Urban Growth in Latin American Cities

Exploring urban dynamics through agent-based simulation

Joana Xavier Barros

Bartlett School of Graduate Studies

Bartlett School of Architecture and Planning

University College London

University of London

A thesis submitted for the degree of Doctor of Philosophy

October 2004

Abstract

The high rates of urban growth in Latin America during the 1960s and 1970s produced rapid urbanisation and housing problems. Planning policies as well as the research community have approached urban growth as a static problem rather than as a spatial form that emerges from the urban development process and that is part of a constant dynamic process. This thesis focuses on a specific kind of urban growth that happens in Latin American cities, called 'peripherisation'. This is characterised by the formation of low-income residential areas in the peripheral ring of the city and a perpetuation of a dynamic core-periphery spatial pattern. The dynamics of growth and change in Latin American cities are explored using agent-based simulation. The objective is to increase the understanding of urban spatial phenomena in Latin American cities, which is essential to providing a basis for future planning actions and policies.

The thesis consists of two parts. The first part presents an overview of urban growth and dynamics in Latin American cities, drawing on previous work on urbanisation in Latin American cities, spontaneous settlements and inner city dynamic processes. The second part focuses on the development of a simulation model based on the theoretical framework established in the first part. A brief review of the literature of automata models is presented, with particular reference to agent-based simulation for land-use dynamics. The Peripherisation Model is introduced, its computer implementation described, and sensitivity analysis tests reported. Simulation exercises were used to revisit assumptions about urbanisation issues in Latin American cities and investigate important aspects of growth and change in these cities. These exercises allowed the problem of urban growth in Latin American cities to be unfolded through their dynamics, relating these dynamics to urban morphology, and thus presenting a new and important perspective on the phenomenon.

Acknowledgments

Firstly, my thanks go to my supervisors Philip Steadman and Mike Batty, for their supervision and support along the development of this research. I would also like to thank Romulo Krafta, for motivating me to start this PhD, and to my former supervisor and friend, Maurício Polidori, for having introduced me to urban studies and motivated me to follow this path.

I cannot fully express my gratitude to Dr. Kim Christensen who has provided me with exceptional help and support throughout the development of this thesis.

Most of my learning process during the last five years was motivated by the research environment provided by CASA. My thanks to everyone I have come across during these years. The talks, discussions, observations, and information I gathered during this time are some of the richest experiences I am taking with me. Thank you especially to Fabiano Sobreira, who greatly contributed to the first phase of this work, and to Nancy Chin and Sinésio Alves Jr., who have helped with the maps presented in the thesis. My thanks also go to Sarah Sheppard and Sonja Curtis, whose efficiency and professionalism helped in making my life at CASA much easier.

My special thanks to the 'corner' people and other members of the 'Pizza Club': Nancy, Junior, Kay, Lily, Elena, Meo and Pooh. Life at CASA is infinitely better with you. To two very special colleagues, Katerina and Theodoris, for whom I have no words to express my love and gratitude for being the best people and friends I could have met.

I would also like to thank my friends. The ones I have met here, for their company and support: Mari, Nikos, Andrea, Andrew, Victor, Lucas, Deo, Rosane, and Giuliano. And to my long-term friends: Ju, Gil, Tom, Marcia, Ana, Bernardo, Paulina, and Nelso, for having always been so close despite the geographic distance between us. My special thanks to Viviane and Silvana, whose help with the programming was essential for the completion of this thesis.

My gratitude also goes to my sponsors: CAPES – Brazilian Research Agency, NEXSUS Project, and CASA. My thanks as well to PRODABEL, who provided me with data and images for the city of Belo Horizonte, Brazil.

And finally, I would like to thank my ever-loving family: my mother Ana Luisa, my father Fernando, and my sisters Flavia and Laura. None of this would be possible if were not for your support and care. I dedicate this thesis to you.

Contents

List of Figures

List of Tables

Chapter 1
Introduction

Rapid urbanisation has been the main theme of urban studies in Latin America since the explosion of rates of growth in the 1960s and 1970s. While studies predicted an unprecedented rate of growth in these cities by the year of 2000, the speed of development has been blamed as the cause of spatial inequalities and problems in these cities. Yet, the actual rates of growth have slowed since the 1980s and studies suggest that the tendency is that the rates will remain as they are. The urban problems, however, have not disappeared in the last two decades, and, despite lower rates of population growth, cities keep growing and developing in the very same way. Hence, the principal problem of urban growth in Latin American cities is no longer the high rates of population growth and rural-urban migration. Rather, it is the spatial pattern of growth and its underlying dynamics of change, the peripherisation process, which enlarges the peripheral rings of cities and metropolises despite the reduction in the overall urban growth rates.

There have been a large number of studies of urbanisation issues in these countries, mainly focusing on the rapidity of growth of cities and the social inequalities in urban space produced by this process. Most of these studies have taken a sociological and political approach, often discussing either the role of the poor and spontaneous settlements, or the State in the context of economic and urban development. Hence, while studies of demographic trends, housing, urban poor and urbanisation proliferated during the 1970s and 1980s, very few studies have been devoted to the morphology and dynamics of Latin American cities to date. Planning policies as well as the research community have approached urban growth as a static problem rather than as a spatial form that emerges from the urban development process and that is part of a constant dynamic process.

The present study looks at issues related to the growth of Latin American cities by investigating the dynamics of this mode of urban growth and change. The objective is to increase the understanding of urban spatial phenomena in Latin American cities, which is essential to providing a basis for future planning actions and policies.

The study of urban dynamics requires tools that allow the exploration of the change phenomenon in time and space. Urban modelling techniques have been traditionally used to explore issues in urban dynamics, and automata models like cellular automata and agent-based models seem to be a particularly suitable approach for this kind of study. Therefore, an agent-based model was used in order to unfold the problem of urban growth of Latin American cities through their dynamics. Agent-based models are based on the understanding that human decision-making plays a major role in urban processes and urban change. Their framework allows interactions between agents and their landscape to be explicitly represented. Hence, this kind of model permits the analysis of dynamic processes that link spatial development with social issues, which is of fundamental importance when dealing with cases of strong social differentiation, as is the case of urban dynamics in Latin American cities.

