more information will be available and the greater the pressure toward a change. This phenomenon is known under the label of *bandwagon pressure*[18]. An example will help make that clearer.

Suppose that country A has some problems in the health care domain and suppose that a consensus has been reached on the problem; for instance, hospital financing costs too much as a share of GDP, denoting that the effectiveness of the current policy is no longer sustainable despite the preference for that policy. To end this bad situation and try to reduce costs, the government has to find a solution. In other words, at this point, the effectiveness of the policy cannot be supported by the preference and, as the different veto players want to make up their mind on the different possibilities and consequences of change, they look at what their neigh-

[17]My brackets

[18]These pressures will be explained in more details Section 3.4.6.

bors that are in the same situation do in order to improve their situation. In other words, depending on their place in the process of diffusion, they use the different mechanisms of diffusion in order to choose the alternative policy. After this step, a game starts between the actors that involves the internal factors; in order to obtain a consensus on the policy to introduce, they define a threshold for a change.

When a process of diffusion is initiated, a process of change begins when the effectiveness of the policy is lower than the preference toward it and the different countries involved are not equal; i.e., they are not homogeneous facing the information (Gilardi, 2010; Volden, Ting and Carpenter, 2008). For example, countries that first change their policy are considered early adopters. At this point in the process, there is less information and it is easier for the country to update its beliefs on the consequences of the change of the policy. If the new policy of these early adopters is more effective, their competitiveness may increase, hence pushing their close competitors to change theirs in hopes of similarly increasing their competitiveness. As the process unfolds, more and more information becomes available and, thus, it is harder for the countries to sort information, so that they decide to change their policy according to the prestige the introduction of this alternative policy can give. In other words, they emulate their neighbors.

Therefore, for a process of policy diffusion to occur, not only must the countries take into account their own internal characteristics, but it should also look at what the others do. In other words, a country must be ready for the change that is the country that is not satisfied with its current policy – because it is ineffective, for example – should evaluate the policy of the others, decide whether or not the country(ies) it is looking at has (have) an acceptable policy; and, in the end, introduce a new policy.

In sum, the process of diffusion occurs through heterogeneous countries, since they have different internal factors that interact with each other according to the different mechanisms of diffusion. As a result, the interactions of heterogeneous countries make diffusion a complex process, since the result of this process can hardly be deduced from these interactions[19].

## 2.7 Conclusion

We start by outlining the evolution of knowledge in the field of policy diffusion at the theoretical as well as at the methodological level. We then explain one of the most widely known definitions of diffusion.

When talking about diffusion, it should now be clear that it is a spatial as well as a temporal process that involves several dimensions. Part of these dimensions are embedded in the two main concepts around which I develop my model; namely, threshold and bandwagon pressures. The former represents the ideal point of a country; that is, some kind of an average value of the internal factors against which a change is decided, and the latter is characterized by the pressures exerted by the neighbors that have already changed their policy. These two concepts influence each other, creating a virtuous circle. In other words, when a country has chosen an alternative policy, depending on the influence of the different mechanisms of diffusion, because a consensus has emerged among the veto players, it increases the number of countries that have changed. Thus, bandwagon pressures becomethe more and more significant, affecting the search for alternatives of the countries that have not yet changed their policy, and thus the mechanism of diffusion. Consequently, we are facing a model of policy change into which diffusion enters with the influence of the different mechanisms on the way countries are looking for information on the new alternative policy.

I then identify four internal mechanisms: the effectiveness of the policy, which can be defined as the attainment of the goals for which the policy has been designed; the ideology or the preference for the policy; political insecurity; that is, the fear of losing governmental power; and institutional power, as characterized by the veto players.

Furthermore, the choice of an alternative is influenced by diffusion mechanisms,

[19]Complexity and the best methodology to study it are the main subjects of the next chapter.

which are grouped in four main categories; i.e., learning defined as an exchange of information between countries that permits the update of beliefs about the alternative policy; competition; that is, economical pressures of the competitors who, by changing their policy, have gained in competitiveness; emulation, defined as the imitation of the most successful peers; and coercion, characterized by a policy change that is constrained by the most powerful actors.

Policy diffusion has been explained with emphasized either on domestic factors (known as *bottom-up* mechanisms) or on international factors; that is, top-down (coercion) or horizontal mechanisms of diffusion (learning, competition, and emulation). “Traditional” studies on diffusion are based on the homogeneity assumption. This implies that all countries have the same odds to enter a process of diffusion and that they are equally affected by this process (Strang and Tuma, 1993). Clearly, each country is different from its neighbors; that is, the countries are heterogeneous. Consequently, his heterogeneity must be part of the study of diffusion! This can be done, at the methodological level, by developing computational models; more precisely, computational agent-based models.

# Chapter 3

# A theory of agent-based modeling

## 3.1 Introduction

This chapter is about a rather new methodology that can be used to tackle one of the major problem of statistical methods; i.e., to take into account the nonlinearity that characterizes social processes. Thanks to the development of personal computers, social sciences can now use this new methodology, namely computational agent-based modeling.

We start this introduction (and our trip through the theory of agent-based models) with a short history of the evolution of computational modeling (of course, the history of computers and informatics is important for the development of computational modeling, but it is too far from our purpose to be of any use here). Three distinct phases characterize the evolution of computational modeling (Macy and Willer, 2002; Gilbert and Troitzsch, 2005):

1. *Macrosimulation*: The first attempts to use computers in research date back to the 1960s. The main idea was to predict the evolution of some parameters based on quantitative assumptions. To do so, simulations of discrete event dynamics were developed. To achieve such prediction goals, were developed huge systems of computed equations. The most famous example of such a model is that developed by the Club of Rome in its report *Limit to growth*, published

in 1972 (Gilbert, 1998). The authors showed that the economic growth of the "Golden Thirties" was unsustainable in the future due to the constraint of limited resources. This first major attempt to predict the evolution of the world has had a huge impact, but was quite a failure because of the problems posed by the choice of assumptions and the problems inherent to prediction, as we will see in Section 3.4.3.

2. *Microsimulation*: Beginning in the 1970s, this kind of simulation is based on low-level entities as a unit of analysis, such as individuals, for example. Each agent is characterized by a set of attributes that are estimated using statistical distribution, as in agent-based models as well as systems of equations and algorithms to approximate the different behaviors of the agents. At each time step, each agent's feature is updated in an independent manner, meaning that the different agents are "socially isolated" (Macy and Willer, 2002, 146). This kind of models uses the bottom-up approach; i.e., the change in the attributes of the different agents – seen as low level entities – gives rise to a macro phenomenon. In such a model, the researcher tries to explain aggregate characteristics at a higher level, such as a region or a country, for example. Here again, their main use is to forecast macro-level events that affect micro-level actions.

3. *Agent-based simulation*: In the 1980s, with the development of personal computers, new computational possibilities could be explored. One important exploration method for researchers was (and still is) the agent-based simulation. Like microsimulation, it is a "bottom-up" approach, but unlike microsimulation, this kind of simulation is based on the interactions and adaptations at the agent's level.

As we can see, the idea of building simulation and computational models is not really new. It is notable that in the USA, the first development of what can be seen as computers put forth the idea of using them for war experiments (nowadays, it corresponds more or less to the first development of prisoner's dilemma games).

This chapter will be concentrated on the explanation of this latter type of simulation. Agent-based models are used in a great variety number of sciences, such as physics or biology, for example, but also in social and human sciences, such as economics, mainly on organizational problems (Thomsen et al., 1999; Rudolph and Repenning, 2002; Rosenkopf and Abrahamson, 1999; Lomi and Larsen, 1996), but also in archaeology with the well-known Anasazi model. The aim of this model was to investigate the sudden vanishing of the Anasazi civilization, a pre-Colombian native tribe in what is now the southwestern United States of America, where they lived from around AD 800 to AD 1350. The target[1]; that is, the "real world" story, was developed using a huge amount of environmental, demographical, and historical files (Epstein, 2006, chap. 4-6).

In the political science field, agent-based modeling is now gaining more and more significance. A large number of works have been developed in international relations using the computational and agent-based approach, mainly in conflict research (see e.g. Cederman, 1997, 2002, 2003; Lustick, Miodownik and Eidelson, 2004) and democracy (see e.g. Elkink, 2006*a*, 2009; Cederman and Gleditsch, 2004; Hegre, 2005)[2].

Yet, at the methodological level, political scientists remain to use "traditional" statistical methods, based on probability and regression and on several global assumptions. Benoît (2001) has listed at least three:

- *Aggregate assumptions*: Aggregation is building categories. In other words, when aggregating, we take the common features of things and then put them in a common category. For example, when we imagine a human body, we imagine legs, arms, eyes, and so on, but not Paul, or Susan. In other words the construction of models is based on the assumptions that only some features are relevant for one purpose (Holland, 1995).

[1] See Section 4.4.8 for an explanation of the building of agent-based models.

[2] We will develop some of these examples in Section 3.4.6

- *The independence of the different variables*: Playing with statistics forces the researcher to develop models using dependent and independent variables. The question that arises from this statement is: *Are the independent variables really independent between them?* Taber and Timpone (1996) answered this question by pointing out the fact that the comprehension of the mechanisms that connect the independent and dependent variables is critical for the interpretation of the prediction of the variance. Often, correlations between these different variables show only a small part of the relation. The use of ABM allows the overtaking of this problem; i.e., in ABM, the dependent and independent variables can be interdependent.

- *Identically distributed observations*: Each random variable has the same probability distribution as the others. This is best expressed with an example. Imagine you are playing "Heads or Tails." At each round, the probability of obtaining "Heads" or "Tails" is the same and is independent from the results of the preceding round. In other words, if, in the ten preceding plays, you have had "Heads," at the next play, you have exactly the same odds to have "Heads" or "Tails." More precisely, the results of the different plays are independent from each other.

Therefore, simulation methodologies are often opposed to statistical procedures. Moreover, all these assumptions must be put aside when using computational methodology, except the first one, as we will explain in Section 4.4.8. In this sense, the computational methodology will give new insights into the understanding of the diffusion process, because it allows us to see the evolution of the process according to the different agents' interactions. However, to test a theory, it is important to keep in mind that computational modeling is one tool among others, because the analysis of social phenomena cannot be studied using a single approach (Taber and Timpone, 1996).

The purpose of this chapter is to develop and explain theoretical bases that are used to build computational and agent-based modeling. After emphasizing the distinction that can be made between computational and agent-based – this distinction is not very easy to make, but it is an important theoretical distinction that needs to be addressed (section 3.2.2), I will highlight the epistemology that undergirds computational modeling (section 3.3). Furthermore, at the end of this chapter, I will explain more narrowly why agent-based models are one of the best tools for the study of policy diffusion and give some examples that will highlight the different possible uses of agent-based models in general and in political science in particular (section 3.4.6; and, for a more precise description of the development and use of computational agent-based modeling: section 3.4.7). A conclusion will follow.

## 3.2 A definition of computational modeling

We start our exploration of the computational agent-based modeling's world with the explanation of the different steps and techniques one should follow for developing computational models in general. In a second step, the necessary distinction between computational models and computational *agent-based* models will be explained.

### 3.2.1 On models

Most scientific work relies on the development of models as an abstraction of the "real-world" phenomena we want to study. Computational and agent-based models are no exception. Figure 3.1 represents the process to follow in order to develop a simulation[3].

To do so, one should start by looking at an interesting "real-world" phenomenon. Once this has been identified, the next step is to develop a theory around this phenomenon. In other words and more precisely (Ahrweiler and Gilbert, 2005; Gilbert and Troitzsch, 2005), we construct the model from a target that corresponds to the "real-world" phenomenon under study; then the target is abstracted; that is, the researcher narrows the target to the relevant characteristics for the aim of her research.

[3]Simulation is often used as a synonym of computational and computational agent-based models. This will be discussed in the next subsection.

Abstraction is simplification, so that the model is a simplification of the "real-world" phenomenon to the useful explanatory variables. The question one should then ask is whether a computational approach is needed. If the answer is positive, then the researcher's thoughts have entered the central gray part of Figure 3.1. At this point, the researcher mathematically transforms the different interactions in a way that they can be computationally used. In a third phase the simulation is run in order to have data that is "compared" with "real-world" empirical data (see Figure 3.2). This is a gross summary of what is developed in the next sections.

To summarize, the development of a computational modeling project must go through several phases: the development of a theory that highlights the "real-world phenomenon," the development of a model; that is, the abstraction of this phenomenon, the development of a computational program, and the test and analysis of the model (Taber and Timpone, 1996).

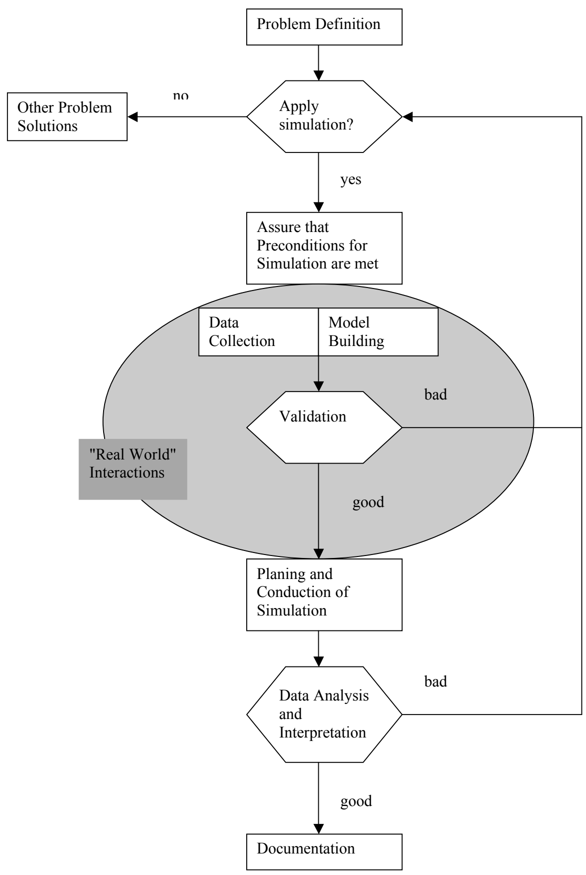



Figure 3.1: Simulation process, (Becker, Niehaves and Klose, 2005, 4)

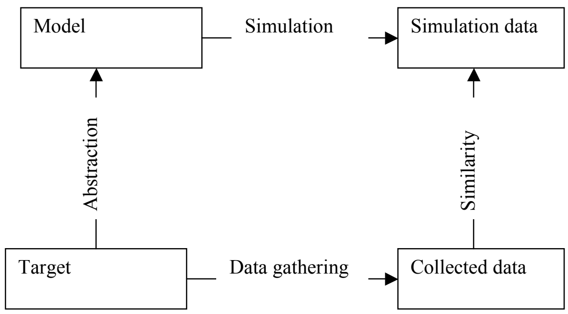



Figure 3.2: The logic of simulation as method, (Gilbert and Troitzsch, 2005, 17)

In order to develop a good model, some basic principles should be followed (Casti, 1997; Axelrod, 2003):

- *Simplicity*: the model should be simple, but not simpler. In Axelrod's words (2003, 6), we must follow the KISS motto: "Keep it simple, stupid!" Not only should the abstraction of the reality (or the target) concentrate on a few simple global patterns, but the results of the simulation should also be concentrated on a few simple explanations, consistent with the observed evolution of the phenomenon under study (Casti, 1997).

- *Clarity*: The description of the model must be unambiguous. The assumptions on which the researcher has decided to concentrate should not be subject to interpretation. One possibility to test the clarity of a model is to program it by using a different programming language and environment.

- *Bias-free*: The theory that is behind the model should be strong. The stronger the theory, the more objective the model and the better it is (Casti, 1997).

- *Tractability*: If the development of the model far exceeds the researcher's capacity in time and money, then it is worth finding another solution, because the model is considered intractable (Casti, 1997).

The goal to achieve when designing an agent-based model is what Goldstone et al. (2005, 425) called "idealized models;" that is models that "are typically motivated to describe domain-general mechanisms with a wide sphere of application." We study here a general model of policy diffusion as the results of interdependencies that exist between countries, and we hope that its results could be more widely used, because, when highlighting a comprehensive structure, idealized models can be very useful in explaining real-world phenomenon, based on few explanatory variables (Goldstone and Janssen, 2005). For example, Axelrod's model of dissemination of culture[4] is very helpful for the understanding of the divergence that still exists in our globalized world.

Since one should decide the level of simplification, and thus of abstraction, of the model, and, based on that, reveal the significant effects, it is important, in the first development of the model, to highlight the key theoretical points on which to base the simplification of the theory and the development of a model (Miller and Page, 2007). This task is difficult. It corresponds to the *art* part of the modeling[5]. A good example of simplification is given by Epstein and Axtell (1996) in their book *Growing artificial societies.* In this book they give birth to a world called *Sugrascape*; agents have several internal features, such as being a parent, vision, metabolism, and so on, and they follow some simple rules. For example, to see if seasonal migrations do exist, they define a seasonal rule. The world is split into two regions; i.e., the summer region and the winter region. At each step, the seasons flip; i.e., winter becomes summer and vice versa, and, as the rate of growing resources depends on the season, it flips too. As a result, some agents become *migrants*; but, more surprisingly, others become *hibernators* (Epstein and Axtell, 1996), that is they rarely migrate. As we can see, this rule is a simple assumption based on the "real world." Thus, by adding more parameters and rules, the authors create a complete world where the interactions of the agents give rise to several histories with birth, death, wars, peace, trade and other phenomena.

This example is a good example of computational agent-based models, because a whole world has been created using simple assumptions to define simple rules, and

[4]See Section 3.4.7 for a description of this model.

[5]For more on the art of the modeling, see section 3.4.5

the interactions between the agents give rise to a macro history that can be compared with the real world.

In the next sub-section, the distinction between computational and agent-based models will be explained. Usually, scholars do not make a strict distinction between these two concepts, because the concept of agent-based modeling is embedded in the one of computational modeling and, as a result, they are often considered synonyms. Nevertheless, as explained below, it is an important distinction to make for our purpose.

### 3.2.2 Computational vs. agent-based modeling

When talking of simulation in a general way, the following definitions can be applied:

1. "Computational models, then, are theories rendered as computer programs" (Taber and Timpone, 1996, 3).

2. "Computational modeling, (...), specifies all formal relationships algorithmically and discovers solutions by "running" the algorithms, that is, by computing the particular solutions for a range of initial conditions" (Taber and Timpone, 1996, 7).

3. "(...) simulation involves creating a computational representation of the underlying theoretical logic that links constructs together within these simplified worlds" (Davis, Eisenhardt and Bingham, 2007, 481).

4. "[A] computer model is equivalent to a formal system; that is, it is closed system whose dynamics and evolution is fully determined by the set of acceptable initial conditions and transformations rules" (Boschetti, McDonald and Gray, 2008, 23).

Thus, a full definition of a computer simulation corresponds to the sum of these four definitions:

> *A computer simulation is a system whose dynamics and evolution is fully determined by the set of acceptable initial conditions and transformations rules, rendered as computer programs that specify all formal relationships algorithmically and discover solutions computing algorithms.*

Following this, some authors, such as Benoît (2001) consider simulation and *computer* simulation as synonym since, to simulate a model, one must build and run a computer program. In other words, we can say that a computational model is constructed and run as a computer program, which is basically instructions that can be read by a computer. More precisely, the strength of a programmed computer lies in its capacity to execute repetitive action (Holland, 1998). And a program consists of a set (or sequence) of instructions that a computer executes indefinitely until a certain condition is satisfied.

At this point, the social component (and the social interactions) are not taken into account. A computational model can represent different things such as, for example, a flight simulator, a video game, or the evolution of the interest rate. In order to introduce social interactions into computational modeling, the "bricks;" that is, the basic components from which we develop our model, should be computationally described (or programmed) with some conditions of interaction that rely on the abstraction of real-world behaviors.

Usually, this basic brick is called *agent*. The model that lets these agents interact is an *agent-based model* (Holland, 1998, 117). In other words, the real world can be described by different interactions that exist between individuals.

An often cited example (Holland, 1998; Zwirn, 2006; Goldstone and Janssen, 2005) for highlighting the building and the behavior of an agent-based model is the operations of an ant colony. Let us describe it quickly. Each ant can be seen as an agent that follows simple interacting rules. As a result of these interactions, the ants are creating colonies, exploring the neighborhood to find food, defending their territory, and so on. This example, as with most of agent-based models, is characterized by the absence of central authority. This feature is expressed in Axelrod's definition of agent-based model as a “bottom-up” process.

We can also find examples in the social science field. The best known example has been developed by the 2005 Nobel Prize in Economics recipient, Thomas Schelling, in his best selling book *Micormotives and macrobehavior* (Schelling, 1978). In this book, Schelling attempts to explain the segregation that has taken place in big U.S. cities by assigning a threshold of similar neighbors that the people agree to support for staying in that particular neighborhood. Schelling developed his model on a checkerboard with two population of agents; i.e., dimes and coins. Thereby, an agent-based model can be developed *without* the computational help. The problem is when the number of agents increases, so does the difficulty of resolving the evolution of the agents' behaviors. In other words, even if agent-based models (ABMs) can be studied *by hand*, it is nearly impossible to determine the calculus of the complex interactions of such models without the help of computers (Holland, 1998, 118). Thanks to the development of personal computers, we have now an excellent tool that allows us to deal with the inherent complexity of ABMs, as we will explain later on.

A formal definition of agent-based models (ABMs) is given by Axelrod (2003, 6): ABM is a type of computational modeling that “is characterized by the existence of many agents who interact with each other with little or no central direction. The emergent properties of an agent-based model are then the results of “bottom-up” processes, rather than “top-down” direction.” Agent-based models can be applied to a variety of interacting systems, such as international relations, ecosystems, immune systems, and so on. Thus, one of the main differences between computer simulations and agent-based models is that agent-based models can be developed without the need of a computer program. It can be stressed here that computational modeling has a broader application as agent-based modeling has, because the latter is limited to the study of the interactions between agents (whatever an agent is; i.e., a firm, a country, an ant, etc. ).

Computational models produce no outputs corresponding to the real world; and, while letting the agents interact, ABMs allow us to highlight the emergence that lies behind the complexity attached to the interactions that characterize every relation in our society. That is the purpose of the next section. More precisely, we will explain the "philosophical" emphasis of emergence and complexity for our purpose.

## 3.3 Toward an epistemology of computational modeling

We saw in Section 3.2.2 that the types of interactions in computational agent-based models are defined as complex. The complexity that characterizes such systems can be simply defined as follows: *The whole represents more than just the sum of each part.* The result of these complex interactions is the emergence that is an unexpected macro result of micro interactions. In social science, the science that heads these concepts is known as generative social science. This chapter offers an understanding of these concepts after an exploration of the importance of semantics in ABMs.

### 3.3.1 All is matter of interpretation

When we run an agent-based model, on the screen we see grids with changing colored cells, maybe evolving charts. How do we really know that the model we create forecasts the evolution of the phenomena we want to explain? Everything is a matter of interpretation.

When developing a computer program, the programmer assigns values to variables. The program consists, therefore, of lines of codes that are transformed into virtual signs by the compiler. In other words, values and variables are manipulated in the computer's memory; that is the agents created in these *in silico* worlds – countries (with policies) in the diffusion model, or ants in the above-cited example – are simply programs "that interact with each other by moving bits and bytes of data around from one memory location to another" (Casti, 1997, 142-144). Following this, we can say that, in their native state, these variables have no meaning at all; they are just syntax (lines of code), that are represented by strings of 0s and 1s in the computer's memory; i.e., only changes of numbers (Holland, 1995). Therefore, it is the act of interpretation, the injection of semantics, so to speak, that allows these electronic worlds to be the virtual counterparts of real-world observations (Casti, 1997).

The basics of programming (and also computers) are mathematics; that is, numbers, which represent a symbolic abstraction of the real world, and consequently a simplification of the real world. When we develop a computational model, we create a mathematical abstraction of the world in a computer. A simple example will help make that clearer. If you imagine the number "one hundred" (100), it is a pure abstraction, only a symbol, an empty hole. To put some meaning into this hole, you have to assign a variable to it (such as peers, francs, or men). Moreover, the computational program can be seen as an algorithm; that is, a set of rules representing mathematical conditions that formalize the process followed by the program. In such a view, the process lies in the world of pure ideas; i.e., pure abstractions (Casti, 1997).

For example, in our computational agent-based model of policy diffusion, the countries that are created within this world are only bits (that is, strings of 0s and 1s), but we give them the appearance of countries by our interpretation of the different parameters. Without being as provocative as Casti (1997) when he argues that "[t]here is no reason at all to think that our every day world has any privileged ontological status and is any more real than the world we can create *in silico*[6] rather than *in vivo*," it is important to be very careful about the interpreted connections between *in silico* and *in vivo worlds*.

Consequently, at this point, the question one should ask is whether the results of the interactions we program – the outputs produced by the model; i.e., the different charts, grids and so on that appear on the screen and that allows us to see the results – have any relevance to our comprehension of the real world. The answer is: It depends.

[6]In other words, the world we create *in* the computer

We have explained that to develop a model we need to abstract a real-world phenomenon; i.e., to simplify the world. To program this abstraction and then to run this program will not yield any information about the real world because the simulation produces an output; that is, the result of the choice of the input and of the interpretation of the phenomenon under study that the modeler made. Nevertheless, this output helps the programmer to understand the consequences – the production of the code – of the computational program "which, in turn, tells us about the appropriateness of the rules we implement and the input we choose" (Boschetti, McDonald and Gray, 2008, 23). As a consequence, information results only from the program or, more precisely, from the written codes.

To go a step further in our understanding of the theory of agent-based models, the next section is devoted to complexity as the concept that describes the mechanisms that define the agents' interactions.

### 3.3.2 Complication vs. complexity

Social scientists have been trained to practice reductionism. Studying and understanding one part of a system after another and then reassembling the acquired knowledge will give a comprehensive understanding of the whole. Therefore, for decades, the world has been studied as a complicated system. A complicated system is linear: Not only does the whole equal the sum of its parts, but if we alter the complication by removing one piece of the system, it will not fundamentally change the behavior of the system. Yet, it should not be treated as complicated, but as complex. Thus, when social scientists decide to reassemble the different parts of a system being studied, they enter the fascinating world of complex systems and, thus, problems appear.

As defined by Simon (cited in Cederman, 1997, 50), a complex system stands for "one made up of a large number of parts that interact in a non simple way. In such a system, the whole is more than just the sum of the parts." Therefore, a complex system needs to be study as a whole.

Studying a complex system one part after another fails because the different interactions that are central to the understanding of the system are nonlinear. For example, studying only agents on the demand side of a market won't give any indications about the evolution of this market because it depends on interactions with suppliers. In a social system, as in an ant colony, the interactions between the people, for example citizens (at the micro level) give rise to political parties. And political parties give rise to governments (the macro level). The form of government changes, as do the laws it promulgates.

The problem of the study of the social science is that, too often, the world is analyzed as a complicated system. In a complicated system, behaviors are often analyzed in a rational choice perspective. Agents in such systems are described as "optimizers;" that is, they try to optimize their utility under the conditions given by the model. In complex system, agents act in a different way to solve their problems. In such a system, the agents evolve according to their *interpretation* of their environment (Page, 2008). Instead of optimizing their behavior, agents adapt the behaviors to the new environment.

The introduction of complexity in the systems under study gives a new methodological orientation. The most visible and important one is the use of computational agent-based modeling. Thus the aim of social science is to understand these micro / macro relationships; and, therefore, to develop explanations of emergence (Gilbert, 1998). Social simulation is a major tool in analyzing macro phenomena that emerge from micro-level situations, meaning that, when some complexity is introduced in a model, the behavior of the agents is no more purely rational, as mentioned here above.

Here above, we explained the importance of heterogeneity for ABMs, which comes from the complexity that is embodied in social processes, such as Schelling's examples[7]. Indeed, if each agent is different from his neighbors, he will act differently and the sum of the different interactions will be largely unanticipated. Thus, the

[7]For a remainder of Schelling's examples, see section 3.2.2 and the footnote 12 on page 14.

understanding of inherent complexity is critical for the comprehension of the concept of emergence, because complexity is the cause of emergence; i.e., the different interactions in a complex system lead to often unpredictable results. More precisely, the local interactions that exist across agents are, as pointed out by Boschetti, MacDonald and Gray (2008, 21), not only the basic mechanisms for emergence to occur, but also "responsible for the immense variety of structures, patterns, and phenomena we see in Nature." In other words and to summarize, the interactions at the micro level give rise to often unexpected macro phenomena. This leads us to the next section, where we try to extend our knowledge of the different tasks of ABMs by the explanation of the concept of emergence.

### 3.3.3 The concept of emergence

We start our explanation of the notion of emergence with a small example. The evolution of nations is an emergent phenomenon[8] (Cederman, 1997). Beside the influence of internal factors, the nations' development process is partly due to interactions between neighbors, whether they be caused by wars, geographical closeness, or ideological proximity. Let us recall from Section 3.3.2 that the different interactions that exist between the agents define the complexity of a system and that, in a complex system, the whole is more than just the sum of its constituents; more precisely, a complex system is a nonlinear system. Thereby, emergence can be defined as "a product of coupled, context-dependent interactions" (Holland, 1998, 121-122). This definition clearly states that emergence is embedded in the concept of complexity and corresponds to its result. More precisely, the emergence of a global pattern corresponds to auto-organization due to interactions between a large number of agents. In other words, emergence denotes a macro-level phenomenon that is not anticipated from the micro-level interactions. For example, the calculation of an equilibrium in a marketplace is defined when supply and demand are equal. However, each agent tries to maximize its utility without computing the market

[8]So is the diffusion process, as it is the purpose of this thesis.

price (Page, 1999).

To explain this concept more clearly, we will use an example taken from the painting technique known as pointillism. Basically, this technique involves painting only dots and using only basic colors. When viewing a pointilistic painting close-up (see Figure 3.3) you will not understand the image; you will only see points of different colors,

Figure 3.3: Zoom on Signac's 1904 painting entitled *La Voile Verte*

but, while moving away, you will see the whole picture (see Figure 3.4). As you move away from the picture, its significance *emerges* and you'll start to understand its meaning.

Figure 3.4: Signac's 1904 painting entitled *La Voile Verte*

The same phenomena is observed in the development of computational agent-based modeling, because the essence of these kinds of models is that macro-level phenomena cannot be deduced from the micro-level behaviors of agents. Remember from Section 3.2.2 that, from the bottom-up approach, emerges a phenomenon that only depends on the connectivity between the agents. Therefore, the different interactions need to be precisely described. Because we don't know the consequences of the agents' interactions, the very core of the emergence lies on the specific nature of those interactions.

For example, in the animal world, we see emergent phenomena such as the organization of an ant colony, or flocks of birds. In each example, there is no leader that organizes the behaviors. Each animal/insect follows simple rules and the interactions between them give rise to an organized structure. An emergent phenomenon is due to the auto-organization (there is no central authority) at the global level of the system under study. Emergent phenomena also occur in the human world, as, for example, traffic jams. This phenomenon appears not only because of accidents. The different speeds of cars can cause traffic jams. In economics, the price formation is also due to an emergent phenomenon (Epstein and Axtell, 1996; Bunn and Day, 2009).

Emergence is characterized by transition phases; that is, a slight change in a parameter introduces a brutal change in the other parameters and in the system itself. To find this point of transition may be very important to act on the system, because its study helps explain the system at this critical point where the chaos becomes order. From this transition phase emerges order. An example can be found in Reeves' best seller *L'heure de s'enivrer* (1986, 106-107). During the winter of 1942, a thousand horses that were trying to escape a bombing-ignited forest fire, swam across Ladoga Lake in Russia. As the horses were swimming, suddenly the lake froze, transforming them into ice sculptures. This happened because the temperature went down quickly that night. As a consequence, the lake did not have the time to freeze and water stayed liquid[9]. But a slight change in the system, here particles of sand in the horses' hair, leads to a brutal transformation: the water froze very quickly, making a thousand horses prisoners. Therefore a chaotic system (liquid water) was transformed into an ordered one (ice).

[9]This process leading to unfreezing water when the temperature is falling below zero is known as *supercooling* and very clear water can stay a long time in that chaoticstate.

This example stresses the importance of the choice of the appropriate level of analysis needed to truly explain a phenomenon. One may ask what is more critical here for the disturbing of this chaotic system: the horses or the particles of sand. Of course, as Reeves explains, the particles of sand are the main driver for supercooling to occur, but, at our level, we can easily imagine that the swimming horses are disturbing the water enough to cause the changing (even if the horses had already swam halfway across the lake before the water froze!). Anyway, to explain this choice, we can rely on an another example (Zwirn, 2006). Flowers, like everything, are composed of quarks and electrons. If one wants to study the odor of flowers, it is unnecessary to study quarks and electrons because they have no odor.
According to the level of analysis, emergence can be separated into two different concepts:

**The *first-order emergence* concept** The *first-order emergence* concept refers to the emergence per se; i.e., emergence, as the unplanned macro-level consequence of micro-level interactions. An important pattern of this concept is that there is no central authority. This means that the agents are not aware of the consequences of their behaviors; only the observer is, because this macro-level property wasn't introduced or modeled in the agents' behavior (Squazzoni, 2008). Let us clarify this with an example. Ants only follow pheromone paths and, without any central authority, they build complex colonies, but they are not aware of that. Only the anthomologist (the person who observes the colony and the ants) is.

**The *second-order emergence* concept** The *second-order emergence* concept is the macro-level property of micro-level interactions yielded by agents that have a higher cognition. This means that a particular agent has the "programmed" ability to influence this macro-level property and, in turn, to be influenced by it. More precisely, there is a feedback loop from the macro level to the micro level (Squazzoni, 2008). For example, global warming can be seen as the macro-level consequence of economic interactions that are, in turn influenced by global warming: the so-called green tech is gaining more and more significance and is slowly modifying the business model. Simply think of all the advertising that promotes *green* consumption!

Holland (1998) reviews some of the main features a model should have in order to "show" emergence:

- The model should model the world
- The model should consist of a limited number of agents interacting with each other
- The organization of the agents adapts as time goes by
- The number of interaction rules designed by the modeler are succinct.

Political scientists include these features in most of their models. This means that they have concepts and theories they use to study emergence in the political world. Unfortunately, they lack the endogenization of this notion of emergence. For example, a great bulk of research in political science relies on game theory, but "game theory takes as given exactly who the actors are in particular setting" (Axelrod, 1997*a*, 125-126).

These two concepts can be embedded in the notion of complex adaptive systems and, more generally, in the generative social sciences, which are the purpose of the next subsection.

### 3.3.4 Some epistemological standpoints

The classical view in most social science stresses that an agent should be analyzed as purely rational, and to be rational, an agent needs to meet at least two main assumptions. First, he has complete information about and complete knowledge of the system he lives in; second, he always tries to maximize his behavior (his utility function[10]) according to this knowledge and information.

[10] We have already explained in the previous chapter. Let us recall that a utility function represents the agent's relative satisfaction with, in our case, the policy.

Computational agent-based models are developed in a system called *Complex Adaptive Systems* (CAS). A complex adaptive system can be described as follows: First, a CAS consists of a network of interacting distinctive agents; second, these dynamic micro-level interactions give rise to an aggregated macro-level behavior; and, third, this emergent behavior can be explained with a global understanding of the micro-level interactions. As a CAS is filled with agents, they too need to fulfill a couple of criteria: "An agent in such a system is adaptive if (...) the actions of the agent in its environment can be assigned a value (performance, utility, payoff, fitness, or the like); and the agent behaves so as to increase this value over time" (Holland and Miller, 1991, 365).

The need for a system as well as for the different agents to fulfill these different criteria clearly show that the paradigm behind the use of ABM is not rational but, rather, adaptive. This does not mean that an adaptive agent has no rationality; rather, it is the degree of rationality that changes. An adaptive agent is characterized by a bounded rationality. This is the core of the development of computational agent-based modeling. Because adaptive agents cannot optimize their behavior on their own, they must look at their neighbors' behavior. In so doing, as shown in sections 3.3.2 and 3.3.3, they "create" emergence through the complexity resulting from their interactions. Thereby, one of the key features of the CAS is the different interactions that exist among the different agents (Miller and Page, 2007).

While interacting, adaptive agents learn from each other. To learn, adaptive agents must look back. In so doing, they adapt their behavior to the evolution of their environment. They must integrate the past to build the future, contrary to the pure rational agents, which try, at each time step, to optimize their behaviors. Therefore, the key distinction between the paradigm underlying agent-based modeling and that underlying the more traditional statistical methodology is the notion of rationality and history[11] i.e., by resolving the same set of equations at each time step, contrary to adaptive agents, which "look backward and learn" (Laver, 2005, 264); i.e., their

[11]This notion will be developed in more details in section 3.4.5

present and future environment is influenced by the history of the model.

It seems that computational agent-based modeling provides a powerful tool that allows the analysis of complex adaptive systems because, as mentioned above, agent-based modeling deals with complex interactions and emergence. We can sum this up in the following motto: "If you didn't grow it, you didn't explain it"[12] (Epstein, 2006, 8). This motto has given rise to the notion of *generative explanation* and, by extension, to the notion of generative social science. More precisely, a computational model grows, because the different agents are interacting – due to interacting agents, a computational model experiment will develop in size and according to changes in the agents' behaviors – and these interactions can "produce the macro-level regularity of interest" (Squazzoni, 2008, 5), which explains the process under study. Therefore, the micro/macro link is central to the understanding and the study of computational agent-based models. Following Epstein (2006), when exploring the theory of this kind of modeling and, more generally, the so-called generative social science, we need to take into account the four main epistemological issues expressed here above:

1. *Generative sufficiency vs. explanatory necessity*: the motto we briefly mentioned above ("If you didn't grow it, you didn't explain it?") means that the emergence of a macroscopic phenomenon is part of the explanation. A model can also lead to the emergence of a totally absurd result. That is why "generative sufficiency is a necessary but not sufficient condition for explanation" (Epstein, 2006, 53). For example, if our policy diffusion model does not show the existence of such process of diffusion, does this mean that the process of diffusion does not exist?

2. *Generative agent-based models vs. explicit mathematical models*: ABMs are based on computer programs which, in turn, are based on recursive functions

[12]This can be logically expressed as follow (Epstein, 2006, 51):

$$\forall x(\neg Gx \supset \neg Ex) \tag{3.1}$$

(equations). In this sense, every ABM is equation based. The opposition here appears in the results given by the different types of models. A huge set of differential equations, as in macro- and micro-simulations[13], may be very difficult to resolve by hand, but may be very easy to resolve with the help of a computer. A good example of such a model composed of huge differential equations is given by the already mentioned Club of Rome report, *Limit to Growth*. Nevertheless, as noted by Epstein (2006, 56) in the case of climate modeling, these large set of nonlinear equations "are not solved analytically, but *approximated*[14] computationally." So, while this distinction is convincing at the theoretical level, it is exaggerated in practice, because computational agent-based models and, thus, computational models are precisely equation based.

The real difference between these two ways of developing a model concerns the taking into account of the (non) linearity. Roughly speaking, and following Holland (1995), by "linear" we mean that the whole is the sum of its parts[15]. Formally, a linear function can be expressed as follows[16]:

$$f(x) = 2x + 1 \tag{3.2}$$

The main methodological tools used in social sciences; for example, regression models, are based on this linear assumption as an approximation of the behavior under study. In other words, this expresses the fact that the effect on the dependent variable is equivalent to the sum of the effects of the selected independent variables (Gilbert and Troitzsch, 2005). For example, the interactions in the diffusion process cannot be obtained by adding the activities of the different countries (say the leaders and the followers); on the contrary, diffusion is representatively a nonlinear phenomenon, because this process plays

[13] See section 3.1
[14] My italic
[15] This corresponds to distinction made in section 3.3.2 between complication and complexity.
[16] Equations 3.2 and 3.3 are examples and are expressed here only for illustrative purpose.

in a complex system, as mentioned in Section 2.6. When stressing nonlinearity, the product of the different variables is important instead of the sum. In other words, the whole is more than just the sum of its parts. Formally, such a function can be written as follows:

$$f(x) = 2x^2 + x - 6 \tag{3.3}$$

3. *Generative explanation vs. deductive explanation*: Generative explanation is often considered nondeductive. The deductive method of doing science implies a beginning in the "real worl" and carefully follows the path to some conclusions about the behavior we want to study (from the general observation to a conclusion about the particular behaviors). For example, we observe the universe – a general observation – to infer conclusions about the behavior of the solar system–a particular system. In other words, "we account an observation as explained precisely when we can *deduce the proposition expressing that observation from other, more general, propositions*" (Epstein, 2006, 10). The problem comes from the recursive functions embedded in every program. Recursion is a way of solving problem by reducing it to one or more simpler sub-problems that are "identical in structure to the original problem" (Roberts, 2006, 1). Therefore, since a program corresponds to a set of recursive functions, a computational agent-based model is a strict deductive model.

4. *Generative explanation vs. inductive explanation*: The generative explanation is often considered noninductive. The induction can be simply defined as the search for general conclusions inferred from particular behaviors. This generative vs. inductive problem can be illustrated by an often-used problem, known as *El Farol Problem*. El Farol is an Irish pub in Santa Fe that is often overcrowded at nights when Irish music is playing. The problem states that it is difficult to decide when to go to this bar because, in general, people do not like it when the bar is too crowded. To decide, people define an acceptable

threshold of consumers as a definition of "crowded," and so the problem can be expressed as follows: if the consumers think that a few of them will go and have a drink, they will all go to the bar, which will be crowded as a result; conversely, if the consumers think the bar will be full, nobody will go and have a drink and the bar will be empty. In other words, from a particular situation, say the bar was crowded/empty last time, the consumers will decide in general what to do.

At the computational level, the agents cannot induce the best-suited behavior, but at the model level the researcher can induce some general behaviors agents will have when facing a choice. Therefore, a computational agent-based model cannot be considered non-inductive.

All this can be summed up in a simple question: *How could the decentralized local interactions of heterogeneous autonomous agents generate the given regularity?* (Epstein, 2006). This is the basic question one should ask before starting to use ABM. The answer may also seem simple: Situate an initial population of autonomous heterogeneous agents in a relevant spatial environment; allow them to interact according to simple local rules; and thereby generate or "grow" the macroscopic regularity from the bottom up (Epstein, 2006). But simple does not mean simplest, because the consequence of these complex interactions – emergence – is difficult to deduce from the behaviors of the agents[17].