Thus, simulation exercises were used to revisit assumptions on urbanisation issues in Latin American cities and investigate important aspects regarding growth and change in these cities. These exercises allowed the problem of urban growth in Latin American cities to be revealed through their dynamics, relating these dynamics to urban morphology, and thus presenting a new and important perspective on the phenomenon.

1.1 Thesis outline

The thesis consists of two parts. Part I reviews literature on the urban growth of Latin American cities and discusses the main issues concerning urban dynamics in Latin American cities. This part is intended to serve as a theoretical framework for the simulation model developed in Part II. Part II describes the simulation model developed from the framework presented in Part I.

1.1.1 Part I

Chapter 2 is an introduction to Latin American cities and their context of urban growth. It reviews the main approaches to the subject and discusses the spatial pattern produced by the rapid urbanization process. The intention of this chapter is to establish the research context in which Latin American cities have been studied, and introduce the main topic of the thesis.

Chapter 3 discusses the dynamics of the peripherisation phenomenon, which is the general process of growth of Latin American cities. This chapter also includes a review of spontaneous settlements and their emergence and evolution within the urban growth process, and a brief discussion of the morphology of the urban fabric as a result of these processes.

Chapter 4 is an overview of inner city change processes in Latin American cities. It reviews the literature on the main processes (filtering, gentrification, movement of elites to the outskirts, and inner city decay), discussing the relevance of each process in the Latin American context.

1.1.2 Part II

Chapter 5 is an overview of urban modelling and simulation within the complexity approach, followed by a review of the literature of automata models, with particular reference to agent-based simulation for land-use and land-cover dynamics.

Building on the material in the previous chapter, and based on the theoretical framework presented in Part I, Chapter 6 describes the proposed agent-based model for urban growth in Latin American cities. This chapter details the logic and implementation of the Peripherisation Model, and presents RePast, which is the agent-based programming environment used to build the model.

Chapter 7 reports the evaluation of the model through sensitivity analysis tests. This chapter is intended to demonstrate the typical behaviour of the model and establish the relationship between the inputs and outputs of the model, allowing a deep knowledge of the inherent characteristics of the simulation model.

Finally, Chapter 8 reports a number of exercises that explore issues in urban dynamics in Latin American cities through the simulation model. The intention of this chapter is to revisit some of the assumptions about urban dynamics in Latin American cities, examined in Part I, and discuss them in the light of the outcomes of the model. Therefore, each exercise explores a feature of the model related to the issues discussed in Part I – respectively, the dynamics of the peripherisation phenomenon, spontaneous settlements, inner city dynamics, and impacts of spatial constraints.

The thesis concludes with a discussion of the contributions of the thesis in Chapter 9. Limitations of the study and suggestions as to the future development of this research are also covered in this discussion.

Part I

Latin American Cities

Chapter 2

The Latin American City

The urbanisation process in cities in developing countries is often insufficiently planned and poorly coordinated, and Third World cities are known for their inherent chaotic and discontinuous spatial patterns, and their rapid and unorganised development process. The fast urbanization process in Latin America attracted the attention of the research community, resulting in a large number of studies between the mid-sixties and the mid-eighties, most of which focussed on urban growth and urbanization.

Although urban growth and urbanization have been frequently studied as a general issue, the focus of these studies differs according to the geographical location of the object of study. Making a rough generalisation, one could say that existing approaches and theories could be divided between First and Third Worlds, or North and South of the planet. It is necessary to note, however, that urban research on 'South' and 'North' have shown different levels of theoretical authority, with a clear predominance of research on urban problems of the north, especially those of European and Anglo-American cities (Klak & Lawson, 1993a).

This is a large discussion that would not be worth mentioning here but for its influence on building of a theory of Latin American cities (see Klak & Lawson, 1993a; 1993b). This theory, emerging from a Northern context, especially that of Anglo-American and European cities, is frequently applied to Third World cities. In this discussion two main groups can be distinguished. The first argue that cities of the Third World present a similar pattern to the pre-industrial city, and correspond, therefore, to an earlier stage in the development of cities in Western developed countries. They believe that the development of Third World cities will in time follow the pattern of American and European cities (see Amato, 1970a; b; London & Flanagan, 1976; Schnore, 1965; Sjoberg, 1980). The second group seem to disagree with this point of view and try to prove that the internal development of Latin American cities, for instance, is not following such a path. They argue that Third World cities are "fundamentally different, qualitatively and quantitatively, from the typical industrialised, capitalist city of the west with which they are so frequently compared" (Payne, 1977, page 8). They also stress the need for a specific model (or theory) of Latin American cities, which would be useful not only to describe these cities but also to explain the processes that shape some of the biggest cities of the world.

Yet the supremacy of Northern urban theory is still evident, and most theories of Latin American cities draw upon existing theories of Northern cities, especially Anglo-American ones.

On the basis of this discussion, the objectives of this chapter are twofold. First, it will introduce and discuss the problem of urban growth and urbanization and examine the spatial pattern of Latin American cities. Second, it will explore these issues within a broader context of urban research, establishing comparisons with research on Western cities, and demonstrating the origin of the discussions within Latin American urban research.

Hence, the first part of this chapter introduces the problem of urban growth and discusses some of the main issues and approaches developed for Latin American cities. The second part of the chapter presents some studies of the Latin American city, more specifically of its internal structure and locational patterns.

2.1 The problem of urban growth

The problem of urban growth has always been a focus of attention for the scientific planning community. The ideal size of the city, the limits of growth, these have always been themes for debate and, despite cities having changed over time, and the variety of problems and approaches, urban growth is still a very contemporary issue.

Contemporary urban growth (in the North) has been mainly seen as a set of three interrelated problems in spatial dynamics: the decline of central or core cities, the emergence of edge cities, and the rapid suburbanization of the periphery of cities (Batty, Xie & Sun, 1999). Among these processes, peripheral growth has been receiving special attention thanks to the investigation of the urban sprawl[1] phenomenon, which has been occurring in both European and Anglo-American cities.

While the problem of urban growth in Europe and North America has been formulated in terms of sprawl, in the Third World[2] and more specifically in Latin America, the main focus has been the rapidity of growth of cities as well as the social inequalities in urban space produced by this process.

During the period between 1950 and 1980, growth rates were very high (see (Hall, 1983; Valladares & Coelho, 1995) and, based on these data, studies anticipated continuing high rates of growth. It was believed that many areas would double in population and a few would triple, creating urban areas that by the year 2000 would be without parallel in history (Hall, 1983).

---

[1] Urban sprawl can be defined as a kind of suburbanization which forms a suburban development that lacks accessibility and open space (Ewing, 1997).

[2] The term "Third World" is understood as the less developed countries of Asia, Africa and Latin America which share, by definition, one characteristic: they are poor compared to most of the rest of the world (Gugler, 1996).

Indeed, Latin America has seen the fastest urban growth in history. Both rates of natural growth and rural-urban migration have been very high. Latin American countries went from being predominantly rural to predominantly urban within a couple of decades, with high concentrations of urban population in cities with more than one million inhabitants (UNCHS, 1996). The urbanisation levels in Brazil, for example, went from 36% in 1950, reaching an even balance between rural and urban populations in the 1970s, going up to 82%, which is its current urbanisation level. This rapid urbanisation produced various kinds of social problems, especially in terms of housing, since employment opportunities did not accompany the rates of growth and the governments of such countries did not manage to provide enough housing and infra-structure to accommodate all the migrants who fuelled the growth of these cities. Cities with millions of inhabitants (mega cities) were produced in few decades, like Mexico City and São Paulo, which are amongst the largest cities in the world.

Within this context, housing was clearly the most important focus of studies of Latin American urbanisation. Other consolidated themes of research in Latin America included the study of urbanisation, urban growth and migration, the international structure of towns and metropolises, economic activities and the labour market, urban planning, urban poverty, urban imagery and the way of life, and urban social movements (Valladares & Coelho, 1995). Most of these studies present a strong sociological and economic approach, often discussing either the role of the poor or the State in the context of economic and urban development, or the question of dependency and underdevelopment in a capitalist context (Valladares & Coelho, 1995). A large number of studies were developed of growth and urbanisation in Latin American cities thanks to the speed and high rates of such growth, focusing on the causes and consequences of these demographic trends. Whilst studies of demographic trends, housing, urban poor and urbanisation proliferated between the 1960s and 1980s, the research community neglected a number of other important issues, such as small and intermediate size cities, commuting patterns, and morphology and internal dynamics of Latin American cities.

However, the fast trend in population growth has changed since 1980. After decades of explosive urbanisation, urban growth rates have slowed, the rate of metropolitan growth has fallen and fertility rates have declined (Valladares & Coelho, 1995). Moreover, rural to urban migration has come to assume a much smaller role in urban population growth and, most recently, the pace of urban expansion has been largely maintained by births in the cities. These new trends have been detected in the period between 1980 to 1990, and have been confirmed by recent censuses.

It is important to stress that urbanisation in Latin America as a whole has not been a homogeneous process, and to recognise the differences that exist between Latin American countries. Virtually all the cities in Latin America with a million or more inhabitants had much slower population growth rates during the 1980s than the average for the period 1950 to 1990. However, the major cities in the southern cone accounted for the slowest rates of population increase during these four decades (UNCHS, 1996). The southern cone countries of Argentina, Chile and Uruguay reached high levels of urbanisation in the 1950s and this process has slowed down in the last two decades (UNCHS, 1995).

Actually, it is possible to identify three groups of nations according to their demographic trends. The first, the most urbanised with more than 80 per cent of their population in urban areas includes the three nations in the Southern cone and Venezuela. The second, with between 50 and 80 per cent in urban areas includes most of the countries that had rapid urban and industrial development during the period 1950-90 and includes Mexico, Brazil, Ecuador, Colombia, Cuba, Bolivia and Peru. The third, with less than 50 per cent of the population in urban areas includes only one country in South America (Paraguay) along with a group of countries in Central America (Costa Rica, El Salvador, Guatemala and Honduras) (UNCHS, 1996).

The changed demographic trends have had impacts on urbanisation patterns. Such patterns can be defined mainly by a process of deconcentration of the population including a fall in the overall rate of population growth, a reduced concentration of population in the core of metropolitan areas coupled with significant growth of small and medium-sized municipalities, and a declining rate of demographic growth in regional capitals and major urban centres.

At first sight the evidence of the slow-down seems to indicate the end of problems caused by rapid urbanisation. However, if one examines the absolute numbers, the urban challenge continues to be enormous. Although growth rates have decreased as a whole, rates of peripheral growth remain high, and secondary cities are growing relatively fast in comparison to metropolitan areas. Furthermore, evidence suggests that the pattern of peripheral growth remains the same and is accompanied by problems related to this kind of growth.

When considering intra-metropolitan population distribution, UNCHS (1996) presents examples of both central cities growing more slowly than suburban rings (or even losing population), of outer suburbs and 'commuting towns' growing more rapidly than inner suburbs and central cities, and in cities in the commuting range of the largest centres sustains population growth rates higher than the metropolitan areas themselves, leading to a process termed 'polarisation reversal'.

Despite the slow-down of the overall rates of urban growth, since the 1970s and throughout the 1980s, the outer rings of nine metropolitan regions in Brazil, for example, have been expanding twice as fast – and in some cases three times as fast – as the metropolitan core (Valladares & Coelho, 1995).

The problem lies in the fact that the fringe of the city, which is mainly composed of low-income settlements, has been growing more than the city as a whole, becoming a major issue in Latin American cities (Dwyer, 1975; Secco & Squeff, 2000; UNCHS, 1982; 1996).

The phenomenon of peripheral growth, which has been recognised by Latin American researchers and planners and termed 'peripherisation'[3], can now be considered as an established process of growth of most Latin American cities. Peripherisation can be defined as a kind of growth process characterised by the expansion of borders of the city thorough the massive formation of peripheral settlements, which are, in most cases, large spontaneous low-income residential areas.

The high rates of urban growth during the 60s and 70s produced a rapid urbanisation, and housing problems on a large scale. The rates have slowed down but the effects of this urbanisation seem to have been to consolidate the process in which these cities are produced as their normal mode of urban growth, and the spatial result has become established as the Latin-American urban pattern.

In the next section this spatial pattern will be examined, and the main theories of the internal structure of Latin American cities explored through the comparison of these theories to others derived from patterns observed in Anglo-American and European cities.