Now we turn to the explanation of (computational) agent-based modeling. As it is the main methodology of this thesis, it is important to describe not only how it links to what has been explained so far, but also its different characteristics.

## 3.4 Why agent-based models?

As an in silico abstraction of the "real world," agent-based models design the behaviors and interactions of different agents according to simple plausible rules. Therefore, these different rules should computationally express the different mechanisms

[17] See Section 3.3.3 if you already forgot.

that drive the agents' behaviors.

Lustick et al. (2004, 212) summarize perfectly the necessity of computational agent-based model in social science when they put emphasis on the fact that this kind of methodology is very helpful "if theoretical expectations are relatively clear, but data are hard to find that reliably match theoretical categories, if available natural experiments do not allow crucial questions to be posed cleanly because of inconvenient confounds, and if key aspects of the phenomenon of interest are relatively rare." Moreover, agent-based modeling implies the direct implementation in a computational program – and, by extension, in the *in silico* world – of the different connections that exist between the agents without any transformations other than the abstractions made for creating our model. In this section, we will more specifically deepen our understanding of agent-based models.

### 3.4.1 On cellular automata

Interactions in *complex adaptive system*[18] are best designed as cellular automata. A cellular automaton is a way of expressing the different interactions of agents on a kind of lattice and this method of designing interactions can be applied to a wide range of situations. There seems to be a broad consensus on the use of cellular automaton in generative social science.

Cellular automata have been (and are) used in modeling a wide range of social phenomena, such as secessionism (Lustick, Miodownik and Eidelson, 2004), extremist opinion propagation (Deffuant, 2006), emergence of regional autonomy movements (Miodownik, 2006), the diffusion of democracy (Elkink, 2006*b*; Cederman and Gleditsch, 2004), the convergence/divergence in cultural habits (Axelrod, 1997*b*), and, in general, for developing a complete world (Epstein and Axtell, 1996). As these examples suggest, cellular automata are well suited for the development and the analysis of nonlinear systems, characterized by emergence of macro phenomena and

[18] A little terminological digression must be made here because some authors (Zwirn, 2006) stresses that adaptive agents *are* complex system, while for other (Holland, 1995, 1998) a complex system is composed of adaptive agents. For our purpose it is much more accurate to follow Holland's view.

agents who follow simple and easily described rule (Taber and Timpone, 1996). In other words, cellular automata are composed with a large number of agents that are locally connected. These connections depend on the type of neighbors as well as on the defined rules of interactions. Thus, a cellular automaton can be defined as a system composed of agents that are characterized by a certain number of finite states that change through time (Zwirn, 2006; Shalizi, 2001). These changes can be of two types (Zwirn, 2006, 65-66):

- First we have the simultaneous change, i.e, at each time step (discrete times), the states change and the result of the process at time $t+1$ only depends on the agents' state at time $t$. It corresponds to the changes in macrosimulations.
- Second, the change can be local; that is, the state of a typical agent is determined by that of its neighbors according to an invariant rule. We develop our model of diffusion according this type of change.

Here we have a first view of the different characteristics that are applied to agent-based models, namely *local interactions* and the *model's time*. These two characteristics will be explained later in this chapter.

Usually, cellular automata are represented as a grid composed of a defined number of cells as shown by Figure 3.6. What should be decided is whether or not this grid is bounded. Normally, in social sciences, to express the world, the grid is defined as non-bounded and is designed as a torus, which is represented by Figure 3.5. A torus is just a different way of expressing a square grid. More precisely, agents at the northeast corner have neighbors at the southwest one. For example, imagine a map of the world. On a map, Russia and the USA are completely opposed, but in reality these two are neighbors. The maps are normally cut at the Bering strait, which links Siberia and Alaska. If the map is wrapped around, Alaska and Siberia become close again. The torus has the same effect on the virtual world; this shape is just a way of getting closer to reality.

As previously mentioned, a cellular automaton is composed of cells that are either active or inactive, depending on the interaction rules we give. How a cellular

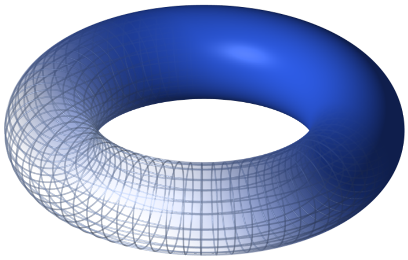


Figure 3.5: Representation of a torus

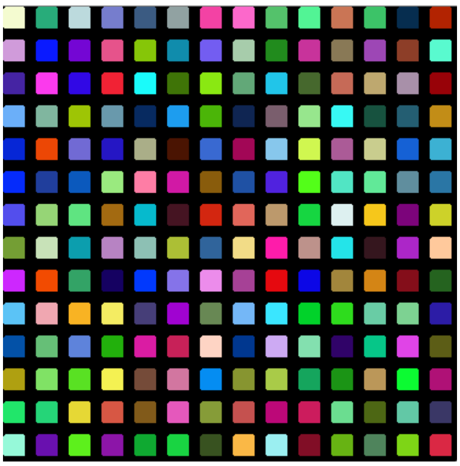


Figure 3.6: Example of a grid

automaton works and the importance of the different states of the cells are best explained with the help of one of the most cited examples of a cellular automaton; i.e., Conway's game of life[19] (see e.g. Zwirn, 2006; Taber and Timpone, 1996; Holland, 1998).

Let us describe briefly how it works. On a grid, each cell can be in two states: dead (the cell is empty) or alive (the cell is occupied). Each cell is taking into account its Moore neighborhood; i.e., the eight adjacent cells. The game follows two simple rules:

1. If a cell is alive (active) at time t, it stays alive (acive) at time $t+1$ if two or three of its immediate neighbors are alive (active), otherwise it will die (inactive);

2. if a cell is dead (inactive) at time t, it will stay dead (inactive) at time $t+1$ except if and only if three of its immediate neighbors are alive (active).

The result of this game can be drawn by hand only for a small amount of occupied cells. Otherwise, the evolution of the states of the different cells must be computationally designed. The game of life is easy to program and, with a computer, we can follow the evolution of the game over a long period of time.

Cellular automata are well suited for the representation of agent-based models, because it is another way of describing networks of connecting agents. Now we simply continue with the exploration of the main characteristics of ABMs.

### 3.4.2 On the different characteristics of agent-based models

We have already explained some basic examples of agent-based models; i.e., the ant colony, the flock of birds, or, in social science, the El Farol problem and the segregation model. All these examples share the same minimal characteristics. These common features are the results of large consensus among scholars in the field of

[19] You can find an example of how Conway's game of life works on the following site: `http://www.bitstorm.org/gameoflife/`

computational agent-based models (see e.g. Epstein, 2006; Gilbert and Troitzsch, 2005; Gilbert, 2008; Miller and Page, 2007). Here is a brief description of the main characteristics of ABMs:

1. *Heterogeneity*: In social sciences, agents (individuals) are normally considered as some kind of *Average Joe*; that is an average representative of the population under study. In economics, such a way of doing is particularly well established. Therefore, the statistical methods of analysis that are commonly used in the social sciences are based on the assumption of homogeneity. This homogeneity is not a common feature observed in the real world, but "rather a necessity imposed on us by our modeling techniques" (Miller and Page, 2007, 84). However, as it is clear that our society is not composed of homogeneous agents, it must be also clear that our classical methodology lacks an important feature of our society, namely the heterogeneity of the different populations. One methodological way to overcome this lack consists in developing computational agent-based models. Indeed, this kind of model allows us to develop and integrate heterogeneous agents in our analysis.
   Following this, the question that one should ask is not whether one should model heterogeneity, but what level of heterogeneity should be introduced in the model. This consists in answering to the following question: *What level of abstraction should characterize an agent?* Part of the answer is given by the already cited KISS motto[20]. The other parts of the answer come from the underlying theory (well developed or not) and from the modeler himself (his experience as an agent-based modeler).

2. *Autonomy*: : As we already mentioned, the ants create a colony without any leader. It is only the local interactions between the ants that give rise to this construction. Therefore, *autonomy* means no central authority. The system is a "bottom-up" structure. Again, in Schelling's segregation model, the two different populations segregate because of the tolerance threshold embedded

[20]"Keep it simple, stupid!" See section 4.4.8

in the agents' characteristics. In such a model, segregation occurs only as a consequence of the interactions between the agents as imposed by the threshold, without any central state.

According to the *second-order emergence concept*, a feedback loop between macro-level patterns and micro-level patterns exists. However, the fact that these two types of patterns grow mutually cannot be seen as the existence of a central authority (Epstein, 2006), since the conditions for interacting are embedded in the agents.

3. *Explicit space*: As we live in the real world, our agents live in their *in silico* world. This latter can have several shapes[21] (grid, torus, etc.) that is composed of several patterns (such as food or intensity threshold). Typically, by living we mean interacting. To live or interact should happen in a defined (or explicit) space. Moreover, the interactions take place between neighbors, thus the neighborhood should be defined with cautiousness (Epstein, 2006). In the above section, 3.4.1, we have discussed more in depth one possible representation of this space above, namely cellular automata.

4. *Bounded rationality and local interactions*: Agent-based modeling is a very well-suited tool for the analysis of bounded rationality. Imagine a computational world built as a grid and composed of heterogeneous agents. We can assign a "view" to each agent; that is, the number of cells with which an agent can interact. Therefore the definition of the agent's vision (its neighborhood) is a way of expressing its bounded rationality (Epstein and Axtell, 1996). In the next chapter this will become even clearer. For now, a brief definition of the two most widely used neighborhoods is given, namely the Moore neighborhood, which is composed of eight closed neighbors, and the Von Neumann neighborhood, which is represented by cells at the four cardinal points. Agents with this kind of vision have a bounded rationality because their vision does not encircle the whole world. Moreover, these different neighborhoods imply

[21] For a brief description, see section 3.4.1

that the agents are local players rather than global ones; therefore, so is the information they can get. The view, or the ability to look for information – in other words, the bounded rationality – must be programmed; i.e., the agents also have "bounded computing power" (Epstein, 2006, 52). In other words, their capacity to evaluate their next move is based on the (finite) number of neighbors with whom they interact.

5. *Non-equilibrium dynamics*: What is important to study is not the equilibrium per se. The statistical methods usually used to analyze social phenomena focus on equilibrium states, and they lack the dynamic of the process (Miller and Page, 2007). The study of equilibria can be done with static models. Here the idea is to analyze the path that leads to a potential equilibrium. While equilibria do not always exist in a system under study, it is critical to understand the dynamics of the system, because it means that the model "produces complexity"[22] (Page, 2008, 133). The use of computational agent-based modeling is well suited for dynamic, heterogeneous, and (sometimes) non-equilibrating worlds (Epstein, 2006). The fact that a world can go toward an equilibrium is important, but not essential in the context of ABM. For example, the prisoner's dilemma as developed by Axelrod (1984) shows that cooperation is possible in an iterated prisoner's dilemma game. Moreover, Axelrod has shown that the TIT for TAT[23] strategy takes over the world. In this model the equilibrium is attained because each agent has the same strategy, but more important than the equilibrium is *how* this pattern emerges.

These five main characteristics can be seen as the main assumptions one should make before starting the development of one's own computational agent-based model. In the next section, we will try to answer another critical question; i.e., for what purpose is a computational agent base developed.

[22]See section 3.2.2 for more on complexity

[23]In the TIT FOR TAT strategy, an agent plays the same move as its opponent, but one step further. More precisely, if the opponent cooperates at time $t$, so does our agent at time $t$ and time $t+1$, even if the opponent defects at time $t+1$.

### 3.4.3 On the different uses of Agent-Based models

All the above explained assumptions create a framework for the development of computational ABMs. The following points enumerate the main uses of computational agent-based models (see e.g. Axelrod, 2003; Gilbert and Troitzsch, 2005), because they too have a significant influence on the design of the computational agent-based model:

1. *Prediction*: In physics, a good example of a predictive model is the one developed by Newton for explaining the elliptical path followed by the planets. This model allows us to calculate the exact position of the planets at any given time in the future. The characteristics of agents (here the planets) and their current behaviors are the main predictors for the future evolution of the model (Casti, 1997). In social sciences, due to the huge number of parameters that play a role at the same time, prediction is a very difficult goal to achieve. However, some tests have been made, notably in economics. For example, one can build a model that tries to predict the short-term evolution of the interest rate (Axelrod, 2003). Nevertheless, prediction remains a hard goal to achieve. Looking into the future using past evolution is not so easy: How can one be sure that past macro behaviors will continue in the future? For this particular task, ABMs help developing scenarios. This possibility is offered because it is easy to slightly change the parameters of the model to see how it behave. Following Axelrod (2003), it is worth noting that prediction is the first thing that comes to the minds of most people when they think of the use of computational models as a research-based tool, even if this use is not the more interesting one.

2. *Interdisciplinary social science*: Most concepts used in ABMs are derived from physics, evolutionary biology, but also economics and sociology[24]. Other examples are the rational vs. adaptive behavior an agent can have, or the diffusion

[24]See Section 3.1.

of innovation (Axelrod, 2005). In other words, the theoretical and technical developments of agent-based models come from a wide range of sciences

3. *To discover new relationships*: This is linked with the study of the *edge of chaos*, that is, the transition phase we explain when describing the concept of emergence in Section 3.3.3. When a system has "switched" from chaos to order, the explanation lies in the slight changes that alter the different parameters, the interactions between and the behaviors of the agents. These new relationships can also help strengthen existing theories or develop new ones.

4. *The existence of proof*: We have already shown how the game of life works. Such a simulation emphasizes the emergence of complex behaviors resulting from simple rules (Axelrod, 2003). In the same way, the segregation model shows that segregated neighborhoods can appear due to simple "thoughts" ("I want 30% of my neighborhood to be composed of neighbors who are like me"). Computational agent-based models are often used to demonstrate the existence of complex behavior arising from simple rules.

Most studied social phenomena need to be analyzed under the aspect of processes. To investigate and theorize about a social process; that is, to consider the evolution or the expansion of the system through time as well as the emergence that arises from the interactions that exist in that system and to decide the level of abstraction the researcher wants may be laborious (Gilbert, 1998). But, too often, simulation is seen as a tool whose purpose is to generate hypotheses that serve the development of theory. Nevertheless, simulation can be used to test theoretical hypotheses in an empirical way. The production of data, as we will see below, is an important advantage of simulation, because there is no missing data. We shall now turn to the evaluation of agent-based models.

### 3.4.4 Evaluating a computational agent-based model

Once the model has been expressed into computational language; that is, once the model has been programmed, the researcher should start to analyze the model; i.e., if the model has been well designed for its purpose (Gilbert, 2008). First, one should go through the program to see whether it has been coded well. Then the simulation should be validated[25]. Two main types of validation can be used: the internal validity; that is, the evaluation of the correct implementation of the model and the external validity; i.e., the comparison between *in silico* and *in vivo* data (Gilbert, 1998). Of course, the first test of a computational model is to run it and see if it produces what it is intended to do.

- *Debugging*: The debugging of a model consists in observing the simulation step by step by placing breakpoints at judicious lines in the program, at the beginning of methods for example. When running the program, the programmer can see the behavior of each parameter or variable one line after another. Debugging increases the knowledge of the internal logic of the program, and the behavior of the program. Therefore, debugging is a synonym of *verification* (Gilbert, 1998). This is an important step for the validation of the model because, as noted by Gilbert (2008), it is unlikely that the first run of a new computational model will be free of bugs. Usually these bugs are the easiest to fix, because when using modern integrated development environments (IDEs[26]), such errors are returned in the console and easily accessible, as in the Eclipse™IDE used to develop the model of diffusion. But the problem still remains. Once the easiest bugs are fixed, one cannot be sure that there are not some left, hidden somewhere deep in the logic of the program. Even when the simulation seems to work well (in other words, the output seems to express what one wants), one should get one's hands dirty and go deeper into the debugging to be sure of the logic of the simulation, because the more com-

[25] Here we are in the central hexagon of the figure 3.1

[26] What an IDE is will be explained in the next chapter. For now you only need to know that it is the interface for writing codes, such as Eclipse.

plex the simulation, the more probable the existence of hidden bugs (Gilbert, 1998). As a consequence, the model may produce results that do not correspond to its description, but to some undesirable influence of some hidden bug (Gilbert, 1998).

While coding, it is important to add comments throughout the program in order to define what the different parameters are and how the different methods (or functions) should behave. The logic of the "thoughts" can be compared, while debugging, to the logic of the simulation run.

- *Internal validity*: Internal validity is defined as a correct implementation of the theoretical model (Axelrod, 2003). This is strongly linked with debugging because, as noticed above, debugging helps the researcher to improve his knowledge of the internal logic of the model.

  To run the model with different random seeds and extreme values to see how it behaves is a good way to test the internal validity of the model because it allows one to evaluate not only the strength of the theory, but also if the model has been well designed for its purpose; that is, if the process under study can be "generated by its underlying assumptions" (Repenning, 2002, 114). This is closely linked with the exploration of the model that we will see below.

- *External validity*: External validity can be defined as "the relation of the model to the empirical reality" (Elkink, 2009, 14). The building of a model is always based on the observation of the real world. When it has been programmed, one wants to see if the model really explained the real world observation under study. The difficulty comes from the embedded nonlinearities[27] that characterize complex adaptive systems, such as agent-based models. In such systems, a small change in the initial condition can lead to totally different outcomes; the use of standard statistical methods can be very hard, but not impossible. Indeed, this type of validation can show the plausibility of the results of a model.

[27]See sections 3.2.2 and 3.3.

This is a first introduction of the different validity tests that can be applied to ABMs. In the next chapters, this knowledge will be deepened with the example of the development of the policy diffusion model. In the following subsection, we will explore the strengths and weaknesses of ABMs.

### 3.4.5 Advantages and weaknesses

A model is an abstraction of reality. A good model does not need to catch the entire complexity of the real world, even for the phenomenon under study (Casti, 1997). The *art* of the researcher lies in the choice of the features of the phenomenon he wants to model. These features must capture the essence of the studied process. If we keep in mind that the interpretation (the semantic as seen in Section 3.3.1) of the computational values is what gives its content to the model, a good model should capture the juice of the phenomenon under study (Casti, 1997). In other words, the (computational) variables used in the building of the model are relevant enough for the asking of interesting questions and, furthermore, for supplying interesting responses to these questions. Abstracting some part of the reality and testing it with a computational model has its advantages. Of course, computational modeling also has weaknesses. These are often the same as other methodologies. The building of computational model is, as was just said, based on an abstraction of the real world and the value of this abstraction depends on the artistic qualities of the researcher. In other words, the researcher builds a model using parameters that allow him to validate the tested theory (de Marchi, 2005).

#### Advantages

1. *The history of the model*: This can be resumed in two words: "History matters!" This is the case not only in describing the evolution of the agents and of the simulation because agents in ABMs not only interact with one another but they also interact with their environment. This interesting property of agent-based models is called "stigmergy," which is defined as "a form of indi-

rect communication between agents that is achieved by agents modifying their environment and also responding to these modifications, for example ants following pheromone trails left by other ants" (Goldstone and Janssen, 2005, 425) and, by so doing, strengthening or modifying the quickest path to food, for example. This notion of stigmergy is of great interest for our purpose. Indeed, diffusion follows *direction(s)* built by the several mechanisms we have described in Chapter 2 (which play the role of pheromone trails in the ants' case). Thus, the behavior of the agents; i.e., the interactions, is influenced by several mechanisms of diffusion, and, as a result, the environment is modified and the way the environment evolves influences the interactions between the agents by modifying which mechanism plays a role. According to this notion, an agent can be seen in two different, but complementary, ways: An agent can be thought as "a collection of preferences, abilities, and information" (Page, 1999, 37), or as a set of historical experience. Moreover, as the behaviors of the agents also modify their environment, not only is the history of agents important for its evolution, but, according to the *second-order emergence concept*, also the "behavior" of the environment (or the world).

2. *The exploration of the model*: One of the great advantage of programming is its flexibility. While programming, the researcher can give standard values to different parameters and these values can be modified throughout different simulation runs. Each simulation run can be seen as a laboratory in which every parameter configuration can be tested (*internal validity*) and "judged in a disciplined empirical way" (Epstein, 2006, 114); that is, the *external validity.* Therefore, a good way to explore the model and to test its strength is to investigate it with extreme values. Moreover, the researcher must experiment with the model by adding "new features to the computational representation" (Davis, Eisenhardt and Bingham, 2007, 493). To program more characteristics allows the model to get closer to the real world. By letting these new properties interact, the researcher may see new emergent patterns. However, one

should be careful before adding too many characteristics. Getting closer to reality hardens the interpretation of the model and, thus, weakens the results. The researcher should spend a lot of time "playing" with the model or experimenting with it, because the possibilities for experimentation are quite infinite and adding new conditions is an easy task that can be achieved simply by modifying the software code. This leads us to the next advantage: there is no missing data.

3. *The problem of missing data*: With standard empirical research, it is common that variables are missing. No such problem happens in computational modeling. Each step of the simulation run produces the model data. Not only does the model produce huge amounts of data, but the data set is complete. One of the great advantages of this kind of methodology is that if one needs more data, one can just run the simulation as many times as necessary to have the required number of data (Axelrod, 2003).

4. *Repeatability and recoverability*: History has only one way. It goes from the past to the future and what happened in the past will not happen again (for example, there is absolutely no possibility for any human being to rediscover America). The situation is different in an *in silico* world, because not only can the history be rerun again and again, but the initial conditions can be changed, as well as the different parameters of the model (Epstein, 2006). The results of such a systematic exploration is that, when using agent-based models, we can follow many historical evolutions and try to answer the "what if" questions. In sum, the use of a computational model allows the researcher to run the history again and again (Page, 1999).

   To rerun history is what we do to explore the model. Rerunning the same experiment with only slight parameter changes is something that cannot be done in the real world, but is necessary to fully understand the processes at work. Also, this is of great importance for the internal validity of the model because rerunning "a problem gives enough data to validate the model, but one should change the parameters, to see if the history we first see is idiosyncratic or typical" (Axelrod, 2003, 8).

5. *The flexibility of the model*: While developing a computational model, the researcher is creating a brand new world. Plus, he has total control over this world and the different parameters used to build it. Therefore, the modeler can be seen as some kind of *deus ex machina*. Following this, we can divide the flexibility into *outer* flexibility (the one the researcher can use to test different configurations) and *inner* flexibility, which corresponds to the behavior of the agents. In this second assertion, the agents are the central units of the model: they can learn, interact with other units, and are subject to historical change. In other words, and as already mentioned, they are complex agents (Benoît, 2001, 14).

6. *The need for precision*: To construct a computational program, one need to be precise. Therefore social theories can be formalized without any ambiguities expressed as source code (Amblard, 2003). The problem arising, in this case, is the trade-off between flexibility and precision. One big strength of computational modeling is precisely to overcome this trade-off, because programming is a very flexible way of encapsulating different behaviors. Nearly everything can be simulated. The difficulty is to stay focus on the studied subject. This is achieved when the assumptions of the model are succinctly and clearly expressed in the program. The program is just another way of mathematically expressing the assumptions of the model. That is why it contains all the relevant information on the assumptions of the model (Miller and Page, 2007). Therefore, not only are computational agent-based models characterized by the above-explained inner and outer flexibility, but they moreover need a high level of precision. The computational transformation of the assumptions is only a way of mathematically expressing them so that the computer can read them. We already explained the fact that the program is no less precise than a mathematical equation.

### Weaknesses

1. *Ad hoc assumptions*: Much of the interactions between the agents are based on what has been called, according to Granovetter (1978), *threshold models.* A threshold is defined as beneficial "the point where the perceived benefits to an individual of doing the thing in question ... exceed the perceived costs" (Granovetter, 1978, 1421). For example, in the model of segregation developed by Schelling (1978), the threshold defines the point at which it becomes beneficial for an agent to move to another neighborhood. The result of such a way of building models is that each agent has the same conditions of change. Therefore, the hypothesis of homogeneous behaviors is realized in computational models. More precisely, heterogeneous agents behave homogeneously because they all face the same condition(s) of change. The agents' heterogeneity and the possibility offered by computational models to make a lot of different experiment with different values help the researcher partially overcome this weakness. In other words, repeatability, recoverability and the flexibility of models counterbalance this assumption problem.

2. *Fragility of results*: If there are too many phenomena under study, the results may be hard to interpret and not correspond to what the researcher wants to explain. The modeler, as the creator of the model, has total control over the different parameters. Therefore their choice and their parameterization influence the behavior of the model. The researcher must balance between the needed abstraction of the reality (or the choice of the different parameters that convey this reality) and the adequacy of the model for the explanation of the phenomenon under study.
   Again, it is important to note that the programmer must be careful and precise while constructing the simulation. Thus, one aspect that should be noted here is that bugs could be the source of this problem because, as noted by Gilbert (2008), bugs alter the behavior of the program and, thus, the produced results. With debugging, the number of bugs follow a decreasing curve that

never reaches zero. If bugs still remain it is hard to be 100% sure about the results. Of course, techniques for reducing the number of bugs exist, and as noted above, the different possibilities for validating the simulation can reduce the importance of this weakness.

Another common objection to computational modeling is that the results are embedded in the program and "thus we can never learn anything new from these techniques" (Miller and Page, 2007, 69). If the second part is false according to the emergence of often-unexpected phenomena, the first one can be true. Nonetheless, every model has built-in features, because of its underlying assumptions. The computer will follow its predetermined program. A mathematical model will follow its predetermined equations.

A third argument often cited here is linked with emergence and the problem of the transition phase. The results of the interactions in a complex system are often unexpected phenomena, and a slight change in the initial conditions can dramatically change the results (Miller and Page, 2007). But this is part of the test for the internal validity. To test the model, one should change the parameters and see if the results are in accordance with the expected outputs.

3. *The problem of the simulation time*: The definition of time is something difficult in the case of computational agent-based models, especially in social systems. It is difficult to make hypotheses on time in the real world (who the next adopter of a policy reform will be, for example) (Amblard, 2003). The problem of the definition of a simulation time is often resolved by using a discrete measure of the time; that is, the evolution of time is represented by a step function because, as the variable representing the time is defined as an integer, the different states of the system are determined for each integer value (Amblard, 2003).

4. *The external validity*: Testing external validity could be of no interest in the case of agent-based models for the reason that, when comparing *real-world* empirical data with that of a simulation, there is a problem of perception.

> The world we see in reality is clearly different from a computational world. Even empirical data only captures a small part of the observed *real world* event. In other words, it is not comparing a *real world* process and its *in silico* results; it is comparing "*what you observe as the real world* with what you observe as the output" (Ahrweiler and Gilbert, 2005, 5). In other words, we consider the model as the *real* world and, with this in mind, test it with the standard statistical methods. Nevertheless, if we compare the results of the empirical tests of the actual *real* world and of the *virtual* real world, we can then have a good idea of the *truth* of the model.

To avoid these weaknesses, the researcher must be rigorous in the conceptualization of the model as well in the development of the model. The different tools for the debugging of the program are, thus, of great importance.

Now that we have a rather large view of computational agent-based models, we will see some examples that have been developed in the field of social sciences, and, more specifically, several examples of agent-based models that analyze diffusion processes in the social sciences.

### 3.4.6 Agent-based modeling and diffusion

In political science, computational models have been used principally in two research fields: international relations (IR), and elections and voting. In the IR literature, agent-based models have been employed especially to study conflicts, both at the international and at the subnational level (see e.g. Lustick, Miodownik and Eidelson, 2004; Cederman, 2002, 2003) while, for the study of elections, they have been used to investigate partisan convergence (Kollman, Miller and Page, 1992), or the relationship between citizens and institutions (Kollman, Miller and Page, 1997).

International relations, by definition, are characterized by interdependencies between countries. One consequence is that social phenomena may spread internationally. Rousseau and van der Veen (2005) have used an agent-based model to study the process leading to the emergence of a shared identity at the international level.

Their model includes two types of agents, leaders and followers, which are defined by their repertoires (the possible identities); their trait values (the characteristics of each repertoire; for example, the repertoire "Religion" can be characterized by the trait values "Catholic," "Jew," "Muslim," etc.); repertoire salience (which repertoire is considered more important); and global bias (that is, which identities are socially more valued). In each iteration the values of these parameters are updated, and the emergence of a shared identity occurs under certain specific conditions. The model shows the probability that a shared identity emerges increases as the size of the repertoire decreases, as the range of the global bias increases, and when leaders are less powerful. For the study of policy diffusion, this research is relevant because it highlights the conditions under which common norms can emerge. As shown in Section 2.5, a policy change can be driven by the imitation of common accepted norms. Also Miodownik (2006) investigates the importance of the collective identity in the emergence of autonomous movements, such as those existing in the Basque Country or in Corsica, for example.

In the IR field, computational agent-based models are also used to study more general patterns, such as the democratization. Based on the observation of the wave of democratization after the Soviet Union collapsed and the idea that implementing democracy in Iraq will lead to a democratic spread in the Middle East, Elkink (2006*a*, 2009) explores the diffusion of democratization and tries to highlight the conditions under which the diffusion of democracy is more likely to occur. Here, again, the justification for the use of an agent-based model is the fact that it allows one to establish a link between the micro level (the individual political life; i.e., voting, debating, protesting, etc.) and macro level (geographical patterns of democratization) patterns (Elkink, 2006*a*, 2009). According to this study, it seems that citizens must have a bias toward democracy; i.e., a preference for democracy, and that an exogenous shock must occur for a wave of democratization to be "launched." Cederman and Gleditsch (2004) also explore the diffusion of democracy. Based on a macro-historical process and starting from a statistical analysis that shows that democracy has spread in waves, they investigate the processes that exist behind these waves through an agent-based model that links micro- and macro-level processes. They postulate what can be called a bandwagon pressure: the more democratic states that surround a nondemocratic state, the higher the probability that the latter will become a democracy. Their model highlights several mechanisms needed for democratization to emerge, especially a collective security mechanism; i.e., a cooperative defense arrangement. The result that emerges from this mechanism is the apparition of "democratic clusters" that help protect democracies in a nondemocratic environment.

Agent-based models have been used to study diffusion not only in political science but also in other fields such as (organizational) economics. Based on threshold models (Granovetter, 1978; Abrahamson and Rosenkopf, 1993), Abrahamson and Rosenkopf (1997) developed a model of the diffusion of innovations that stressed the existence of a bandwagon process and how the structure of social networks influences this process. The basic question is: Why do certain innovations diffuse and eventually become taken for granted, whereas others do not? Their central argument is that "social-network effects must be incorporated into theories that explain when and to what extent innovations diffuse" (Abrahamson and Rosenkopf, 1997, 290). The model works as a self-reinforcing process; that is, the more adopters, the more information is available; thus, the stronger bandwagon pressure, and the stronger bandwagon pressure, the greater the incentive for a change (Abrahamson and Rosenkopf, 1997). Organizations, before adopting an innovation, fix their threshold by assessing their potential profits (losses) from the innovation. Since they are uncertain about the future, the process of diffusion is path dependent; that is, the former adopters have an influence on the development of the process because they influence the behavior of later adopters (Abrahamson and Rosenkopf, 1997). The model also includes a "network dimension." The network is composed of a strong center that is linked to a weak periphery, and the position an adopter has in the network determines the information he receives (Abrahamson and Rosenkopf, 1997). The more a potential adopter communicates with others, the greater bandwagon pressure. The simulation results show that social networks are important variables for explaining diffusion, because they work as communication channels – here learning is taken into acount. However, as we show in Section 2.5, several diffusion mechanisms exist. In a later article Rosenkopf and Abrahamson (1999) integrate the reputation of the adopters and informational influences in the model. They conclude that both factors influence the diffusion of innovations in a significant way, which explains why an innovation is still adopted even when some adopters have had an unsatisfactory experience with it.

Bullnheimer, Dawid and Zeller (1998) study the diffusion of innovations using only a learning process. Their study shows that learning by imitation has a positive effect not only on production efficiency but also on profits of firms. Looking at available information, firms iteratively choose the competitive technology that suits them best. By doing so, firms enter into an adaptive learning process.

Deffuant, Huet and Amblard (2005) also propose a threshold model to study the diffusion of innovations. They build a general model that takes into account both social value and individual payoffs. Their model works like Abrahamson and Rosenkopf's. The more adopters of the innovation, the greater the pressure is for adoption. They develop a model that stands at the crossroad of the cognitive agent approach and the physics-inspired model of cellular automata[28]. After developing the main parameters of the model and the comportment of the agents, they perform several runs and study the results by observing "the average final number of adopters over several runs for different values of the main parameters—in particular, the definition of the a priori distribution of social values and the function of individual benefit evaluation" (Deffuant, Huet and Amblard, 2005, 1042). Their main results show that adopters take up easily innovations about which they are best informed, that a minority can block or, on the contrary, force the adoption of an innovation, and that uncertainty is negatively correlated to the level of adoption. Here we see that

[28]See section 3.4.1.

they also introduce an uncertainty term in their model that can be considered close to the ambiguity term of the above model. In the end these two models are not so different from one another.

In his model of the dissemination of culture, Axelrod (1997*b*) investigates why differences between agents still persist despite the fact that "people tend to become more alike (...) when they interact" (Axelrod, 1997*b*, 203). In order to respond to this interrogation, In order to respond to this interrogation, Axelrod has developed an agent-based model, because the existing explanations of the differences did not take into account one of the most important patterns in social life; i.e., communication. "The model of social influence offered here abstracts this fundamental principle to say that communication is most effective between similar people" (Axelrod, 1997*b*, 205). This is a common feature of all the models we have described. Here culture is defined as a list of cultural characteristics, such as religion and language. The probability that two agents interact increases with their similarity, and is proportional to the number of cultural features they have in common. The logical conclusion of this simulation should be convergence, because multiplying interactions with similar neighbors increases the similarity, as underlined by the bandwagon theory. So, at the end of the process, all agents should be similar. A first result of the models is that local convergence is compatible with global polarization.

A quick look at all these examples show that they contain the principal characteristics of agent-based computations. For example the Axelrod's model of dissemination of culture is composed of heterogeneous and autonomous agents that interact on simple rules of interaction. And the emerging result is quite unexpected, as it will be deepened in the next section.

### 3.4.7 In-depth: Axelrod's dissemination of culture model

We briefly expose in the above subsection the main arguments and conclusions of one of the leading agent-based example in the literature; namely Axelrod's dissemination of culture model (Axelrod, 1997*b*). In the present subsection, I will explore this model more deeply in order to highlight the construction of an agent-based model.

As already mentioned, this model deals with the question of the differences that persist in a "convergent" world. In other words, if one assumes that people tend to converge in their attitudes or beliefs when they interact, one must find an explanation of the persisting divergences between these agents. For example, Axelrod cites the *state formation* as processes partly driven by the share of common habits and languages. If some shared features help creating a state, regional differences still exist. In other words, common habits are a necessary but not sufficient condition for convergence. Therefore, divergence still persists.

Axelrod starts by asking what the most generic term that specifies the influence between people is. For him, this influence is best expressed by the term *culture*. Central to this idea of dissemination is the principle of human communication, which is "a process in which participants create and share information with one another in order to reach a mutual understanding" (Rogers, 2003, 5). This exchange of ideas that characterized the dissemination process is more frequent among people who share some common *features*, such as beliefs, education, and the like, as already mentioned.

In order to be able to computationally define his idea of culture, Axelrod has made some simple assumptions about what "builds" culture. More precisely, culture is assumed to have two basic properties: people communicate more with people who share most of their cultural traits, and the cultural distance between two people tends to decrease as they interact (Axelrod, 1997*b*). After exploring several models of diffusion in different fields (political science, anthropology, sociology, biology), Axelrod highlights the two principal weaknesses of the research, which are the interdependence between the different cultural features of an actor and the impact of the agents' similarity. To overcome these weaknesses, he proposes the development of an agent-based model.

First of all, *culture* must be computationally defined. This is done by defining the

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 47915 | 07982 | 77785 | 21612 | 47150 | 89321 | 63528 | 47793 | 03741 | 82574 |
| 10748 | 88936 | 01313 | 59316 | 47445 | 90082 | 27753 | 42657 | 01255 | 93320 |
| 70954 | 22446 | 31201 | 01180 | 20638 | 28356 | 42940 | 88786 | 86066 | 98070 |
| 06865 | 00013 | 97137 | 67556 | 37096 | 77500 | 17083 | 74593 | 60482 | 00049 |
| 89650 | 09313 | 67959 | 30446 | 01151 | 84366 | 10378 | 53515 | 16401 | 63722 |
| 54764 | 86218 | 00954 | 22845 | 62902 | 49985 | 77417 | 43254 | 33649 | 10579 |
| 10956 | 52610 | 68968 | 91660 | 09199 | 99174 | 89339 | 30968 | 21230 | 29734 |
| 07114 | 30073 | 40666 | 29350 | 80645 | 11890 | 65514 | 48965 | 45395 | 14394 |
| 69761 | 53743 | 77800 | 02737 | 71448 | 93604 | 40796 | 72326 | 88180 | 08077 |
| 58839 | 87747 | 62945 | 19469 | 40766 | 83282 | 68810 | 78511 | 73375 | 50563 |

Table 3.1: A typical initial set of cultures

culture of an agent as "a list of *features* or dimensions of culture. For each feature there is a set of traits, which are the alternative values the features may have" (Axelrod, 1997*b*, 208). If we suppose that five features compose a culture, this latter can be formally expressed as follows:

62971

In other words, this list of five digits represents the culture of an agent and each number represents the proportion of (or the degree of) the value a feature can have. An example will make that clearer.

Suppose that the culture of an agent is composed of the following five features: religion, health, political "orientation," wealth, and education. For example, the religious feature of an agent can be indexed from *atheist* (0) to *orthodox* (10), or the wealth can range from *poor* (0) to *rich* (10). Therefore, we can translate this five-digit list into an agent's culture. As an example, take the underlined agent in the northwest corner of Table 3.1. This agent is religious, not very healthy, close to the extreme right movement, quite rich (upper middle class), but has only finished middle school. As noted by Axelrod (1997*b*, 2018), "[this] formulation allows one to define the degree of cultural similarity between two individuals as the percentage of their features that have the identical traits." For example, in the table 3.1, the underlined agent (47915) and its right neighbor (07982) have 40% cultural similarity (2 on 5 traits are similar).

In this model, the world (or the territory in Axelrod's word) is a 10X10 grid, populated with 100 agents. Each agent has a cultural "identity" defined by 5 features, and takes its Von Neumann neighborhood into account; that is, the neighbors that are at the four cardinal points. Moreover, the world is bounded; that is, the agents in the corners have only two neighbors and the ones forming the border of the world have three neighbors. Table 3.1 shows a typical set of initial cultures. Now that the model has been described, we need to explain the different interactions between the agents. This process of interaction corresponds to a series of repeated steps that Axelrod (1997*b*, 208) expresses as follow:

- "Step 1. At random, pick a site to be active, and pick one of its neighbors.
- Step 2. With a probability equal to their cultural similarity, these two sites interact. An interaction consists of selecting at random a feature on which the active site and its neighbors differ (if there is one), and changing the active site's trait on this feature to the neighbor's trait on this feature."

As an example, take again the underlined agent in Table 3.1. It has a 40% percent chance to interact with its right neighbor. If they do interact, the will take one of the three different features of its neighbor and replace it in its own culture. Suppose that the first feature is the active site's trait; then the underlined agent's first trait will become 0. Thus its culture is now 07915 and their cultural similarity has increased to 60%, "making it even easier for them to converge still further" (Axelrod, 1997*b*, 209).

Intuitively, these interactions should lead to the convergence in this world and the emergence of one cultural region. Surprisingly the results of this model show the emergence of a few number of stable regions.

In a second step, and in order to explore and validate the model, the main parameters (the number of features, the size of the world, and the number of neighbors) are modified. The main results are summarized as follows:

- *Changes in the number of features and traits*: To construct Table 3.1, the culture is composed of five features, each chosen from 10 possible traits. Axelrod allows the scope of features and traits to vary between 5 and 15, and he tests the model with the values of 5, 10, and 15 features and traits, which gives 9 possible combinations to test the model.

If the number of features (and/or possible traits per feature) is increased, that is if the culture is more diverse, the odds of interactions between the agents should increase and the number of stable regions should also increase. In other words, the increasing cultural diversity should lead to a decreasing convergence. In this model, increasing the number of features leads to an increasing convergence, up to the reach of one stable region. If we increase the number of traits, the result has the opposite effect. In other words, increasing the number of traits leads to an increasing number of stable regions: "Having more features (i.e., dimensions) in the culture actually makes for fewer stable regions, but having more alternatives on each feature makes for more stable regions" (Axelrod, 1997*b*, 213).

- *Changes in the size of the world*: At the beginning, 10X10 sites compose the size of the territory. Surprisingly, the size of the territory has no significant impact. Therefore, a more detailed analysis of the effect of the size is done. By only varying the size of the territory, all things being equal, Axelrod shows that the number of stable regions increases as the size of the territory increases, but, surprisingly, as the size of the territory is still increasing, after reaching a peak, the number of stable regions declines.

- *Changes in the number of neighbors*: We explained above that an agent is interacting with its Von Neumann neighbors. The number of neighbors is then increased by also interacting with the Moore neighbors (the eight adjacent agents) and by mixing these two kinds of neighborhoods (that is, 12 neighbors). The result is that "larger neighborhoods result in fewer stable regions. (...). Thus, when interactions can occur at greater distances, cultural convergence is easier" (Axelrod, 1997*b*, 213).

To conclude, this model shows three important results (Axelrod, 1997*b*, 223):

1. "Local convergence can lead to global polarization.
2. The interplay between the different features of culture can shape the process of social influence.
3. Even simple mechanisms of change can give counterintuitive results in which large territories generate surprisingly little polarization."

A computational agent-based model composed of autonomous and heterogeneous agents and that follows a simple rule that can be summed up in a single sentence – "with probability equal to their cultural similarity, a randomly chosen site will adopt one of the cultural features of a randomly chosen neighbor" (Axelrod, 1997*b*, 208) – can lead to the emergence of surprising results.

Mixed with a threshold model, this way of building a computational model can give important insights for the development of computational agent-based models. Starting from a simple question on the possible results of the different interactions between people, Axelrod has researched a general theoretical concept that describes at best this communication process: culture. Based on simple assumptions on the role of this concept, his model showed important counterintuitive results.