2.2 The spatial pattern of Latin American cities

Latin American cities have clearly segregated areas of land use, as described by Gilbert (1987):

> "There are industrial zones which accommodate modern factories, well developed commercial and rental centres, high-income residential areas, zones of government and private offices, and large swathes of low-income residential development. Some parts of the cities are well ordered and regulated, others lack services and appear to have developed spontaneously." (Gilbert, 1987, page 181)

---

3 'Periferização', in Portuguese.

The segregated pattern probably originated in the colonial city. At that time, the morphology of the Latin American city was created by a traditional grid plan that prevailed so long as the characteristics of urbanisation were slow growth, minimal industrialisation, limited provision of public services, and restricted mobility (Griffin & Ford, 1980). The morphology and organization of the colonial city is described by Gilbert (1987):

> "Most Latin American cities have a central core which developed during the colonial period sometimes on the site of an existing settlement, as in Mexico City and Cuzco, but most frequently in a new location. The centre piece of the Spanish American city was the colonial *plaza mayor*, a large square flanked by the cathedral or church, government offices, and other public buildings. […]. Portuguese colonial settlements usually lacked a central plaza, were often less regular in design, and were more likely to follow the dictates of the terrain, even if the differences with the Spanish settlements have often been exaggerated." (Gilbert, 1987, page 182)

The colonial city spatial pattern was already segregated. While the high-income groups (elite) were located close to the square, which was the governmental, commercial and ecclesiastic centre of the city, the lower income areas were further out at the edge of the city (Gilbert, 1987; Griffin & Ford, 1980).

The spatial pattern of the Latin American city evolved from the colonial pattern to a core-periphery one. This change occurred in the late nineteenth century when some structural changes began in most Latin American cities. According to Amato (1970b), the breakdown of the colonial land use model was caused by the movement of the upper class who, during the twentieth century, have gradually migrated away from the central city to occupy suburban housing. This movement will be further examined in Chapter 4, together with other important processes of inner city change.

Caldeira (2000), in her study of the evolution of segregated patterns for São Paulo, suggests that the present pattern of segregation in that city is no longer the core-periphery. According to her, throughout the last century Sao Paulo's urban space has expressed the city's social segregation in at least three different forms (Caldeira, 2000). The first phase, dated from the late nineteenth century to the 1940s, produced a condensed city in which different social groups were packed into a small urban area and were segregated by type of housing. The second phase, from the 1940s to the 1980s, was characterized by the center-periphery pattern. During this period, the city had different social groups separated by great distances. While the middle and upper classes were located in central and well-equipped neighbourhoods, lower income groups were living on the city's outskirts. This is the best known pattern of Latin American cities, and it is still widely considered as their predominant urban form. Despite this fact, Caldeira (2000) suggests that a third form has been taking shape since the 1980s and argues that it has already exerted considerable influence on São Paulo and its metropolitan region. According to her, the recent transformations are superimposed on the center-periphery pattern and are justified by the fear of violent crime. In this pattern, different social groups are again closer to one another but are separated by walls and technologies of security. Moreover, the groups tend not to circulate or interact in common areas. The main instruments for this new pattern of spatial segregation are what Caldeira (2000) calls 'fortified enclaves', which are privatized, enclosed, and monitored spaces for residence, consumption, leisure, and work. These spaces "appeal to those who are abandoning the traditional public sphere of the streets to the poor, the marginalized, and the homeless" (Caldeira, 2000, page 213).

The last phase of transformation observed by Caldeira (2000) for São Paulo is not a general rule for all Latin American cities, since São Paulo is one of the largest and most developed cities in the continent and, therefore, its evolution differs from smaller and less developed cities. However, the way São Paulo develops in space shows the path of development these other cities might follow.

As mentioned earlier, there is very little research on the spatial structures of Latin American cities. In the following section some of the few relevant conceptual and descriptive models are presented.

2.2.1 Models of the internal structure of Latin American cities

Models can be defined in a number of different ways. They can be seen as examples, schemata, or theories. Models have been used in a number of fields as tools to describe different systems. Specifically in urban studies, schematic models of urban internal structure are attempts to summarise the morphological and social character of cities in the form of diagrams. The most important schematic models of the internal structure of cities were developed during the 1920s at the Chicago School of Urban Sociology. This line of research is known as Urban Ecology, since it described the city by drawing analogies between the urban and natural worlds. These models were based on Anglo American cities and had a profound influence on the understanding of the structure and development processes of cities in general. Although models of Latin American cities were clearly based on ideas developed by urban ecologists, they had neither the impact on urban theory nor the power of generalisation of the models of Anglo American cities. Most 'models' of Latin American cities can be considered as case studies, in that they describe the internal structure of single cities without attempting any generalisation. The model published by Griffin and Ford in 1980 is an exception of this rule and, hence, is the most important model of Latin American cities. In order to give the background to Griffin and Ford's model, a brief introduction will be given to two models in the field of Urban Ecology.

The first of these and perhaps the most important is the concentric rings model developed by Burgess (1925). The model was concerned with the physical growth and expansion of cities and the related processes of 'urban metabolism' and mobility. Burgess suggested that the typical process of expansion of the city could be best illustrated by a series of concentric circles. He proposed that the city be divided in five rings, organised into zones differing in age and character and located in a definite order from the city centre (see Figure 2.1). The five zones were (in order from the centre): an *inner central zone* or CBD; a *transition zone* of mixed land uses in which deteriorating residential properties predominate; a *working class residential zone inhabited by people who have escaped from the area of deterioration*; a *zone of better housing* characterised by single-family dwellings, and exclusive residences and high class apartment buildings; and a *fringe zone* of suburban and satellite communities forming dormitory suburbs for people working in the central city (Burgess, 1925; Garner, 1967). This analysis was largely based on a casual observation of Chicago.

For Burgess the expansion of cities happened through a tendency for each inner zone to extend its area by the invasion of the next outer zone, in a process he called 'succession-invasion'. It is clear that Burgess conceived his idea as a 'growth model', that is, taking in consideration a dynamic process, rather than as a static or cross – sectional representation of urban structure.