## 3.5 Conclusion

The aim of this chapter was to provide an overview of the main theoretical bases that form computational agent-based models and to give a better understanding of this methodology as a powerful tool for the study of social phenomena. We also develop arguments in favor of a broader application of such a method in political science.

Despite several weaknesses and problems – especially the fact that computational agent-based models cannot capture the full history, because the *real-world* process under study also corresponds to part of reality – this kind of methodology is a critical tool, not only in theory development by helping us experimenting unusual hypotheses, but also in scientific progress because of repeatability and recoverability of the results, among other things (Epstein, 2006). Moreover, it may now be clear from Section 3.3 that simulation is a scientific tool that is build partly with the help of deduction – because of the recursion, and partly with the help of induction – for the choice and the development of the conditions of change. Thus, this combination should help and allow the researcher to develop precise and well structured computational models, because "the simulated data comes from a rigorously specified set of rules rather than direct measurement of the real world" (Axelrod, 2003, 5). In other words, "simulation is a third way of doing science" (Axelrod, 2003, 5).

Computational modeling as a scientific field is lacking some important features, the most important one is the lack of standardization of programming tools (Axelrod, 2005). But efforts for more standardization are made in this direction. For example, new toolkits for ABM are now developed using Java™, an object-oriented programming[29] language, and the toroidal shape of the world is now widely used.

Going through this chapter allows us to draw the 4 main assumptions of ABMs upon which scholars have reached a consensus (see e.g. Macy and Willer, 2002; Axelrod, 1997*a*, 2003):

1. *Agents are autonomous*: There is no central authority

2. *Agents are interdependent*: The action of an agent has consequences on the behaviors of the other.

3. *Agents follow simple rules*: Even if they follow simple rules, their interactions create behaviors that are far from simple. The chess game (Zwirn, 2006, 28) is a good example: Each piece follows a simple rule on the chessboard, but their moves create an infinity of possible games.

4. *Agents are adaptive and backward-looking*: They follow the IF/THEN rule. This is an action reaction rule. It works as follows: IF a certain condition is

[29]Object-oriented programing is well suited for the development of agent-based model as we will explain in the next chapter

> true/false, THEN execute the defined action: "IF [] THEN [] clauses are the heart of the flexible, conditional responses that give digital computers their tremendous power" (Holland, 1998, 223).

Now that we have a broad theoretical view of how a computational agent-based model is developed, we will link, in the next chapter, the theoretical aspects of diffusion and those of computational agent-based modeling by developing our own computational model of policy diffusion.

# Chapter 4

# A computational model of policy diffusion

## 4.1 Introduction

Throughout Chapter 2, we developed a theoretical framework of policy diffusion based on several internal factors (Section 2.4) and on different interactions, which exist between countries and are expressed by several mechanisms of diffusion (Section 2.5) in order to explain how a process of diffusion can occur (Section 2.6). The fact that, in this world, heterogeneous agents interact according to these different mechanisms means that diffusion is a nonlinear process, and, thus, that this process evolves in a complex system.

In a second step, we highlight the theoretical underpinning of computational agent-based modeling, with a particular emphasis on the two main concept that drives the micro/macro links, namely complexity and emergence. Therefore, based upon Chapter 3, it should now be clear that the best way to study complex systems is computational agent-based modeling, since it allows the researcher to take into account the complexity (Section 3.2.2) of the world and the resulting nonlinearity of the interactions (Section 3.3.4).

The aim of the present chapter is to combine what we have learned, up to now, in order to develop our own computational agent-based model (ABM). The help of computers for developing ABM implies some basic knowledge derived from computer science, which is highlighted in Section 4.2.

The process of diffusion occurs between countries in an explicit space (the world). Hence, we need to explain how the country (and its parameters) and the world are computationally defined and operationalized (Section 4.3). Moreover, we need to define the different phases a country has to go through in order to change its policy (Section 4.4), method by method. This is an important step, since it shows the logic of the computational implementation of the model.

Let us start by outlining the different technical tools needed. This first section is a bit technical, but is necessary to fully understand the development and implementation of the model.

## 4.2 The methodological tools

This section is devoted to a brief presentation of the different tools involved in the development and the execution of a computational agent-based model. As this methodology is not well known in the political science community, a better comprehension of at least the programming language; i.e., Java, the Integrated Development Environment (IDE) (Eclipse), and the toolbox (RePast) is needed. As the purpose of this thesis is political science and not computational science, we will not go too deep into the technical explanations[1].

### 4.2.1 Object-oriented programming and Java™

To be able to communicate in a foreign country, you often need to learn a new language. This is also the case with a computer. To communicate with it and ask it to execute the required tasks, you have to acquire the basics of a programming language. Thus, a model with the help of Java as a programming language is

[1] See Appendix A for deeper information on these tools.

developed, since it takes full advantage of object-oriented programming.

### Object-oriented programing

The general idea of object-oriented programming (OOP) is that *everything is an object*, meaning that each part of the program can be constructed as an object. To make that clearer, we can rely on real world examples.

In the real world, we are surrounded with objects, such as cars, televisions, and so forth. For instance, a car can perform a number of tasks, such as accelerate, brake, and/or turn and it is made of other objects, like wheels, seats, tires, windows, and so on. All these objects are assembled and interact to help perform the different tasks a car is supposed to do. If we go a step further and take the distinct object "seat," for instance, it is also made of several objects and can perform a certain number of tasks, and the same reflection can be applied to the other constituents of the car.

Moreover, the car explained above is only a car in general, an abstract car that exists solely in the imagination. A brand-X car that is parked in the street, however, is a real car. This real car is an *instance* of the car, an example that we can describe with more details – its brand, its exact color, the size of the wheels, the number of doors, its maximum speed, and so on. OOP works in exactly the same manner.

This example shows us the difference between a class and an object: "Car" corresponds to a class; "your car," or the car parked in the street, to an instance of that class. "Car" is just an abstraction, a mental representation; "my car" is a real car that I can drive, that accelerates or brakes. In other words, an object is an example of a class that has been precisely defined with useful parameters. More precisely, and to use object-oriented termination, an object is an *instance* of a class. Furthermore, the main mechanisms that characterize object-oriented programming – as well as Java, since it is an object-oriented language – that are useful for us, since they are used with RePast, are inheritance because a child class automatically has the methods of a parent class and polymorphism [2] that is the possibility to rewrite

[2]See appendix A for more information on the different mechanisms of OOP.

or *override* the methods of a class

### Java

Java has proven to be well suited for the development of Web applications and, as its use increases, more and more programmers find this language interesting for other applications. Therefore, Java has several advantages – portability, which means that Java works on all platforms; speed, which represents the fact the compilation time is rather short; and security – that made it so interesting for other uses than solely the development of Web applets, as shown in Figure 4.2.1.

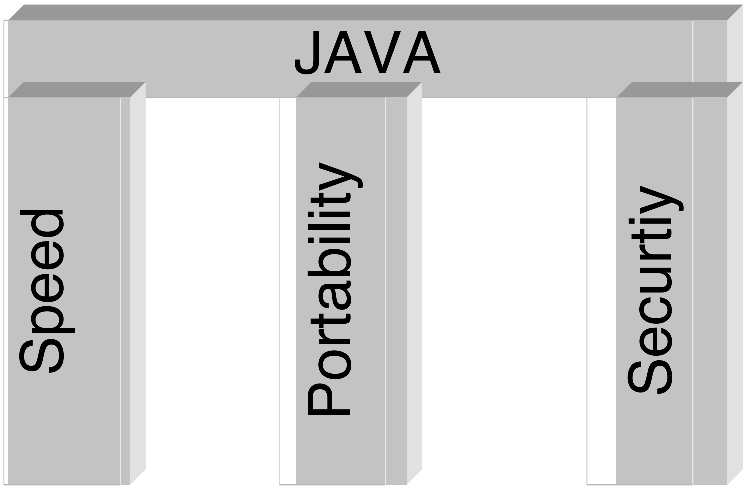



Figure 4.1: The strengths of Java$^{TM}$(Niemeyer and Jonathan, 2005, 7)

Java comes with large libraries of already defined classes. Moreover, as Java is not a spoken idiom, but a write-only language, we need to have some kind of an exercise book to develop a program. More precisely, *Integrated Development Environments*, or IDEs, are used, which basically correspond to programs used to develop software. There are several IDEs that exist; such as, for example, NetBeans, which was developed at Sun Microsystems, or JEdit. For our purposes, Eclipse has been chosen.

### 4.2.2 Eclipse

Eclipse, as an IDE, can handle several programming languages, such as C++, C# or Python, but is attached to Java. The fact that Eclipse is so widely used comes from its several advantages:

- It is a free open-source IDE downloadable from the site http://www.eclipse.org/;
- as it is Java-based, it is well-suited for programming in that language;
- and one of its great instrument is the programming assistant, which gives several possible solutions when the programer is facing a problem.

The Eclipse environment is composed of several windows or *views*. Here will be briefly presented the most important[3]: Usually, on the left, a Navigator window shows the hierarchy of projects and classes that have been developed; in the middle, the Editor window is the place where the code is written; and in the bottom of the perspective, you have the console where the results and / or the errors of the compilation are shown. Figure 4.2 shows an example of the Eclipse environment and its different windows.
After this presentation of the exercise book, we will emphasize the fact that some chapters have already been filled with an existing Java-based toolbox, which helps the development of computational agent-based models, namely RePast.

### 4.2.3 RePastJ_3.1[4] as an agent-based toolkit

We have seen in Section 3.2.2 that agent-based models can be developed without the help of computers; but, because they are facing an increasing complexity, the different behaviors cannot be studied by hand, so that the help of computers is needed to overcome this complexity. Therefore, the **Re**cursive **P**orous **A**gent **S**imulation

[3]As the Eclipse workspace can be customized according to the will and needs of the programmer, this presentation is of course subjective and corresponds to the needs of the author. Nevertheless, this view shows the main useful windows.
[4]RePastJ is freely downloadable at http://repast.sourceforge.net/repast_3/index.html.

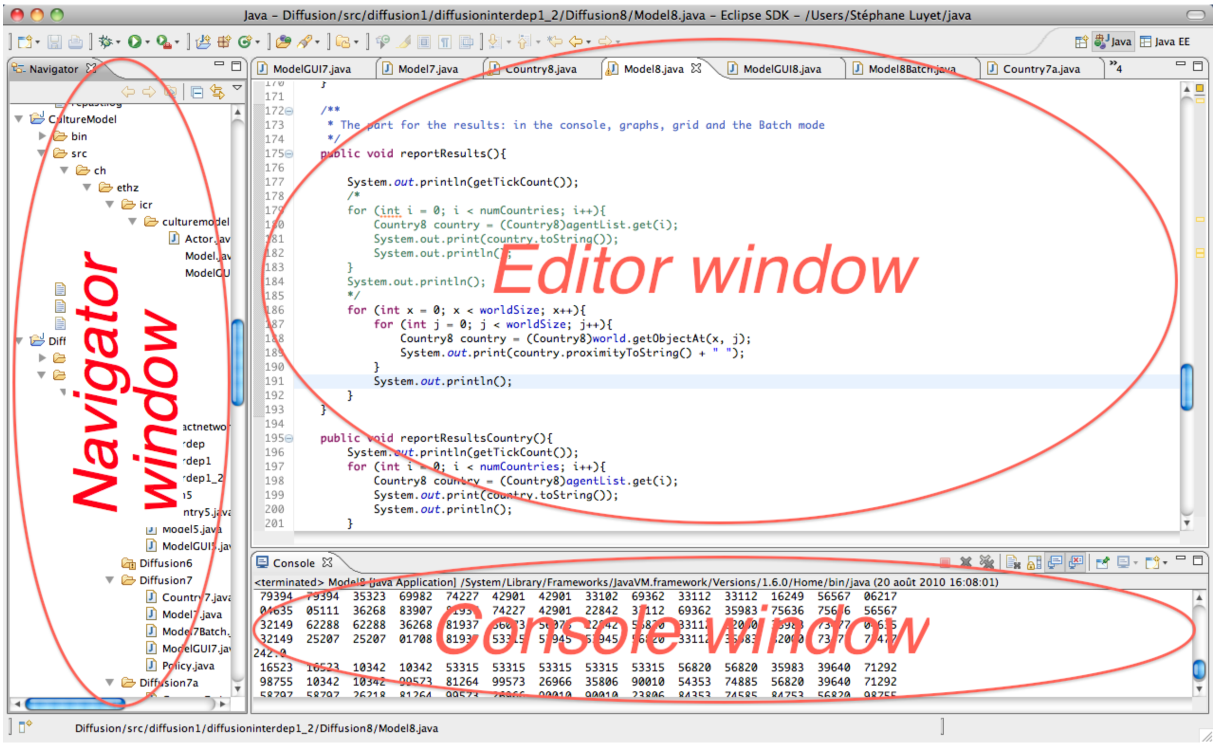



Figure 4.2: The Eclipse environment

Toolkit or RePast has been developed to ease the building and the analysis of agent-based models. RePast supplies an extensive Java API of already defined classes, meaning that the basic architecture of Java classes needed to create one's own computational model have already been programmed at the University of Chicago, as for instance the basic architecture of visualization and editing tools have already been programmed. Put simply, RePast takes advantages of the object-oriented programming concepts of inheritance and polymorphism.

Now that we have a language, a book in which to write and some of the chapters partly written, it is time to explain how to fill in the gaps. In the next two sections, the operationalization and the implementation – the writing of the program – of the different theoretical parameters explained in Chapter 2 will be discussed.

### 4.2.4 The *Model Exploration Module* or MEME

Batch models[5] are mainly run to extract data produced by different simulations. One of the main limitations of the batch model encountered here is that not only can one not alter the random seeds, but one also should explore the model by hand, meaning that one runs one batch model and then manually changes the wanted parameter in the program and run the model with the new values and so forth, which may be long and especially fastidious.

Moreover, to be able to analyze this data, we need a tool that can not only collect this data but also allows for the organization of the dataset. To achieve this aim, an application has been developed by AITIA International inc., a Hungarian company active in the field of artificial intelligence and that develops others interesting tools for computational agent-based modeling under the name of **M**ulti-**A**gent **S**imulation **S**uite or MASS, such as a programming language and an environment for the development of agent-based models[6]. The MEME module is one of the tools of this MASS suite.

Moreover, the MEME is a well-suited tool to use with RePastJ and provides such facilities for collecting and organizing the dataset, since it comes with the ability to export datasets as CSV files that can be used with other software (Bocsi et al., 2010), such as Stata, for instance. In other words, with the batch mode, we can alter the parameters, but not at the same time. That is why we need another tool that allows us to alter the different parameters and that gives us a complete database. That is the purpose of the MEME *application.*

This application works as follows. When the needed model and parameters have been chosen and defined within the MEME module, the latter becomes automatically attached to RePastJ. Moreover, when launched, MEME is capable of running without the help and interaction of the user. In other words, it is able to run in the background as a separate application (IVÁNYI et al., 2007).

[5] See Section 4.4.8 for a brief description of the use of Batch models.

[6] For information on the different tools provided under the *label* MASS, see the following link: http://mass.aitia.ai/.

At the end of the runs, the results are automatically stored in a database with all the parameters and variables of interest the user has decided to alter and to analyze. Besides such automatically created databases, the MEME application can manage imported database of RePast results or other CSV files. Moreover, the database automatically generated when running the MEME in order to obtain simulation results can be exported to CSV files (IVÁNYI et al., 2007). Such export allows the researcher to analyze this data empirically with other statistical software, such as Stata or R, for example. As a result the arborescence (Figure 4.3) of the different parameters – in our case neighborhoods, sizes of the world, sizes of the proximity array and the number of traits in order to have the values of the interested variables, namely the number of regions and the average effectiveness – gives us a total of 540 runs.

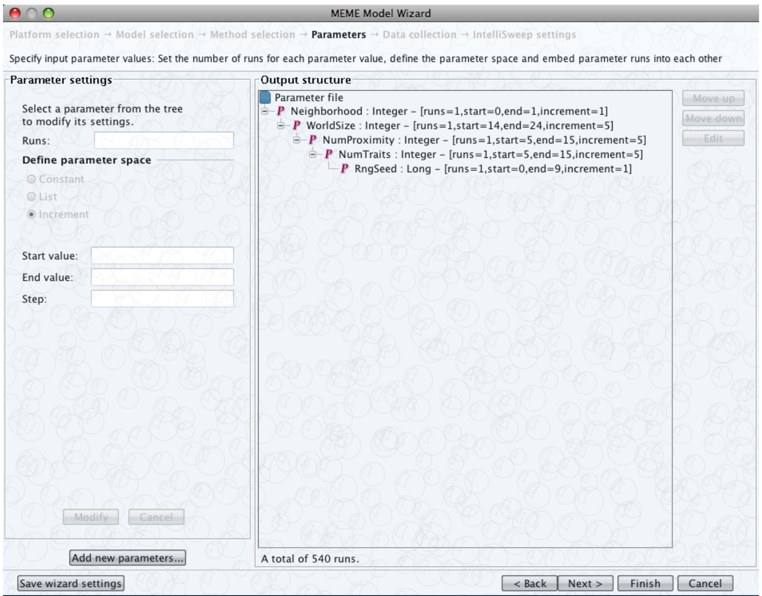



Figure 4.3: The arborescence of parameters in MEME

## 4.3 The description of the parameters

In Chapter 2, we explained that a policy change can occur depending on the evolution of some factors internal to a country, but that this change is subject to external influences as expressed by different mechanisms of diffusion. Furthermore, we have stressed that these interdependencies make diffusion complex, so that this process need to be studied with the help of computational agent-based modeling, discussed in Chapter 3. The next subsections will be devoted to the merging of diffusion and agent-based model, which serves the explanation of the development of the model.

### 4.3.1 The agents[7]

As the process of diffusion occurs between countries, they correspond to the agents in the model. They have internal and external characteristics, as shown in Table 4.1.

| Threshold | Bandwagon pressures |
|---|---|
| Policy Preference | The share of neighbors |
| Policy Effectiveness | that have changed |
| Institutional Constraints | their policy |
| Political insecurity | Proximity array |
| Policy (current and alternative) | |

Table 4.1: The characteristics of the agents

From Section 2.4, we have emphasized the following internal determinants: Ideology, political insecurity, effectiveness, and institutional constraints and external determinants: the share of neighbors and the proximity array. Here is explained the operationalization these internal and external determinants. The special case of political insecurity will be taken into account in a different manner, since it depends on the time of the elections[8] as shown in the description of the preference here below. Let us start by having a quick look at the internal characteristics.

- *The preference for the policy*: Agents have specific preferences over the current policy. As we have argued in Section 2.4.1, by changing the current policy, policy makers seek their ideal point on the left-right continuum. However, this ideal point is not fixed forever, since it moves following the results of elections and/or voting, which roughly represent the political insecurity.

[7]Here we have the description of the operationalization of the parameters theoretically explained in Section 2.4

[8]For a reminder of the influence of elections, see Section 2.4.2.

At the beginning of the simulation, preferences are drawn randomly from a normal distribution with mean 0 and standard deviation 0.2, truncated at 1 and $-1$. 1 means that the agent has extremely strong preferences for the current policy, while $-1$ means that preferences are entirely against it. Moreover, the preference for the current policy is fixed for a period of five steps, which is supposed to reflect the fact that policy makers' preferences change principally when there is alternation in government, which does not happen every year. Of course, this is a rather sketchy operationalization of political insecurity. We have tried the possibility for the countries to have a legislature randomly chosen in a uniform distribution truncated at 0 and 10. However, this implementation was not concluding. Since time remains a major problem in computational agent-based models, we have tried to minimize this problem by developing the parameter *elections* so that the user of the program can modify it by himself. Put differently, this parameter can be changed by hand in the model. Nevertheless, we assume at the beginning that, for every five time steps, there is a possibility of a radical change in the preference, meaning a majority change. That is why the preference is redrawn from a random distribution every five steps. In other words, every five steps, the *old* policy preference is replaced by a new one drawn randomly from a normal distribution with a mean of 0 and a standard deviation of 0.2. The change in preferences is, therefore, not biased in a specific direction.

- *The effectiveness of the policy*: In Section 2.4.3, we have defined policy effectiveness as the attainment of the desired outcome. For example, since aging policies should help old people to maintain their purchasing power after retirement, they are ineffective if the target population is becoming impoverished. At the beginning of a run, each country has its own policy effectiveness, which is drawn randomly from a normal distribution with mean 0.0 and standard deviation 0.4, truncated at 1 and $-1$. 1 means that the policy is entirely effective, while $-1$ means that it is entirely ineffective. In other words, the effectiveness can be either positive or negative. The fact that the effectiveness of the policy can move along this dimension expresses the possible diffusion of a policy even if it is not very effective or a bad idea (Gilardi, 2010). Policy effectiveness has a rather large standard deviation throughout the world, meaning that the effectiveness difference between the countries is quite wide. For instance, this can be interpreted as the global benefits of different welfare state arrangements that are more effective in some countries than in others, due to various factors such as population aging, public finance rises, and generally postindustrial developments (Pierson, 2001).

At each step, a variable randomly drawn from a normal distribution with mean $-0.01$ and standard deviation 0.03 is added to the policy effectiveness. Therefore the effectiveness is likely to decrease, which is in line with what is observed in reality. Indeed, we have emphasized in Section 2.4.3 that postindustrial developments have induced new challenges that call for new more effective policies.

- *The institutional constraints*: Agents face specific institutional constraints, which determine the probability with which a law can be passed. Conceptually, this can be linked with Tsebelis (2002)'s veto players[9]. At the beginning of the simulation, institutional constraints are drawn randomly from a normal distribution with mean 0.0 and standard deviation 0.3, truncated at 1 and $-1$. 1 actually means that there are no institutional constraints and, therefore, that a policy proposal faces no obstacles to be voted into law. The institutional constraints are fixed through the entire simulation. This shows the institutional stability of the different countries.

- *The policy*: As one of the purposes of this thesis is to simulate the diffusion of policies, the different countries are also characterized by their (current and, after the change, alternative) policies. At the beginning of the simulation run, each country has its own policy, as characterized by the color on the grid. In

[9]For a reminder, see section 2.4.4

other words, there are 196 policies, as shown in Section 4.3.2.

Here below, we briefly explain the theoretical pertinence and operationalisation of the external characteristics.

- *The share of neighbors*: In Sections 2.3.3 and 3.4.6, we have highlighted the substantial weight of the different neighbors in the process of diffusion. Moreover, we have defined bandwagon pressures as the more neighbors that have changed their policy, the higher the chance for a country to introduce the alternative policy. Hence, to be precise, we integrate into the model the share of neighbors that have changed their policy. Formally, this can be stated as follows:

$$\text{share of neighbors} = \frac{N_c}{N} \tag{4.1}$$

  where $N_c$ = the number of neighbors that have changed their policy and $N$ = the total number of neighbors.

  Furthermore, the geographical proximity corresponds to the definition of the neighborhood used – the Moore neighborhood (8 adjacent cells) or the Von Neumann neighborhood (cells at 4 cardinal points). Thus, the neighborhood is a way of expressing the assumption of the bounded rationality, as explained in Section 2.5.1.

- *The proximity array*: In Section 2.3.3, the neighborhood was defined not only as purely geographical, but also as taking into account several other dimensions, such as the culture, the dominant religion, the economic proximity, and so forth. Thus, as we have already showed in Section 2.3.3, to share a common border is a necessary but not sufficient condition to define a neighborhood.

  Besides the geographical border, the neighborhood is defined as a proximity array, with several dimensions, each representing a possible common feature countries may share, such as economy, religion, history, and so forth. Furthermore, computationally, this proximity array is based on the definition of culture as expressed by Axelrod (1997*b*) in his model of the dissemination of culture[10] and as depicted in Table 5.1.

| | | | | |
|---|---|---|---|---|
| 3 | 7 | 9 | 4 | 6 |

Table 4.2: An example of a proximity array

At the beginning of a run, each country has a defined proximity array. Each trait–each number–that defines a dimension in the array is randomly chosen from a uniform distribution. Additionally, the length of the array and the number of possible traits can be fixed manually by the researcher using a slider ranging from 1 to 25. This is a way to fine-tune the proximity. At the initialization, the length of the array is arbitrarily fixed at 5 cells and there are 10 possible traits.

Moreover, we assume that the countries have a bounded rationality as they interact with their defined closest neighborhood. However they are adaptive, since they react to the information given by their environment as to which is the potentially more effective policy (Section 3.3.4 and 4.1).

Now that we have defined and explained the main features of the agents, it is time to explore the wonderful world in which they can freely interact.

### 4.3.2 The world

We develop a toroidal world. This kind of shape is now well diffused in computational agent-based models. A toroidal shape means that our world is wrapped around. Figure 4.4 shows what a torus[11] looks like.

The advantage of using such a shape for developing our world is that there is no borders. Agents at the northeast corner have neighbors at the southwest one. As already explained in Section 3.4.1, a torus can be approximated with the example of a map of the world. On a map, Russia and the USA are completely opposed, but

[10] More on that in Section 3.4.7.

[11] This kind of shape has already been presented in Section 3.4.1

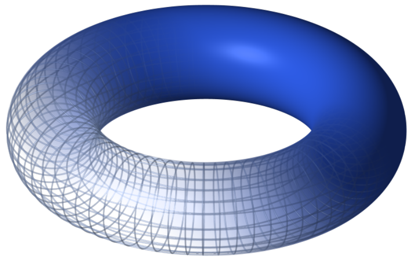


Figure 4.4: Representation of a torus

in reality these two are neighbors. The maps are normally cut at the Bering Strait, which links Siberia and Alaska. If the map is wrapped around, Alaska and Siberia become close again. The torus has the same effect on the virtual world. This shape is just a way of getting closer to reality.

To develop the model, a square grid[12] composed of a certain number of cells is created; each cell representing a country. The size of the world can be chosen with the help of a slider between 10 and 100. The initial size is arbitrarily fixed at 14, which yields 196 cells (countries), which is more or less the actual current number of countries[13].

Figure 4.5 is a typical representation of the world at the setup.

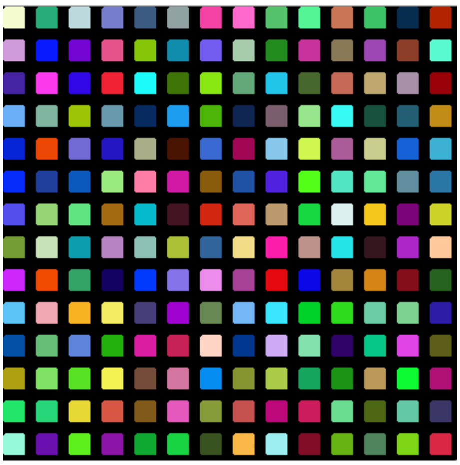


Figure 4.5: The world at the start of a run

[12]This kind of lattice is the usual way for creating computational agent-based models, as expressed in Section 3.4.1.

[13]The United Nations has 192 members (http://www.un.org). However, the US State Department recognizes 194 *independent* countries (http://www.state.gov/s/inr/rls/4250.htm). So that our world composed with 196 countries is a rather good approximation of the current world.

Now that we have a world filled with heterogeneous countries, we need to give them the basic conditions for changing their current policy, since it has become ineffective. In other words, how do the agents interact?

### 4.3.3 The interactions

The agent level is the very core of the computational agent-based model, since the flow of information is gathered and handled through their interactions that are endogenous within the world, and since this flow shapes the interrelations of the agents (Epstein, 2006). The different parameters evolve according to the rules we have defined: the effectiveness change at each step, every X steps – the time between the elections – for the preference, depending on the choice of the researcher, and the political constraints are fixed for the entire run. Then, for a change to occur, a country must respect the following conditions:

1. *The agent is ready for change*: An agent is ready for changes when its effectiveness is lower than its preference for the current policy. This means that if the policy goes ineffective, the agent starts looking at its neighbors to gather an idea of their current situation. This models the idea that the impact of effectiveness on policy change depends on preferences. In other words, if policy makers have strong preferences in favor of the current policy, this must be very ineffective in order to be abandoned. Thus, policy makers will accept high levels of ineffectiveness, since they are ideologically (or electorally) biased in its favor. By contrast, policy makers who have not-as-strong preferences for the policy will be willing to abandon it at lower levels of ineffectiveness.

2. *The choice*: When a country is ready for change, it goes through an intermediate phase, in which the country starts looking at its neighborhood. To show the importance of the neighbors in the policy choice, the neighborhood of the agents can be composed of either the Moore Neighbors (the eight adjacent cells) or the Von Neumann Neighborhood (the four cardinal-point cells).

By defining these neighborhoods, we assume that the agents have a bounded rationality. We assign a “view” to each agent; that is, the number of cells with which an agent can interact (4 or 8 neighbors). Therefore the definition of the agent’s vision (its neighborhood) is a way of expressing its bounded rationality (Meseguer, 2005; Epstein and Axtell, 1996)[14]. Each agent looks at its neighbors and search for the one(s) that has (have) already changed its (their) policy. In so doing the agent can see whether the new policy of its neighbors that have already changed their policy is more (less) effective.

As Granovetter (1978, 1421) argues, threshold models are well suited for studying the diffusion of innovations, because it takes into account “the variation of norms and preferences within the interacting group.” The threshold is defined as “the point where the perceived benefits to an individual of doing the thing in question ... exceed the perceived costs.” This model is also well suited for dichotomous dependent variables: in our model, each agent can have either its current policy or an alternative policy.

The choice variable ($CV$) is then calculated as follows: This variable is composed of two elements. The first one is the difference between the average effectiveness of the neighbors that have changed their policy and the current effectiveness. In other words, the average effectiveness of the neighbors who already have changed their policy ($\frac{\sum_{c=1}^{n} E_c}{N_c}$) is compared to the one ($e$) of the agent. By looking at all the neighbors that have changed their policy, the country updates its beliefs on the potential consequences of a possible change. If the result is greater than 0, it means that, in general, the policy of the neighbors that have changed their policy is more effective. In the second part, this subtraction is weighted with the number of neighbors that have already changed their policy ($N_c$) divided by the number of neighbors (N; that is, $N = 8$ in the case of the Moore neighborhood and $N = 4$ in the case of the Von Neumann neighborhood). This corresponds to a bandwagon pressure

[14]See point 4 in Section 4.1

(Abrahamson and Rosenkopf, 1993, 1997) that represents a self-reinforcing process: the more neighbors that have changed their policy, the higher the probability of choice.

The choice variable (CV) can be mathematically written as follow:

$$CV = \Big( \big( \frac{\sum_{c=1}^{n} E_c}{N_c} - e \big) \big( \frac{N_c}{N} \big) \Big) \tag{4.2}$$

And the condition for a choice to occur is given by the following expression:

$$CV > threshold \tag{4.3}$$

This *threshold* variable is randomly chosen in a uniform distribution truncated at 2 and -2[15], which represents the point from which the countries start looking at their neighbors.

Equation 4.3 shows that, at each step, and for each country, a choice variable is compared to the threshold and it represents the point from which the countries start looking at their neighbors. In other words, Equation 4.3 expresses that a country chooses the most effective policy of its neighborhood when the potential gain in effectiveness exceeds the expected costs of introducing the alternative policy that are, at this point, randomly defined.

3. *The change*: When the country has chosen the policy of its most effective and similar neighbor, this policy is introduced, if the condition for a change expressed below is respected. The change variable is composed of three parts:

   (a) A baseline probability: the baseline probability is arbitrarily fixed at 0.05, since there is a small amount of diligent agents in the process of diffusion that will introduce the policy even if no one else wants to do so (Simmons and Elkins, 2004).

   (b) The average effectiveness among the similar neighbors: this is introduced

[15]-2 and 2 correspond to the extreme results of the equation 4.2

to take into account the fact that a policy can be introduced if it is in line with the preference of the policy makers, even if it is not effective (Braun and Gilardi, 2006). Therefore, the country calculates whether the difference between the average effectiveness of similar neighbors who have changed their policy ($\frac{\sum_{s=1}^{c} E_c^s}{N_c^s}$) and the current effectiveness ($e$). Here, by comparing itself with the similar neighbor(s), it has acquired the conviction that the introduction of the policy of this neighbor is the best solution, and, at this point, since the two interacting countries have been influenced by their shared information, they become more similar. The division by the number of neighbors that have changed their policy is justified, since they provides information about the potential alternative policy. If this difference is lower than 0, the alternative policy is ineffective compared to the current one and the chance of success decreases. At this point, diffusion enters the model, since it is defined as a process whereby the choices of a country are influenced by those in other countries (Section 2.3), expressed here by the comparison of the effectivenesses.

(c) <u>The weighted institutional constraints</u>: in Section 2.4.4, we stress the importance of the institutional constraints for the interactions, as expressed by the veto players. Therefore a change is possible only if the different veto players have found some kind of consensus. More precisely, the institutional constraints must be overcome for an alternative policy to be introduced. This institutional constraints parameter is then weighted with the share of neighbors that have already changed their policy ($\frac{N_c}{N}$). This expresses the fact that the internal political *game* is influenced by the information found in other countries. In other words, the more information is found abroad, the more this information influences the veto players, and the more its weight in the policy changes decisions.

Thus, the change variable can be formally expressed as follows:

$$\text{change} = \Big(0.05 + \big(\frac{\sum_{s=1}^{c} E_c^s}{N_c^s} - e\big) + \Big(\text{institutional constraints}\big)\big(\frac{N_c}{N}\big)\Big) \tag{4.4}$$

We transform Equation 4.4 as a logit. Consequently, the choice variable is now transformed as a probability and is bounded towards 0 and 1:

$$\text{p(change)} = \frac{e^{(\text{change})}}{1 + e^{(\text{change})}} \tag{4.5}$$

wherein the probability of success equals the change variable, as defined in Equation 4.4.

Therefore, in this model, the change is seen as a success. For that reason, at each time step, each country that has chosen its alternative policy – the most effective policy among the similar neighbors – experiences a Bernoulli trial with a probability of success equals to the change variable, as defined in Equation 4.4. The probability function of the Bernoulli distribution is expressed as follows:

$$f(x;p) = \begin{cases} p & \text{if } x = 1; \\ 1-p & \text{if } x = 0; \\ 0 & \text{otherwise.} \end{cases} \tag{4.6}$$

where $p$ is the probability of success. Equation 4.6 stresses that the Bernoulli random variable can have only two values, 0 and 1, where 1 means success.

In sum, a country has a chance of changing the current policy – a chance of success – that corresponds to a probability defined by the logit of the change variable. In other words, a country introduces the most effective policy if the institutional constraints are overcome.

In this section, we have highlighted the operationalization, mainly based on the normal distribution of the different main parameters we will use in the building of a computational agent-based model of policy diffusion. We also define how they should evolve in our world. This, put together, allows us to create basic countries. In a second step, we explain the computational world in which our basic countries will interact. Moreover, we describe the different interactions between the countries. The share of neighbors that have already changed their policy mainly conditions these interactions.

Therefore, for a change to occur, a country traverses three different phases[16]:

1. The country is *ready for change* if the effectiveness of its current policy is lower than its preference level. In other words, the country is ready for change if the current policy is ineffective despite the preference in favor of the current policy;

2. if the country is ready for a change, it looks at what the neighbors that have already changed their policy do and then *chooses* the most effective policy among them. Then it searches for the most similar one;

3. if the country has chosen an alternative policy, it will increase its similarity with the "policy sender" and have a chance to successfully introduce this chosen policy; that is to change the current policy[17] – even if it is not the most effective one – if the institutional constraints are overcome;

In the next section, we will turn to the computational implementation of these three phases. More precisely, we will explain the main methods we program to construct our model.

[16]Here we try to apply the KISS motto explained in Section 4.4.8, so that our model can be summed up in three simple sentences!

[17]To be precise, the country introduces the policy of its most similar neighbor that has changed its policy.

## 4.4 The implementation[18]

This section explores how the different conditions of change are programmed. The program must respect the logic of the algorithm. In this model, the program follows the three main conditions for a change to occur; that is, ready, choose, change. The slight changes that appear in the program are easily comprehensible.

In the next subsections, we will explore the different classes and objects[19] that compose the codes of the diffusion model[20]. We start with the methods of the *Country* class.

### 4.4.1 The *Country* class

The class *Country* is the basic block of my program, since the country corresponds to the agent that populates the world. Moreover, in this class are defined the main mechanisms and conditions for the interactions.

Beside the several parameters that describe the country, we have to explain in the next subsections the different methods that illustrate the behavior(s) of the countries. In other words, in the next subsection, how the different agents interact will be computationally illustrated.

### 4.4.2 The parameters of the class *Country*

In Section 4.3.1, we emphasized the creation of the different parameters as purely random. Thus, in the computational development of the model, a method for the creation of normal distributions is needed. This method is expressed as follows:

```
public static double createNormalDistribution(double mean, double sd){
  Random.createNormal(mean, sd);
  double param = Random.normal.nextDouble(mean, sd);
  if (param > 1){
```

[18]This section may seem a little bit redundant, but it is necessary to highlight that the computational development of the model strictly correspond to the definition of the *algorithm* explained in Section 4.3.3

[19]Remember from Section 4.2.1 that an object is an instance of a class!

[20]To have a comprehensive view of the program, that is all the parameters and methods, you will find all the codes in Appendix C.

```
    param = 1;
  } else if (param < -1){
    param = -1;
  }
  return param;
}
```

What this method says is that the normal distribution is characterized by its mean and standard deviation. For our purpose, the distribution is also truncated at 1 and -1.

Therefore, when objects of the class *Country* are created, each parameter is drawn randomly from that normal distribution, and the mean and standard deviation of the different parameters need to be initialized (these means and standard deviations correspond to the arguments *mean* and *sd* in the method *createNormalDistribution(double mean, double sd)*):

- The mean and the standard deviation of the effectiveness of the policy (*policyEffectiveness*) are set to 0.0 and 0.4, respectively (this is also the case for the best effectiveness – *bestEffectiveness*; that is, the effectiveness introduced after a change). The change parameter for the effectiveness has a mean set to -0.01, since the effectiveness is likely to decrease and has a standard deviation of 0.03.

- The same is done for the preference parameter. A first difference lies, however, in the initialization: the mean and standard deviation are set to 0.0 and 0.2, respectively.

  ```
  public double changePreference(){
    if (model.getTickCount()%model.elections == 0){
      policyPreference = createNormalDistribution(policyPreferenceMean,
          policyPreferenceSD);
    }
    return policyPreference;
    }
  ```

  The second distinction comes from the evolution. As the preference changes every *model.elections* steps, we use the operator modulo (%), that gives the rest of a division, with the time of the model. In other word, programing (time of the model)%model.elections == 0 means that, if the division by *model.elections* gives no rest, the condition is applied – here, the change in the preference. *model.elections* is a parameter defined as a slider in the program. At the initialization, we arbitrarily fixed the time between the elections at 5 steps. However to test the model, it can be modified to between 1 and 35 steps

- Again, the same logic applies to the institutional constraint parameters (*politicalConstraints*), with a mean set to 0.0 and a standard deviation set to 0.3. Since this parameter is fixed for the entire simulation, there is no parameter of change defined.

### 4.4.3 The neighborhood

The neighborhood is central to the interactions developed in this model. As two neighborhoods are defined – the Moore and the Von Neumann neighborhood – Java provides the SWITCH statement to select from among a number of alternatives of this type.

```
switch (model.neighborhood){
case Model.MOORE:
  //[some code];
  break;
case Model.VON_NEUMANN:
  //[some code];
  break;
}
```

SWITCH blocks are composed of three expressions[21]:

1. <u>switch</u>: switch evaluates the expression between the parentheses;
2. <u>case</u>: The first block case is executed if it corresponds to a known value; if not, it goes to the next case block; and so forth until there is no more case block;
3. <u>break</u>: the break statement has here a great significance, since it causes the switch statement to end after its execution and the program goes on with the

[21]The *break* statement is not always used.

next method. In other words, the BREAK statement causes the program to go on with the lines after the SWITCH.

The break is here rather important, as the behavior of the model depends on the type of the chosen neighborhood defined at the beginning of a run. To not use the break statement will cause the program to run the Moore and the Von Neumann neighborhood one after the other, even if only one neighborhood is chosen. As a consequence, the results of the model would be totally mixed up.

### 4.4.4 The *reset()* method

At each time step, the count of the number of changed neighbors is cleared; that is, at each time step the vector that contains the neighbors that have changed their policy is emptied. This is done through the *reset()* method:

```
public void reset (){
  switch (model.neighborhood){
  case Model.MOORE:
    countChangedNeighbors(x, y).clear();
    break;
  case Model.VON_NEUMANN:
    countChangedNeighbors(x, y).clear();
    break;
  }
}
```

This method expresses the fact that the countries that have changed at time $t-1$ are not automatically the same ones that change at time $t$.

### 4.4.5 The *ready()* method

In Section 4.3.3, we explain that a country is ready for a change if the effectiveness of the current policy is lower than the preference level; that is, a change must be envisaged despite the preference in favor of this policy.

```
public boolean ready(){
  switch (model.neighborhood){
  case Model.MOORE:
    if (policyEffectiveness < policyPreference){
```

```
      return true;
    }
    break;
  case Model.VON_NEUMANN:
    if (policyEffectiveness < policyPreference){
      return true;
    }
    break;
  }
  return false;
}
```

This method is easy to understand, as it corresponds to the computational expression of the first step defined in Section 4.3.3. If the effectiveness of the current policy is lower than the preference the agent/country has for the policy, then this condition is true and the country chooses an alternative policy; otherwise, this condition is false.

### 4.4.6 The *chooseAlternativePolicy()* method

The method is written as follow:

```
public boolean chooseAlternativePolicy(){
  double choiceThreshold = 0.0;
  switch (model.neighborhood){
  case Model.MOORE:
    choiceThreshold = Random.uniform.nextDoubleFromTo(-2.0, 2.0);
    choiceVariable =calculateChoiceVariable(x, y);
    if (choiceVariable > choiceThreshold){
      findSimilar(x, y);
      return true;
    }
    break;
  case Model.VON_NEUMANN:
    choiceThreshold = Random.uniform.nextDoubleFromTo(-2.0, 2.0);
    choiceVariable =calculateChoiceVariable(x, y);
    if (choiceVariable > choiceThreshold){
      findSimilar(x, y);
      return true;
    }
    break;
  }
  return false;
}
```

Thus, the *chooseAlternativePolicy()* method stresses that if the *choiceVariable* parameter is greater than a threshold randomly defined in a uniform distribution truncated to 2.0 and -2.0, which corresponds to the extreme value *choiceVaraible* can have.