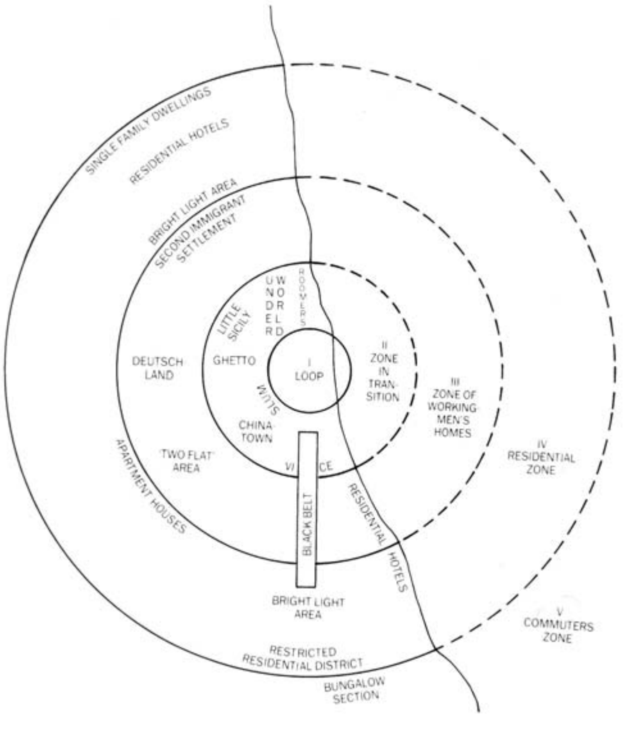


Figure 2.1 - Burgess's concentric rings model. Source: Burgess (1925, page 55)

The second model worth mentioning here is the Sector model developed by Hoyt (Hoyt, 1939). This model assumes that the internal structure of the city is conditioned by the disposition of routes radiating outwards from the city centre. Therefore, it is developed upon the idea that similar land uses concentrate along particular routes from the city centre forming 'sectors', as can be observed in Figure 2.2.

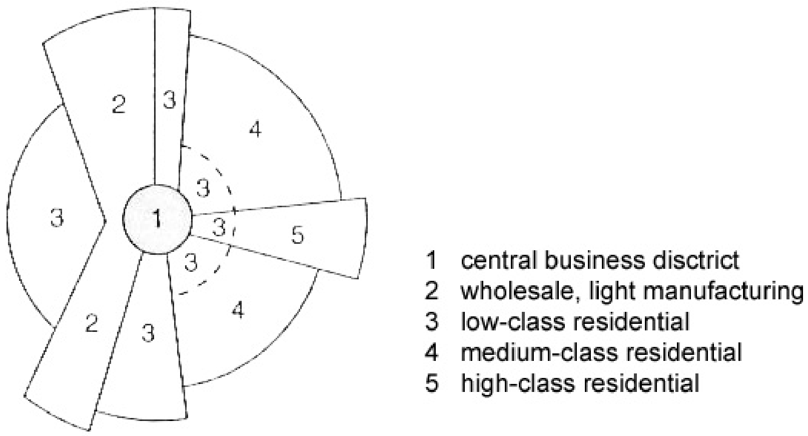


Figure 2.2 - Hoyt's sector model. Source: Pacione (2001, page 101).

These models were widely criticised but are still considered 'classical models' of urban growth. However, the similarities of these with the model developed by Griffin and Ford (1980) for internal spatial structure of Latin American cities is what makes them relevant to the present investigation. Their Latin American model combines elements from the classical models but modifies these to illustrate similarities as well as differences between cities in the two Americas. The main objective of the authors was to create a model simple enough to offer insights into the dynamics of Latin American urbanisation.

Griffin and Ford's model is characterised by one dominant elite residential sector and a commercial spine as well as series of concentric rings in which residential quality decreases with distance from the city centre. The relative size of each zone is a function of the age of the city and of the rate of population growth in relation to the economic capacity of the city to effectively absorb additional residents and to extend public services (Griffin & Ford, 1980).

As in Burgess's model, the first element of Griffin and Ford's model is the CBD. The city core of Latin American cities remains an important employment, commercial and entertainment node for the city, in contrast to Anglo-American cities. In most large Latin American cities the CBD presents a zone of transition, similar to the one proposed by Burgess, but less important and less visible than Anglo-American urban centres.

The second dominant element of Latin American urban structure in the model is a commercial spine surrounded by an elite residential sector, referred to as the 'spine sector'. According to Griffin and Ford, the spine sector is present in all Latin American cities, and is essentially an extension of the CBD, containing all the most important urban amenities (including boulevards, golf courses, major parks, museums, zoos, theatres, and restaurants) as well as all the professionally built upper-class and upper-middle class housing stock (Griffin & Ford, 1980). In relation to the classical models of urban growth of Anglo American cities, the spine is a Hoytian sector.

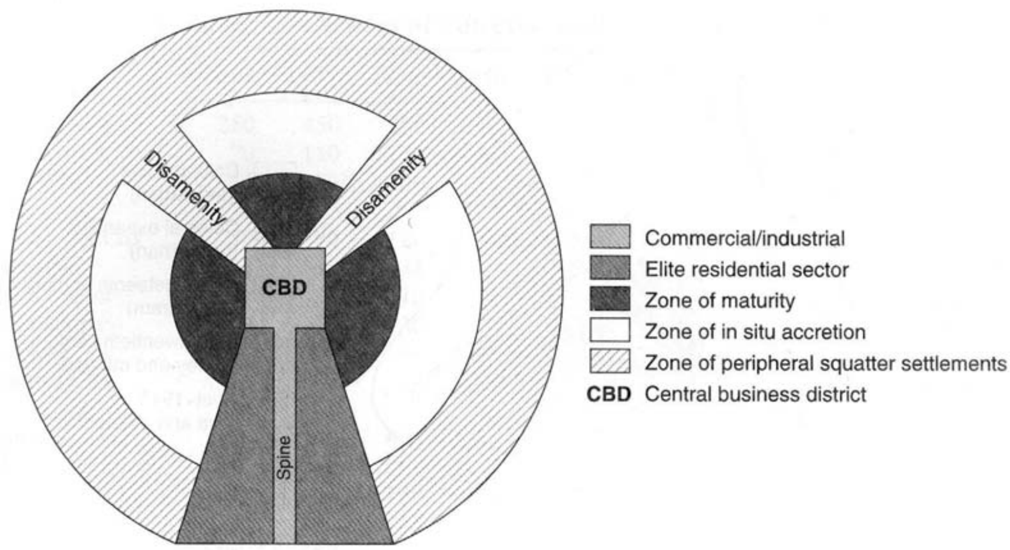


Figure 2.3 - Griffin and Ford's model of Latin American cities. Source: Griffin and Ford (1980, page 406).

Griffin and Ford suggest that the process of formation of the spine sector differs substantially from the processes shaping the rest of the city. The market for residential and commercial space is largely the same as that operating in Anglo-

American cities. This means that similar processes of filtering-down of housing, for example, happen within this area between high and middle-to-high income groups.

Although the spine sector is an important and, in some cases, dominant morphological feature of the Latin American city, it contains only a small percentage of the total metropolitan population. In spite of this fact, since it is the residential location of the elite, it is a key element of the internal structure of Latin American cities.

According to the model, outside the spine sector the structure of a Latin American city consists of a series of three concentric zones with socio-economic characteristics roughly the opposite of those postulated for Anglo-American cities in the Burgess model, where the rich live near the centre and the poor live at the periphery.

Three distinctive residential zones or rings typify the Latin American city: a zone of maturity, a zone of *in situ* accretion, and a zone of peripheral squatter settlements. All three zones are characterised by mixed land uses. Griffin and Ford relate the size of each zone to the speed of population growth in each country of Latin America. They suggest that the relative size of the three zones is a function of the rate of in-migration compared with the pace of individual on-site improvements in housing, and the ability of a city to expand urban services. Thus, the larger the maturity zone, for example, the slower is the growth in population. On the other hand, cities with rapid population growth would present larger zones of peripheral settlement. The authors stress that although these zones are generally present, sectors with high disamenity (flood-prone river channels, steep slopes, or canyons, for example) often remain unimproved even when they are centrally located. In what follows each of these three zones will be described further.

As an inner city location carries a positive connotation in the Latin American context, the *zone of maturity* is the area of 'better residences' in the city outside the spine sector. The zone is typically an area of gradually improved, significantly upgraded, self-built housing, although it may also include large numbers of traditional, filtered-down houses. The zone is fully serviced with paved streets, lighting, public transportation, schools, and sewerage.

The second ring is the *zone of in situ accretion* and represents the area where the process of assimilation between the inner and outer zones is occurring most dramatically. Thus, the zone of *in situ* accretion in most Latin American cities has a modest residential quality but shows signs of transition to a zone of maturity. It is characterised by wide variations in type, size, and quality of housing in which some of the housing is 'completed' and similar to that found in the zone of maturity, although for example a hovel may be located on a block with good housing. In terms of services, often only the main thoroughfares are paved, but there are small shops and schools. Improvement, nonetheless, is evidenced by the expansion of public services and by the overall betterment of housing quality and landscaping. The landscape reflects, to a certain extent, the uneven assimilation of residents into the economic and social structure of the city. The rate of improvement in the accretion zone is based on the ability of a city to provide services and continued economic mobility to the residents. An interesting feature of the zone of *in situ* accretion is that it is likely that large government-sponsored building projects are located there.

The *zone of peripheral squatter settlements*, as the name suggests, accommodates the impoverished recent in-migrants to the city and is the worst section of the city in terms of housing quality and public services. Besides the fact that quality of life is marginal and the physical conditions of housing, services and landscape are bad, the neighbourhood will eventually be transformed and improved. Griffin and Ford suggest that parts of the zone of maturity and, more recently, the zone of *in situ* accretion once possessed the characteristics now found in the peripheral zone and, thus, this is part of the process of growth of Latin American cities.

As the Latin American city has changed over time, the model of Griffin and Ford has been reviewed to incorporate these changes. Ford (1996) proposed a new and improved model in 1996. In the new model, Ford added some new elements without changing the main features. The revised model keeps the central business district, the spine sector, the three concentric rings and sector of disamenity, but incorporates some changes as shown in Figure 2.4.

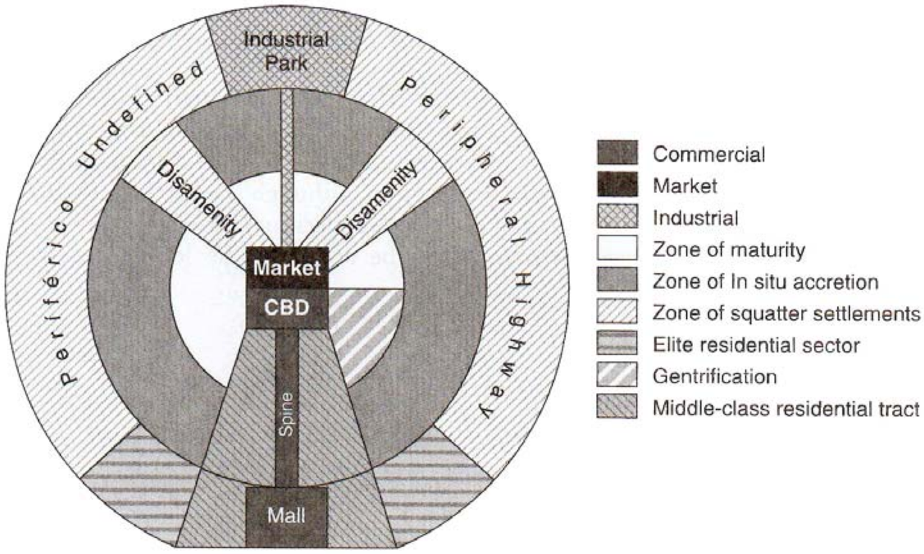


Figure 2.4 - Ford's model of Latin American cities. Source: Ford (1996, page 438).

The first change is related to the centre of the city, which is divided into two parts: CBD and market. According to Ford, this reflects the development of a new modern segment in the CBD, clearly separated from the traditional market and shopping areas.

The new model incorporates a shopping mall or 'edge city' sector, located near the spine sector and elite residential areas. According to the model, many Latin American cities now have major suburban nodes that compete with the downtown, although not yet as ubiquitous and numerous as in North America. A separate industrial sector is also incorporated into the model in a suburban location. Both shopping mall and industrial park are connected by ring roads that pass through the periphery of the urban area. These are labelled *periférico* in the model.