This means that if the *choiceVariable* is greater than the threshold chosen randomly from an uniform distribution truncated at 2.0 and -2.0[22], that is the *choiceThreshold* parameter, then the country looks for the most similar neighbor(s) through the *findSimilar(int pos, int pos)* method and the condition is true.

This method needs a deeper explanation, as it uses several other important methods.

### The *findSimilar(int pos, int pos)* method

With this method, the country loops through the neighbors that have already changed their policy. If two agents have the same number of similar proximity traits in common, they then are considered as similar[23].

```
public Vector findSimilar(int px, int py){
  int numSimilar = similarNeighbors.size();
  numSimilar = 0;
  switch (model.neighborhood){
  case Model.MOORE:
    Iterator it = countChangedNeighbors(px, py).iterator();
    while (it.hasNext()){
      Country similarNeighbor = (Country)it.next();
      if (countAlikeDimensions(similarNeighbor) ==
        similarNeighbor.countAlikeDimensions(this)){
        similarNeighbors.add(similarNeighbor);
      }
    }
    break;
  case Model.VON_NEUMANN:
    Iterator vnIt = countChangedNeighbors(px, py).iterator();
    while (vnIt.hasNext()){
      Country similarNeighbor = (Country)vnIt.next();
      if (countAlikeDimensions(similarNeighbor) ==
        similarNeighbor.countAlikeDimensions(this)){
        similarNeighbors.add(similarNeighbor);
      }
    }
```

[22] We use here a uniform distribution truncated in -2.0 and 2-0 to illustrate the fact that not all countries are facing the same threshold.

[23] Look at the subsection 4.3.1 for a reminder

```
    break;
  }
  return similarNeighbors;
}
```

The number of similar neighbors is, at each time step, reset to 0, as the similarity increases through the simulation. In other words, the neighbor(s) with which the country shares the same number of similar features of the proximity array may be a different one at time $t+1$ than at time $t$ as the interactions imply changes in the proximity array of the agents.

This method calls for two explanations; i.e., an explanation of the *countChangedNeighbors(x, y)* method and how the *countAlikeDimensions(Country neighbor)* works. One of the most important method of this model is the *countChangedNeighbors(x, y)* method, because a change can occur only if there is at least one country that is in a sufficiently bad situation that it has to change its policy.

### The *countAlikeDimensions(Country neighbor)* method

This method is used to evaluate the number of like features the two countries that are interacting have in common. More precisely, this method gives the percentage of similarity between two interacting countries.

```
public double countAlikeDimensions(Country n){
  int same = 0;
  for (int i = 0; i < model.numProximity; i++){
    if (proximity[i] == n.proximity[i]){
      same++;
    }
  }
  return (double)same/(double)model.numProximity;
}
```

At the beginning, the number of like features is set to 0. This is justified since, at each time step, the similarity may increase. Thus, each feature is compared with the one of the neighbor – Country n – and if both have the same trait, the same variable is increased by one. When the entire proximity array has been evaluated, this method returns the share of like dimensions.

### The *countChangedNeighbors(int pos, int pos)* method

To count the number of neighbors that have already introduced an alternative policy, first the number of changed neighbors is set to 0, then we loop through the neighbors. If the country has the same effectiveness as some neighbor(s) and has updated its policy – it has changed its color; the neighbor is considered as having changed its policy.

To take into account the neighbors that have already changed their policy is a little bit tricky. The method *changePolicy()* cannot be directly used[24] because it causes the program to generate a *stack overflow* error–a programming error caused by a too deep recursion[25]. Therefore, it seems that the best approximation of the change is to express it with the effectiveness and the color, which are the main characteristics of a policy.

```
public Vector<Country> countChangedNeighbors(int px, int py){
  switch (model.neighborhood){
  case Model.MOORE:
    neighbors = model.world.getMooreNeighbors(px, py, false);
    numChangedNeighbors = 0;
    Iterator it = neighbors.iterator();
    while(it.hasNext()){
      Country changedCountry = (Country)it.next();
      if (changedCountry.bestEffectiveness == policyEffectiveness
        && changedCountry.updatePolicyColor() == true){
        changedNeighbors.add(changedCountry);
        numChangedNeighbors++;
      }
    }
    break;
  case Model.VON_NEUMANN:
    neighbors = model.world.getVonNeumannNeighbors(px, py, false);
    vnNumChangedNeighbors = 0;
    Iterator vnIt = neighbors.iterator();
    while(vnIt.hasNext()){
      Country changedCountry = (Country)vnIt.next();
      if (changedCountry.bestEffectiveness == policyEffectiveness
        && changedCountry.updatePolicyColor() == true){
        changedNeighbors.add(changedCountry);
```

[24]This method will be explained latter in this section.

[25]For example, if we define the change using the *changePolicy()* method, the program will throw a *StackOverFlowError*, because the condition for a change to occur uses the *countChangedNeighbors(x, y)* method that use the *changePolicy()* method, that uses the *countChangedNeighbors(x, y)* method and so on. There is no way out of this loop!

```
        vnNumChangedNeighbors++;
      }
    }
    break;
  }
  return changedNeighbors;
}
```

Thus, if a neighbor has introduced the best effectiveness, which corresponds to the current policy effectiveness and if this neighbor has updated its color, then it is supposed to have changed its policy and the number of neighbors that have changed their policy is incremented by one. This condition is tested for all the neighbors (4 or 8, depending on the type of neighborhood) and, at the end, the array that stores the neighbors that have changed their policy is returned.

### The *calculateChoiceVariable(int pos, int pos)* method

As its name indicates, this method has been developed in order to calculate the choice variable.

```
public double calculateChoiceVariable(int px, int py){
  double pCV = 0.0;
  switch(model.neighborhood){
  case Model.MOORE:
    Vector moooreNeighbors = model.world.getMooreNeighbors(px, py, false);
    numNeighbors = moooreNeighbors.size();
    numChangedNeighbors = countChangedNeighbors(px, py).size();
    meanEffectiveness = calculateMeanEffective(px, py);
    pCV = ((meanEffectiveness-policyEffectiveness)*(numChangedNeighbors
        /numNeighbors));
  break;
  case Model.VON_NEUMANN:
    Vector vonNeumannNeighbors = model.world.getVonNeumannNeighbors
    (px, py, false);
    vnNumNeighbors = vonNeumannNeighbors.size();
    vnNumChangedNeighbors = countChangedNeighbors(px, py).size();
    meanEffectiveness = calculateMeanEffective(px, py);
    pCV = ((meanEffectiveness-policyEffectiveness)*(vnNumChangedNeighbors/
        vnNumNeighbors));
  break;
  }
  return pCV;
}
```

At each iteration, the choice variable – the $pCV$ parameter – is initialized at 0. In other words, this variable is calculated at each time step. Then we count the number of neighbors (according to the chosen neighborhood – Moore or Von Neumann) and the number of neighbors that have changed their policy, as the country is interacting with these neighbors. We also need to calculate the average effectiveness of the neighbors that have already introduced an alternative policy, since it is compared with that of the current policy

The *choiceVariable* parameter is then calculated as follows: the current policy effectiveness of the country is subtracted from the average effectiveness. The result is weighted with the proportion of neighbors that have changed their policy. This is the computational development of Equation 4.2

### The *calculateMeanEffectiveness(int pos, int pos)* method

The above block of code has explained the calculus of the choice variable. In this case, the calculus has intervened using the method *calculateMeanEffectiveness(int pos, int pos)* the result of which gives the average effectiveness among the neighbors that have *already* changed their policy.

```
public double calculateMeanEffective(int px, int py){
  meanEffectiveness = 0.0;
  switch(model.neighborhood){
  case Model.MOORE:
    numChangedNeighbors = countChangedNeighbors(px, py).size();
    Iterator it = countChangedNeighbors(px, py).iterator();
    while(it.hasNext()){
      Country changedNeighbor = (Country)it.next();
      for (int i = 0; i < numChangedNeighbors; i++){
        meanEffectiveness = (meanEffectiveness +
        changedNeighbor.policyEffectiveness)/numChangedNeighbors;
      }
    }
    break;
  case Model.VON_NEUMANN:
    vnNumChangedNeighbors = countChangedNeighbors(px, py).size();
    Iterator vnIt = countChangedNeighbors(px, py).iterator();
    while(vnIt.hasNext()){
      Country changedNeighbor = (Country)vnIt.next();
      for (int i = 0; i < vnNumChangedNeighbors; i++){
        meanEffectiveness = (meanEffectiveness +
```

```
        changedNeighbor.policyEffectiveness)/vnNumChangedNeighbors;
      }
  }
  break;
  }
  return meanEffectiveness;
}
```

At each time step, the average effectiveness is set to 0. Since the effectiveness evolves through the entire simulation, it is normal to recalculate the average effectiveness at each time step. Then, the number of changed neighbors is calculated. This is done by looping through them and summing their policy effectiveness. Finally, to obtain the mean, the total sum of the different policy effectivenesses is divided by the number of changed neighbors.

### 4.4.7 The *changePolicy()* method

Now that the choice has been computationally explained, we need to turn to the programming of the change, as it is constructed within the *Country* class.

#### The *changePolicy()* method

The method is, to a certain degree, easy to explain. The country can change its policy if the *changeLogit* is equal to 1; that is, if the country experiences a success from the Bernoulli distribution, as explained in Section 4.3.3.

Thus, a country has a greater chance to introduce the alternative policy – the most effective policy among the policies of the similar neighbors that have already changed their policy – if the institutional constraints have been successfully overcome and if the alternative policy is more effective than the current one.

In the *changePolicy* method, the change is true if country tosses 1 from the Bernoulli distribution, where 1 means success.

```
public boolean changePolicy(){
  double logit = 0.0;
  double changeLogit = 0.0;
  switch (model.neighborhood){
  case Model.MOORE:
```

```
    logit = calculateLogit();
    changeLogit = createBernoulli(1, logit);
    if (changeLogit == 1){
      return true;
    }
    break;
  case Model.VON_NEUMANN:
    logit = calculateLogit();
    changeLogit = createBernoulli(1, logit);
    if (changeLogit == 1){
      return true;
    }
    break;
  }
  return false;
}
```

Several pieces of information comprise the method that leads to a concrete change. These pieces of information need a deeper explanation in order to fully understand the computational logic behind a change. We will describe computationally in the following parts how we calculate the logit, that is the probability of success, and how the Bernoulli distribution is created.

### The *createBernoulli(int, logit)* method

The RePast *Random* class contains number of different distributions, such as for example the normal distribution, the Pareto distribution or the Student distribution, but unfortunately it does not contain an already created Bernoulli distribution.
To attain our goal, we have to use the binomial distribution that "counts the number of successes in $n$ Bernoulli trials" (Verzani, 2005, 150). The binomial distribution has two parameters that need to be determined: the $n$ number of trials and the success probability, $p$. These two variables have been denoted $n$ and *logit* in the code below.

```
public double createBernoulli(int n, double logit){
  logit = calculateLogit();
  Random.createBinomial(n, logit);
  double change = Random.binomial.nextInt(n, logit);
  return change;
}
```

The toss of a coin is a usual example for illustrating the binomial distribution. Suppose that a success is to obtain "tails" when tossing a coin. The binomial distribution expresses the probability of successes tossing the coin n times.
If the number of trials equals 1, then the binomial distribution logically corresponds to a Bernoulli one. In other words, a Bernoulli distribution is just a simpler version of the binomial, which is exactly what is done in the *changePolicy()* method.
The next step is, thus, the calculation of the probability of success, defined as a logit and, with great imagination, it is called *logit* in the *calculateLogit()* method explained here below.

### The *calculateLogit()* method

Above, we explained that the Bernoulli trial is a simplification of the binomial distribution, with $n$, the number of trial, equal to 1. At this point the probability of success is still missing. Therefore, its calculation is the subject of the next explanation.

```
public double calculateLogit(){
  pcv = 0.0;
  changeVariable = 0.0;
  switch (model.neighborhood){
  case Model.MOORE:
    neighbors = model.world.getMooreNeighbors(x, y, false);
    numNeighbors = neighbors.size();
    changedNeighbors = countChangedNeighbors(x, y);
    numChangedNeighbors = changedNeighbors.size();
    changeVariable = calculateChangeVariable(x, y);
    double beta = numChangedNeighbors/numNeighbors;
    double z = (0.05+changeVariable+(politicalConstraints*beta));
    pcv = ((Math.pow(E, z))/(1+Math.pow(E, z)));
    break;
  case Model.VON_NEUMANN:
    vnNeighbors = model.world.getVonNeumannNeighbors(x, y, false);
    vnNumNeighbors = vnNeighbors.size();
    vnChangedNeighbors = countChangedNeighbors(x, y);
    vnNumChangedNeighbors = changedNeighbors.size();
    changeVariable = calculateChangeVariable(x, y);
    double vnBeta = vnNumChangedNeighbors/vnNumNeighbors;
    double vnZ = (0.05+changeVariable+(politicalConstraints*vnBeta));
    pcv = ((Math.pow(E, vnZ))/(1+Math.pow(E, vnZ)));
    break;
```

```
  }
  return pcv;
}
```

This method is the computational expression of the building of a logit as explained in Section 4.3.3 and, hence, corresponds to a slightly modified version of Equation 4.5.

A baseline of 0.05 is specified and the institutional constraints variable – the *politicalConstraints* parameter – is weighted with the number of changed neighbors – the (vn)beta variable[26]. To these two parameters we add a *changeVariable* variable; that is, the difference between the best effectiveness and the current effectiveness. If this difference is lower than 0, it reduces the logit, and thus the chance of successfully change policy, without completely excluding it, since the possibility for introducing policies that aren't totally effective is still open.

### The *calculateChangeVariable(int pos, int pos)* method

This *changeVariable* variable corresponds to the best effectiveness calculated among the similar neighbors that have already changed their policy subtracted with the current policy effectiveness.

```
public double calculateChangeVariable(int px, int py){
  double effective = 0.0;
  changeVariable = 0.0;
  switch (model.neighborhood){
  case Model.MOORE:
    effective = calculateBestEffectiveness(px, py);
    changeVariable = (effective-policyEffectiveness);
    break;
  case Model.VON_NEUMANN:
    effective = calculateBestEffectiveness(px, py);
    changeVariable = (effective-policyEffectiveness);
    break;
  }
  return changeVariable;
}
```

[26]The *vn* is for Von Neumann, meaning that the *vnBeta* parameter is the *beta* parameter calculated for the Von Neumann neighborhood.

If the result – the difference between the best effectiveness and the current effectiveness – is lower than 0, this means that the alternative policy is not so interesting and that the alternative is not really worth it. However, as already stressed in Section 2.3.4, a government may introduce a new policy even if it seems ineffective or is a bad idea, in order to satisfy its electorate, for example.

### The *calculateBestEffectiveness(int pos, int pos)* method

The search for the best effectiveness is a significant step in the change process, because this method allows the modification of the key variables of a policy – the effectiveness and the color of the (current and alternative) policy, since they represent the two main characteristics of a policy.

```
public double calculateBestEffectiveness(int px,int py){
  newColor = color;
  bestEffectiveness = getPolicyEffectiveness();
  switch (model.neighborhood){
  case Model.MOORE:
    Iterator it =findSimilar(px, py).iterator();
    while(it.hasNext()){
      Country effectiveNeighbor = (Country)it.next();
      double mostEffective = effectiveNeighbor.getPolicyEffectiveness();
      if (mostEffective > bestEffectiveness){
        bestEffectiveness = mostEffective;
        newColor = effectiveNeighbor.color;
        proximate(effectiveNeighbor);
      }
    }
    break;
  case Model.VON_NEUMANN:
    Iterator vnIt =findSimilar(px, py).iterator();
    while(vnIt.hasNext()){
      Country effectiveNeighbor = (Country)vnIt.next();
      double mostEffective = effectiveNeighbor.getPolicyEffectiveness();
      if (mostEffective > bestEffectiveness){
        bestEffectiveness = mostEffective;
        newColor = effectiveNeighbor.color;
        proximate(effectiveNeighbor);
      }
    }
  }
return bestEffectiveness;
}
```

First, the new parameters *newColor* and *bestEffectiveness*–the parameters that define the change – are set equal to the current one. Then we loop through the similar neighbors that have changed their policy to search for the most effective one. To do so, the most effective policy is set equal to the effectiveness of the policy of the neighbor the country is comparing itself with. If this latter is greater than that of the current policy of the country, then the country introduces this best effectiveness, which becomes the current effectiveness. Moreover, the country changes its color; that is, its new color becomes the color of its most effective neighbor, which increases its proximity to the neighboring country.

This increase of the proximity between the two countries is introduced at this point, since we assume that they have been mutually influenced by their exchange of information. Thus, we need to describe the routine of the increase in the proximity.

### The *proximate(Country)* method

This method has an important role to play, since it allows an increase in the number of similar features with the neighbor – Country n – of the proximity array.

```
public boolean proximate(Country n){
  int[] different = new int[model.numProximity];
  int numDifferent = 0;
  for (int i = 0; i < model.numProximity; i++){
    if(proximity[i] != n.proximity[i]){
      different[numDifferent]=i;
      numDifferent++;
    }
  }
  if (numDifferent > 0){
    int feature = different [model.getNextIntFromTo(0,
        numDifferent-1)];
    n.proximity[feature]=proximity[feature];
    return true;
  }
  return false;
}
```

The proximity array is defined as certain number of proximity features and each feature–each cell of the array–represents a dimension of the proximity (cultural, economical, religious and so forth, as defined in Section 2.3.3)–the *numProximity*.

This feature corresponds to an integer that can be randomly chosen in a possible number of traits, as explained in Section 3.4.7 and shown in Table 5.1
First, the number of different features is set to 0. Then we loop through the proximity array and, if we find two features with the same trait, then the number of similar features is incremented.
In a second step, if the number of different features is greater than 0; that is, if a difference between the arrays still persists, a feature is randomly chosen among the ones that are still different between the country and the neighbor and the country adopts this chosen feature, thus becoming more similar to the neighbor.

Here the conditions of change expressed in Section 4.3.3 have been computationally developed. We observe the three different steps for a change to occur, which can be stated as follows: The country is ready for a change, so it chooses an alternative policy, which is introduced if the condition of change is respected. In other words, when the conditions for a change are respected, the country updates the main policy parameters – the policy effectiveness, the policy preference, and the color associated with the policy.
IIn the next section, the different *Model* classes are described, since they constitute the computational location for the countries to interact. More precisely, in the *Model* class, we define the different steps of the simulation: ready, choose, change.
The *ModelGUI* class serves to create the different charts and grid used to see the evolution of the different parameters.

### 4.4.8 The different *Model* classes[27]

In RePast, as we have already mentioned, in order to run a simulation, we need to define the different Model classes. In this simulation, they are of three types: the simple Model class, the GUI[28], and the Batch.

[27]For more technical explanation, look at the RePast tutorial following the link http://repast.sourceforge.net/repast_3/tutorials.html.
[28]GUI means Graphical Users Interface.

### The *Model* class

In the *Model* class, we build the first skeleton of our own *Model* class, based on the RePast preconstructed *SimpleModel* class[29]. At this stage of the program, we define three basic tasks of the model:

1. The **setting-up** of the model: In the *setup()* method, we give a name to the model and the different initial values of the different parameters of the model. The chosen initial value are set as follows:

    - The neighborhood is defined as the Moore neighborhood; that is, the eight adjacent cells of a country;
    - The number of features that defines the proximity array is set to five, as in Table 5.1;
    - The number of traits; that is, the number of different value a feature can have is set to 10;
    - and the size of the world is set to 14. This means that the world is a 14 by 14 squared lattice. RePast comes with a number of predefined shapes for creating the world, such as grid and torus[30];
    - the time between elections is set to 5.

    At the setup, the countries are created and they fill the world.

2. The **building** of the model: The *buildModel()* method is used to construct the world as we wanted. The *World* object is built with its size and filled with agents. The different agent objects – the *Country* objects – are created with all their attributes (effectiveness, preference, color and so on) and randomly placed in the torus that represents the world.

3. The **time steps** of the model: At each *tick* the model performs some of the actions that are defined in the *step()* method.

[29]The *Model* class **extends** the *SimpleModel* class, in Java language.

[30]Let us remember from Section 4.3.2 that the world is created as a torus, meaning that the opposite corners are neighbors!

```
 public void step(){
  resetChange();
  changeParam();
  readyForChange();
  choose();
  change();
  reportResults();
}
```

Each method that is developed in the *step()* method are played one after another. The *reset()*, *readyForChange()*, *choose()* and *change()* methods all iterate through the agents in the world. Then they execute the specified behavior on each agent.

In the *readyForChange()* method, the *ready()* method defined in the *Country* class is called on the agents. The *chooseAlternativePolicy()* method in the *choose()* one, if *ready()* is true; and if *chooseAlternativePolicy()* is true, then the *changepolicy()* is executed in the *change()* method. And if this last one is true, the two methods that update the characteristics of the policy–*updatePolicyEffectiveness()* and *updatePolicyColor()* are completed.

When this kind of model is launched, it only displays what the programmer wants to be explicitly printed in the console through the *reportResult()* method.

```
public void reportResults(){
  System.out.println(getTickCount());
  for (int i = 0; i < numCountries; i++){
    Country country = (Country)agentList.get(i);
    System.out.print(country.toString());
    System.out.println();
    }
    System.out.println();
    for (int x = 0; x < worldSize; x++){
      for (int j = 0; j < worldSize; j++){
        Country country = (Country)world.getObjectAt(x, j);
        System.out.print(country.proximityToString() + "&");
    }
    System.out.println();
  }
}
```

This method stresses that, for each country, what is defined in the *toString()* and *promximityToString()* methods of the class *Country* must be printed in the console through the *System.out.print(String)*[31] method.

Thus, in the class *Country*, we define two *toString()* methods:

```
public String toString(){
  return "[Country(" + countryID + "):e:"+
  policyEffectiveness+",p:"+policyPreference+
     ",c:"+politicalConstraints+"]";
}

public String proximityToString(){
  String close = "␣";
  for (int i = 0; i < model.numProximity; i++){
    close = close + proximity[i];
  }
  return close;
}
```

The *toString()* method stresses that the different parameters of the country are printed in the console, along with the ID number, the effectiveness ($e$) of the policy and its preference ($p$), as well as the institutional constraints ($c$). The *proximity-ToString()* method does the same, except that it prints the proximity array of each country. In Figure 4.6, we have an example of the representation of the results of these two methods.

### The *ModelGUI* class

In this class, the idea is to create the different visual objects that appear on the screen, such as the display and the different graphs. In the GUI part of the model are also defined the different sliders used for the exploration of the model. The GUI part is created using the same methods as in Subsection 4.4.8. Since it inherits from the *Model* class, the initialization uses the same parameters, but, because we need more objects, such as the sliders for example, the different methods need to be overridden. In the *setup()* method, the different sliders are created and, in the *buildDisplay()* method, the different graphs and the display are created and their adaptation through time is executed in the *step()* method.

The different objects used to create the GUI part of the model are briefly explained

[31] The *println* part of that method only say that an empty line should be printed.


Figure 4.6: The results of the *reportResult()* method

here:

- The *OpenSequenceGraph* object: This object plots the different variables of interest of the model versus the *tick* – the time – of the model. The addition of the different sequences (the *Sequence* interfaces) will show the evolution of the variables.

- The *DisplaySurface* object: Displays correspond to the different possible graphical presentations of the agents and their environment.

- The *RangePropertyDescriptor* object: The different *RangePropertyDescriptor* objects allows researchers to alter the different parameters, since they create the available sliders in the settings window (Figure 4.7).

  On Figure 4.7, we can see the different parameters of our model – the size of the world, the neighborhood, the number of the features of the proximity array

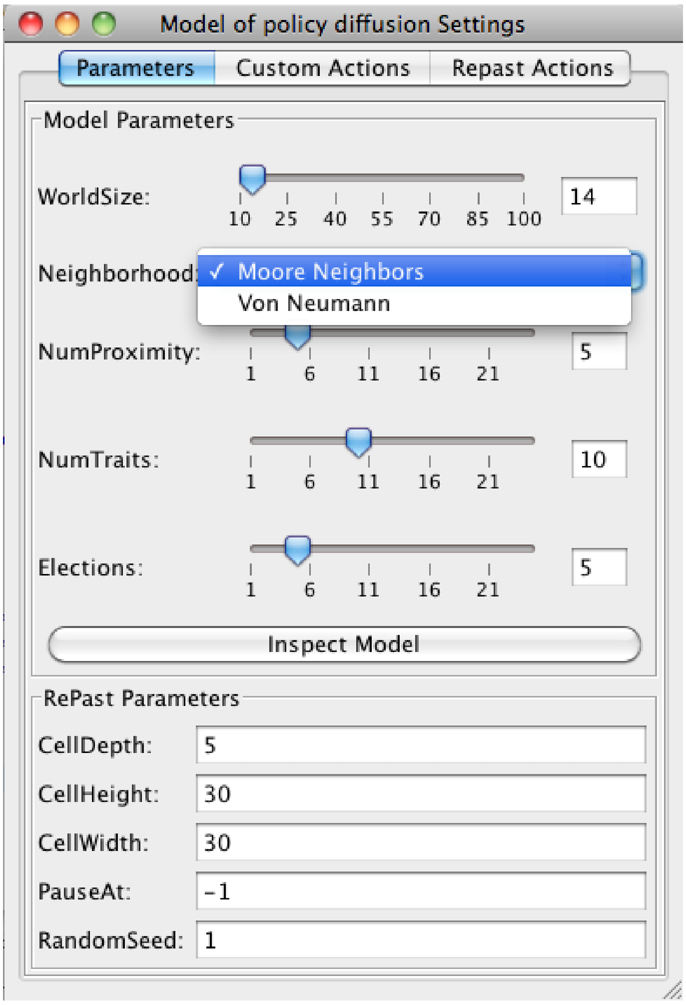



Figure 4.7: The parameters settings

and the number of the different traits per feature. The idea behind the use of the sliders is that these parameters should be altered in order to validate the model.

### The *ModelBatch* class

Batch runs of simulations are done usually for long/large simulations. These types of models are created in order to collect the different data. The main parameter of this class of models is the *DataRecorder* object.

With the instantiation of this object, the file to write the data out is defined, as well as the name of the file. For example, in my model, the data is stored in saved as data.csv files.

## 4.5 Conclusion

What first comes to mind at the beginning of this chapter is that we need some technological tools in order to be able to develop computational agent-based models. Besides the obvious, a computer, we need some software[32]. More precisely, we present Eclipse as the IDE used to write our code lines, code that was developed using a Java-based existing toolkit, RePast. It eases the building of our model, as it relies on two basic specificities of object-oriented programming – inheritance, the possibility to navigate through the axis generalization/specialization for the development of the different classes, and polymorphism; that is, the possibility to override the different methods. Moreover, RePast comes with an API of already built classes and methods for the development of computational ABMs.

In Section 4.4.8, we highlight the development of agent-based models based on simple rules that can be summed up in a minimum of sentences. In this chapter, we explain the different steps of our model of policy diffusion. More precisely, we highlight the three main phases a country has to go through in order to change its policy. These three steps are described here:

1. A country is ready for change if the effectiveness of the current policy is lower than the preference for it;
2. If the first step is true, the country searches for the most effective policy among the like neighbors that have already changed their policy;
3. If it has found the most effective policy, the country introduces it.

Therefore, we have described the different methods we built and that a country must follow in order to change its policy. These methods – which correspond to the sentences enumerated here above – are: ready(), choose(), change(). Furthermore, we explain more in detail the different components of these methods, which express the different conditions of change that are expressed at the theoretical level in Section 4.3.3. Thus, all the interactions are defined within different *Model class*, but

[32] In its large acceptance, it only means *program*

the conditions for these interactions to take place are defined within the *Country* class. In other words, these conditions are embedded in the countries at the creation. In order to have a comprehensive view of the links between the theoretical and computational development of the condition for a change to occur, a summary has been built into Table 4.3.

In addition, throughout the description of the implementation of the model, the particular place of the neighborhood has been emphasized. The Moore and Von Neumann neighborhoods represent the geographical environment of a country. Nevertheless, as stressed in Section 2.3.3, the neighborhood is more than just the geographical neighbors. Following this, we have developed a proximity array that represents the other dimensions of the proximity and that is used to increase the similarity between the countries. Hence, the integration of the neighborhood in the conditions of change is, as expressed at theoretical level, a good approximation of the interactions that exist among the countries and the proportion of neighbors that have changed their policy as a good estimation of the weight they can have on the internal decision.

It should now be clear that computational ABM is a well-suited tool for the study of policy diffusion, since the process of diffusion has the main characteristics of a computational agent-based model; that is, heterogeneity (heterogeneous agents), autonomy[33], explicit space (the world), and bounded rationality (the neighborhood). Now that the model has been programmed, we need to run it and see what happens when these virtual countries interact. This is the topic of the next chapter.

[33]Since we assume that coercion plays no role in our model, there is no central authority!

| Conditions | Formulae | Computations |
|---|---|---|
| A country is ready for a change | | *ready()* |
| A country chooses an alternative policy | | *chooseAlternativePolicy()* |
| The choice variable is calculated as follow | | |
| The difference between the average effectiveness of the neighbors that have changed their policy and the current effectiveness | $\frac{\sum_{i=1}^{n} E_c}{N_c}$ | *calculateMeanEffectiveness()* |
| | $N_c$ | *countChangedNeighbors()* |
| | $CV = \Big( \big( \frac{\sum_{c=1}^{n} E_c}{N_c} - e \big) \big( \frac{N_c}{N} \big) \Big)$ | *calculateChoiceVariable()* |
| The condition for a change to occur | $CV >$ threshold | choiceTreshold= Random. *uniform*.nextDoubleFrom To(-2.0,2.0) |
| The change of the policy | | |
| The average effectiveness among the similar neighbors | $\frac{\sum_{s=1}^{c} E_s^c}{N_c}$ | *calculateBestEffectiveness (int, int)* |
| The weighted institutional constraints | $\big(\text{institutional constraints}\big)\big(\frac{N_c}{N}\big)$ | *politicalConstraints (vn)beta* |
| The increase in the proximity | $\boxed{3}\boxed{7}\boxed{9}\boxed{4}\boxed{6} \Rightarrow \boxed{3}\boxed{6}\boxed{8}\boxed{4}\boxed{5}$ | *proximate()* |
| The transformation of the change variable as a logit | $\text{p(change)} = \frac{e^{(\text{change})}}{1+e^{(\text{change})}}$ | *calculateLogit()* |
| The Bernoulli trial | $f(x;p) = \begin{cases} p & \text{if } x = 1; \\ 1-p & \text{if } x = 0; \\ 0 & \text{otherwise.} \end{cases}$ | *createBernoulli(1, logit)* |
| The introduction of the new policy | | *updateEffectiveness updatePolicyColor()* |

Table 4.3: Summary of the computational implementation

# Chapter 5

# The results of the model

## 5.1 Introduction

As seen in Chapter 3, agent-based modeling has a great advantage over more conventional methods, as it can take into account the nonlinearity[1] and the interdependencies that usually exist in social life. Since diffusion involves interdependencies between countries and since computational agent-based modeling provides the methodological tools usually used to study these interactions, it should now be clear that one of the best methods to complete the traditional methodological arsenal of the political scientist, and that is used in this study of the process of diffusion, is computational agent-based modeling.

In Chapter 4, the algorithm of the model and its computational implementation have been described. The time has come to let the different countries interact and try to see what will *emerge* from their interactions.

Among the different advantages stressed in Section 3.4.5, one of great interest for our purpose is that computational agent-based modeling enables the researcher to rerun history as many times as needed. Moreover, it allows the researcher to run the same history with slight changes in the parameters. Thus, the observed differences should come only from the fact that the countries interact with a greater (lesser)

[1]As explained in Section 3.3.2, non-linearity means that the whole is more than just the sum of its part. More precisely the different interactions cannot just be added to obtain the result at the macro level.

number of neighbors, and not from exogenous disturbances.

Consequentially, one should alter the different parameters to see how the model would evolve. As a result, a comparison between multiple runs, each with different initial conditions, is possible. This can be done through the batch mode, a kind of model that allows for the collection of data[2]. However, if one has to change the different parameters *by hand*, it would take a lot of time to obtain the needed data. That is why we have used the **M**odel **E**xploration **M**odul**E** (MEME).

This chapter is structured as follows. First, we will explain the results of an *emblematic* run for both neighborhoods. More precisely, we will explain the evolution through time of the process of diffusion and its consequences, when the countries are interacting (section 5.2). Moreover, what an emblematic run is, and why it is considered emblematic, must be emphasized. Section 5.3 will bring an overview of the results when the different parameters are modified. More precisely, depending upon the parameter change, we will give and compare the results of the dependent variables, namely the number of regions and the average effectiveness. A conclusion will sum this up and extend the discussion on diffusion by trying to draw some parallels between the *in silico* and *real* worlds

In other words, we will try to see how our computational model would help comprehend diffusion in the real world. To achieve this goal, the first thing to understand is how the model should behave when the countries comprising it interact with each other in the different environments – the Moore and Von Neumann neighborhood.

## 5.2 The results of an emblematic run

This first section describes the behavior of the countries in the model. In other words, this section is dedicated to the description of the evolution of the interactions in an *emblematic* run and to the results that arise at the macro level.

In order to obtain some data, we launch batch runs with the Moore neighborhood and the Von Neumann neighborhood, each run characterized by the same initial

[2] See section 4.4.8

conditions. Thus, since one of the advantages of computational agent-based modeling, for testing theories, is to let the different agents interact and see how they evolve, this section will highlight auto-organization in this world. In other words, the idea is to emphasize what *emerges*[3] from these interactions.

Before explaining the history of the countries, a brief clarification of what *emblematic* means is required. Since, at the start of a simulation, the evolution of the interactions – the comprehensive history of the countries – is mainly unknown, the settings of the parameters at the beginning of a run define its frame. Thus, *emblematic* refers, here, to the setting of values that are interesting enough to let fascinating patterns emerge and that allow the linkage with some real-world scheme.

Consequently, the size of the world is arbitrarily fixed at 196 countries; that is, a 14 by 14 grid. This corresponds more or less to the size of the real world, as we already noted in Section 4.3.2. The notion of geographical contiguity is expressed by the different neighborhoods – the possible neighbors a country can interact with – and, to demonstrate the need to extend the concept of neighborhood to other dimensions than only geography – as listed at the beginning of Section 2.3.3, a proximity array composed of five different cells representing these dimensions has been created.

Moreover, each cell of the proximity array is filled with a possible trait that is arbitrarily predetermined at 10[4]. Table 5.1 shows how one can see the different dimensions of the emblematic array, since we can identify at least five main aspects that are important in the context of policy diffusion (see e.g. Boschma, 2005; Beck, Gleditsch and Beardsley, 2006).[5]

[3]For a reminder of the different concepts of computational agent-based modeling, and specifically complexity and emergence, see section 3.3.2 and 3.3.3

[4]In Java, the first position in an array is identified as 0, not 1.

[5]Of course, there may be more dimensions; that is why we allow researchers to alter their possible numbers.

| Dimensions | Number of possible traits | Literature |
|---|---|---|
| Budget Policy | From public debts (0) to public surplus (9) | Gilardi and Wasserfallen (2010) |
| Economy | From market-oriented (0) to state oriented (9) | Martin (2009) |
| Political system | From autocracy (0) to democracy (9) | Gleditsch and Ward (2006) |
| Demography | From little populated (0) to populated (9) | Volden (2006) |
| Ideology | From left (0) to right (9) | Grossback, Nicholson-Crotty and Peterson (2004) |

Table 5.1: Examples of dimensions in the *emblematic* run

We reproduce here the example of the proximity array of Section 4.3.1:

| | | | | |
|---|---|---|---|---|
| 3 | 7 | 9 | 4 | 6 |

As a result, we can portray the corresponding country as follows: it has public debts and a rather state-oriented economical system. Moreover, it is a democracy governed by the right and is sufficiently populated.

Now that the initial values of the different parameters have been explained, we will run the model and see what happens when the different countries interact. More precisely, we will see the emergent results of these interactions.

### 5.2.1 How does the model evolve?

In Section 3.4.1, we define the Moore neighborhood as the eight adjacent cells (the red cells) and the Von Neumann neighborhood as the 4 neighbors (the blue cells) of the analyzed agent (the green cell), as shown in Tables 5.2 and 5.3.

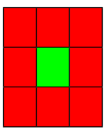


Table 5.2: The Moore neighborhood

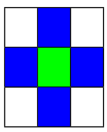


Table 5.3: The Von Neumann neighborhood

Thus, Tables 5.2 and 5.3 specify the interacting area of a country.

This computational world corresponds to an aggregation of several such neighborhoods, and the toroidal representation of the world as defined in Section 4.3.2 is a particular form of network in the sense that each agent is connected with its eight (four) closest neighbors, each of its neighbors is identically linked with its eight (four) closest neighbors, and so forth. As the world is wrapped around[6], the agents in such a world are strongly interconnected.

In Figure 5.1, on which we can see the grid that represents the world (on the left), the different graphs that display the count of the number of regions (at the center), and the evolution of the average effectiveness (on the right) within the world, as well as the RePastJ toolbar (at the top of the toolbar) that allows the different manipulations of the world, such as start, stop, pause, and so forth; gives a picture at the start of a run; that is, the capture of the screen when the *ModelGUI* class is launched; or, more precisely, when the different objects of the *ModelGUI* class are created at the initialization.

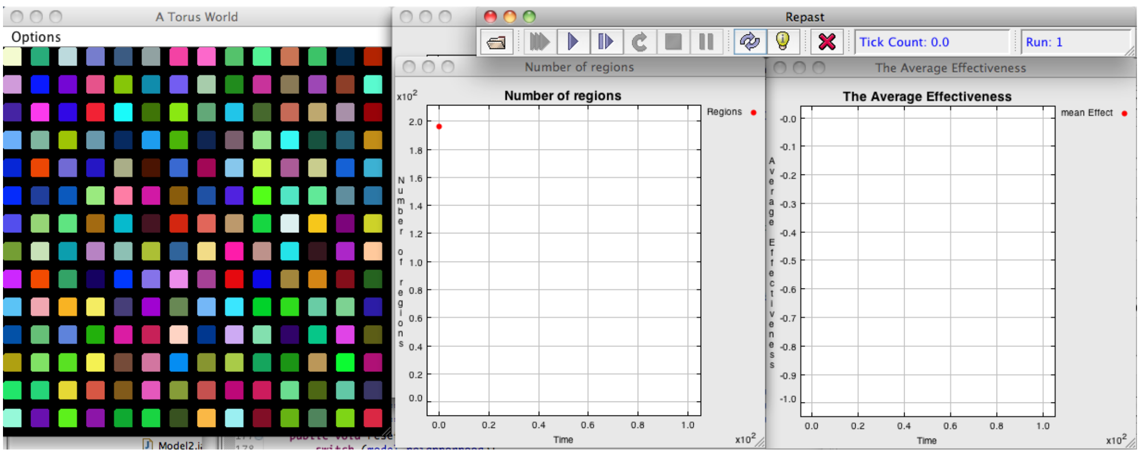



Figure 5.1: The start of a run

With the RePast toolbar, there are two different types of starting the simulation, one iteration after another (), or once and for all (). Anyway, a controlled step-by–step simulation or a simulation launched one time give the same results, as

[6]See Section 4.3.2 for an explanation of the advantages of the use of a toroidal shape of the world.

shown in Figures 5.2 and 5.3

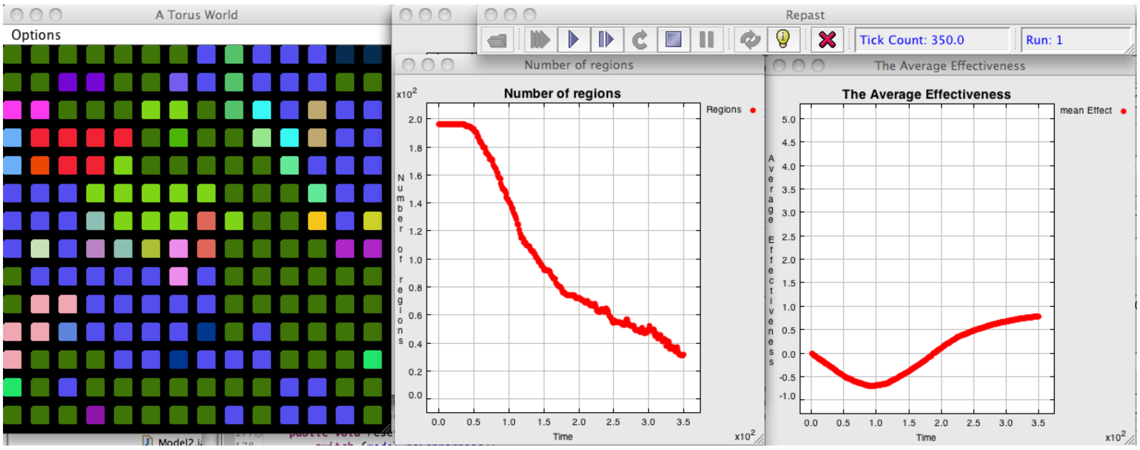



Figure 5.2: The end of a run (Moore neighborhood)

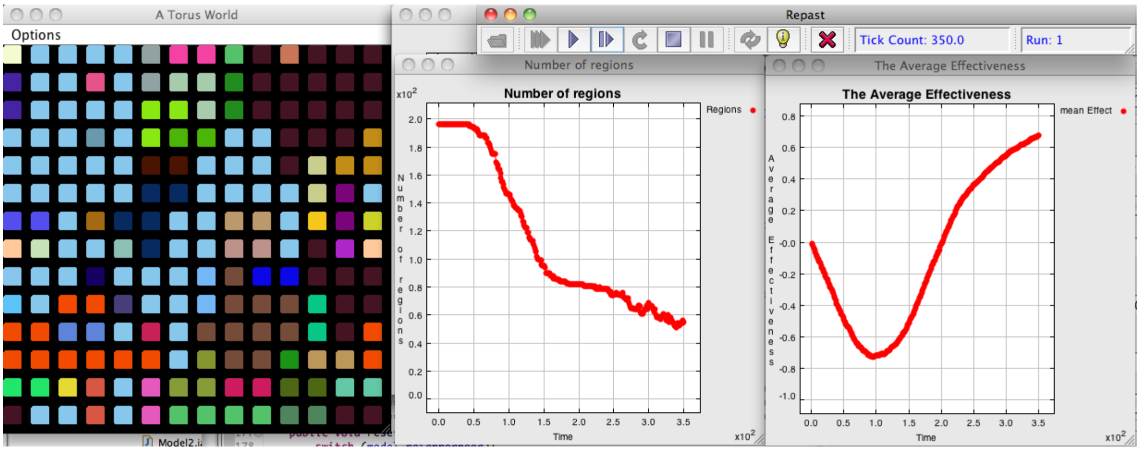



Figure 5.3: The end of a run (Von Neumann)

What we see in Figures 5.2 and 5.3 is the results of a simulation when the countries have interacted with their neighbors 350 times. The number of steps in a simulation is, here, arbitrarily fixed[7]. To attain these results, at each time step, each country goes through the *step()* method of the *Model* class, which was presented in Section 4.4.8.