Another change in the model is the introduction of an area of middle-income housing tracts, that is, middle-class settlements which are located as close as possible to the elite sector and periférico, 'in order to ensure access, status, and protection'. Finally, the model includes a section of gentrification in the zone of maturity near the CBD and the elite sector, which is an area of old and historical neighbourhoods undergoing refurbishment.

It seems that this second version of the model includes issues that are still very controversial, and it appears that the changes serve to diminish the model's power of generalisation. As can be observed, the revised model incorporates some changes suggesting that the Latin American city is converging towards the Anglo-American model. Some of the issues, like gentrification for example, are going to be discussed in Chapter 4. Other issues, as for instance the location of shopping malls still deserve further investigation. Although the first generation shopping malls have indeed been located on the fringe of Latin American cities, as the model suggests, recent shopping malls built in the last 10 years in some of the biggest cities of Brazil[4], for instance, are located in very central areas. This new trend certainly needs verification and deserves further investigation, but it may suggest that the Latin American city is, again, presenting a divergent development from the Anglo-American model.

2.3 Summary

This chapter has presented an overview of urban growth in Latin American cities and reviewed the main approaches to the subject. It has been shown that the demographic trends that caused rapid urbanization in Latin America have changed since 1980, but the dynamic urban growth pattern developed during the rapid urbanization period seems to have become consolidated as the normal mode of growth of these cities. Therefore, there are still important issues to be investigated,

[4] Rio de Janeiro, Sao, Paulo, Belo Horizonte and Porto Alegre have examples of recent built shopping malls located in central areas.

especially related to the spatial pattern and dynamics involved in the growth process.

The second part of this chapter has examined the spatial pattern of Latin American cities and reviewed the most important schematic models developed for these cities, bringing to light some of the main dynamic processes that shape them.

The discussion of the internal structure of Latin American cities also brought up the issue of differences or similarities in patterns of development between the cities of the South and North, as commented earlier in this chapter. Among the evidence for a single pattern of urbanisation, are the following:

- movement of high-income groups to suburban housing;
- massive peripheral growth with subsequent problems of sustainability, transport and the control of growth;
- decline of inner city areas and gentrification, and;
- location of shopping malls in peripheral areas.

The next two chapters will examine the main dynamic processes in Latin American cities. Chapter 3 discusses the peripherisation phenomenon, which is the general process by which Latin American cities grow and produce the core-periphery pattern. This chapter will also analyse the emergence and evolution of spontaneous settlements within this growth context. Chapter 4 will look into inner city processes of change that break down the strict core-periphery pattern.

Chapter 3

The Peripherisation Phenomenon

The phenomenon of peripheral growth, which has been recognised by Latin American researchers and planners and termed 'peripherisation', can now be considered as an established process of growth of most Latin American cities. This chapter is intended to explain the main mechanisms that produce the peripherisation phenomenon.

The development of spontaneous settlements within the context of global spatial change is a key element for the understanding of the dynamics of the process. Therefore, it is necessary to study the physical/spatial elements that comprise this development process and the dynamic of its formation, growth and consolidation. The study of spontaneous settlement in the context of urban growth is important not only for the purposes of planning but also for understanding the nature and scope of urban problems in developing countries. The need to discuss the dimensions of urban growth in developing countries, along with the implications of the corresponding mushrooming of spontaneous settlements, has been addressed by UNCHS (1982; 2003).

The next sections review the problem of spontaneous settlements, examine their main characteristics, and explore locational issues. On the basis of the studies developed for spontaneous settlements, the second part of this chapter examines the dynamics of peripheral growth and urban transformation resulting from the process of expansion, in which the development of low-income settlements plays an important role.

3.1 Spontaneous settlements

The most obvious difference between cities of the two Americas and between Latin America and Europe is the phenomenon of squatter settlements and self-built housing (Griffin & Ford, 1980). Squatter settlements similar to those in Latin American cities, have occurred in Europe and Japan as an ephemeral process resulting from new industrialisation as in England, or in transition periods of renovation as during the execution of Hausmann's plan in Paris (Kostof & Tobias, 1991). However, they do not contribute to the shaping of the character of rapidly growing urban areas, as they do in Latin-American countries, where they have affected the internal structure of the city.

Spontaneous settlements have been studied especially as a problem of housing in the Third World, and usually addressing the question of illegal occupation of land. There are a number of terms, most of them local, which are used to designate these settlements: slums, squatter settlements, uncontrolled settlements, shantytowns, *favelas, villas, callampas, campamentos, barriadas, pueblos jovenes, colonias, bidonvilles* and so on (Dwyer, 1975).

In the present study the term *spontaneous settlement* will be used to designate low-income settlements in Latin America, for two reasons. Firstly because the term 'spontaneous' relates to the mode of formation and development of the settlement, and refers also to the resulting morphology. Secondly, this term is preferable to its closest alternative, 'squatter settlements', because the term 'squatter' carries legal implications which are outside the scope of the present study.

A spontaneous settlement, as a general concept, is understood as any settlement whose development is left entirely to individuals who live on the land, without any kind of plan or control (Kostof & Tobias, 1991). The definition of spontaneous settlements within the Latin American context, however, is a difficult task. Not only is there a lot of confusion arising from the multiple terms used to describe them, but also the physical and social characteristics of these settlements vary from place to place (UNCHS, 1982).

At present, most of the literature classifies settlements as legal or illegal, or as formal or informal, on the basis of tenure status. Indeed, most of them present some kind of illegality, but often the land legislation is complex and creates confusion about what is a legal or illegal land occupation (UNCHS, 1982). The problem of such classification is that not only is legality very difficult to define, but also many settlements have a mix of legal and illegal characteristics (UNCHS, 1982).

Since the issue of legality is not a central theme for the present study, two kinds of spontaneous settlements will be considered, classified according to locational and morphological characteristics. The first are *inner city settlements*, usually called 'slums' in English. These settlements are located close to the urban core and generally present high densities since most of these settlements are old and have reached the limits of their growth (UNCHS, 1982). The second kind are called *peripheral settlements*, which are usually located at the edges of growing cities. Peripheral settlements tend to be newer than inner city settlements and also less dense. Some of these settlements might present semi-rural characteristics, but this is not a general rule (UNCHS, 1982). The most common term for peripheral spontaneous settlements in the literature is 'squatter settlements', which are defined as illegal occupation of land and therefore are not necessarily located at the city fringe.