Let us recall that this method works as follows: At each time step, each country first resets to 0 its count of neighbors that have changed their policy, meaning that an agent that has changed its policy at time $t-1$ is no longer considered a changed

[7] 350 has been defined, as it is sufficiently high to let interesting patterns emerge.

country at time $t$. In other words, at time $t$, it is again at odds with changing its policy. Then the effectiveness and the preference variables are updated. With the third step begins an evaluation of the country's situation, with the comparison between the effectiveness of and the preference for the current policy. If the effectiveness level is lower than the preference level, then the country begins to choose an alternative policy by looking at what its neighbors do.

If the country has chosen an alternative and more effective policy, the next step is to introduce this policy, which corresponds to the most effective policy among the similar neighbor(s)[8]. The last method allows the printing of the different parameters in the console as explained in Section 4.4.8.

Thus, when the countries have interacted a certain number of times (350 in our case), we see the development of some strong patterns at the macro level. Moreover, as seen in Section 3.3.3, we assist with the emergence of the macro phenomenon; that is, diffusion as an unexpected result at the macro level from micro interactions, which will be explained according to the different aspects of the definition of diffusion explained in Section 2.3. More precisely, the following section will emphasize the evolution of the process of diffusion from the computational model point of view:

- Diffusion and the temporality: In Section 2.3.2, we present the classical theoretical evolution of the process of diffusion as a S-shaped curve. More precisely, the proportion of adopters through time follows this kind of curve. Therefore, at the beginning of the process, no country has changed its policy, meaning that the number of adopters equals 0. When the process unfolds, the number of adopters grows following a S-shaped curve.

  Moreover, in the model, as we concentrate not on the countries, but on the regions – a group of countries characterized by the same color – the result of the process is an inverted S-shaped curve, as shown in Figures 5.4 and 5.5, which represents the decreasing number of regions.

[8] We do not go in more depth here in the different steps toward a change, since these steps have already been described, at the conceptual level, Section 2.6, as an algorithm Section 4.3.3, and at the methodological level Section 4.3

In this figure, we see that the number of regions decreases through time. This means that the number of policies in the world are diminishing. Thus, some policies are spreading whereas others, supposedly less effective, simply disappear. In other words, the number of countries that changed their policy for a more effective one is increasing and the countries are aggregating in clusters that are defined by their new policy. Consequently, the progress of the process of diffusion results in the diminution of the number of regions, because they change their color for that of the alternative policy.

Furthermore, in Section 2.3.2, we have stressed that each mechanism of diffusion has its own temporality (see e.g. Dobbin, Simmons and Garrett, 2007) and that each mechanism has its own duration (Shipan and Volden, 2008). As a result, the inverted S-shaped curve can be segmented into three parts.

In other words, as in the *traditional* S-shaped curve, two points where the slope of the curve changes can be observed. After a slow takeoff at the beginning of the process, the first point corresponds to an acceleration of the process of diffusion; that is, the number of policies is decreasing sharply while, conversely, the number of changing countries is increasing sharply. This is consistent with the consequences of bandwagon pressures; that is, the more countries that have changed their polices the higher the incentive to change, which is an often-used concept in computational agent-based modeling (see e.g. Abrahamson and Rosenkopf, 1997; Elkink, 2009; Cederman and Gleditsch, 2004).

The second point corresponds to a slowing down in the process of diffusion. At this point the decrease in the number of regions is slowing down. In other words, the curve of decreasing number of regions flattens because it moves toward the maximum number of possible adopters, conversely, the minimum number of possible regions.

Contrary to the traditional S-shape, where the path at the beginning of the model – the takeoff of the curve – takes more time than at the end, here we

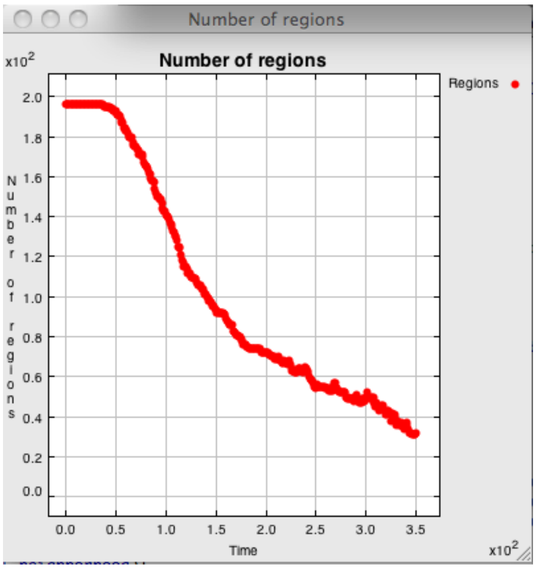



Figure 5.4: The number of regions with the Moore neighborhood

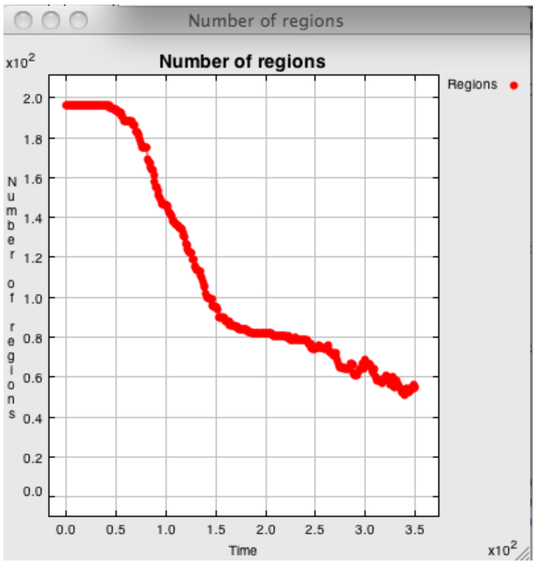
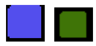



Figure 5.5: The number of regions with the Von Neumann neighborhood

see that the takeoff appears rather early in the model and has a very short life in opposition to the path at the end of the process (Shipan and Volden, 2008). Consequently, learning should have a very short life; imitation, a longer life. Moreover, the fact that the slope of the curve gets very sharp at the beginning of the process comes from the strong interconnection of the countries in the world, which facilitates the spread of the change (Rogers, 2003). However, the disentanglement of these different aspects are rather difficult since they are interrelated and the model provides a global view of the evolution of the process. Thus, the fact that the number of regions in the world is diminishing implies the political clustering of the world, since a policy is defined by its color. The next point will show how this is spatially expressed.

- Diffusion and the spatiality: In Section 2.3.3, we emphasized that the process of diffusion occurs through space and that one of the main points to pay attention to was the definition of the neighborhood, since it is defined by more than just the geographical border. Consequently, in Section 4.3.1, we explained the computational development of the proximity array as a complement of the geographical neighborhood and, in Section 4.4, its computational implementation.

Furthermore, the effect of the interactions on the shape of the space is shown in Figures B.1 and 5.6[9]. In this latter figure, we see the emergence of two kinds of policy. Since a policy is expressed by its color, the two emerging policies are



when the Moore neighbors are taken into account and when the countries interact with their Von Neumann neighbors

The emergence of clusters as a result of the process of diffusion calls for at least the following remark:

- We have just seen that the evolution of the number of regions is influenced by bandwagon pressures. As a result, the slope of the curve tends to get steeper, since the more countries that have changed their policy the higher the pressure to change, and the lower the number of regions, meaning that few policies spread over the countries, leading to the convergence.

  We expressed in Sections 2.3.2 and 2.3.3, that convergence is one possible result of the process of diffusion. However, most of the time, the convergence is not total; that is, divergence still exists in the convergence. At the size of our computational world (Figures 5.6 and 5.7), we find the same result; that is, the emergence of two main policies besides the existence of smaller regions with other policies. In that way, we can observe, from Figure B.1, the limited convergence that exists in the world, which also comes from the proximity array.

[9]The same explanation applies in the case of the Von Neumann neighborhood as shown in Figures B.2 and 5.7

If we look closely at the proximity arrays, we can observe that only a few countries are totally alike; that is, only a few countries have fully converged. Thus, most of them keep at least one particular differing feature. Consequently, even if the countries are clustering – converging – at the political level, they are still keeping some divergent features, whether economical, cultural, or ideological, in the other dimensions.

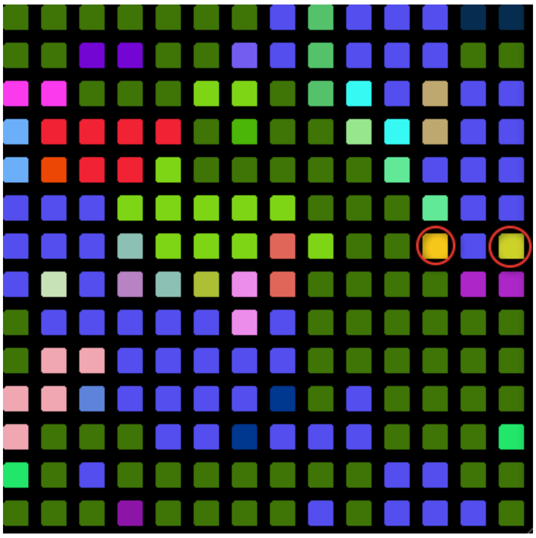


Figure 5.6: The clusters with the Moore neighborhood

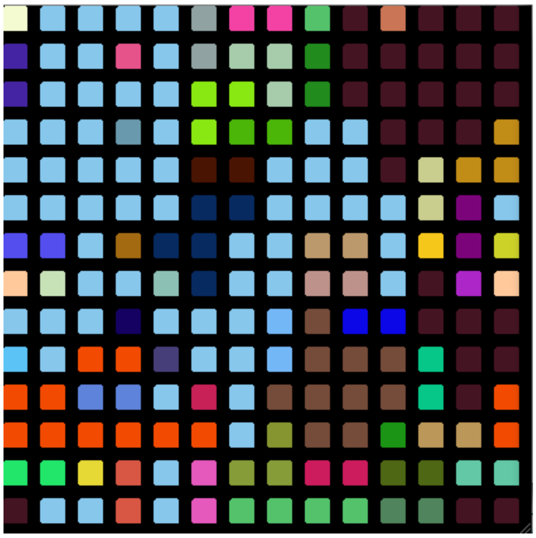


Figure 5.7: The clusters with the Von Neumann neighborhood

The fact that, in Figures 5.6 and 5.7, we clearly see the emergence of two main policies is in line with several phenomena observed in the real world. Several theoretical and empirical works have emphasized this result. For instance, the development of welfare states, as shown by the famous "three worlds" typology developed by Esping-Andersen (1990) is characterized by significant geographical clustering, as well as the diffusion of democracies that spread under the force of bandwagon pressures (Elkink, 2009; Gleditsch and Ward, 2006), or the development of different health care systems (Palier, 2004). Moreover, the existence of some main types, such as NHS or liberal in the health care domain, for instance, does not mean that countries with the same health care type are totally identical. In sum, the results shown on the grid are derived from the ones of the temporal diffusion dimension. The fact that the decreasing number of regions is a result of the increase in the number of adopters means that few policies gain in importance and spread through time in the world.

The next part is devoted to the explanation of how the conditionality of the process of diffusion is expressed in our world; as we have seen, diffusion is conditional because of the temporality of the different mechanisms of diffusion and the internal differences that exist between the countries.

- Diffusion and the conditionality: Conditionality has been defined in Section 2.3.4 as the fact that the countries may be differently affected by the mechanisms of diffusion because they are facing different internal conditions. In other words, different countries do not go through the process of diffusion in the same manner. Consequently, internal factors determine the way a country interacts with its neighbors, as explained in Section 2.4. By so doing, these factors shape their environment, as is shown in the explanations of the temporal and spatial results.

  Moreover, the environment – the results of the evolution of the clustering in the world – shapes the interactions between the agents, since it influences the choice of the effectiveness and the introduction of the alternative policy through the calculi of the average effectiveness and the share of neighbors that have already changed their policy as shown in Equation 4.4. In other words, the behaviors of the agents can shape the environment, which in turn shapes the agents, since the evolution of the process of diffusion modifies the neighborhood. This circle of agents who influence their environment and have been influenced by it has been labeled *stigmergy* (Section 3.4.5). In short, the conditionality of the process of diffusion corresponds to the *diffusionist* expression of stigmergy.

  Furthermore, from the second order emergence concept[10], not only does the history of the countries draw the outlines of the world, but also the history of the world itself. In other words, the history of the agents determine the

[10]See Section 3.3.3 for a reminder

clustering of the world and this clustering – the history of the world – results in the J-shaped curve of the average effectiveness of the world, as shown in Graphs 5.8 and 5.9.

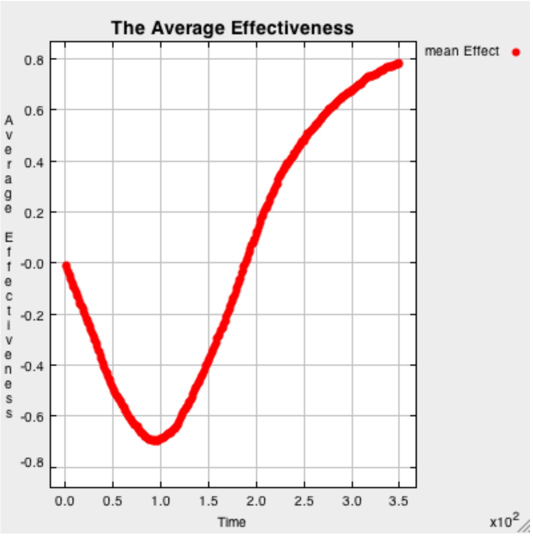



Figure 5.8: The average effectiveness with the Moore neighborhood

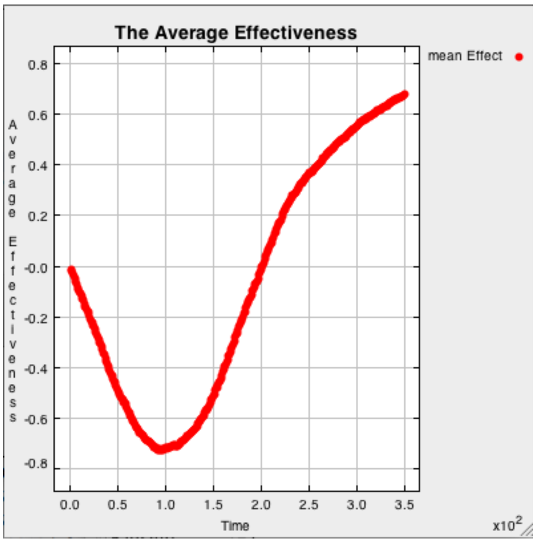



Figure 5.9: The average effectiveness with the Von Neumann neighborhood

This curve highlights the fact that the average effectiveness decreases early in the process of diffusion, which is not surprising, since, for each country, effectiveness is likely to decrease,[11] up to a point where the effective policies are numbered enough to favor the introduction of a more effective policy, which is another way of expressing the results of bandwagon pressures.

Put differently, the number of countries that have changed to a more effective policy are sufficient to induce the average effectiveness level towards more effective. Consequently, the fact that the average effectiveness in the world decreases at the beginning of the process suggest that the leaders are outnumbered by the countries that have not changed their policy. In other words, the number of ineffective policies is greater than the number of effective policies and the lowest point in Graphs 5.8 and 5.9 corresponds to the point where the effective policies in the world are numerous enough to induce countries

[11] Remember, from Section 4.3.1, that at each step a variable randomly drawn from a normal distribution with mean -0.01 is added to the policy effectiveness.

towards the introduction of even more effective policies.

Furthermore, if the effectiveness is lower than the preference[12], a country introduces the most effective policy among its similar neighbors that have already changed their policy. Nonetheless, this policy does not necessarily correspond to the most effective policy per se. For instance, a country can still introduce an alternative with an effectiveness level below 0, but greater than the old one. However, what is striking here is that despite this decreasing tendency, we assist at the emergence of a rather maximal average effectiveness.

From the definition of conditionality established in Section 2.3.4, we saw that countries facing the same degree of interdependence with the same neighbor may be differently affected by this interdependence, since they are characterized by varying internal factors. This is expressed in Equation 4.4. The average effectiveness of the similar neighbors ($\frac{\sum_{s=1}^{c} E_c^s}{N_c^s}$) and the share of neighbors that have already changed their policy ($\frac{N_c}{N}$) correspond to the expression of the degree of interdependence. In other words, if two countries face the same share of neighbors and/or the same average effectiveness, their current effectiveness and their institutional constraints *condition* the probability of success of the introduction of the alternative policy. Thereby, the country with the lower effectiveness and/or the lower institutional constraint level has a higher chance to introduce the alternative policy. In other words, a country with fewer veto players and a ineffective policy sees its chance of introducing a more effective policy increasing. As a result, at the global level, the average effectiveness decreases at the beginning of the process, since fewer countries have changed their policy for a more effective one and the alternative policies are still not the most effective ones.

Nevertheless, global effectiveness, under the impact of bandwagon pressure, increases toward total effectiveness. However, the curve slows down at the end of the process, never attaining total effectiveness because, as shown in the

[12] A quick look at Section 4.3.3 will refresh the knowledge on the different conditions for a change to occur.

explanation of the results of the temporality and the spatiality of diffusion, some countries/regions remain ineffective.

In sum, when facing different internal factors, the countries are unlikely influenced by their neighborhood. As a result, the average effectiveness in the world tends to follow a J-shaped curve. In other words, at the beginning of diffusion process, the world becomes ineffective. This seems to better correspond with the results of the introduction of a new policy. In Meseguer's words (2006*a*, 42), "(...) many innovations produce results along a J-curve; that is, immediately after implementation results are bad or even a recession is induced, and only after a while do policies deliver good results."
Moreover, it is important to stress that one should fight again the propensity to compare the inverted S-shaped curve with the J-shaped curve, even if the evolution of both curves is influenced by bandwagon pressures. Differently said, these two graphs express different aspects of the process of diffusion, owing to the fact that Figures 5.4 and 5.5 highlights the number of regions, as defined by the clusters of alike colors (represented in Figures 5.6 and 5.7); that is, the few policies that diffuse more, and Figures 5.8 and 5.9 show the results of this diffusion on policy effectiveness. In other words, even if the effectiveness tends to decrease, the most effective policies diffuse. These two effects do not have the same time horizon.
Now that the behavior of the model has been explained at the more technical level, we will, in the next section, discuss the main results considering that the process of diffusion is, here, driven by the mechanism of learning.

### 5.2.2 Discussion: learning and the different outcomes

Up to now, we have explained the evolution of the process of diffusion on the technical level. In line with the theoretical expectations and empirical evidence, the results of the process in this computational world are threefold: partial convergence (following a S-curve); clustering around a few policies; and, since not all the countries have the same time horizon for the choice and the change, the average effectiveness follows a J-shaped curve that goes toward maximum effectiveness. However, the process has not been *categorized* and, furthermore, the implications of these results for the study of diffusion must be emphasized. In other words, the meanings of these results for the understanding of diffusion must be highlighted.

So far, this question was answered at the computational level, as we have described how the behaviors of countries when they interact in order to update their belief on the different consequences – effectiveness – of a policy lead to the emergence of policy diffusion.

Nevertheless, the mechanism(s) that is (are) at play is (are) difficult to highlight, as has already been stressed. For instance, Gilardi (2010) when studying the diffusion of unemployment benefit retrenchment in the Organization of Economic Co-operation and Development (OECD) countries, has developed the dependent variable as probable imitation, meaning that interdependencies are characterized by imitation. Then, the essence of these interdependencies can be *interpreted* as learning.

In the model presented in this work, bandwagon pressures; that is, the strength of the share of neighbors with an alternative policy, have a great impact throughout the process of change. We have seen that diffusion is an emergent phenomenon that occurs between interacting agents that aim at changing their current policy by seeking the best alternative solution(s). Therefore, to characterize diffusion, different mechanisms can be highlighted. The evolution of bandwagon pressure for one part – and the study of the interplay between the average effectiveness and the current one, on the other – may be used for the study of these mechanisms. Since the difference between the average effectiveness and the current one is used to update beliefs, bandwagon pressures are the variable that must be used to help the disentanglement of the different mechanism. More precisely, when only a few neighbors have changed their policy, bandwagon pressures are weak, and the update of the beliefs is more prominent in the decision of change, meaning that learning is the main mechanism at play. Thus, when more and more neighbors change their policy, their weight is increasing. When all neighbors have introduced the same alternative policy, bandwagon pressures have a greater weight than that of the beliefs update, meaning that emulation is now at work. However, as already mentioned in the above section, since we only have data at the global level, this interplay between the different mechanisms cannot be analyzed.

Anyway, if we assume that the comparison of the average effectiveness of the (similar) neighbors that have a new policy and the current one is used to reinforce the beliefs on the necessity of change, and as we do not have the number of changed neighbors at the country level, we will concentrate our discussion on the update part, since it can be characterized as learning. In other words, the process in, and the results of, the computational model can be seen as diffusion driven by learning; more precisely, bounded rational learning, with Bayesian updating, since the computational countries updates their beliefs on the outcomes – the effectiveness – of the alternative policy by looking at the experience of their neighbors, estimated as the difference between the average effectiveness of the similar neighbors that have changed their policy and the current one. Hence, this corresponds to the definition of learning given in Section 2.5.1.

Learning, as the main driver of policy diffusion, is one of the most-studied mechanisms (see e.g. Volden, 2006; Braun and Gilardi, 2006; Gilardi, Füglister and Luyet, 2009). This research subscribes to that trend. More precisely, in this model, we develop a bounded rational version of learning, with Bayesian updating. Therefore, the operationalization of this version of learning calls for at least three remarks:

1. *Boundedness*: The process of diffusion is, here, driven by the bounded version of learning since a country can interact only with its closest neighbors, depending on the type of neighborhood (Moore or Von Neumann). As already stressed in Section 4.1, learning, in this sense, corresponds to the view of an agent;

2. *Rationality*: Each country learns from its neighbors in a rational way, as each seeks the most effective policy, considering the local interactions induced by the boundedness of learning;

3. *Bayesian* learning: Bayesian learning means that, at each time step, a country updates its beliefs on the (con)current policy according to more consistent data. In other words, a country gives greater value to the policy experience observed in other countries than to the prior beliefs it had on the consequences of this policy (Meseguer, 2006*a*), as the process unfolds. Indeed, at each time step, the effectiveness of the current policy is estimated in comparison with the policy outcomes – the effectiveness – of the neighbors.

Now what can our model say about learning, according to the different results of the process of diffusion?

**Learning, partial convergence and clusters**

Since we have here a broader description of policy diffusion process; that is, between countries, we assist in a partial convergence at the political level. In other words, our world is diverging at the global level and converging at the regional level (Meseguer, 2006*a*; Axelrod, 1997*b*). Thus partial convergence and clustering are the *results*, the outcomes of the *process* of diffusion (Gilardi, 2011). A country that changes its policy becomes more similar to the *sender* of the relevant information on the alternative policy. In other words, a country becomes more like its neighbors at the political level, leaving other dimensions divergent.

Even in a strongly interconnected world, diffusion by learning can give rise to politically divergent regions (Figures 5.6 and 5.7). And at a more micro level; that is, at the country level, from Tables B.1 and B.2, convergence[13] is the rule. However, even if learning leads countries to convergence in other dimensions than the political one, divergence persists. Put differently, from the micro-level interactions, diffusion as driven by learning emerges, and the outcomes of such a process are in line with theoretical expectations and empirical evidence, resulting in global divergence and local convergence (Axelrod, 2003; Meseguer, 2006*a*).

The classic result of the S-shaped curve in policy diffusion literature is convergence,

[13]At least one trait is still different.

since at the end of the process the curve flattens when the number of potential adopters becomes very small. Moreover, in this representation, all potential adopters are at the same odds of adopting (Berry and Berry, 2006). As a result, the curve trends toward total adoption. With our model, the fact that learning is bounded induces different chances of success and, thus, partial convergence. Moreover, convergence is, as already mentioned, partial between countries, but also between regions, as expressed by Figures 5.6 and 5.7.

Moreover, Meseguer (2006*a*) has questioned the idea that countries in a region learn from a prominent example and, at the world level, countries may learn from an entire successful region. Thus, by challenging the regionalization of policy diffusion, she emphasized local convergence in globally divergent world. For instance, at a country level in Latin America, Chile is a prominent model to learn from and, at the regional level, the group of nations called the *Asian Tigers* may be relevant.

Nevertheless, the results of the computational model suggest that learning does not need a prominent example in order to drive the process of policy diffusion. The estimation of average effectiveness is a sufficient option. Moreover, the different political regions, as unfolded from the process of diffusion, do not have stable borders. In other words, according to the different estimations of policy effectiveness, the different regions are permeable to the different emergent policies, meaning that countries at the borders do interact with each other, with the result of moving the policy boundaries.

In the introduction, we highlighted the existence of the phenomenon of diffusion using the example of the spread of antismoking bans, an example well-suited for our purpose. Thus, for instance, antismoking bans have spread throughout the European Union. This process started in Ireland in 2004[14] and reached Italy in 2005. A few months after its introduction in Italy, this policy passed through the southern border of Switzerland, since the Tessin canton was interested in the introduction of antismoking bans and promulgated a cantonal law to that effect in 2007. Then this

[14] `http://www.otc.ie`

idea spread throughout the country,[15] ending in the creation of a Swiss federal law in May 2010. Thus, following Meseguer (2006*a*), the explanatory power of our version of bounded rational learning goes beyond diffusion by learning from a prominent example, since it may explain regional as well as global policy diffusion as a result of local interactions.

Moreover, countries usually learn from the most effective examples (Meseguer, 2006*a*; Gilardi, 2005; Gilardi, Füglister and Luyet, 2009; Volden, 2006; Shipan and Volden, 2008; Simmons and Elkins, 2004; Elkins and Simmons, 2005). Thus, the effectiveness should go toward its maximum.

**Learning and the evolution of the average effectiveness**

In this computational world, diverging policies evolve toward the maximum effectiveness possible. This, of course, is the expected result of the process of diffusion since each country, when introducing an alternative policy, introduces the most effective policy. However, what is striking here is the process that leads to total effectiveness. From Figures 5.8 and 5.9, what can this particular behavior teach us about the process of diffusion in general and learning in particular? We present a Bayesian version of learning, meaning that prior beliefs of the policy makers in the likely consequences of policy are updated after looking at its results in other countries. Thus, these later beliefs are taken into account in the process of policy change (Meseguer, 2003, 2006*b*; Gilardi, 2010).

In this model, the Bayesian updating rule is, here, expressed as the updating of beliefs after a country has compared the effectiveness of the possible alternative policies and its current one. If this difference is positive, it will introduce the policy of its most similar neighbors. What this result suggests is that learning has an impact on the choice and the change of an alternative policy. However, learning has no impact on the results of the policy. What a country can do is change its policy as often as needed; that is, when the policy becomes ineffective and the institutional

[15] More than one third of the cantons had introduced such a policy before the introduction of the federal law.

constraints are overcome, in order to introduce the best solution available at that moment. A few time steps later, however, this solution may be totally ineffective.
The fact that Bayesian learning is bounded limits the number of experiences to learn from. Thus, countries may learn from not-so-effective examples. Therefore, the evolution of the average effectiveness, which follows a J-shaped curve, indicates that not only is time needed for a policy to deploy its effects, it also takes time for a country to find the best-suited policy. Moreover, this J-shaped curve seems to be a perfectly logical result. Since the future remains unknown, one can learn only from past experience and the time needed to introduce a new, supposedly more effective, policy may be long. For instance, the introduction of old-age insurance in Switzerland took more than 20 years between its introduction into the federal constitution in 1925 and its promulgation into law in 1948. Thereafter, in 1950, the first revision of this insurance was completed. This is a rather good example of the problem of the evolution of a policy's effectiveness.
At this point the individual behaviors of the countries are difficult to predict. What we have done here is deduced the countries' learning behaviors from the global results. Therefore, this can be done only if the model is stable.

## 5.3 The validity tests

The different validities have been explained Section 3.4.4. We will first highlight how we test for the internal validity of our model, meaning that we will see if the model has been correctly implemented. In other words, we will *explore* the model to see if the consequences of the modification of the different parameters lead to the same results as Sections 5.2.1, which emphasized the progress of the model with emblematic default parameters.
However, 540 runs are the number of runs needed to obtain the results of the variables for all possible combinations with the chosen changes in the parameters we

use. In brief, the size of the world (*WorldSize*)[16] is changed by 5 units from 14 to 24, the size of the proximity array (*NumProximity*) by 5 units from 5 to 15 as well as the number of traits (*NumTraits*), the random seeds (*RandomSeed*) by 1 unit from 0 to 9 and the two neighborhoods (*Neighborhood*)–the Moore and the Von Neumann[17]. Formally, we have $3 * 3 * 3 * 10 * 2 = 540$ runs.

A first step toward a comprehensive validity test is made in Section 5.3.1, since, for each country, the different parameters – the size of the world, the size of the proximity array, and the number of possible traits – are altered and we will compare the different results to see if the model exhibits the same behavior under these changing conditions. However, an important parameter still needs to be changed in order to have the full picture, namely the random seeds, which represent the random implementation of the agent in the world and which will be the purpose of Section 5.3.3. To facilitate the legibility of the results, they will be presented graphically. First, only the results when the neighborhood is changed will be compared.

### 5.3.1 The validity test with different neighborhoods

The correct implementation of the algorithm of the model developed in Section 4.4.8 and the results obtained according to the interactions with different types of neighbors need to be tested and compared. This will allow us to highlight not only the importance, or the weight, of the number of neighbors a country can interact with – in other words, the impact of bandwagon pressures throughout the entire process of diffusion – but also if the model behaves as it should according to the description of the algorithm (Section 4.3.3); that is, if the model delivers the intended outcomes.

Table 5.4 gives the initial situation[18] in the different worlds. Consequently, there are 196 different regions, as each country has its own policy (characterized by its own

[16]In parentheses, we have the different parameters as they appear in the parameters settings window as shown Figure 4.7.

[17]These changes and the corresponding result are explained in more detail Section 5.3.3.

[18]As default parameters, we have the size of the world set to 196 countries, the size of the proximity array set to 5 cells, each can be filled with a number chosen between 1 and 10.

| Neighborhood | | |
|---|---|---|
| | Moore | Von Neumann |
| Number of regions | 196 | 196 |
| Average effectiveness | -0.0097812 | -0.097812 |

Table 5.4: The number of regions and the average effectiveness at the start of a run

color) and the average effectiveness is near 0, since the effectiveness of the different countries are randomly drawn from a normal distribution with a mean of 0.
the completion of a emblematic run; that is, after 350 iterations, the situation is of course a little bit different, as Table 5.5 shows us.

| Neighborhood | | |
|---|---|---|
| | Moore | Von Neumann |
| Number of regions | 37 | 53 |
| Average effectiveness | 0.7491 | 0.6783 |

Table 5.5: The number of regions and the average effectiveness at the end of a run

What can be observed in Table 5.5 is that the countries tend to be more clustered in the case of the Moore neighborhood than in the case of the Von Neumann neighborhood and the average effectiveness is higher when countries interact with their Moore neighbors than with their Von Neumann neighbors. This means that the higher the number of neighbors taken into account, the higher the political similarity and the more effective the different policies. Therefore, the comparison of the results in the number of regions is a representation of the spread of few policies that have been explained at the spatial level in Section 5.2.1. Moreover, this can be interpreted as the importance of the available information.
When countries have more available information, the higher the chance of introducing a more effective policy, as these results clearly show, since the average effectiveness is higher when the countries communicate with more neighbors (the eight adjacent countries in our model). In addition, the chance to find a most similar neighbor that has already introduced the alternative policy[19] is higher when a country can interact with a more extended neighborhood, as is the case with the Moore

[19]See Section 4.3.3 for a reminder of the importance of the similar neighbors

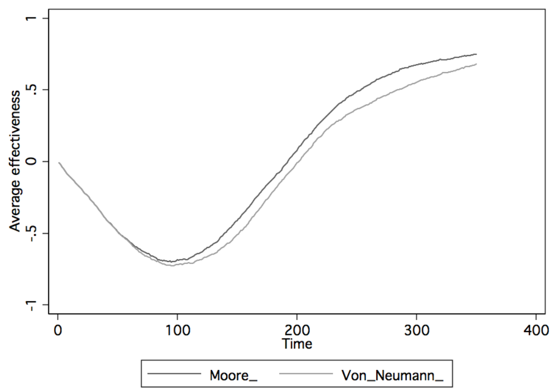



Figure 5.10: The evolution of the average effectiveness in different environment

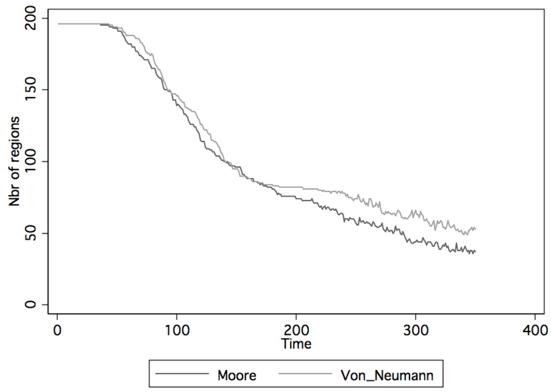



Figure 5.11: The evolution of the number of regions in different environment

neighborhood. However, more parameters need to be altered to have a broader picture of these tests.

### 5.3.2 The validity tests with country parameters

To fully validate these first conclusions – and the model in general – we need to combine the different parameter changes with one important parameter that has not changed, namely the size of the world, the possible size of the proximity array, and the number of possible traits.

Thus, to validate the model, we systematically alter the different parameters to see if the number of regions and the average effectiveness at the end of the simulation ensues from a particular articulation of the initial parameters or if the results repeat and, thus, are independent from exogenous factors. Each parameter is modified; all else being equal for each parameter, we choose the following variations:

- <u>worldSize</u>: The size of the world is incremented by 5 units from 14 to 24; that is, we create 14 by 14 grids, 19 by 19 grids, and 24 by 24 grids. In other words, we test the evolution of the average effectiveness and the clustering for worlds populated with 196, 361, and 576 countries, respectively. For each neighborhood, we obtain the following graphs (Figure 5.12).

  Clearly, the number of regions is lower and the average effectiveness better in

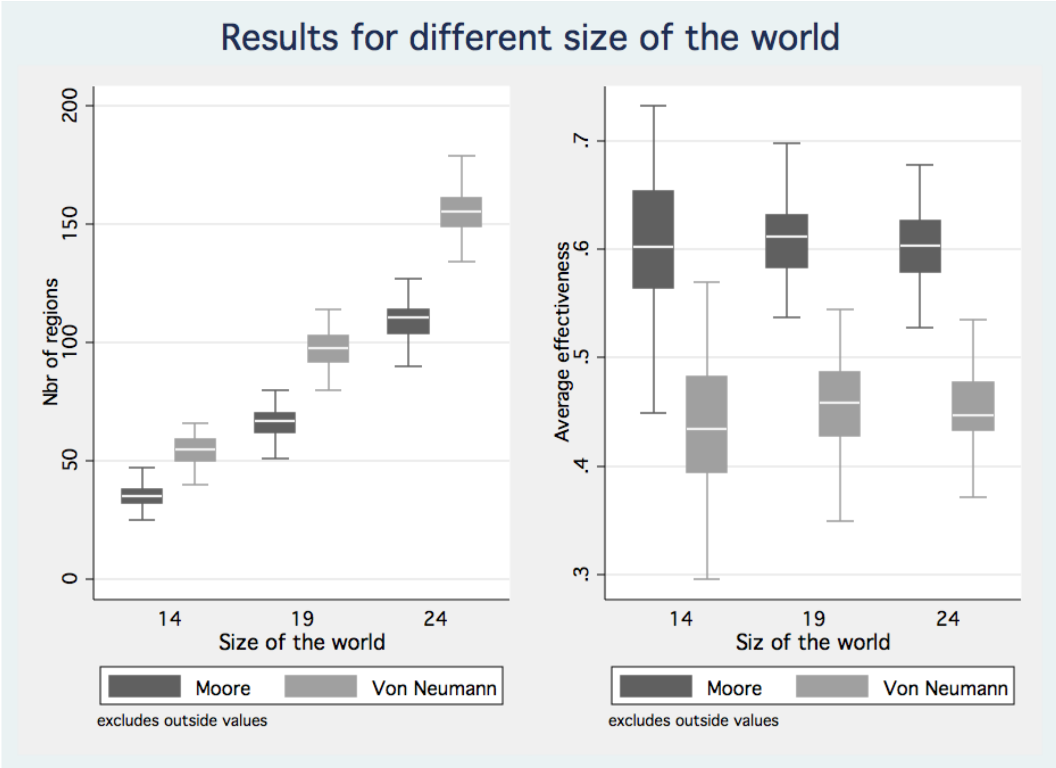



Figure 5.12: The results with different size of the world

the case of the Moore neighborhood than the Von Neumann neighborhood. These results are in line with what we explained here above, namely, when countries interact with their Moore neighbors the world is more clustered and the policies that diffuse are more effective.

- numProximity: The number of possible cells in the proximity array is gradually increased from 5 to 15. Thus this can be seen as a refinement of the potential dimensions of the proximity. Table 5.6 shows us examples of different proximity array sizes, each with 10 possible traits.

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 3 | 7 | 9 | 4 | 6 | | | | | | | | | | |
| 2 | 9 | 6 | 4 | 4 | 9 | 9 | 4 | 5 | 2 | | | | | |
| 3 | 2 | 4 | 0 | 2 | 4 | 6 | 0 | 0 | 4 | 2 | 1 | 0 | 3 | 7 |

Table 5.6: Examples of the proximity array with different sizes

The results when running the simulations with the changes in the number of cells of the proximity array are represented in the graphs of Figure 5.13. Here again, the results shows us that the more interactions, the more effective the world and the more clustered it is. In other words, to have more interaction possibilities – a higher number of neighbors – allows an easier updating of the beliefs about the current policy and, thus, favors the change.

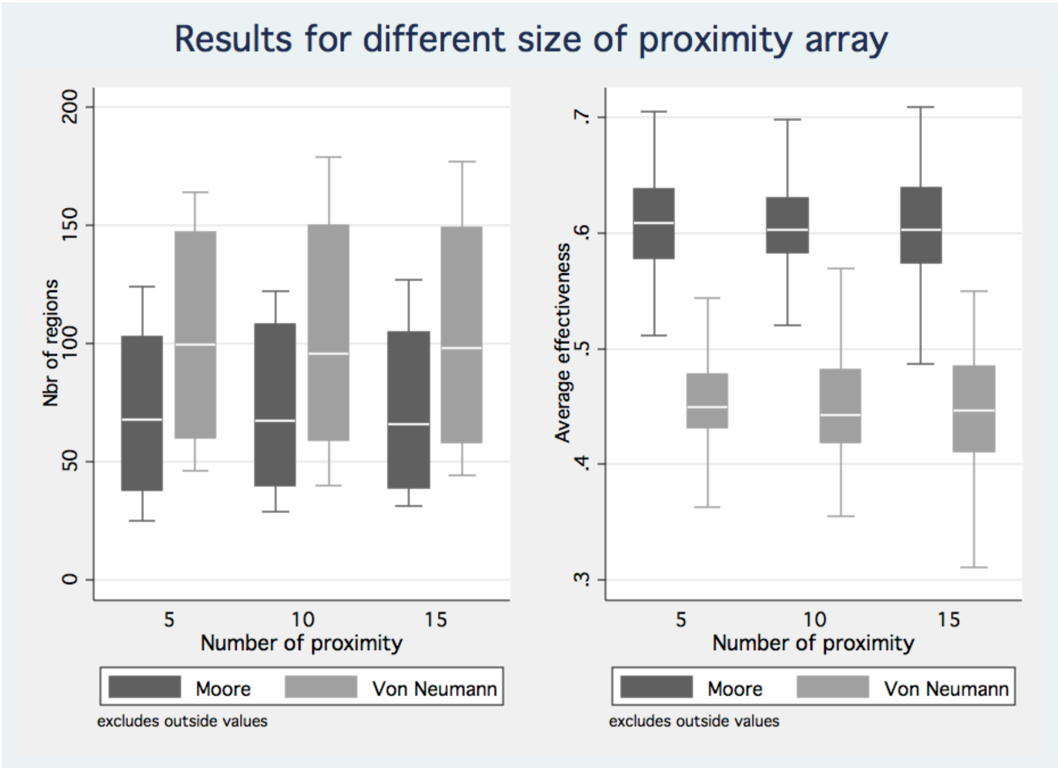



Figure 5.13: The results with different size of the proximity array

- numTraits: The number of traits is increased, in increments of 5, from 5 to 15. The possible values of the different proximity dimensions can be seen as the fine tuning of these dimensions.

  Once again, the results of the different simulations are in line with those in the emblematic runs we explained in Sections 5.2.1. The interactions between countries and their Moore neighbors are more successful in terms of diffusion – less regions for more effectiveness – than between countries and their Von Neumann neighbors (Figure 5.14).

In Tables 5.7 and 5.8, the results for the number of regions and the average effectiveness with different parameters changes are highlighted. The only parameter that

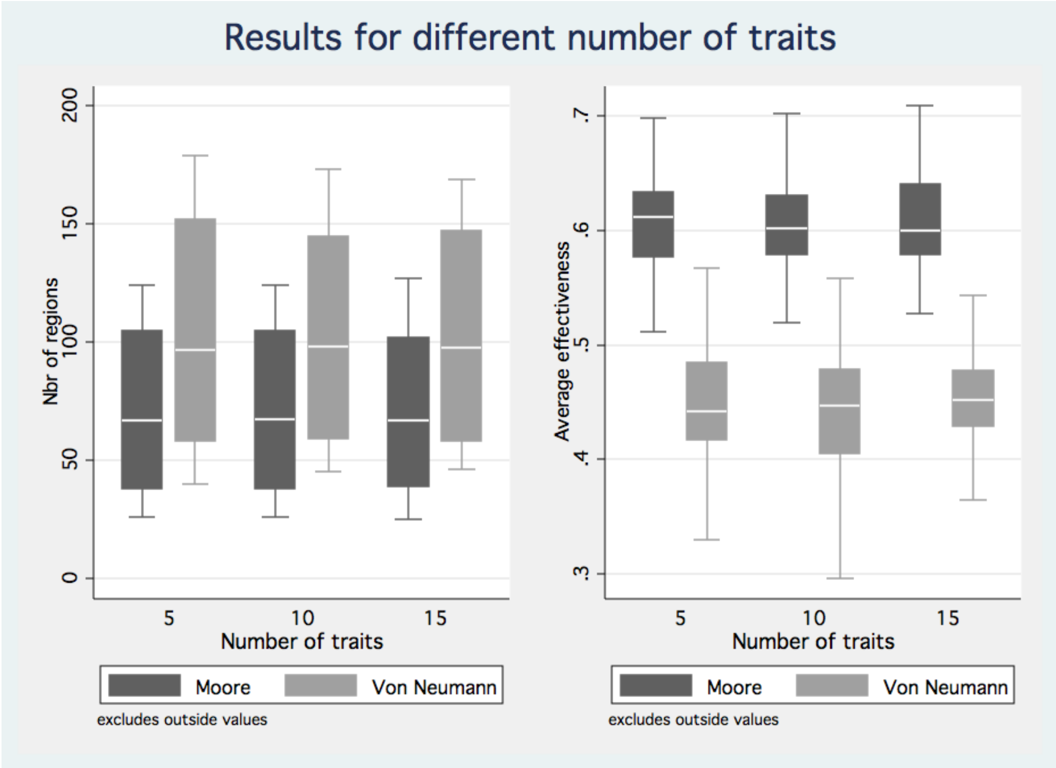



Figure 5.14: The results with different number of traits

does not change is the random implementation of the agents in the world, namely the random seeds, which has a default value of 1.

In these two tables, we have a rather broad view of the results of the two main dependent variables; that is the number of regions and the average effectiveness, if we alter the different parameters – worldSize, numProximity, and numTraits. What can be observed in these two tables is that the number of regions is systematically lower when the Moore neighbors are taken into account. Moreover, the average effectiveness is systematically higher under the same conditions.

Additionally, when the countries interact with their Moore neighbors, the lower the number of regions as the process unfolds indicates that less policies diffuse, but their *territory* is greater. As shown in Section 5.2.1, this can be related to theoretical and empirical studies that highlight the existence of some typological aspects in the real world. For instance, Esping-Andersen (1990) has shown the geographical clustering of the welfare states with its famous the typology of the three worlds of the welfare states or the different typology of the health care systems (Palier, 2004).

| | | worldSize | | | | | |
|---|---|---|---|---|---|---|---|
| | | 14 | | 19 | | 24 | |
| numProximity | numTraits | Moore | Von Neumann | Moore | Von Neumann | Moore | Von Neumann |
| 5 | 5 | 25 | 50 | 71 | 94 | 114 | 166 |
| | 10 | 37 | 53 | 87 | 98 | 131 | 153 |
| | 15 | 35 | 57 | 70 | 103 | 119 | 166 |
| 10 | 5 | 46 | 61 | 75 | 104 | 132 | 169 |
| | 10 | 41 | 63 | 70 | 93 | 132 | 177 |
| | 15 | 44 | 59 | 79 | 102 | 110 | 164 |
| 15 | 5 | 44 | 49 | 79 | 106 | 114 | 178 |
| | 10 | 40 | 55 | 77 | 108 | 118 | 164 |
| | 15 | 44 | 46 | 73 | 97 | 112 | 160 |

Table 5.7: The results for the number of regions

| | | worldSize | | | | | |
|---|---|---|---|---|---|---|---|
| | | 14 | | 19 | | 24 | |
| numProximity | numTraits | Moore | Von Neumann | Moore | Von Neumann | Moore | Von Neumann |
| 5 | 5 | 0.828166411 | 0.688748162 | 0.730580286 | 0.672842763 | 0.755897128 | 0.65929895 |
| | 10 | 0.749080501 | 0.678361238 | 0.688611475 | 0.648187811 | 0.728494121 | 0.654499277 |
| | 15 | 0.792405414 | 0.688907047 | 0.740206357 | 0.652871495 | 0.735263733 | 0.643404431 |
| 10 | 5 | 0.699105831 | 0.578184354 | 0.736570451 | 0.62848327 | 0.747543789 | 0.621999321 |
| | 10 | 0.702970072 | 0.595075094 | 0.719553179 | 0.675769875 | 0.725760192 | 0.596452334 |
| | 15 | 0.670808638 | 0.553114794 | 0.755897128 | 0.633728413 | 0.747595076 | 0.639487552 |
| 15 | 5 | 0.716917889 | 0.68866259 | 0.740206357 | 0.650794949 | 0.745555123 | 0.594734246 |
| | 10 | 0.738227108 | 0.617898402 | 0.736570451 | 0.611210479 | 0.734407861 | 0.64724315 |
| | 15 | 0.741836118 | 0.688906586 | 0.719553179 | 0.671454133 | 0.743192061 | 0.652422968 |

Table 5.8: The results for the average effectiveness

### 5.3.3 The internal validity with the random seeds parameter

Modifying the initial conditions of the model by changing its parameters is the cornerstone of several advantages[20] attached to computational agent-based modeling. Since history matters[21] the possibility of exploring the model allows the researcher to rerun history. Consequently, to test for internal validity is a way of fully using the advantages of computational agent-based modeling, as highlighted in Section 3.4.5. Therefore, history is rerun with parameter changes that allow the researcher to test if the model has been well developed for its purpose. In sum, testing the internal validity consists in altering the initial conditions of the model to see its behavior and, by systematically modifying the parameters, the internal validity shows if the model has the same behavior under different starting conditions, which emphasizes the (in)correct implementation of the model.

In Section 5.3.1, as previously mentioned, a first attempt to compare the results of the model has been made, as emphasized in Figures 5.16 and 5.17. The results are that the effectiveness is greater when the countries interact with more neighbors and the number of regions in the world are lower if the countries interact with the Moore neighborhood. Consequently, the model is stable and internally valid, since, with changes in the parameters, these results are the same as in the emblematic runs explained in Section 5.2.1.

Therefore, if the model had evolved differently under different conditions, it would have meant that the results of the model do not only ensue from the different interactions between the different countries, but also from changes in the environment that are totally exogenous from the embedded conditions of the countries, which contradict the assumption that no central authority gives order to influence the interactions in computational agent-based model.

- rngSeed: The random implementation of the countries in the world is increased

[20] Section 3.4.5 provides a large overview of these advantages.

[21] Depending on the initial implementation of the different countries, the clustering of the world is different. This means that the way the different countries are aggregated as regions is the same but the place where these regions form in the world is different.

by one unit from 0 to 9. This is a crucial test, since, if we modify the *environment* of the countries, that is we alter their position in the world, it should, according to the notion of *stigmergy*, influence the interactions between the countries, since the behavior of the agents affects the environment; this, in turn, should influence the environment, but not necessarily the results.

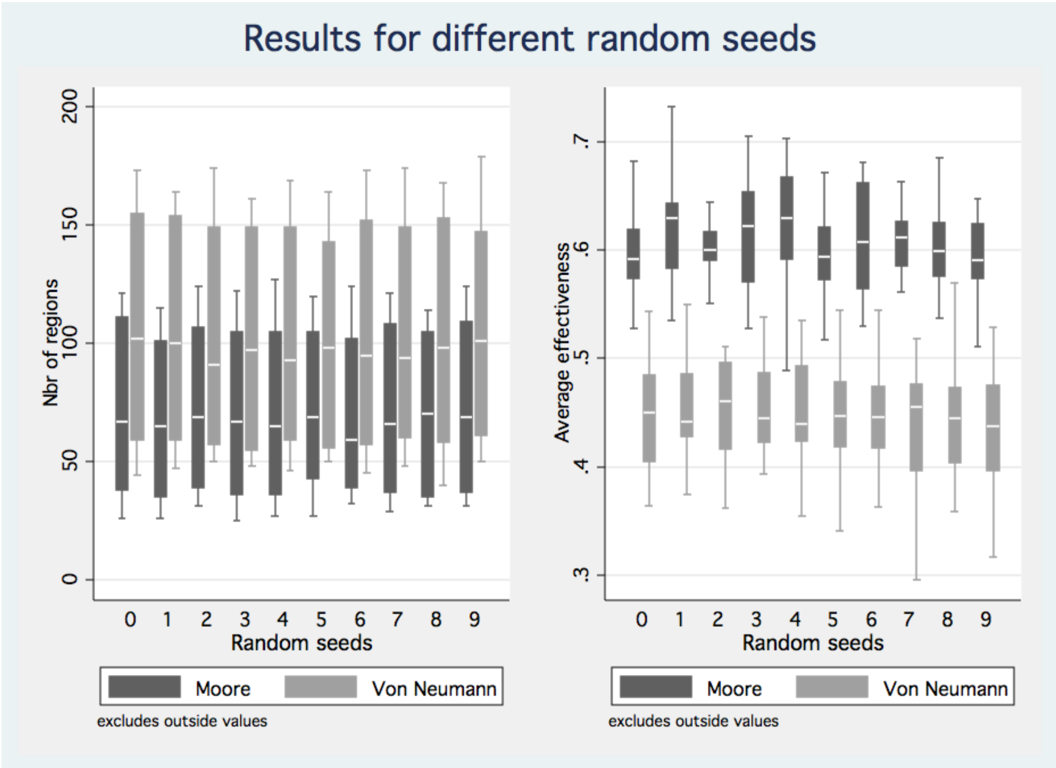



Figure 5.15: The results with the different random seeds

Figure 5.15 highlights the different results when modifying the countries' implementation of the policies. We clearly show that the interactions, when modifying solely the random seeds, lead to the same results as when altering the other parameters, all else being equal. Yet, when modifying the random implementation of the countries, still more effective policies diffuse when interacting with the Moore neighbors and less policies spread, since the number of regions is lower in the case of the Moore neighborhood.

This is an important result, for the reason that it shows very good stability of results. In other words, to reposition the agents elsewhere on the grid does not change the outcomes, meaning that the model has the same behavior.

In the different graphs of Figures 5.16 and 5.17, the results of different simulations are summarized. The different random seeds compose the box plot – we see the dispersion of the results for the different random seeds – and the parameters that have been altered are the number of proximities, the traits, and the size of the world. As an expression of this notion of *stigmergy*, we see on the different graphs that the different initial condition shapes the evolution of the different dependent variables – the number of regions and the average effectiveness of the world, as there is some dispersion in the results according to the different random seeds. However, the results at the end of the runs are always the same: the number of regions are lower and the average effectiveness is greater in the case of the Moore neighborhood.

## 5.4 Conclusion

In this chapter, we start by outlining the results of the emblematic run. By *emblematic*, we mean that the different parameters are set to default values that are sufficiently interesting for our purpose to let fascinating patterns emerge from the interactions of the agents. For that reason, we set the size of the world to 14, which means that the world is a 14 by 14 toroidal grid populated with 196 countries. Each of these countries are then built with a proximity array composed of five cells and each of these cells can have a possible value randomly chosen in an uniform distribution truncated to 0 and 9, which gives 10 possible values.

Moreover, from Chapter 3, we can say that one of the aims of computational agent-based models is not only to emphasize the results that emerge from these interactions, but also to highlight the *process* that leads to this emergence. Following this, in Sections 5.2.1, we explained and analyzed the development of the model in depth. Subsequently, we tried to describe the process of diffusion at the temporal, spatial, and conditional level. To summarize, at the temporal level, we assist with the diminishing number of different policies through time, which means that the world is clustering, but mainly at the political level. In other words, we assist in the spread of some policies, since different countries introduce the same policy as the more effective alternative, but not necessarily at the same time. Consequently, the average effectiveness in the world is increasing. However, at the beginning of the process, the average effectiveness decreases, owing to the fact that the process of diffusion affects the countries differently due to their internal factors.

Since this evolution of the process can be applied to both neighborhoods, it is a first step in order to test the internal validity of the model. Therefore, more has been done to fully validate the model. We thus ran the MEME module to extract data with different initial conditions. After analyzing the dataset, we compared the different results and found that there were no significant differences, meaning that the behavior of the model is independent from exogenous condition. In sum, our model of policy diffusion is internally valid.

The results emphasized the importance of the neighborhood in the process of diffusion and the resulting temporal and spatial clustering. This can rely on several studies on the diffusion of democracy, where the creation of different regions composed by neighbors with the same regime has been highlighted as a consequence of bandwagon pressures (Wejnert, 2005; Lustick, Miodownik and Eidelson, 2004; Gleditsch and Ward, 2006). Yet bandwagon pressures play an important role in the process of diffusion, since the number of regions are diminishing through time and we assist at the emergence of few policies in our world.

In the case of the conditionality of the process of diffusion, we have explained how the process evolves. However, we can only conjecture about the fact that the countries are differently influenced by the mechanisms of diffusion. Since the model has been developed in order to give the global evolution of the process of diffusion, we miss at this point the data that allows us to disentangle the different mechanisms of diffusion at the country level. Yet, this leads us to a first exploration of the limits of the model[22]. The limit imposed by the type of data we have drawn from our model are very influential in the explanation of the following limits:

- The S-shaped curve has a shorter path in the beginning than in the end, this

[22]Here, we briefly highlight the different limits of the model. More will be said in the next section.

is in contradiction with the empirical results explained in Section 2.3.3. This is strongly linked with the problem of time, which is a recurrent problem of computational agent-based modeling (section 3.4.5). The fact that, in the model, the first change in the slope of the S-curve comes after 43 steps on 350 in the case of the Moore neighborhood if we launch a emblematic run is hard to interpret. Nevertheless, we can say that the first path, which should correspond to learning, is shorter as highlighted in empirical research.

- This latter remark on the temporal horizon of the learning highlights a second important limit of the model, namely the disentanglement of the different mechanisms, which is very difficult. This comes more from a conceptual problem in the development of the model. More precisely, we have developed the batch model for a global observation of the process of diffusion, since we gathered data of the number of regions and the average effectiveness. In a sense, we put aside the micro level, and start observing what emerges at the macro level from interactions at the meso level.

- Another problem arises from the analysis of the meso level. The fact that the average effectiveness follows a J-shaped curve is an important result. However, we can only analysis this result at the global level. Therefore, the above mentioned problem of the disentanglement of the mechanisms of diffusion remains a open question. At this point, we can only speculate on the fact that some countries remarking the gain of effectiveness of the early adopters, also want to change their policy and start a competing process up to the point where some policies are considered as an accepted norm or an *ideal* of effective policy.

These limits shape the future development of this work, described in more details in the next concluding chapter. At this point and to paraphrase Schelling's best selling book *Micormotives and Macrobehavior* (Schelling, 1978), what we observe here are mesomotives and macrobehaviors, since we observe the emergence of a macro phenomenon, namely diffusion, from the interactions of countries.

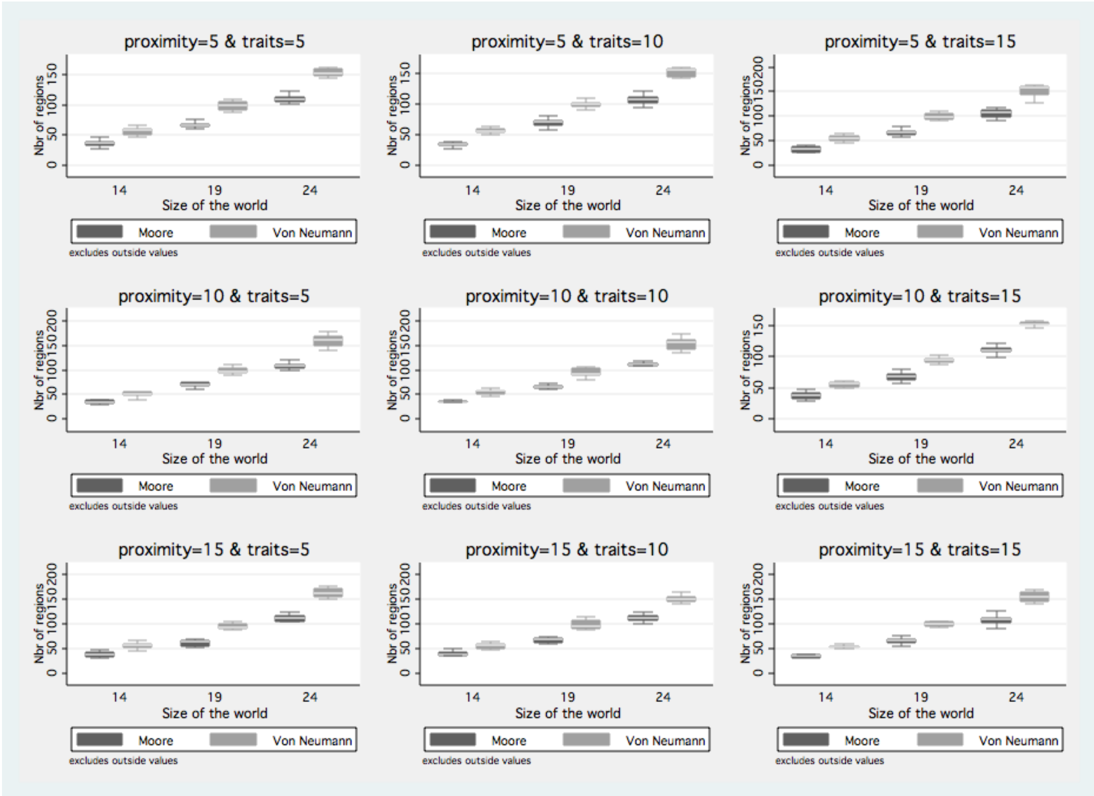



Figure 5.16: The different numbers of regions according to the different initial conditions

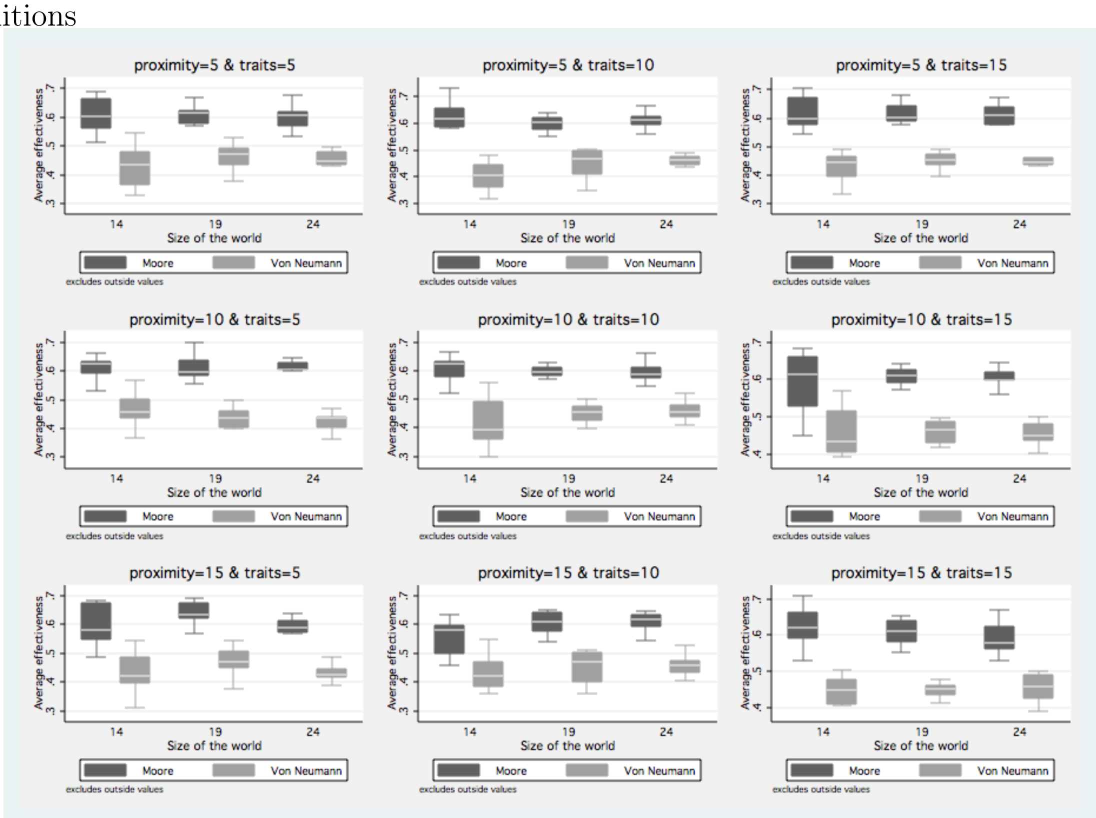



Figure 5.17: The different average effectiveness according to the different initial conditions

# Chapter 6

# Conclusion

Now that we are at the end of our trip through this first attempt to develop a computational agent-based model of policy diffusion, it is time for, first, a backward look and, second, to emphasize the main development of the model.

## 6.1 What have we learned?

We base this dissertation on the following definition of diffusion:

> *"International policy diffusion occurs when government policy decisions in a given country are systematically conditioned by prior policy choices made in other countries"* (Simmons, Dobbin and Garrett, 2006, 787).

This definition implies that diffusion is a phenomenon that takes place at three different levels. First, the temporality of the process is taken into account by the fact that changes at time $t-1$ have an impact on the decisions of the country at time $t$. All through the progress of the process, prior choices made elsewhere influence the decisions made in the country. This means that the path dependency of the process of diffusion serves as a *springboard* for the future development of diffusion. According to its theory, the phenomenon of diffusion should follow a S-shaped curve. This curve highlights the fact that the number of early adopters is rather small. As the number of adopters increases, the potential advantages of the changes become more and more obvious, resulting in a steep increase in the number of adopters up to the point where almost all potential adopters have introduced the change. In other words, the result of the theoretical S-shaped curve is the total convergence of the adopters.

In our model, the countries are represented by the color that characterizes the policy. Since a color also defines a region, we have at the beginning a number of regions that is equal to the number of countries. When the process is launched, the countries start introducing an alternative and more effective policy. As a result, the number of regions decreases, highlighting the fact that only a few policies are spreading. This is shown with an inverted S-curve. Moreover, this political convergence is reduced by the divergence that still exists among the different dimensions of the proximity. Thus, there is divergence in the convergence, which is a well-known phenomenon in the study of diffusion (Simmons and Elkins, 2004; Axelrod, 1997*b*). A analysis of Figure B.1 and B.2 gives a better view of this partial local convergence result, since only a few proximity arrays are fully similar.

Secondly, diffusion occurs in a defined space that can be the world, the European Union, or a group of regions. The space involves the notion of neighborhood(s) with which a country interacts. Thus, the neighborhood should be defined. To take into account the particularity of the *environment* of the country; that is, the fact that more than the geographical borders are important, we have decided to separate the notion of neighborhood into two concepts:

1. <u>The geography</u>: Geography corresponds to the Moore or Von Neumann neighbors; that is. the eight or four adjacent cells;

2. <u>The proximity</u>: Proximity is defined as an array composed of a certain number of cells that emphasizes other dimensions of the neighborhood, such as economy, culture, and so forth.

Most of the studies involving diffusion are based on the fact that the more the neighbors that have changed the greater the pressure for a change, which has been defined as bandwagon pressures (see Sections 2.3.3, 3.4.6, and 4.3). In our model, the evolution of the process of diffusion ends in a clustering of the world with few policies that emerge. Furthermore, this result emphasizes the fact that bounded learning as a mechanism of diffusion limits convergence.

The last characteristic of this definition of diffusion is the fact that policy makers react differently to the same influences than their neighbors do. In other words, because of their internal differences, the countries are unlikely influenced by the behavior of their neighbors. As a result, in the context of our computational world, the average effectiveness follows a J-shaped curve, meaning that, following Meseguer (2006*a*), it takes time for the different policies to deliver the intended outcomes, at least at the global level – the macro level. This result comes from the expression of the change variable. The concept of conditionality (Section 2.3.4) expresses the fact that two countries with the same degree of interdependence may be differently affected by the process of diffusion. More precisely, it comes from the third part of Equation 4.4; that is $\left((\text{institutional constraints})\left(\frac{N_c}{N}\right)\right)$. In words, if a country shares the same number of changed neighbors – the degree of interdependencies – the change is influenced by the institutional constraints that correspond to the veto players, all else being equal.

To build a comprehensive theoretical framework, the theoretical model of policy diffusion developed by Braun and Gilardi (2006) has largely inspired this work and we have stressed that the accurate method is computational agent-based modeling, since diffusion occurs between heterogeneous interdependent countries and because the results of the process are mainly unknown at the macro level. Computational agent-based modeling as a method in political science is rather new; however, it is a growing research field. The two main concepts that are linked with the use of computational agent-based model are *complexity* and *emergence*. The first one implies that the whole is more than just the sum of its parts. From that is derived the concept of emergence, which has been defined as an unexpected result at the macro level from micro interactions. Moreover, we have developed the three main conditions a country should go through in order to change its policy and that can be summed up in three main methods: *ready()*, *choose()* and *change()*. Thus, when a country is first ready for a change, it chooses the alternative policy and changes to it. More precisely, if the effectiveness of the policy is lower than the preference for the current policy, the country is ready for a change.

Before changing its policy, the country has to choose an alternative, supposedly more effective, policy by looking at its (Moore or Von Neumann) neighbors that have already changed their policy. Finally, if this alternative policy is more effective than the current one and the institutional constraints are overcome – the veto players must agree upon the political change – then the most effective policy among similar neighbors that have changed their policy is introduced.

Not only can the results of this computational agent-based model be expressed as the evolution of the process, but also as the outcome of the process when this latter is stopped. Thus, when altering the different parameters, we highlight the behavior of the model under several different initial circumstances. Moreover, this allows us to test for the internal validity of the model; that is, its correct implementation.

By modifying the main parameters – all else being equal – and by combining these alterations, we have emphasized the fact that our model is internally valid, since different initial values of the parameters give the same results: the number of regions at the end of a run are systematically lower when more neighbors are taken into account, meaning that fewer policies are spreading when countries interact with their Moore neighbors than with their Von Neumann neighbors and the average effectiveness is greater under the same conditions.

Thus, to develop our computational agent-based model and in order to be able to study diffusion after defining the different characteristics of the countries/agents – the effectiveness of the policy, the preference for the policy, the institutional constraints, and, of course, a policy – and the particular shape of the world, which is a toroidal or wrapped-around shape, we have described both at the theoretical and computational[1] levels how the countries should interact; that is, the different conditions for a policy change to be introduced.

In this thesis, we have tried to highlight the influence of neighboring countries through different mechanisms of diffusion between countries. Moreover, we have emphasized the macro patterns that emerge from these interactions depending on the internal factors – mainly the institutional constraints and if these patterns are (in)dependent from exogenous shocks.

Therefore, in Chapter 3, we highlighted the necessity to develop a computational agent-based model based on simple, but not simpler, conditions for the different agents to interact. In Section 4.4.8, we label this simplicity principle under the name of the KISS, or *"Keep it simple, Stupid!"* (Axelrod, 2003). Based on this motto and to achieve the development and the results summarized above, the conditions for a change to occur are threefold:

- A country is <u>ready</u> for a change if the effectiveness level is lower than the preference level, meaning that the current policy is so ineffective that the country has to change it despite its preference in its favor.

- If the above condition is true, the country starts looking at what its (Moore or Von Neumann, depending on the chosen type) neighbors do. More precisely, the country is looking at neighbors that have already changed their policy[2] to see if the new policy is more effective than its current one. If it is the case, it <u>chooses</u> this policy as the alternative

- If the country has chosen the most effective policy among its neighbors, it successfully <u>changes</u> its policy by introducing the policy of its most similar neighbor that has already changed its policy if the institutional constraints are overcome.

In developing such a theoretical model, the question of its use, besides theory testing, can be investigated. In other words, we need to challenge such a model with

[1]The complete code of the program can be found in Appendix C.

[2]This corresponds to the update of the beliefs about the *bien-fondé* of the policy change.

its application in, or the piece of understanding of the real world, it brings. In other words, the main results must be discussed according to the theory. The model presented here is in line with theoretical expectations and empirical evidence; more precisely a sort of S-curve evolution, partial local convergence, and global divergence. Moreover, the average effectiveness tends to follow a J-shaped curve.

Since we cannot gather data on the composition and evolution of the neighborhood at the country level, we have limited our discussion to only one possible mechanism of diffusion and, thus, we have stressed that this mechanism at play here can be characterized as learning in its bounded rational version with Bayesian updating.

As we already mentioned in Section 5.2.2, bandwagon pressures; that is, the evolution of the weight of the neighbors that have changed their policy in the process of policy change are essential in explaining the transition from learning to emulation. Hence, the results of the computational model suggest that learning does not need a prominent example, but rather an estimation of the average effectiveness of the alternative policy in order to drive the process of policy diffusion. Moreover, the different political regions, as unfolded from the process of diffusion, do not have stable borders. In other words, according to different estimations of the policy effectiveness, the different regions are permeable to the different emergent policies, meaning that countries at the borders do interact with each other, with the result of moving the *policy borders*. Furthermore, since the beliefs on the effectiveness of the policies are updated using Bayesian learning, the fact that the progress of the average effectiveness following a J-shaped curve indicates that, not only is time needed for a policy to deploy its effects, but it also takes time for a country to find the best-suited policy. Moreover, as the future is vast, one can only learn from past experience. Thus, time is needed to introduce a new policy too. Consequently, this J-shaped curve seems to be a perfectly logical result.

At this point we will briefly challenge the main results of the model:

- In Section 2.3, we stressed out the different dimensions of policy diffusion process. Our results are in line with the theoretical expectations developed in

Section 2.3. The temporal level is emphasized by the behavior of the agents of the process that follow a S-shaped curve (see e.g. Rogers, 2003; Berry and Berry, 2006; Greenhalgh et al., 2005). For instance, Gilardi, Füglister and Luyet (2009) show that the introduction of Diagnosis Related Groups (DRGs) as a hospital financing method follows a S-shaped curve if the introduction of this policy is an effective experience elsewhere and if the current policy of the country is not so effective, meaning the hospital financing policy in OECD countries do converge.

However, in our model, the combination of an ineffective current policy and the introduction of a more effective policy found in the neighbors that have already changed their policy results in the decrease in the number of regions through time following an inverted S-shaped curve, meaning the world is clustering around only a few policies. These two results represent both sides of the same coin.

Consequently, the interdependencies between different agents not only lead to the inverted S-curve, but also to the emergence of like regions. For instance – in the case of the diffusion of democratization – Cederman and Gleditsch (2004), using computational agent-based modeling, highlight the potential clustering of the democracies; and, using a more traditional, empirical methodology, Gleditsch and Ward (2006) show the regional convergence of the diffusion of democratization. In this thesis we have developed a computational version of bounded rational learning, since a country typically seeks the most effective policy among its neighbors that have already changed, which is rational, and *bounded* comes from the fact that a country interacts with a defined number of neighbors.

The process of diffusion driven by rational bounded learning leads to the clustering of the world as expressed by the inverted S-curve and, at the visual level, by regions that are defined by the same color. More precisely, convergence is the *results* of the *process* of diffusion, meaning that "diffusion is *not* equivalent to convergence" (Gilardi, 2011, 2).

Moreover, the computational process of diffusion explained here does not attain the total convergence. The inverted S-curve goes toward 1, without reaching it, and the world will not be covered by only one policy. Thus, the result of this process that is driven by bounded learning is *partial local* convergence. Not only are proximity arrays, which are represented in Tables B.1 and B.2, rarely identical; but also, at the more global level, cleavages persist on the grid.

- Average effectiveness follows a J-shaped curve. At the beginning of the process of diffusion, the number of countries that have changed their policy is outnumbered by the countries that still have a less-effective policy, resulting in a decrease in average effectiveness. We also show that this result comes from the fact that, facing the same degree of interdependence, the country with fewer veto players and a less-effective policy will more quickly change its policy.

  Only few studies have studied the conditional nature of diffusion (Shipan and Volden, 2008; Gilardi, 2010). However, the fact that a country with a less-effective policy is more subject to change is a well-known result (see e.g. Volden, 2006; Dobbin, Simmons and Garrett, 2007; Füglister, 2009). Furthermore, the impact of the veto players on the policy change has been emphasized as the more veto players, the more difficult the policy change (see e.g. Gilardi, 2005; Gilardi, Füglister and Luyet, 2009; Henisz, 2004).

  Our model combines these two effects by underlining that it may takes more time to introduce a more effective policy depending on the strength of the veto players. As a result the average effectiveness in the world follows a J-shaped curve, meaning that it takes time for the different policies that are spreading to deploy their beneficial effects.

  Furthermore, the bounded version of rational learning can be represented as Bayesian updating (Gilardi, 2011). More precisely, at each time step, the beliefs about the effectiveness of the (current and alternative) policy update according to the consequences of the change in neighboring countries, which is represented by the difference between the average effectiveness and the current effectiveness. Thus the choice occurs if the average effectiveness of the policy(ies) of the neighbors that have already changed is greater than that of the current policy. Since the alternative policy is more effective than the current policy, but is not the most effective, a change may introduce a policy that is still largely ineffective.

Consequently, the J-shaped form of the curve of the average effectiveness comes not only from the blocking of the institutional constraints, but also from the bounded version of learning, which limits the possibilities for updating the beliefs, meaning that the better policy is not the best policy.

In sum, we already stressed in Section 3.4.6 that, in political sciences, computational agent-based modeling is mainly used in the subfield of international relations to study the creation of nations, conflicts, and also the diffusion of democracy. Thus, we have extended here the use of computational agent-based modeling to the diffusion of policies by developing a general model.

The main point here is that micro-level interactions lead to the emergence of global patterns, since diffusion is characterized by interdependencies. More precisely, interaction between the countries, which is defined by three main conditions for a change to occur – ready, choose, change – leads to a process of diffusion following an inverted S-shaped curve and to an average effectiveness following a J-shaped curve. Furthermore, the number of neighbors involved in the process has an impact on the results of the process since, the more neighbors a country interacts with, the more effective the new policy and the lower the number of policies that are spreading. The principal argument in favor of such a model is that we obtain the same results as with empirical methods, but we do not have to analyze separately the effects of internal and external – international – factors to display the progress of the process of diffusion. Furthermore, tests for internal validity show the remarkable stability of

the model, which is something interesting in the case of complex adaptive systems, since slight changes in the initial conditions can lead to important changes at the macro level. However, this model has its limits.
Since we can analyze our model only at the global level, we miss the extraction of data at the country level. More precisely, the evolution of the different parameters – policy effectiveness, policy preference, and the different alternative policies, as well as the evolution of the proximity array and the neighborhood – cannot be extracted at the country level with our model. Because of that, and this is the first limitation of our model, we cannot disentangle the different mechanisms of diffusion, meaning that the place of the different mechanisms of diffusion is here highly conjectural. Secondly, we cannot test the external validity of our model, meaning that we need to empirically test the plausibility of the model:

- First, along the process of diffusion, the countries may be differently affected by their interactions with their neighbors. This result – that countries facing the same degree of interdependence are differently affected by the mechanisms of diffusion due to their internal heterogeneity – has been highlighted here, which is in line with several studies that have gone in this promising direction (see e.g. Volden, Ting and Carpenter, 2008; Gilardi, 2010; Franzese and Hays, 2008).

  Nonetheless, at the theoretical level, the main point that needs to be addressed concerns the disentanglement of the different mechanisms of diffusion. In other words, which mechanism is at play, and when, remains an open question. As we already mentioned in Section 5.2.2, bandwagon pressures; that is the evolution of the weight of the neighbors that have changed their policy in the process of policy change are essential in explaining the transition from learning to emulation.

- At the methodological level, the main problem – at this point in the development of the model – concerns the test of external validity. This problem ensues from the same lack of the theoretical level mentioned here above, namely the lack of data gathered at the country level. Since the links between micro and macro level cannot be studied using empirical data analysis, as explained in Section 3.4.4, the external – empirical – validation of the model needs to be done in two parts: firstly, at the lower level; ad secondly, at the global level using time series methodology (Elkink, 2009).

The underlined lacks and limits of the model serve as a basis for the development of future research. Several developments should be interesting. However, two fundamental developments deserve particular attention: the development of the batch model to allow the gathering of data at the country level and the development of dynamic networks to study the disentanglement of the different mechanisms of diffusion.

## 6.2 Future research perspectives

In this dissertation, we only focus on horizontal diffusion (that is diffusion from same level agent). In following works, and to even better understand the diffusion process, we need to study vertical diffusion; that is, diffusion between different states that are not at the same level. This is a research direction in the field of diffusion in the federal states (such as Switzerland and the USA, for example) (Füglister, 2009; Volden, Ting and Carpenter, 2008; Gilardi, 2010).

1. In order to see if one's own computational model is empirical, the external validity of the model should be tested. The problem comes here from the fact that computational agent-based modeling is a methodology that allows the study of nonlinear models, which is not the case of more traditional methodology. So arises the question: How can such a model be empirically tested? One possible way is to treat the model as the real world and then develop time-series models, since it is the standard methodology in the study of diffusion. However, this strategy seems to lead to very weak (Elkink, 2009). Another strategy that seems more promising can be to study the two levels of analysis

– micro and macro – separately and then put that together to highlight the main pieces of evidence that corroborate the main mechanisms.

2. Secondly, we need to collect data at the country level in order to analyze the entanglement of the different mechanisms of diffusion. This can be done by analyzing the evolution of the share of neighbors with an alternative policy. For instance, we can imagine that learning characterizes the interactions when the number of neighbors with a new policy is less than 3, meaning that the weight of the comparison of the different effectiveness is more important, and emulation when the number of neighbors with an alternative policy is greater than four, because the weight of the neighbors that have changed their policy is greater.

3. Nevertheless, since few studies have tried to disentangle the different mechanisms of diffusion (Boehmke and Witmer, 2004; Shipan and Volden, 2008; Gilardi, 2010), another promising way seems to be that of developing dynamic networks. Indeed, using network analysis, Cao (2010), by studying the diffusion of capital tax policy, shows that this diffusion comes partly from competition between key actors at the country level and partly from learning and emulation between countries. Thus, overlapping networks should be one interesting way to study the entanglement of policy diffusion mechanisms.

Throughout this thesis, we explain the building and use of a computational agent-based model for the study of policy diffusion. However, such a model has broader use; that is it can be apply to a wider range of diffusion phenomena. For instance, democracy has already been studied as an emergent phenomenon pushed by bandwagon pressures (see e.g. Cederman and Gleditsch, 2004), resulting in spatial segregation and adoption following an S-shaped curve. Besides, measures of the effectiveness of democracy may also follow a J-shaped curve, meaning that the introduction of democracy may lead to the following result: Time is needed in order to fully understand its principles and use its tools. In other words, the passage from autocracy to democracy may be tumultuous.

Moreover, besides the study of the potential effects of policy diffusion at a lower level; that is inside the countries, and to turn back to the basis of chaos theory, we may rely on fractal mathematics. In short, part of the object looks like the object itself. For instance, a branch of a tree looks like the tree itself. The idea is to see if the evolution of the effectiveness at a lower level, the country level, also follows a J-shaped curve. In other words, it would be interesting to see if the evolution of average effectiveness at the global level corresponds to the aggregation of the development of effectiveness at the country level.

What is great about social science is that it is always evolving. The future generation benefits from the knowledge of the current generation as the latter has benefited from the knowledge of the past generation. History is not always an eternal beginning and this circle also applies to the scientific evolution. In every aspect of the scientific life, we have a lot to learn. In this sense, the evolution of computer performance will give us new theoretical and methodological insight to develop our knowledge of the basics mechanisms that characterize the social life in general, and policy diffusion in particular (as it is the main subject of this thesis).

With this in mind, we hope that political scientists will pay greater attention to fields that may be far from their traditional *reservoir* of knowledge and explore chaos in other scientific fields to find new orders for the development of the comprehension of political phenomena that imply so many intricate interactions that can no longer be studied only as complicated systems.

# Appendices

# Appendix A

# The methodological Tools

## A.1 Object-Oriented Programming and Java

Herein are developed the main characteristics of object-oriented programming:

- *Inheritance*: The concept of inheritance is, again, based on real-world observations. Indeed, children receive genetic characteristics from their parents; sometimes, houses and money. After a child has inherited, she has the same characteristics as her parents. Similarly, the different classes create a genealogical tree, so that we have parent classes (the base class) and children classes (subclasses or derived classes). So is the basic explanation of the concept.

  The child can be improved with her own characteristics, as in the real world. A child is the genetic combination of her two parents, but she also develops her own personality. The difference is that, in object-oriented programming, a child class can, generally, have only one parent[1].

  In Figure A.1, a scheme represents the concept of inheritance. In other words, inheritance can be explained through the development of a hierarchy of classes. The derived class[2] has all the variables and methods of the parent class. In object-oriented programming, at the top of each hierarchy exists the class from

[1]The notion of multiple inheritance–the possibility to inherit from more than one superclass–is supported by some object-oriented language, such as Eiffel, Python, C++, but not in Java.

[2]A derived class can be the parent class of the next level in the hierarchy.

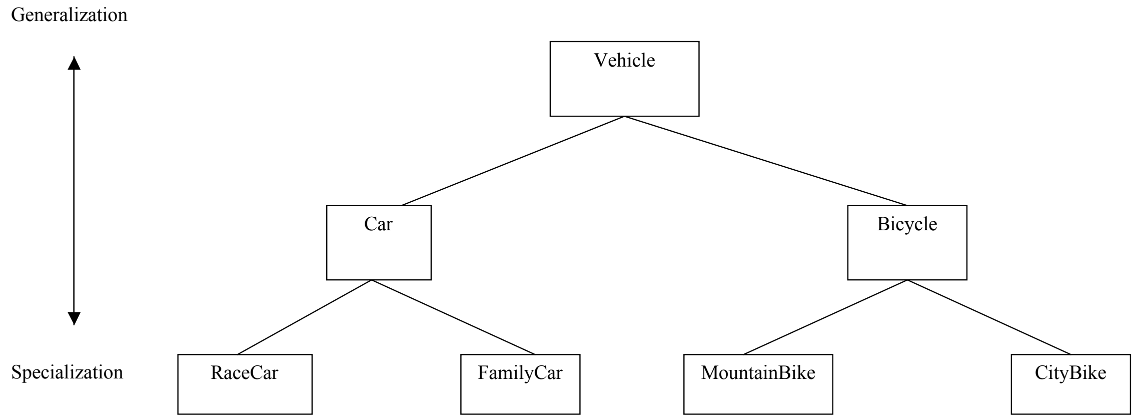



Figure A.1: A Class hierarchy

which all other classes are derived. For instance, this class is the *Object* class in Java. Implicitly, each class derives from the *Object* class.

In Figure A.1, a Vehicle class that includes all types of vehicles has been defined; this class is then used to create more specific classes – Car and Bicycle. These two classes are, in turn, used to define more detailed classes, and so forth. In other words, inheritance means moving between generalization and specialization[3] (Meyer, 2009, 594).

- *Polymorphism*: Polymorphism is derived from inheritance. The best way to explain this concept is to use it with an example. Imagine a parent class called Vehicle. This class has some methods, such as *go()*, *brake()*, *turn()* and so forth. TThe different subclasses – the children classes – through inheritance will have the same methods. However, these methods can be implemented to perform exactly the same tasks in the subclasses, or they can be rewritten – *overridden* in the Java terminology – to perform more specific task.

  If we create the following two subclasses, *Car* and *Bicycle*, their method *break()* (), for example, does exactly the same; that is, stop the vehicle, but the way the brakes are used to stop each vehicle is different. Thus, the method *brake()*

[3]This move between generalization and specialization can be expressed with the example of *matriochkas*–the famous russian dolls. The smallest one has only few details and corresponds to the class Vehicle. As the dolls get bigger, they become more detailed and the biggest one is the most detailed one.

must be overridden in order to take into account that difference. For instance, you cannot put the brakes of a bicycle on a race car. You simply won't be able to stop the car. To not rewrite the methods for corresponding to the needs will have the same result: the designed class won't do what you expect it to.

## A.2 Java as a programing language

In the beginning of the 1990s, under the lead of James Gosling at Sun Microsystems[4], a team of programmers developed a programming language for the programming of information devices (such as cellular phones) and home appliances (television or washing machines, for example) (Savitch, 2006). The idea was to develop a portable language that can work on every platform (Linux, MacOS, Windows) (Delannoy, 2007).

With the development of the internet and applets[5], Java has gained an increasing notoriety. Moreover, Java was used by several major companies in the industry such as IBM, INTEL, and Microsoft (Niemeyer and Jonathan, 2005, 3).

Since the launch of Java 1.0 in 1996, it has been regularly updated. Currently, version 6.21 is the latest version used. However, the next version, which should be a major update, should be inaugurated by the end of 2010.

As an object-oriented programming language, Java uses the main mechanisms described in Section 4.2.1, but it is not a fully object-oriented language, since it does not support multiple inheritance, for example.

In sum, Java has proved to be well suited for the development of Web applications and, as its use increases, more and more programmers find this language interesting for other applications. Therefore, Java has several advantages – portability, speed and security – that made it so interesting for other uses than solely the development of Web applets, as shown in Figure 4.2.1.

[4]Sun Microsystems is a US firm that produce softwares and hardwares, which has been acquired in 2009 by Oracle, a leading supplier of information management software. More on Sun and Java: http://www.oracle.com/technetwork/java/index.html.

[5]An applet can be defined as a "little Java application to be run from on a Web browser" (Savitch, 2006, 4)

The main advantages of Java are:

- *Portability*: Basically, portability as one of the great advantage of Java, means that it works on all platforms (Linux, MacOS, Windows). The motto that lies behind this advantage can be summarized as "Write once, run anywhere" (Meyer, 2009, 747) an advertising slogan developed by Sun Microsystems that is also known by the acronym of WORA.

- *Speed*: The problem of speed not only corresponds to a compilation time that should not be excessively long, but also to the other resources – such as the allocation of memory space, for instance – that are needed to run a software program without any problem, especially if you are working in network. At its debut Java was rather slow compared to the competing languages. However, the developers have made a great attempt to increase the speed of Java. The best example is that the video game Quake2 has been transferred to Java (Niemeyer and Jonathan, 2005, 8).

- *Security*: The idea that lies behind the concept of security is the idea of protecting the software against potential external attacks. This is more important if you are working in networks. The java.security package provides the needed classes to build the security skeleton of the software.

Java also comes with large libraries of already defined classes. You can have almost anything that you need. These libraries are very important for the programming tasks, as you can modify predefined classes through the use of inheritance and polymorphism for your own needs, thus saving a lot of time. However, their use may be a bit complicated for a newcomer, as they are huge, even if they are well structured[6]. It is like learning using an encyclopedia in another language. The *Application Programming Interface* or API – the *real* name of the library of classes – is structured with the help of packages, which are a method of grouping different

[6]The Java api can be found on the following website: http://download.oracle.com/javase/6/docs/api/.

classes, just as books are organized by general themes in public libraries. Instead of economics, history, mathematics and so forth, you will find the groups of classes that allow for the development of applets, the group that provides classes for security, for the development of user interfaces, and so on.
Using the keyword *import* with the needed package allows the programmer to use the predefined classes of the package in his program (Meyer, 2009). For instance the following import statement:

import uchicago.src.sim.gui.Drawable;

allows the researcher to use the class *Drawable* of the RePast's package uchicago.src.sim.gui, which gives the classes for the development of graphical users interface.

## A.3 Eclipse as an Integrated Development Environment[7]

Eclipse is an integrated development environment (IDE), which basically corresponds to a program used to develop software. Even if this IDE can handle several programming languages such as C++, C# or Python, Java is attached to Eclipse. In other words, Eclipse is mainly written in Java.
At the beginning, Eclipse was largely supported and financed by IBM. Several increases, since the launch of the first version in 2001, made Eclipse a very highly appreciated tool by the community of the Java programmers. Even though IBM is still working on its development, Eclipse is now managed as a foundation whose members are, among others, Cisco, Motorola, and Research In Motion (RIM).
The fact that Eclipse is so widely used comes from its several advantages:

- It is a free open-source IDE downloadable from the site http://www.eclipse.org/, and as the Eclipse community is ever growing, it is often updated. Furthermore, the documentation and tutorials are well developed and, as already

[7]This subsection is mainly based on the Eclipse tutorial written by Holzner (2004)

mentioned, even if it is Java-based, it supports several other programing language;

- Because Eclipse is Java-based, it is well suited for programming in that language; one of its great instruments is the programming assistant, which gives several possible solutions when the programmer is facing a problem. The knowledge of the different Java packages and classes is very important. However, this instrument can help target needs through the huge amount of information given by the different Java libraries.

## A.4 RePastJ

RePast is defined as an open-source "software framework for creating agent-based simulations using the Java language" (Collier, 2002, XX). It was developed at the Social Science Research Computing Center at the University of Chicago and, later, at the Argonne National Laboratory which is, according to its website[8], one of the largest U.S. national laboratories for science and engineering research.
RePast supplies an extensive Java API of already defined classes suited for the development of computational agent-based models. In other words, the basic architecture of Java classes needed to create one's own computational model have already been programmed at the University of Chicago, meaning that the basic architecture of visualization and editing tools have already been programmed. Put simply, RePast takes advantages of the object-oriented programming concepts of inheritance and polymorphism.
Besides these extensive libraries, it also has a large "How To" documentation that helps the beginners to acquire the basics of RePast functioning and programming. In addition, since its beginning, RePast has a very active and helpful mailing list (Tobias and Hofmann, 2004).
According to Collier (2002), the design of RePast goals, besides the ease of learning

[8]http://www.anl.gov/

and of use, are based on the following criteria:

- Abstraction corresponds to the fact that the essential elements for the construction of an agent-based model are developed as Java classes, since the concept of class is a conceptual representation of the target. Thus, RePast comes with an API of generic classes that define the tools for the creation of a computational ABM, such as the spaces, the display, the methods to collect data, and so forth[9].

- Flexibility and extensibility: These criteria rely mainly on the concept of polymorphism and inheritance that characterize object-oriented programming language, and, thus, Java. Through inheritance, the subclasses have the same parameters as the parent classes. Of course, Java is flexible enough to allow the introduction of one's own parameters. Moreover, with polymorphism, the different methods can be overridden in order to fit the needs of the modeler.

- Performance and scalability: This corresponds to the problem of speed explained above. The use of RePast should perform similarly to the ABM toolkit; which, according to Collier (2002), seems to be the case. In addition, as it is Java-based, it benefits from the improvement in the performance of Java.

- Interoperability: With RePast several external tools can be connected, such as the R statistics environment, for example. It can also support Geographic Information Systems (GIS). With such tool you can merge statistical analysis with cartography and data management in general.

To sum up the preceding sections, our model will be built using *Java* as programming language and *RePastJ* as an agent-based toolkit. Java is a low-level object oriented programming language that is platform-independent, secure, and well suited for developing agent-based models because each agent can be constructed as an object. RePastJ, the RePast implementation for Java, is an open-source software framework

[9]The RePast api can be found here: http://repast.sourceforge.net/api/index.html.

for developing agent-based simulations. It is fully object-oriented and has a very active mailing list, which is very helpful for the many questions that can arise. The model is developed in the free Integrated Development Environment (IDE) Eclipse, since it is an open-source, well-developed, and widely used IDE. It is also particularly well suited for programming in Java.

# Appendix B

# Examples of the countries expressed by their proximity array

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 58152 | 21102 | 98252 | 10690 | 98652 | 58602 | 93400 | 93400 | 93400 | 98282 | 98182 | 98252 | 89548 | 98252 |
| 58853 | 58152 | 50652 | 98188 | 58602 | 98188 | 93188 | 35806 | 93480 | 98282 | 09315 | 58182 | 99649 | 98852 |
| 58853 | 58853 | 58152 | 29064 | 98188 | 98188 | 98070 | 98070 | 38682 | 64482 | 09315 | 58182 | 99649 | 98887 |
| 98197 | 64294 | 64294 | 29064 | 02712 | 98188 | 35415 | 38682 | 38682 | 64482 | 98182 | 68185 | 98187 | 80535 |
| 98197 | 98197 | 95192 | 65865 | 02712 | 98188 | 78285 | 38282 | 38682 | 38682 | 68185 | 98182 | 80535 | 98187 |
| 98187 | 95187 | 95412 | 95412 | 65865 | 78185 | 78285 | 78285 | 68915 | 68185 | 68187 | 65987 | 60861 | 98187 |
| 04275 | 25412 | 75492 | 86001 | 79092 | 75192 | 78292 | 70221 | 70221 | 64911 | 74762 | 74762 | 65987 | 65187 |
| 65187 | 25412 | 25412 | 75192 | 86001 | 86001 | 75192 | 35112 | 69091 | 65917 | 64911 | 65987 | 65987 | 65187 |
| 37780 | 68587 | 32828 | 39867 | 75192 | 75192 | 75192 | 75192 | 38019 | 65917 | 65917 | 66672 | 98987 | 37780 |
| 72264 | 78507 | 39867 | 32828 | 99305 | 75192 | 75192 | 22904 | 38619 | 39099 | 65619 | 38619 | 98957 | 66598 |
| 72264 | 58507 | 78187 | 98107 | 28477 | 99305 | 22904 | 38619 | 33090 | 38619 | 38859 | 38619 | 66598 | 98957 |
| 78957 | 05111 | 38507 | 83907 | 99305 | 28477 | 37675 | 38810 | 38859 | 38859 | 38619 | 98957 | 78957 | 54637 |
| 78957 | 32127 | 58602 | 38507 | 38507 | 58602 | 22779 | 22779 | 38859 | 93409 | 38619 | 78252 | 78252 | 54637 |
| 32127 | 21102 | 58602 | 01708 | 58602 | 58602 | 07243 | 93400 | 93400 | 93400 | 98282 | 89548 | 98252 | 78252 |

Table B.1: The different proximity arrays at the end of a Moore

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 16533 | 48324 | 48324 | 92263 | 92263 | 65277 | 21272 | 74905 | 05287 | 50984 | 24389 | 26389 | 36610 | 02825 |
| 02825 | 02825 | 09875 | 69675 | 69675 | 65277 | 21272 | 35806 | 25377 | 24389 | 26389 | 26389 | 36610 | 02825 |
| 02825 | 02825 | 02825 | 69675 | 69675 | 69279 | 95277 | 95277 | 95375 | 04385 | 59814 | 26839 | 99649 | 02825 |
| 42325 | 64294 | 42325 | 29024 | 69378 | 69309 | 35415 | 95277 | 23761 | 04385 | 59814 | 66809 | 21535 | 21535 |
| 42325 | 42325 | 02315 | 09318 | 09307 | 29307 | 29307 | 99784 | 09507 | 92163 | 62805 | 62805 | 62805 | 62305 |
| 42904 | 42904 | 28984 | 28984 | 86047 | 25327 | 09307 | 09207 | 09507 | 09507 | 20534 | 62805 | 50410 | 50410 |
| 54075 | 46442 | 28984 | 13593 | 86047 | 25327 | 29827 | 09507 | 09507 | 09507 | 62509 | 62509 | 78103 | 85778 |
| 54075 | 46442 | 46442 | 13593 | 24627 | 29827 | 29827 | 09507 | 75697 | 75697 | 62509 | 11220 | 78103 | 85778 |
| 25728 | 35822 | 35822 | 24627 | 25327 | 29827 | 70351 | 27099 | 62509 | 62509 | 62509 | 62509 | 45465 | 25728 |
| 25728 | 24625 | 24728 | 24627 | 25327 | 25327 | 70351 | 27099 | 61048 | 62509 | 62509 | 62509 | 45465 | 25728 |
| 72264 | 24625 | 24728 | 24327 | 25327 | 05227 | 05227 | 05227 | 61048 | 62509 | 62509 | 16249 | 76596 | 54210 |
| 24625 | 24625 | 24625 | 24627 | 93029 | 93029 | 37675 | 05227 | 05287 | 29614 | 29614 | 25993 | 76596 | 54210 |
| 24625 | 24625 | 24625 | 24625 | 81800 | 57033 | 57033 | 22149 | 05287 | 05287 | 05287 | 25993 | 04681 | 24625 |
| 24625 | 24625 | 24625 | 81800 | 81800 | 65277 | 07243 | 74905 | 05287 | 05287 | 04387 | 04387 | 04681 | 24625 |

Table B.2: The different proximity arrays at the end of a Von Neumann

# Appendix C

# The program of the model of policy diffusion

```java
package diffusion1.diffusioninterdep1_2.diffusion_phd;


import java.awt.Color;
import java.util.Hashtable;
import java.util.Iterator;
import java.util.Vector;

import uchicago.src.reflector.DescriptorContainer;
import uchicago.src.sim.gui.DisplayConstants;
import uchicago.src.sim.gui.Drawable;
import uchicago.src.sim.gui.SimGraphics;
 * The problem is to know if and how to develop a class Policy: ideology, color, ...
import uchicago.src.sim.util.Random;

public class Country implements Drawable, DescriptorContainer{
    static final double E = Math.E;
    // the proximity array
    int[] proximity;
    int numCountries; // the number of countries in the world
    // the localization of the country on the grid
    int x, y, countryID;
    // each agent is surrounded by max 8 neighbors, so handle to the neighbor(s)
    Country aNeighbor; //the neighbors of the country is of class Country
    Vector <Country> neighbors; // the list of neighbors
    Vector <Country> vnNeighbors;

    Vector <Country> changedNeighbors; // the list of neighbors that have changed their policy
    Vector <Country> vnChangedNeighbors;

    Vector <Country> similarNeighbors; // the list of neighbors that are similar (among the list of changed neighbors)
    Vector <Country> vnSimilarNeighbors;

    int numNeighbors; // to sum up the neighbors
    int vnNumNeighbors;
    int numChangedNeighbors; // to sum up the neighbors that have changed = changedNeighbors.count
    int vnNumChangedNeighbors;
```

```java
    int region;
    int numColor;
    // the effectiveness
    double policyEffectiveness;
    double policyEffectivenessMean = 0.0;
    double policyEffectivenessSD = 0.4;
    double bestEffectiveness;
    // the parameters to modify policy effectiveness
    double effectivenessChange;
    double effectivenessChangeMean = -0.01;
    double effectivenessChangeSD = 0.03;
    // the policy preference
    double policyPreference;
    double policyPreferenceMean = 0.0;
    double policyPreferenceSD = 0.2;
    // the preference (not the ideology) changes
    double policyPreferenceChange;
    double policyPreferenceChangeMean = 0.00;
    double policyPreferenceChangeSD = 0.02;
    // the parameters to create the political constraints
    double politicalConstraints; // strength of the veto players: the greater the stronger!
    double politicalConstraintsMean = 0.0;
    double politcalConstraintsSD = 0.3;

    public double choiceVariable;
    public double changeVariable;
    public double meanEffectiveness;

    Color color;
    Color newColor;

    double pcv;

    Model model; // handle to the model

    Hashtable descriptors;

    public Country(int id, Color color, Model m){
        this.x = x;
        this.y = y;
        countryID = id;
        model = m;

```

```java
        this.color = color;

        neighbors = new Vector<Country>();
        vnNeighbors = new Vector<Country>();

        similarNeighbors = new Vector<Country>();
        vnSimilarNeighbors = new Vector<Country>();

        changedNeighbors = new Vector<Country>();
        vnChangedNeighbors = new Vector<Country>();

        proximity = new int [model.numProximity];
        for (int i = 0; i < model.numProximity; i++){
            proximity[i] = model.getNextIntFromTo(0, model.numTraits - 1);
        }

        this.policyEffectiveness = createNormalDistribution(policyEffectivenessMean, policyEffectivenessSD);
        this.policyPreference = createNormalDistribution(policyPreferenceMean, policyPreferenceSD);
        this.politicalConstraints = createNormalDistribution(politicalConstraintsMean, politcalConstraintsSD);
        descriptors = new Hashtable();
    }

    /**
     * Setting the agent position on the grid
     */
    public final void placeTo(int a, int b){
        x = a;
        y = b;
    }

    /***************************************************************************************
     *  The normal distribution                    *
     *                                                        *

```

```java
**************************************************************************************/

    public static double createNormalDistribution(double mean, double sd){
        Random.createNormal(mean, sd);
        double param = Random.normal.nextDouble(mean, sd);
        if (param > 1){
            param = 1;
        } else if (param < -1){
            param = -1;
        }

        return param;
    }

    public double changeParams(double param, double change, double changeMean, double changeSD){
        change = createNormalDistribution(changeMean, changeSD);
        param = param+change;

        if(param > 1){
            param = 1;
        } else if (param < -1){
            param = -1;
        }

        return param;
    }

    public double changePreference(){
        if (model.getTickCount()%model.elections == 0){
            policyPreference = createNormalDistribution(policyPreferenceMean,
                    policyPreferenceSD);
        }
        return policyPreference;
    }

    public double changeEffectiveness(){
        policyEffectiveness = changeParams(policyEffectiveness, effectivenessChange,
                effectivenessChangeMean,
```

```java
effectivenessChangeSD);
        return policyEffectiveness;
    }

    /
***************************************************************
*****************************************************/
    /*************************** Calculation of the readiness
***********************************************************/
    /
***************************************************************
*****************************************************/
    public void reset (){
        switch (model.neighborhood){
        case Model.MOORE:
            countChangedNeighbors(x, y).clear();
            break;
        case Model.VON_NEUMANN:
            countChangedNeighbors(x, y).clear();
            break;
        }
    }
    /*
     * When the effectiveness of the current policy is lower than the preference,
     * that is when the policy is so ineffective that it is not ideologically sustainable,
     * the country is ready for change => true!
     */
    public boolean ready(){
        switch (model.neighborhood){
        case Model.MOORE:
            if (policyEffectiveness < policyPreference){
                return true;
            }
            break;
        case Model.VON_NEUMANN:
            if (policyEffectiveness < policyPreference){
                return true;
            }
            break;
        }
        return false;
    }
```

```java

    /********************************************************************************************************************/
    /*************************** Choice of the neighbor to interact with ************************************************/
    /********************************************************************************************************************/

    /*
     * To count the neighbors that have changed their policy, we loop through them (according to the chosen neighborhood,
     * here the Moore neighborhood) and we create a list with them. For a change to occur, the country must have changed
     * its policy: effectiveness and color.
     *
     * the problem is that this method must return an array with max 8 objects in it. Here as it is programmed yet,
     * it returns a cumulative numbers of changed neighbors. It must returns the number of changed neighbors PER
     * country!!!!!!!
     *
     * With this condition we suppose that the introduction of the alternative effectiveness means the policy change,
     * this is done in order to avoid the StackOverFlow error, that an infinite recursive loop!
     */
    public Vector<Country> countChangedNeighbors(int px, int py){
        switch (model.neighborhood){
        case Model.MOORE:
            neighbors = model.world.getMooreNeighbors(px, py, false);
            numChangedNeighbors = 0;
            Iterator it = neighbors.iterator();
            while(it.hasNext()){
                Country changedCountry = (Country)it.next();
                if (changedCountry.bestEffectiveness == policyEffectiveness
                        && changedCountry.updatePolicyColor() == true){
                    changedNeighbors.add(changedCountry);
                    numChangedNeighbors++;
```

```java
                }
            }
            break;
        case Model.VON_NEUMANN:
            neighbors = model.world.getVonNeumannNeighbors(px, py, false);
            vnNumChangedNeighbors = 0;
            Iterator vnIt = neighbors.iterator();
            while(vnIt.hasNext()){
                Country changedCountry = (Country)vnIt.next();
                if (changedCountry.bestEffectiveness == policyEffectiveness
                        && changedCountry.updatePolicyColor() == true){
                    changedNeighbors.add(changedCountry);
                    vnNumChangedNeighbors++;
                }
            }
            break;
        }

        return changedNeighbors;
    }
    /*
     * The country looks for the most similar neighbors among the ones that have changed their policy.
     * First we define the similar neighbors as the ones that have changed their policy....
     * better than introduce the policy of the most similar: count the number of changed neighbors, loop
     * through them to find the most effective one. When the condition of choice is ok, introduce the policy
     * of the most effective neighbor and then increase the proximity between these 2 countries!! this is an important
     * difference with Axelrod's model: the country is not randomly chosen in the world but is a specific one!
     *
     * This methods returns the neighbor that is similar among the neighbors that have changed their policy
     */
    public Vector findSimilar(int px, int py){
        int numSimilar = similarNeighbors.size();
        numSimilar = 0;
        switch (model.neighborhood){
```

```
        case Model.MOORE:
            Iterator it = countChangedNeighbors(px, py).iterator();
            while (it.hasNext()){
                Country similarNeighbor = (Country)it.next();
                if (countAlikeDimensions(similarNeighbor) ==
                    similarNeighbor.countAlikeDimensions(this)) {
                    similarNeighbors.add(similarNeighbor);
                }
            }
            break;
        case Model.VON_NEUMANN:
            Iterator vnIt = countChangedNeighbors(px, py).iterator();
            while (vnIt.hasNext()){
                Country similarNeighbor = (Country)vnIt.next();
                if (countAlikeDimensions(similarNeighbor) ==
                    similarNeighbor.countAlikeDimensions(this)) {
                    similarNeighbors.add(similarNeighbor);
                }
            }
            break;
        }
        return similarNeighbors;
    }

    /*
     * To choose an alternative policy, a country must look at its neighbors (function of the chosen neighborhood).
     * While looking among its neighbors, it is searching for the neighbors that have changed their policy.
     * Among the neighbors that have changed their policy, the country looks for the similar one. When this latter
     * is found, the country "store" this effectiveness level...
     */
    public boolean chooseAlternativePolicy(){
        double choiceThreshold = 0.0;
        switch (model.neighborhood){
        case Model.MOORE:
            choiceThreshold = Random.uniform.nextDoubleFromTo(-2.0, 2.0);
```

```java
            choiceVariable =calculateChoiceVariable(x, y);
            if (choiceVariable > choiceThreshold){
                findSimilar(x, y);
                return true;
            }
            break;
        case Model.VON_NEUMANN:
            choiceThreshold = Random.uniform.nextDoubleFromTo(-2.0, 2.0);
            choiceVariable =calculateChoiceVariable(x, y);
            if (choiceVariable > choiceThreshold){
                findSimilar(x, y);
                return true;
            }
            break;
        }
        return false;
    }

    /********************************************************************************************************/
    /********************* Calculation of the choice variable ***********************************************/
    /********************************************************************************************************/

    /*
     * This method is used to calculate the diffusion variable. This variable is corresponding
     * a simplified computational notation of the change equation of the theoretical model. To develop
     * this variable,we consider the payoffs as fixed. To calculate it, we loop through the
     * neighbors, calculate the proportion of neighbors that have changed their policy and introduce
     * it the calculus of this variable. This diffusion variable is used in the determination of
     * choice. We calculate the mean effectiveness of the changed neighbors and then compare to ours!
     *
     * choiceVariable = 0.05+x;
     * Cf simmons and elkins 2004: "In this stylized scenario,
```

```
one can see that a small set of actors (about 5%) would
     * adopt the policy even if no one else is expected to do so."
     */
    public double calculateChoiceVariable(int px, int py){
        double pCV = 0.0;
        switch(model.neighborhood){
        case Model.MOORE:
            Vector moooreNeighbors = model.world.getMooreNeighbors(px, py, false);
            numNeighbors = moooreNeighbors.size();
            numChangedNeighbors = countChangedNeighbors(px, py).size();
            meanEffectiveness = calculateMeanEffective(px, py);
            pCV = ((meanEffectiveness-policyEffectiveness)*(numChangedNeighbors
                    /numNeighbors));
        break;
        case Model.VON_NEUMANN:
            Vector vonNeumannNeighbors = model.world.getVonNeumannNeighbors(px, py, false);
            vnNumNeighbors = vonNeumannNeighbors.size();
            vnNumChangedNeighbors = countChangedNeighbors(px, py).size();
            meanEffectiveness = calculateMeanEffective(px, py);
            pCV = ((meanEffectiveness-policyEffectiveness)*(vnNumChangedNeighbors
                    /vnNumNeighbors));

        break;
        }
        return pCV;
    }

    /*
     * With this method we calculate the mean effectiveness. We loop through the changed neighbors!
     */
    public double calculateMeanEffective(int px, int py){
        meanEffectiveness = 0.0;
        switch(model.neighborhood){
        case Model.MOORE:
            numChangedNeighbors = countChangedNeighbors(px, py).size();
```

```java
                Iterator it = countChangedNeighbors(px, py).iterator();
                while(it.hasNext()){
                    Country changedNeighbor = (Country)it.next();
                    for (int i = 0; i < numChangedNeighbors; i++){
                        meanEffectiveness = (meanEffectiveness +
                        changedNeighbor.policyEffectiveness)/numChangedNeighbors;
                    }
                }
                break;
            case Model.VON_NEUMANN:
                vnNumChangedNeighbors = countChangedNeighbors(px, py).size();
                Iterator vnIt = countChangedNeighbors(px, py).iterator();
                while(vnIt.hasNext()){
                    Country changedNeighbor = (Country)vnIt.next();
                    for (int i = 0; i < vnNumChangedNeighbors; i++){
                        meanEffectiveness = (meanEffectiveness +
                        changedNeighbor.policyEffectiveness)/vnNumChangedNeighbors;
                    }
                }
                break;
        }

        return meanEffectiveness;
    }

    /*****************************************************************************************************/
    /******************** Axelrod's routine from Cederman and Girardin's culturemodel *********************/
    /*****************************************************************************************************/
    /*
     * At the beginning of each interaction, a country picks up a neighbors and comparison of the similarity:
     * 0.0 means no similar traits, and 1.0 means all traits the same. Here it depends on the neighborhood!
```

```
     */
    public double countAlikeDimensions(Country n){
        int same = 0;
        for (int i = 0; i < model.numProximity; i++){
            if (proximity[i] == n.proximity[i]){
                same++;
            }
        }
        return (double)same/(double)model.numProximity;
    }
    /*
     * At what step should we introduce this function?
     * The Axelrod's algorithm for rendering neighbors more alike. Each
     */
    public boolean proximate(Country n){
        int[] different = new int[model.numProximity];
        int numDifferent = 0;
        for (int i = 0; i < model.numProximity; i++){
            if(proximity[i] != n.proximity[i]){
                different[numDifferent]=i;
                numDifferent++;
            }
        }
        if (numDifferent > 0){
            int feature = different [model.getNextIntFromTo(0,
                    numDifferent-1)];
            n.proximity[feature]=proximity[feature];
            return true;
        }
        return false;
    }

    /*
     * This method counts the % of alike proximity dimensions
     */
    public double countProximity(Country n){
        if (n != null){
            double near = countAlikeDimensions(n);
            return near;
        } else {
            return 1.0;
        }
    }
```

```java

    /***************************************************************************************************************/
    /************************** Calculation of the change routine ***********************************************/
    /***************************************************************************************************************/

    public double calculateChangeVariable(int px, int py){
        double effective = 0.0;
        changeVariable = 0.0;
        switch (model.neighborhood){
        case Model.MOORE:
            effective = calculateBestEffectiveness(px, py);
            changeVariable = (effective-policyEffectiveness);
            break;
        case Model.VON_NEUMANN:
            effective = calculateBestEffectiveness(px, py);
            changeVariable = (effective-policyEffectiveness);
            break;
        }
        return changeVariable;
    }

    /*
     * If the country has chosen an alternative policy (choose = true), a country changes its current
     * policy if the change variable is lower that the political constraints divided by the costs
     * (see Braun and Gilardi for more info!)
     */

    public boolean changePolicy(){
        double logit = 0.0;
        double changeLogit = 0.0;
        switch (model.neighborhood){
        case Model.MOORE:
            logit = calculateLogit();
            changeLogit = createBernoulli(1, logit);
            if (changeLogit == 1){
                return true;
            }
```

```
                break;
            case Model.VON_NEUMANN:
                logit = calculateLogit();
                changeLogit = createBernoulli(1, logit);
                if (changeLogit == 1){
                    return true;
                }
                break;
            }
            return false;
        }

        /*
         * search effective neighbor, increase similarity and introduce policy!
         */
        public double calculateBestEffectiveness(int px,int py){
            newColor = color;
            bestEffectiveness = getPolicyEffectiveness();
            switch (model.neighborhood){
            case Model.MOORE:
                Iterator it =findSimilar(px, py).iterator();
                while(it.hasNext()){
                    Country effectiveNeighbor = (Country)it.next();
                    double mostEffective = effectiveNeighbor.getPolicyEffectiveness();
                    if (mostEffective > bestEffectiveness){
                        bestEffectiveness = mostEffective;
                        newColor = effectiveNeighbor.color;
                        proximate(effectiveNeighbor);
                    }
                }
                break;
            case Model.VON_NEUMANN:
                Iterator vnIt =findSimilar(px, py).iterator();
                while(vnIt.hasNext()){
                    Country effectiveNeighbor = (Country)vnIt.next();
                    double mostEffective = effectiveNeighbor.getPolicyEffectiveness();
                    if (mostEffective > bestEffectiveness){
                        bestEffectiveness = mostEffective;
                        newColor = effectiveNeighbor.color;
                        proximate(effectiveNeighbor);
```

```java
                }
            }
        }

    return bestEffectiveness;
    }

    /*
     * this method returns true if the newPolicy is assigned to policy, that is if the current policy
     * is the policy of the most effective neighbors has calculated in the updatePolicy(int px, int py)
     * method!
     */

    public boolean updatePolicyEffectiveness(){
        policyEffectiveness = bestEffectiveness;
        return true;
    }

    public boolean updatePolicyColor(){
        color=newColor;
        return true;
    }

    public boolean updatePreference(){
        policyPreference = changeParams(policyPreference, policyPreferenceChange,
                policyPreferenceChangeMean, policyPreferenceSD);
        return true;
    }

    /************************************************************************************************************/
    /****************** The construction of the Bernoulli trial *************************************************/
    /************************************************************************************************************/
    /*
     * Bernoulli distribution: discrete probability distribution that takes value 1 with probability p
```

```
     * and 0 with probability q=1-p.
     * the binomial distribution gives the probability distribution of success in a sequence of n
     * independent y/n experiments, each of which yields success with probability p. If the number
     * of experiment = 1, the binomial distribution is a Bernoulli trial!
     */
    public double createBernoulli(int n, double logit){
        logit = calculateLogit();
        Random.createBinomial(n, logit);
        double change = Random.binomial.nextInt(n, logit);
        return change;
    }
    /*
     * the idea:
     */

    public double calculateLogit(){
        pcv = 0.0;
        changeVariable = 0.0;
        switch (model.neighborhood){
        case Model.MOORE:
            neighbors = model.world.getMooreNeighbors(x, y, false);
            numNeighbors = neighbors.size();
            changedNeighbors = countChangedNeighbors(x, y);
            numChangedNeighbors = changedNeighbors.size();
            changeVariable = calculateChangeVariable(x, y);
            double beta = numChangedNeighbors/numNeighbors;
            double z = (0.05+changeVariable+(politicalConstraints*beta));
            pcv = ((Math.pow(E, z))/(1+Math.pow(E, z)));
            break;
        case Model.VON_NEUMANN:
            vnNeighbors = model.world.getVonNeumannNeighbors(x, y, false);
            vnNumNeighbors = vnNeighbors.size();
            vnChangedNeighbors = countChangedNeighbors(x, y);
            vnNumChangedNeighbors = changedNeighbors.size();
            changeVariable = calculateChangeVariable(x, y);
            double vnBeta = vnNumChangedNeighbors/vnNumNeighbors;
            double vnZ = (0.05+changeVariable+
```

```
(politicalConstraints*vnBeta));
            pcv = ((Math.pow(E, vnZ))/(1+Math.pow(E, vnZ)));
            break;
        }
        return pcv;
    }

    /
*****************************************************************
*******************************************
     ******************************** To be printed in the
console ********************************************
************************************************************************************
************************************************************************************************************/

    /*
     * the proximity array
     */
    public String proximityToString(){
        String close = " ";
        for (int i = 0; i < model.numProximity; i++){
            close = close + proximity[i];
        }
        return close;
    }

    /*
     * the different variables (effectiveness, preference,
political constraints)
     */
    public String toString(){
        return "  [Country (" + countryID + "): e: " +
        policyEffectiveness + ", p: " + policyPreference +
         ", c: " + politicalConstraints + "]";
    }

    /*
     * The color on the grid
     */
    public void draw(SimGraphics g) {
```

```java
        g.setDrawingParameters
(DisplayConstants.CELL_WIDTH*2/3,
DisplayConstants.CELL_HEIGHT*2/3,
DisplayConstants.CELL_DEPTH*2/3);
        g.drawFastRoundRect(color);
    }

    /
**************************************************************
********************************************/
    /************************     the getters and
setters
********************************************/
    /
**************************************************************
********************************************/

    public double getPolicyEffectivenessChange(){
        return effectivenessChange;
    }
    public void setPolicyEffectivenessChange(double pec){
        effectivenessChange = pec;
    }
    public double getPolicyPreference(){
        return policyPreference;
    }
    public void setPolicyPreference(double pp){
        policyPreference = pp;
    }
    public double getPolicyPreferenceMean(){
        return policyPreferenceMean;
    }
    public void setPolicyPreferenceMean(double ppm){
        policyPreferenceMean = ppm;
    }
    public double getPolicyPreferenceSD(){
        return policyPreferenceSD;
    }
    public void setPolicyPreferenceSD(double ppsd){
        policyPreferenceSD = ppsd;
    }
    public double getPoliticalConstraints(){
        return politicalConstraints;
    }
```

```
    public void setPoliticalConstraints(double pc){
        politicalConstraints = pc;
    }
    public int getX() {
        return x;
    }
    public int getY() {
        return y;
    }
    public double getPolicyEffectiveness(){
        return policyEffectiveness;
    }
    public void setPolicyEffectiveness(double pe){
        policyEffectiveness = pe;
    }
    public double getPolicyEffectivenessMean(){
        return policyEffectivenessMean;
    }
    public void setPolicyEfectivenessMean(double pem){
        policyEffectivenessMean = pem;
    }
    public double getPolicyEffectivenessSD(){
        return policyEffectivenessSD;
    }
    public void setPolicyEfectivenessSD(double pesd){
        policyEffectivenessSD = pesd;
    }
    public double getBestEffectiveness(){
        return bestEffectiveness;
    }
    public void setBestEffectiveness(double be){
        bestEffectiveness = be;
    }
    public Hashtable getParameterDescriptors() {
        return descriptors;
    }

    public void setColor(Color color){
        this.color = color;
    }
    public Color getColor(){
        return color;
    }
    public Color getNewColor(){
```

```
        return newColor;
    }
    public void setNewColor(Color nc){
        newColor = nc;
    }

    public double getChoiceVariable(){
        return choiceVariable;
    }
    public void setChoiceVariable(double cv){
        choiceVariable = cv;
    }
    public double getChangeVariable(){
        return changeVariable;
    }
    public void setChangeVariable(double cv){
        changeVariable = cv;
    }
    public double getMeanEffectiveness(){
        return meanEffectiveness;
    }
    public void setMeanEffectiveness(double ae){
        meanEffectiveness = ae;
    }
    Country getNeighbor(int pos){
        final int bounds[][] = {{1,1}, {1,0}, {0,1}, {0,-1},
{1,-1}, {-1,1}, {-1,0}, {-1,-1}};

        int xx = x+bounds[pos][0];
        int yy = y+bounds[pos][1];
        if (xx>0 && xx < model.world.getSizeX() &&
                yy>0 && yy< model.world.getSizeY()){
            return (Country)model.world.getObjectAt(xx, yy);

        }
        return null;
    }
    public Vector getChangedNeighbors(){
        return changedNeighbors;
    }
    public void setChangedNeighbors(Vector cn){
        changedNeighbors = cn;
    }
    public Vector getSimilarNeighbors(){
```

```
        return similarNeighbors;
    }
    public void setSimilarNeighbors(Vector sn){
        similarNeighbors = sn;
    }
    public Vector getVNNeighbors(){
        return vnNeighbors;
    }
    public void setVNNeighbors(Vector vnn){
        vnNeighbors = vnn;
    }
    public Vector getVNChangedNeighbors(){
        return vnChangedNeighbors;
    }
    public void setVNChangedNeighbors(Vector vncn){
        vnChangedNeighbors = vncn;
    }
    public Vector getVNSimilarNeighbors(){
        return vnSimilarNeighbors;
    }
    public void setVNSimilarNeighbors(Vector vnsn){
        vnSimilarNeighbors = vnsn;
    }
    public int getRegion(){
        return region;
    }
    public void setRegion(int r){
        region = r;
    }
    public int getNumColor (){
        return numColor;
    }
    public void setNumColor(int nc){
        numColor = nc;
    }

    public double getNumChangedNeighbors(){
        numChangedNeighbors = 0;
        if(updatePolicyColor() == true){
            numChangedNeighbors++;
        }

        return (double)(numChangedNeighbors/
model.numCountries);
```

```
    }
    public double getVNNumChangedNeighbors(){
        vnNumChangedNeighbors = 0;
        if (updatePolicyColor() == true){
            vnNumChangedNeighbors++;
        }
        return (double)(vnNumChangedNeighbors/
model.numCountries);
    }
    public void setVNNumChangedNeighbors(int vnncn){
        vnNumChangedNeighbors = vnncn;
    }
    public double getPCV(){
        return pcv;
    }
    public void setPCV(double probvar){
        pcv = probvar;
    }
}
```

```java
package diffusion1.diffusioninterdep1_2.diffusion_phd;

import java.awt.Color;
import java.util.*;

import uchicago.src.sim.engine.SimInit;
import uchicago.src.sim.engine.SimpleModel;
import uchicago.src.sim.space.Object2DTorus;
import uchicago.src.sim.util.SimUtilities;

public class Model extends SimpleModel {
    // the definition of the different kinds of neighborhood
    public static final int MOORE = 0;
    public static final int VON_NEUMANN = 1;
    int neighborhood;
    // the shape of the world
    Object2DTorus world;
    int worldSize;
    // the time between elections
    int elections;
    // the number of countries
    int numCountries;
    int numNeighbors;
    // the countries that have changed the policy
    int numChangedCountries;
    // the average effectiveness and preference
    double meanEffectiveness;
    double averagePreference;
    // the different dimension of the proximity array!
    int numProximity;
    // the different values of the dimensions for creating the
proximity "index"
    int numTraits;

    Color color;
    int numColors;
    int[] num;

    Country country;

    public Model(){
        super();

    }
```

```java

    public void setup(){
        super.setup();
        // setting the name of the model
        name = "Model of policy diffusion";
        // the initial size of the world
        worldSize = 14;
        // at the beginning, we set the Moore neighborhood as the default neighborhood
        neighborhood = MOORE;
        // the initial number of possible dimensions that compose the proximity
        numProximity = 5;
        // the initial number of possible values per dimension
        numTraits = 10;
        // at the beginning the initial time between elections
        elections=5;
    }

    public void buildModel(){
        super.buildModel();
        // the creation of the world
        world = new Object2DTorus(worldSize, worldSize);
        // the number of countries
        numCountries = worldSize*worldSize;

        numColors = numCountries;
        for (int i = 0; i<numColors; i++){
            num = new int[i];
        }

        int countryID = 0;

        for (int i = 0; i < numCountries; i++){
            countryID++;
            color = new Color(getNextIntFromTo(0, 255), getNextIntFromTo(0, 255), getNextIntFromTo(0,255));
            final Country country = new Country (countryID, color, this);
            agentList.add(country);
            country.setColor(color);
        }

        SimUtilities.shuffle(agentList);
```

```java

        for (int x = 0; x < worldSize; x++){
            for (int y = 0; y < worldSize; y++){
                Country country = (Country)agentList.get(x*worldSize+y);
                world.putObjectAt(x, y, country);
                country.placeTo(x, y);
            }
        }

        for (int x = 0; x < worldSize; x++){
            for (int y = 0; y < worldSize; y++){
                Country country = (Country)world.getObjectAt(x, y);
                switch (neighborhood){
                case MOORE:
                    country.neighbors.addAll(world.getMooreNeighbors(x, y, false));
                    break;
                case VON_NEUMANN:
                    country.neighbors.addAll(world.getVonNeumannNeighbors(x, y, false));
                    break;
                }
            }
        }
    }

    public void step(){
        resetChange();
        changeParam();
        readyForChange();
        choose();
        change();
        reportResults();
    }

    public void resetChange(){
        for(int i = 0; i < numCountries; i++){
            Country country = (Country)agentList.get(i);
            country.reset();
        }
    }

```

```java
    public void changeParam(){
        for (int i = 0; i < numCountries; i++){
            Country country = (Country)agentList.get(i);
            country.changeEffectiveness();
            country.changePreference();
        }

    }

    public void readyForChange(){
        for (int i = 0; i< numCountries; i++){
            Country country = (Country)agentList.get(i);
            country.ready();
        }
    }

    public void choose(){
        for (int i = 0; i < numCountries; i++){
            Country country = (Country)agentList.get(i);
            if (country.ready() == true){
                country.chooseAlternativePolicy();
            }
        }
    }

    public void change(){
        for (int i = 0; i < numCountries; i++){
            Country country = (Country)agentList.get(i);
            if (country.chooseAlternativePolicy() == true){
                country.changePolicy();
            }
        }

        for (int i = 0; i < numCountries; i++){
            Country country = (Country)agentList.get(i);
            if (country.changePolicy() == true){
                country.updatePolicyEffectiveness();
                country.updatePreference();
            }
        }
    }

    public double calculateMeanEffectiveness(){
        double meanEffect = 0.0;
```

```
        for (int i = 0; i < numCountries; i++){
            Country country = (Country)agentList.get(i);
            meanEffect = meanEffect +country.policyEffectiveness;
            meanEffectiveness = meanEffect/(double) numCountries;
        }

        return meanEffectiveness;
    }

    /**
     * count the number of regions according to the color!
     *
     */
    public void markRegion(Country n, int numRegions){
        n.region = numRegions;
        Iterator it = n.neighbors.iterator();
        while(it.hasNext()){
            Country neighbor = (Country)it.next();
            if ((neighbor.region == 0) && (n.color == neighbor.color)){
                markRegion(neighbor, numRegions);
            }
        }
    }

    public int regionCounter(){
        Iterator it = agentList.iterator();
        while(it.hasNext()){
            Country neighbor = (Country)it.next();
            neighbor.region = 0;
        }
        int numRegions = 0;
        it = agentList.iterator();
        while (it.hasNext()){
            Country neighbor = (Country)it.next();
            if (neighbor.region == 0){
                numRegions++;
                markRegion(neighbor, numRegions);
            }
        }
        return numRegions;
    }
```

```java

    /**
     * The part for the results: in the console, graphs, grid and the Batch mode
     */
    public void reportResults(){

        System.out.println(getTickCount());

        for (int i = 0; i < numCountries; i++){
            Country country = (Country)agentList.get(i);
            System.out.print(country.toString());
            System.out.println();
        }
        System.out.println();

        for (int x = 0; x < worldSize; x++){
            for (int j = 0; j < worldSize; j++){
                Country country = (Country)world.getObjectAt(x, j);
                System.out.print(country.proximityToString() + " &");
            }
            System.out.println();
        }

    }

    /*
     * the get and set methods
     */
    public int getNumNeighbors(){
        return numNeighbors;
    }
    public void setNumNeihbors(int nn){
        numNeighbors=nn;
    }

    public int getNumChangedCountries(){
        return numChangedCountries;
    }
    public void setNumChangedCountries(int ncc){
        numChangedCountries = ncc;
    }
```

```java

    public double getMeanEffectiveness(){
        return meanEffectiveness;
    }
    public void setMeanEffectiveness (double me){
        meanEffectiveness = me;
    }
    public double getAveragePreference(){
        for (int i = 0; i < numCountries; i++){
            Country country = (Country)agentList.get(i);
            averagePreference = averagePreference +country.policyPreference;
        }
        return averagePreference;
    }

    public int getNumColors(){
        return numColors;
    }
    public void setNumColors(int nc){
        numColors = nc;
    }
    public int[] getNum(){
        return num;
    }
    public void setNum(int[] no){
        num=no;
    }
    public int getNumProximity(){
        return numProximity;
    }

    public void setNumProximity(int np){
        numProximity = np;
    }
    public int getElections(){
        return elections;
    }
    public void setElelctions(int elect){
        elections=elect;
    }

    public static void main (String args[]){
        final SimInit init = new SimInit();
```

```
        Model m = new Model();
        init.loadModel(m, null, false);
    }
}

```

```java
package diffusion1.diffusioninterdep1_2.diffusion_phd;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Hashtable;

import uchicago.src.reflector.ListPropertyDescriptor;
import uchicago.src.reflector.RangePropertyDescriptor;
import uchicago.src.sim.analysis.OpenSequenceGraph;
import uchicago.src.sim.analysis.Sequence;
import uchicago.src.sim.engine.Controller;
import uchicago.src.sim.engine.SimInit;
import uchicago.src.sim.gui.DisplayConstants;
import uchicago.src.sim.gui.DisplaySurface;
import uchicago.src.sim.gui.Object2DDisplay;

public class ModelGUI extends Model{

    DisplaySurface dSurf;
    OpenSequenceGraph graphNeigh;
    OpenSequenceGraph graphEffect;
    OpenSequenceGraph graphPref;
    OpenSequenceGraph graphRegion;

    boolean countRegions;
    boolean countColorRegions;

    public ModelGUI(){
        super();
        Controller.ALPHA_ORDER = false;
        Controller.CONSOLE_ERR = false;
        Controller.CONSOLE_OUT = false;
    }

    /**
     *
     */

    public void setup(){
        super.setup();

        countRegions = true;
        countColorRegions = true;
        //countProxiRegions = true;
```

```

        DisplayConstants.CELL_WIDTH = 30;
        DisplayConstants.CELL_HEIGHT = 30;

        if (dSurf != null){
            dSurf.dispose();
        }

        if (graphNeigh != null){
            graphNeigh.dispose();
        }
        if (graphEffect != null){
            graphEffect.dispose();
        }
        if (graphRegion != null){
            graphRegion.dispose();
        }
        if (graphPref != null){
            graphPref.dispose();
        }

        params = new String[]{"WorldSize", "Neighborhood", "NumProximity", "NumTraits", "Elections"};
        // the different sliders
            // to choose the size of the world
        final RangePropertyDescriptor pdWorldSize = new RangePropertyDescriptor("WorldSize", 10, 100, 15);
        descriptors.put("WorldSize", pdWorldSize);

        Hashtable neighborType = new Hashtable();
        neighborType.put(new Integer(MOORE), "Moore Neighbors");
        neighborType.put(new Integer(VON_NEUMANN), "Von Neumann");
        ListPropertyDescriptor pdNeighborType = new ListPropertyDescriptor("Neighborhood", neighborType);
        descriptors.put("Neighborhood", pdNeighborType);

            // the choice the number of proximity features
        final RangePropertyDescriptor pdNumProximity = new RangePropertyDescriptor("NumProximity", 1, 25, 5);
        descriptors.put("NumProximity", pdNumProximity);
            // the number of possible values for the dimensions
        final RangePropertyDescriptor pdNumTraits = new
```

```java
RangePropertyDescriptor("NumTraits", 1, 25, 5);
        descriptors.put("NumTraits", pdNumTraits);

        final RangePropertyDescriptor pdElections = new RangePropertyDescriptor("Elections", 1, 25, 5);
        descriptors.put("Elections", pdElections);

        dSurf = new DisplaySurface(this, "A Torus World");
        registerDisplaySurface("Main", dSurf);

         modelManipulator.addButton("Refresh", new ActionListener() {
                public void actionPerformed(ActionEvent evt) {
                    dSurf.repaint();;
                }
            });

    }

    /*
     * At each time step, must count the neighbor that have changed! cumulative! but it shouldn't be linear...
     * At each time step, the number of changed neighbor is added to the precedent sum! -> at 0, for example, we have
     * 2, at 2, we have 7
     */
    /**
     * count the number of countries that have changed their policy
     */

    public double countAlikeCountries(){
        double numChangedCountries = 0;
        for (int i = 0; i < numCountries; i++){
            Country alikeNeighbors = (Country)agentList.get(i);
            if (alikeNeighbors.updatePolicyEffectiveness() == true){// && alikeNeighbors.updatePolicyColor() == true
                numChangedCountries = numChangedCountries +alikeNeighbors.getChangedNeighbors().size();
            }

        }
    return (numChangedCountries/(double)numCountries);
    }
```

```java

    class PropSeq implements Sequence{
        public double getSValue(){
            return countAlikeCountries();
        }
    }

    /**
     * count the number of regions: countries must have introduce the best effectiveness and the appropriated color
     */

    class Seq implements Sequence{
        public double getSValue(){
            return (double)regionCounter();
        }
    }

    /**
     * calculate the evolution of the average effectiveness of the world through the run
     */

    class EffectSeq implements Sequence{
        public double getSValue(){
            return calculateMeanEffectiveness();
        }
    }

    public double calculateMeanPreference(){
        double meanPreference = 0.0;
        for (int i = 0; i < numCountries; i++){
            Country country = (Country)agentList.get(i);
            meanPreference = meanPreference +country.policyPreference;
        }
        return meanPreference/(double)numCountries;
    }

    class PrefSeq implements Sequence{
        public double getSValue(){
            return calculateMeanPreference();
        }
    }
```

```
    /**
     *
     */

    public void buildModel(){
        super.buildModel();
        buildDisplay();
    }

    public void buildDisplay(){

        Object2DDisplay display = new Object2DDisplay(world);
        display.setObjectList(agentList);
        dSurf.addDisplayableProbeable(display, "Display");
        addSimEventListener(dSurf);
        dSurf.display();

        graphNeigh = new OpenSequenceGraph("Proportion of Neighbors", this);
        graphNeigh.setXRange(0.0, 100.0);
        graphNeigh.setYRange(0.0, (double)numCountries);
        graphNeigh.setAxisTitles("Time", "Nbr of countries");
        graphNeigh.addSequence("nbr of changed Neighbors", new PropSeq());

        graphEffect = new OpenSequenceGraph("The Average Effectiveness", this);
        graphEffect.setXRange(0.0, 100.0);
        graphEffect.setYRange(-1.0, (double) meanEffectiveness);
        graphEffect.setAxisTitles("Time", "Average Effectiveness");
        graphEffect.addSequence("mean Effect", new EffectSeq());

        graphPref = new OpenSequenceGraph("The Average Preference", this);
        graphPref.setXRange(0.0, 100.0);
        graphPref.setYRange(-1.0, (double) averagePreference);
        graphPref.setAxisTitles("Time", "Average Preference");
        graphPref.addSequence("mean Pref", new PrefSeq());

        graphNeigh.display();
        graphEffect.display();
```

```
        graphPref.display();

        if (countRegions){
            graphRegion = new OpenSequenceGraph("Number of regions", this);
            graphRegion.setXRange(0, 100);
            graphRegion.setYRange(0.0, (double)numCountries);
            graphRegion.setAxisTitles("Time", "Number of regions");
            graphRegion.addSequence("Regions", new Seq());
            graphRegion.display();
            graphRegion.step();
        }
    }

    public void step(){
        super.step();

        graphNeigh.step();
        graphEffect.step();
        graphPref.step();
        dSurf.updateDisplay();

        if (countRegions){
            graphRegion.step();
        }
    }

    public void postStep(){

    }

    /*
     * The getters and setters
     */
    public int getNeighborhood(){
        return neighborhood;
    }
    public void setNeighborhood(int n){
        neighborhood = n;
    }
    public int getNumCountries(){
        return numCountries;
    }
```

```java
    public void setNumCountries(int nc){
        numCountries = nc;
    }
    public int getWorldSize(){
        return worldSize;
    }
    public void setWorldSize(int ws){
        worldSize = ws;
    }

    public int getNumTraits(){
        return numTraits;
    }

    public void setNumTraits(int nt){
        numTraits= nt;
    }
    public int getElections(){
        return elections;
    }
    public void setElections(int e){
        elections = e;
    }

    public static void main (String args[]){
        SimInit init = new SimInit();
        ModelGUI mGUI = new ModelGUI();
        init.loadModel(mGUI, null, false);
    }

}
```

```java
package diffusion1.diffusioninterdep1_2.diffusion_phd;

import java.util.ArrayList;
import java.util.Iterator;
import uchicago.src.sim.analysis.LocalDataRecorder;
import uchicago.src.sim.engine.Controller;
import uchicago.src.sim.engine.SimInit;
import uchicago.src.sim.space.Object2DTorus;
import uchicago.src.sim.util.SimUtilities;

public class ModelBatch extends ModelGUI{

    int numOfTimeSteps;
    LocalDataRecorder recorder;
    Object2DTorus world2;
    ArrayList agentList2;

    public ModelBatch(){
        super();
        Controller.ALPHA_ORDER = false;
        Controller.CONSOLE_ERR = false;
        Controller.CONSOLE_OUT = false;
    }

    public void setup(){
        super.setup();

        //params = new String[]{"WorldSize", "Neighborhood", "Topology", "NumFeatures", "NumTraits", "NumRegions", "PolicyEffectiveness"};
        params = new String[]{"WorldSize", "NumFeatures", "NumTraits", "NumRegion", "PolicyEffectiveness"};
        numOfTimeSteps = 350;
    }

    public void buildModel(){
        super.buildModel();
        setStoppingTime(numOfTimeSteps);
        recorder = new LocalDataRecorder("./data.csv", this);
        recorder.createNumericDataSource("NumRegion", this, "computeNumRegions");
        recorder.createNumericDataSource("PolicyEffectiveness", this, "computeEffectiveness");
        recorder.createNumericDataSource("Effectiveness",
```

```
this, "computeCountryEffectiveness");
        /*
        for (int i = 0; i < numCountries; i++){
            Country c = (Country)agentList.get(i);
            recorder.createNumericDataSource(new String("Country "+c.countryID), c, "changeEffectiveness");
        }
        */
        recorder.setDelimeter("; ");
    }

    public void step(){
        super.step();
        recorder.record();
        recorder.write();
    }

    public final void atEnd(){
        super.atEnd();
        recorder.record();
        recorder.write();
    }

    public void markRegion2(Country n, int numRegions){
        n.region = numRegions;
        Iterator it = n.neighbors.iterator();
        while(it.hasNext()){
            Country neighbor = (Country)it.next();
            if (neighbor.region == 0 && neighbor.newColor == n.color){
                markRegion2(neighbor, numRegions);
            }
        }
    }

    public int computeNumRegions(){
        Iterator it = agentList.iterator();
        while(it.hasNext()){
            Country neighbor = (Country)it.next();
            neighbor.region = 0;
        }
        int numRegions = 0;
        it = agentList.iterator();
        while (it.hasNext()){
```

```java
            Country neighbor = (Country)it.next();
            if (neighbor.region == 0){
                numRegions++;
                markRegion2(neighbor, numRegions);
            }
        }
        return numRegions;
    }

    public double computeEffectiveness(){
        return meanEffectiveness;
    }

    public double computeCountryEffectiveness(){
        for (int i = 0; i < numCountries; ){
            Country  country = (Country)agentList.get(i);
            double logit = country.calculateLogit();
            return logit;
        }
        return 0.0;
    }


    /**
     * the getters and setters
     */
    public int getNumOfTimeSteps(){
        return numOfTimeSteps;
    }
    public void setNumOfTimeSteps(int nots){
        numOfTimeSteps = nots;
    }

    public static void main (String args[]){
        final SimInit init = new SimInit();
        String parameterFile = SimUtilities.getDataFileName("params.csv");
        final ModelBatch mb = new ModelBatch();
        init.loadModel(mb, parameterFile, true);
    }

}
```

# Bibliography

Abrahamson, Eric and Lori Rosenkopf. 1993. "Institutional and Competitive Band wagons: Using Mathematical Modeling as a Tool to Explore Innovation Diffusion." *Academy of Management Review* 21(1):254–285.

Abrahamson, Eric and Lori Rosenkopf. 1997. "Social Network Effects on the Extent of Innovation Diffusion: A Computer Simulation." *Organization Science* 8(3):289–309.

Adner, Ron. 2002. "WHEN ARE TECHNOLOGIES DISRUPTIVE? A DEMAND-BASED VIEW OF THE EMERGENCE OF COMPETITION." *Strategic Management Journal* 23(8):667–668.

Ahrweiler, Petra and Nigel Gilbert. 2005. "Caffè Nero: The Evaluation of Social Simulation." *Journal of Artificial Societies and Social Simulation* 8 (4).

Amblard, Frédéric. 2003. Comprendre le fonctionnement des simulations individus-centrées: Application à des modèles de dynamiques d'opinions PhD thesis Université Blaise Pascal Clermont-Ferrand: .

Amin, Ash and Frank Wilkinson. 1999. "Learning, proximity and industrial performance: an introduction." *Cambridge Journal of Economics* 23:121–125.

Armingeon, Klaus, Michelle Beyeler, Denis Bouget, Jane Lewis and Peter Taylor-Gooby, eds. 2004. *The OECD and European Welfare States*. Cheltenham: Edward Elgar.

Axelrod, Robert. 1984. *The Evolution of Cooperation.* Unites States of America: Basic Books.

Axelrod, Robert. 1997*a*. *The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration.* Princeton: Princeton University Press.

Axelrod, Robert. 1997*b*. "The Dissemination of Culture: A Model with Local Convergence and Global Polarization." *The Journal of Conflict Resolution* 41(2):203–226.

Axelrod, Robert. 2003. "Advancing the Art of Simulation in the Social Sciences." *Journal of the Japanese Society for Management Information System* 12(3):16–22.

Axelrod, Robert. 2005. *AGENT-BASED MODELING AS A BRIDGE BETWEEN DISCIPLINES*. Vol. 2 of *Handbook in Economics Series* North Holland Press.

Beck, Nathaniel, Kristian S. Gleditsch and Kyle Beardsley. 2006. "Space is More than Geography: Using Spatial Econometrics in the Study of Political Economy." *International Studies Quarterly* 50(1):27–44.

Becker, Joerg, Bjoern Niehaves and Karsten Klose. 2005. "A Framework for Epistemological Perspectives on Simulation." *Journal of Artificial Societies and Social Simulation* 8 (4).

Benoît, Kenneth. 2001. "Simulation Methodologies for Political Scientists." *The Political Methodologist* 10(1):12–16.

Berry, Frances Stokes and William D. Berry. 1990. "State Lottery Adoptions as Policy Innovations: An Event History Analysis." *American Political Science Review* 84(2):395–415.

Berry, Frances Stokes and William D. Berry. 2006. Innovation and Diffusion Models in Policy Diffusion. In *Theories of the Policy Process*, ed. Paul A. Sabatier. Westview Press pp. 223–260.

Besley, Timothy and Anne Case. 1995. "Incumbent Behavior: Vote-Seeking, Tax-Setting, and Yardstick Competition." *The American Economic Review* 85(1):25–45.

Bocsi, Rajmund, Viktor Erdélyi, Gábor Ferschi, László Gulyás, Márton Iványi and Zsolt Kúti. 2010. *MEME Model Exploration Module User Manual.* Budapest: AITIA.

Boehmke, Frederick J. and Richard Witmer. 2004. "Disentangling Diffusion: The Effects of Social Learning and Economic Competition on State Policy Innovation and Expansion." *Political Research Quaterly* 57(1):39–51.

Bonabeau, Eric. 2002. "Agent-based modeling: Methods and Techniques for simulating human systems." *Proceedings of the National Academy of Sciences* 99(3):7280–7287.

Bonoli, Giuliano. 2001. *Political Institutions, Veto Points, and the Process of Welfare State Adaptation.* Oxford University Press chapter 8, pp. 238–264.

Bonoli, Giuliano. 2003. "Two Worlds of Pension Reform in Western Europe." *Comparative Politics* 35(4):399–416.

Boschetti, Fabio, David McDonald and Randall Gray. 2008. "Complexity of a Modelling Exercise: A Discussion of the Role of Computer Simulation in Complex System Science." *Complexity* 13:21–28.

Boschma, Ron A. 2005. "Proximity and Innovation: A Critical Assessment." *Regional Studies* 39(1):61–74.

Botcheva, Liliana, Martin. and Lisa L. Martin. 2001. "Institutional Effects on State Behavior. Convergence and Divergence." *International Studies Quarterly* 45(1):1–26.

Braun, Dietmar and Fabrizio Gilardi. 2006. "Taking "Galton's problem" seriously.

Towards a theory of policy diffusion." *Journal of Theoretical Politics* 18(3):298–322.

Braun, Dietmar, Fabrizio Gilardi, Katharina Füglister and Stéphane Luyet. 2007. "Ex Pluribus Unum: Integrating the Different Strands of Policy Diffusion Theory." *Politische Vierteljahresschrift* 38:39–55.

Brooks, Sarah M. 2005. "Interdependent and Domestic Foundations of Policy Change: The Diffusion of Pension Privatization Around the World." *International Studies Quarterly* 49(2):273–294.

Brooks, Sarah M. 2007. "When Does Diffusion Matter? Explaining the Spread of Structural Pension Reforms Across Nations." *The Journal of Politics* 69(3):701–715.

Bullnheimer, B., H. Dawid and R. Zeller. 1998. "Learning from Own and Foreign Experience: Technological Adaptation by Imitating Firms." *Computational and Mathematical Organization Theory* 4(3):267–282.

Bunn, Derek W. and Christopher J. Day. 2009. "Computational modelling of price formation in the electricity pool of England and Wales." *Journal of Economic Dynamics and Control* 33(2):363–376.

Cao, Xun. 2010. "Networks as Channels of Policy Diffusion: Explaining World-wide Changes in Capital Taxation, 1998-2006." *International Studies Quarterly* 54(3):823–854.

Case, Anne C., Harvey S. Rosen and James R. Hines Jr. 1993. "Budget spillovers and fiscal policy interdependence: Evidence from the states." *Jounal of Public Economics* 52(3):285–307.

Casti, John L. 1997. *Would-be worlds: How Simulation is Changing the Frontiers of Science*. New-York - Chichester - Brsibane - Toronto - Singapore - Weinheim: John Wiley & Sons, Inc.

Cederman, Lars-Erik. 1997. *Emergent Actors in World Politics: How states & nations develop & dissolve*. Princeton Studies in Complexity Princeton University Press.

Cederman, Lars-Erik. 2002. "Endogenizing geopolitical boundaries with agent-based modeling." *Proceedings of the National Academy of Sciences* 99(3):7296–7303.

Cederman, Lars-Erik. 2003. "Modeling the Size of Wars: From Billiard Balls to Sandpiles." *American Political Science Review* 97(1):135–150.

Cederman, Lars-Erik. 2005. "Computational Models of Social Forms: Advancing Generative Process Theory." *American Journal of Sociology* 110(4):864–893.

Cederman, Lars-Erik and Kristian Skrede Gleditsch. 2004. "Conquest and Regime Change: An Evolutionary Model of the Spread of Democracy and Peace." *International Studies Quarterly* 48(3):603–629.

Collier, David and Richard E. Messick. 1975. "Prerequisites Versus Diffusion: Testing Alternative Explanations of Social Security Adoption." *American Political Science Review* 69:1299–1315.

Collier, Nicholson. 2002. "RePast: An Extensible Framework for Agent Simulation.". **URL:** *http://www2.econ.iastate.edu/tesfatsi/RepastTutorial.Collier.pdf*

Davis, Jason P., Kathleen Eisenhardt and Christopher B. Bingham. 2007. "Developing Theory through Simulation Methods." *Academy of Management Review* 32(2):480–499.

de Marchi, Scott. 2005. *Computational and Mathematical Modeling in the Social Sciences*. New York: Cambridge University Press.

Deffuant, Guillaume. 2006. "Comparing Extremism Propagation Patterns in Continuous Opinion Models." *Journal of Artificial Societies and Social Simulation* 9(3).

Deffuant, Guillaume, Sylvie Huet and Frédéric Amblard. 2005. "An Individual-Based Model of Innovation Diffusion Mixing Social Value and Individual Benefit." *American Journal of Sociology* 110(4):1041–1069.

Delannoy, Claude. 2007. *Programmer en Java*. 5è ed. Eyrolles.

Dobbin, Frank, Beht Simmons and Geoffrey Garrett. 2007. "The Global Diffusion of Public Policies: Social Construction, Corecion, Competition or Learning?" *Annual Review of Sociology* 33:449–472.

Elkink, Jos. 2006*a*. An agent-based model of democratic diffusion. "Diffusion of Policies and Institutions" workshop, ECPR Joint Sessions of Workshops Nicosia, Cyprus: .

Elkink, Jos. 2006*b*. "Spreading democracy: An agent-based model of the diffusion of democracy." *Midwest Political Science Association Conference* .

Elkink, Jos. 2009. An attitude diffusion model of the international clustering of political regimes PhD thesis School of Politics and International Relations University College Dublin.

Elkins, Zachary, Andrew T. Guzman and Beth Simmons. 2006. "Competing for Capital: The Diffusion of Bilateral Investment Treaties, 1960–2000." *International Organization* 60(4):811–846.

Elkins, Zachary and Beth Simmons. 2005. "On Waves, Clusters, and Diffusion: A Conceptual Framework." *The Annals of the American Academy of Political and Social Sciences* 598(1):33–51.

Epstein, Joshua M. 2005. Remarks on the Foundations of Agent-Based Generative Social Science. In *HandBook on Computational Economics*, ed. Kenneth L. Judd and Leigh Tesfatsion. Washington, D.C.: North Holland Press.

Epstein, Joshua M. 2006. *Generative Social Science: Studies in Agent-Based Computational Modeling*. Princeton University Press.

Epstein, Joshua M. and Robert Axtell. 1996. *Growing artificial societies*. Washington DC: Brookings Institution.

Esping-Andersen, Gøsta. 1990. *The Three Worlds of Welfare Capitalism*. Princeton: Princeton University Press.

Franzese, Robert J. and Jude C. Hays. 2008. "Interdependence in Comparative Politics: Substance, Theory, Empirics, Substance." *Comparative Political Studies* 41(4-5):742–780.

Füglister, Katharina. 2009. Where Does Learning Take Place? The Role of Intergovernmental Cooperation for Policy Diffusion in Federal States. Midwest Political Science Association Chicago, IL: .

Ganghof, Steffen. 2003. "Promises and Pitfalls of Veto Player Analysis." *Swiss Political Science Review* 9(2):1–25.

Gilardi, Fabrizio. 2005. "The Institutional Foundations of Regulatory Capitalism: The Diffusion of Independent Regulatory Agencies in Western Europe." *The Annals of the American Academy of Political and Social Sciences* 598(1):84–101.

Gilardi, Fabrizio. 2010. "Who learns from what in policy diffusion processes?" *American Journal of Political Science* 54(3):650–666.

Gilardi, Fabrizio. 2011. Transnational diffusion: Norms, ideas, and policies. In *Paper prepared for Handbook of International Relations*, ed. Walter Carlsnaes, Thomas Risse and Beth Simmons. SAGE Publications.

Gilardi, Fabrizio and Fabio Wasserfallen. 2009. The diffusion of tax policies: competition and networks. American Political Science Association Toronto: .

Gilardi, Fabrizio and Fabio Wasserfallen. 2010. How Cooperation Attenuates Tax Competition. Paper for presentation at the American Poltical Science Association (APSA) Washington, D.C.: .

Gilardi, Fabrizio and Katharina Füglister. 2008. "Empirical modeling of policy diffusion in federal states: the dyadic approach." *Swiss Political Science Review* 14:413–450.

Gilardi, Fabrizio, Katharina Füglister and Stéphane Luyet. 2009. "Learning from Others: The Diffusion of Hospital Financing Reforms in OECD Countries." *Comparative Political Studies* 42(4):549–573.

Gilbert, Nigel. 1998. The simulation of social processes. Clermont-Ferrand, France: SMAGET Conference.

Gilbert, Nigel. 2008. *Agent-Based Models.* Vol. 153 Thousand Oaks - New Delhi - London - Singapore: Sage Publications.

Gilbert, Nigel and Klaus G. Troitzsch. 2005. *Simulation for the Social Scientist.* Maidenhead: Open University Press.

Gilbert, Nigel and Pietro Terna. 2000. "How to build and use agent-based models in social science." *Mind and Society* 1(1):57–72.

Gleditsch, Kristian Skrede and Michael D. Ward. 2006. "Diffusion in the International Context of Democratization." *International Organization* 60(4):911–933.

Goldstone, Robert L. and Marco A. Janssen. 2005. "Computational models of collective behavior." *Trends in Cognitive Sciences* 9(9):424–430.

Granovetter, Mark. 1978. "Threshold Models in Collective Behavior." *American Journal of Sociology* 83(6):1420–1443.

Gray, Virginia. 1973. "Innovation in the States: A Diffusion Study." *The American Political Science Review* 67(4):1174–1185.

Greenhalgh, Trisha, Glenn Robert, Paul Bate, Fraser Macfarlane and Olivia Kyriakidou. 2005. *Diffusion of Innovations in Health Service Organisation: A systematic literature review.* Blackwell Publishing.

Grossback, Lawrence J., Sean Nicholson-Crotty and David A. M. Peterson. 2004. "Ideology and Learning in Policy Diffusion." *American Politics Research* 32(5):521–545.

Hacker, Jacob S. 2004. "Privatizing Risk without Privatizing the Welfare State: The Hidden Politics of Social Policy Retrenchment in the United States." *American Political Science Review* 98(2):243–260.

Hallerberg, Mark and Scott Basinger. 1998. "Internationalization and Changes in Tax Policy in OECD Countries: The Importance of Domestic Veto Players." *Comparative Political Studies* 31(3):321–352.

Hegre, Hårvard. 2005. Simulating the Evolution of Global Democracy Levels. American Political Science Association Chigago (IL): .

Henisz, Witold J., Bennet A. Zelner and Guillén Mauro F. 2005. "The Worldwide Diffusion of Market-Oriented Infrastructure Reform, 1977–1999." *American Sociological Review* 70(6):871–897.

Henisz, Witold Jerzy. 2004. "Political Institutions and Policy Volatility." *Economics and Politics* 16(1):1–27.

Hennessy, Alexandra. 2009. Policy Diffusion or Common Shock? The Sources of Pension Reforms in Europe. 67th Annual National Conference of the Midwest Political Science Association Chicago (Il): .

Holland, John. 1995. *Hidden order: how adaptation builds complexity.* Reading, MA: Addison Wesley.

Holland, John. 1998. *Emergence: From Chaos to Order*. Cambridge, Massachusetts: Perseus Books.

Holland, John H. and John H. Miller. 1991. "Artificial Adaptive Agents in Economic Theory." *The American Economic Review* 81:365–370.

Holzinger, Katharina and Christoph Knill. 2005. "Causes and conditions of cross-national policy convergence." *Journal of European Public Policy* 12(5):775–796.

Holzner, Steve. 2004. *Eclipse.* Editions O'Reilly.

Häusermann, Silja and Bruno Palier. 2008. "The politics of employment-friendly welfare reforms in post-industrial economies." *Socio-Economic Review* 6(3):559–586.

IVÁNYI, M.D., L. GULYÁS, R. BOCSI, G. SZEMES and R. MÉSZÁROS. 2007. MODEL EXPLORATION MODULE. In *Proceedings of the Agent 2007 Conference on Complex Interaction and Social Emergence*. Argonne National Laboratory and Northwestern University.

Karch, Andrew. 2007. "Emerging Issues and Future Directions in State Policy Diffusion Research." *State Politics and Policy Quarterly* 7(1):54–80.

Kastener, Scott L. and Chad Rector. 2003. "International Regimes, Domestic Veto-Players, and Capital Controls Policy Stability." *International Studies Quarterly* 47(1):1–22.

Kollman, Kenneth, John H. Miller and Scott E. Page. 1992. "Adaptive Parties in Spatial Elections." *American Political Science Review* 86(4):929–937.

Kollman, Kenneth, John H. Miller and Scott E. Page. 1997. "Political Institutions and Sorting in a Tiebout model." *American Economic Review* 87(5):977–992.

Kollman, Kenneth, John H. Miller and Scott E. Page. 1998. "Political Parties and Electoral Landscapes." *British Journal of Political Science* 28:139–158.

Laver, Michael. 2005. "Policy and the Dynamics of Political Competition." *American Political Science Review* 99:263–281.

Lee, Chang Kil and David Strang. 2006. "The International Diffusion of Public-Sector Downsizing: Network Emulation and Theory-Driven Learning." *International Organization* 60(4):883–909.

Levi-Faur, David. 2005. "The Global Diffusion of Regulatory Capitalism." *The Annals of the American Academy of Political and Social Sciences* 598(1):12–32.

Lomi, Alessandro and Erik K. Larsen. 1996. "Interacting Locally and Evolving Globally: A Computational Approach to the Dynamics of Organizational Populations." *Academy of Management Journal* 39(4):1287–1321.

Lustick, Ian S., Dan Miodownik and Roy J. Eidelson. 2004. "Secessionism in Multicultural States: Does Sharing Power Prevent or Encourage It?" *American political Science Review* 98(2):209–229.

Macy, Michael W. and Robert Willer. 2002. "FROM FACTORS TO ACTORS: Computational Sociology and Agent-Based Modeling." *Annual Review of Sociology* 28:143–166.

Martin, Christian W. 2009. Conditional diffusion: Smoke free air legislation and tabacco taxation policies in the United States 1970-2006. Annual National Conference of the Midwest Political Science Association Chicago (IL): .

McVoy, Edgar C. 1940. "Patterns of Diffusion in the United Sates." *American Sociological Review* 5(2):219–227.

Meseguer, Covadonga. 2003. Learning and Economic Policy Choices: a Bayesian approach. Eui working paper Robert Schuman Centre for Advanced Studies, European University Institute.

Meseguer, Covadonga. 2004. "What Role for Learning? The Diffusion of Privatisation in OECD and Latin American Countries." *Journal of Public Policy* 24(3):299–325.

Meseguer, Covadonga. 2005. "Policy Learning, Policy Diffusion, and the Making of a New Order." *The Annals of the American Academy of Political and Social Sciences* 598(1):67–82.

Meseguer, Covadonga. 2006*a*. "Learning and Economic Policy Choices." *European Journal of Political Economy* 22:156–178.

Meseguer, Covadonga. 2006*b*. "Rational Learning and Bounded Learning in the Diffusion of Policy Innovations." *Rationality and Society* 18(1):35–66.

Meyer, Bertrand. 2009. *Touch of Class: Learning to Program Well with Objects and Contracts*. Springer-Verlag.

Miller, John H. and Scott E. Page. 2007. *Complex Adaptive Systems: An Introduction to Computational Models of Social Life*. Princeton studies in complexity Princeton University Press.

Miodownik, Dan. 2006. "Cultural Differences and Economic Incentives: an Agent-Based Study of Their Impact on the Emergence of Regional Autonomy Movements." *Journal of Artificial Societies and Social Simulation* 9(4).

Niemeyer, Patrick and Knudsen Jonathan. 2005. *Learning Java, Third Edition*. O'Reilly Media Inc.

OECD. 2006. "OECD Health Data.".

O'Loughlin, J., M.D. Ward, C.L. Lofdahl, J.S. Cohen, D.S. Brown, D. Reilly, K.S. Gleditsch and M. Shin. 1998. "The diffusion of democracy, 1946- 1994." *Annals of the Association of American Geographers* 88(4):545–574.

Page, Scott E. 1999. "Computational Models from A to Z." *Complexity* 5:35–41.

Page, Scott E. 2008. "Uncertainty, Difficulty, and Complexity." *Journal of Theoretical Politics* 20:115–149.

Palier, Bruno. 2004. *La réforme des systèmes de santé*. Paris: Presses Universitaires de France.

Pierson, Paul. 2001. Post-Industrial Pressures on the Mature Welfare States. In *The New Politics of the Welfare State*, ed. Paul Pierson. Oxford: Oxford University Press pp. 80–104.

Polillo, Simone and Mauro F. Guillén. 2005. “Glaobalization Pressures and the State: The Worldwide Spread of Central Bank Interdependence.” *American Journal of Sociology* 110(6):1764–1802.

Radaelli, Claudio M. 2005. “Diffusion without convergence: how political context shapes the adoption of regulatory impact assessment.” *Journal of European Public Policy* 12(5):924–943.

Reeves, Hubert. 1986. *L'heure de s'enivrer: L'univers a-t-il un sens?* Points Sciences Editions du Seuil.

Repenning, Nelson P. 2002. “A Simulation-Based Approach to Understanding the Dynamics of Innovation Implementation.” *Organization Science* 13(2):109–127.

Roberts, Eric. 2006. *Thinking Recursively with Java.* John Wiley & Sons, Inc.

Rogers, Everett M. 2003. *Diffusion of Innovations (5th edition).* New-York: Free Press.

Rosenkopf, Lori and Eric Abrahamson. 1999. “Modeling Reputational and Informational Influences in Threshold Models of Bandwagon Innovation Diffusion.” *Computational and Mathematical Organization Theory* 5(4):361–384.

Rousseau, David and A. Maurits van der Veen. 2005. “The Emergence of Shared Identity.” *Journal of Conflict Resolution* 49(5):686–712.

Rudolph, Jenny P. and Nelson P. Repenning. 2002. “Disaster Dynamics: Understanding the Role of Quantity in Organizational Collapse.” *Administrative Science Quarterly* 47(1):1–30.

Sastry, M. Anjali. 1997. “Problems and Paradoxes in a Model of Ponctuated Organizational Change.” *Administrative Science Quarterly* 42(3):237–275.

Savitch, Walter. 2006. *Absolut Java (2nd edition).* San Francisco: Addison Wesley.

Schelling, Thomas C. 1978. *Micromotives and Macrobehavior.* New-York, London: W. W. Norton.

Shalizi, Cosma Rohilla. 2001. Causal Architecture, Complexity and Self-Organization in Time Series and Cellular Automata PhD thesis Center for Complex Social Studies, University of Michigan.

Sharman, Jason C. 2010. "Dysfunctional Policy Transfer in National Tax Blacklists." *Governance* 23(4):623–639.

Shipan, Charles R. and Craig Volden. 2006. "Bottom-Up Federalism: The Diffusion of Antismoking Policies from U.S. Cities to States." *American Journal of Political Science* 50(4):825–843.

Shipan, Charles R. and Craig Volden. 2008. "The Mechanisms of Policy Diffusion." *American Journal of Political Science* 52(4):840–857.

Simmons, Beht A., Frank Dobbin and Geoffrey Garrett. 2008. *The Global Diffusion of Markets and Democracy.* Cambridge: Cambridge University Press.

Simmons, Beth, Frank Dobbin and Geoffrey Garrett. 2006. "Introduction: The International Diffusion of Liberalism." *International Organization* 60(4):781–810.

Simmons, Beth and Zachary Elkins. 2004. "The Globalization of Liberalization: Policy Diffusion in the International Political Economy." *American Political Science Review* 98(1):171–189.

Squazzoni, Flaminio. 2008. "The Maicro-Macro Link in Social Simulation." *Sociologica* 1.

Strang, David and John W. Meyer. 1993. "Institutional conditions for diffusion." *Theory and Society* 22(4):487–511.

Strang, David and Nancy Tuma. 1993. "Spatial and Temporal Heterogeneity in Diffusion." *American Journal of Sociology* 99(3):614–639.

Strang, David and Sarah A. Soule. 1998. "Diffusion in organizations and social movements: From Hybrid Corn to Poison Pills." *Annual Review of Sociology* 24(??):265–290.

Swank, Duane. 2005. Policy Diffusion, Globalization, and Welfare State Retrenchment in 18 Capitalist Countries, 1976-2001. American Political Science Association Washington DC: .

Swank, Duane. 2006. "Tax Policy in an Era of Internationalization: Explaining the Spread of Neoliberalism." *International Organization* 60(4):847–882.

Swank, Duane. 2008. *Tax policy in an era of internalization: an assessment of a conditional diffusion model of the spread of neoliberalism.* Cambridge University Press chapter 2, pp. 64–103.

Swank, Duane and Sven Steinmo. 2002. "The New Political Economy of Taxation in Advanced Capitalist Democracies." *American Journal of Political Science* 46(3):642–655.

Taber, Charles S. and Richard J. Timpone. 1996. *Computational Modeling.* Vol. 113 Thousand Oaks - London - New Delhi: Sage Publications.

Thomsen, Jan, Raymond E. Levitt, John C. Kunz, Clifford I. Nass and Douglas B. Fridsma. 1999. "A Trajectory for Validating Computational Emulation Models of Organizations." *Computational and Mathematical Organization Theory* 5(4):3085–3401.

Tobias, Robert and Carole Hofmann. 2004. "Evaluation of free Java-libraries for social-scientific agent based simulation." *Journal of Artificial Socities and Social Simulation* 7(1).

Tsebelis, George. 2002. *Veto Players: How Political Institutions Work?* Princeton: Princeton University Press.

Verzani, John. 2005. *Using R for Introductory Statisitcs.*

Volden, Craig. 2006. "States as Policy Laboratories: Emulating Success in the Children's Health Insurance Program." *American Journal of Political Science* 50(2):294–312.

Volden, Craig. 2010. Failures: Diffusion, Learning, and Policy Abandonment. American Political Science Association Annual Meetings Washington, D.C.: .

Volden, Craig, Michael M. Ting and Daniel P. Carpenter. 2008. "A Formal Model of Learning and Policy Diffusion." *American Political Science Review* 102(3):319–332.

Walker, Jack L. 1969. "The Diffusion of Innovations among the American States." *The American Political Science Review* 63(3):319–363.

Way, Christopher R. 2005. "Political Insecurity and the Diffusion of Financial Market Regulation." *The Annals of the American Academy of Political and Social Sciences* 598(1):125–144.

Wejnert, Barbara. 2005. "Diffusion, Development, and Democracy, 1800-1999." *American Sociological Review* 70(1):53–81.

Weyland, Kurt. 2002*a*. The Diffusion of Innovations: A Theoretical Analysis. American Political Science Association Boston (MA): .

Weyland, Kurt. 2002*b*. Learning From Foreign Models in Latin American Policy Reform. American Political Science Association Boston (MA), USA: .

Weyland, Kurt. 2004. *Learning From Foreign Models in Latin American Policy Reform*. Washington D.C.: Woodrow Wilson Center Press.

Williamson, Jeffrey G. 1996. "Globalization, Convergence, and History." *The Journal of Economic History* 56(2):277–306.

Zwirn, Hervé P. 2006. *Les Systèmes Complexes: Mathématiques et biologie*. Paris: Odile Jacob.