The prototypal squatter settlement was built by owner-occupiers who invaded the land illegally. Although they can be found in Latin America (UNCHS, 1982), organised invasions are very rare. In most cases, settlements grow by gradual accretion rather than planned invasion.

Another kind of occupation that is worth mentioning is the quasi-legal subdivision (also called a 'housing tract'). These are very common in Latin America and present morphological features very like those of squatter settlements. The difference is that these settlements are developed by landlords, who sell small plots of undeveloped land to low-income people. Usually they lack infrastructure and permission to develop and, hence, the problems and characteristics of these settlements are essentially the same as squatter settlements. The literature usually does not distinguish between quasi-legal subdivisions and squatter settlements and, therefore, for the purposes of the present research, they will both be considered as 'peripheral spontaneous settlements'.

A general description of squatter settlements is given by UNCHS, which clearly illustrates their ambiguous character:

> "Squatter settlements are mainly uncontrolled low-income residential areas with an ambiguous legal status regarding land occupation; they are to a large extent built by the inhabitants themselves using their own means and are usually poorly equipped with public utilities and community services. The usual image of a squatter settlement is of a poor, underserviced, overcrowded and dilapidated settlement consisting of make-shift, improvised housing areas. The land occupied by squatter settlements is often but not always, further from the city centre than is the case with slums. Often, but not always, the houses are built and occupied by their owners. The land is often occupied illegally, while in many other cases the legality of occupation is complicated or unclear." (UNCHS, 1982, page 15)

In most cases, plots are defined as a result of a mutual agreement between neighbours and are therefore determined largely by the date of the residents' arrival at the settlement site. The result of this process is that plots of various shapes and sizes are packed together with little provision for a circulation network and even less provision for public open space. Not all plots have access to roads or walkways, and the number of roads suitable for motor vehicles is very limited. Roads and pedestrian paths in both inner city and peripheral spontaneous settlements are seldom planned but rather evolve as people and vehicle pass between buildings.

Spontaneous settlements are found both in small groups as well as in large clusters. Peripheral settlements tend to be larger than inner city spontaneous settlements since there is greater availability of large areas of land and no limitation to growth. On the contrary, more centrally located sites of limited extent are able to support only a few dwellings.

Generally speaking, no land-use controls are applied to spontaneous settlements, since they are considered illegal, with the exception, in some cases, of controls designed to curb the further extension of the settlements. In the absence of regulations, the land use pattern of squatter areas develops gradually as the settlements grow, and follows closely the private and collective needs of inhabitants. Hence, most settlements are essentially residential, mixed with small local business.

In terms of urban services, peripheral spontaneous settlements are characterised by a lack of adequate public utilities and services. Some of the most common problems are: inadequate and unsafe water supply, inadequate sewage and garbage disposal systems, lack of health and educational facilities, poor roads and transportation services and lack of recreational facilities. The residents provide for themselves the community facilities and services they lack by a variety of methods, many of them informal or even illegal. Health and educational facilities are more difficult for low-income groups to initiate and provide for themselves and are therefore most lacking in slum and squatter settlements. The result is that educational facilities and services are almost non-existent (UNCHS, 1982).

Transportation is another problem, since residents of peripheral spontaneous settlements are often extremely dependent on public transport. They mainly use buses or equivalent services provided by private companies: minibuses, collective taxis, taxis, rickshaws and so on. Some walk or use such private means of transport as bicycles and scooters. Since the bus service does not serve all settlements, they offer a large potential for the further development of private and informal companies. Thus, collective taxis, often remodelled pick-ups or minibuses, are very common, though a more expensive alternative than public transport (Eyre, 1972; UNCHS, 1982).

3.1.1 Location of spontaneous settlements

There is as yet no generally accepted theory of the location of spontaneous settlements. However, there seems to be some agreement about some important factors influencing location.

The first of these factors is *land availability*. An important issue in the location of low-income residential areas is the understanding that "whatever the mechanism by which the poor obtain land they occupy the areas which other groups have left unoccupied" (Gilbert, 1987, page 182). This means that whether the land is legally occupied or not, the location of low-income areas will be always the lowest value land available. Low-income groups occupy land that is unpopular among other groups, either because of its bad location or physical characteristics that make it unsuitable for urban uses. In many cases such land is liable to flood, suffers from some kind of pollution, or presents physical characteristics that make it difficult to service.

Thus patterns of availability of unused land play some part in determining patterns of spontaneous settlement. Land being held vacant for speculative purposes, including individual inner city plots, is frequently at risk, though for tactical purposes State or Church land is usually preferred by spontaneous settlers to private property (Dwyer, 1975).

The most significant kind of available land is that which has a very low value for more regular urban uses. Thus, spontaneous settlements are frequently located on steep hillsides adjacent to cities, on land subject to flooding, on swampland or even on islands of garbage. The settlements sited on these and similar kinds of low-value urban land usually provide the most complicated environmental problems together with problems in the provision of services, and in the health and safety of the population (Dwyer, 1975).

A second factor suggested as a major controlling element in the location of spontaneous settlements is the proximity of areas of high-intensity mixed land use, usually for job opportunities (Dwyer, 1975; Ulack, 1978). The idea that job location is an important factor in location of squatter settlements is very similar to all major theories of residential location. The difference lies in the fact that typically about 60% of the inhabitants of spontaneous settlements are unemployed or underemployed (UNCHS, 1982). It seems that in these cases, the right term is proximity of job opportunity rather than availability of jobs as such and therefore this factor has less weight than in classical theories of residential location.

Ulack (1978) suggests that the main reasons for the location of the spontaneous settlements are closeness to markets, closeness to work, and the availability of marginal land. It seems that although generally the preference is for a central location, the determining factor for the actual location of settlements is the availability of land, and thus, most new settlements are located on the urban fringe.

What seems to be a matter of general agreement is that the older the settlement the more centrally located it is. Studies also relate greater age to higher quality of life in those settlements. Ulack (1978), in his study of whether spontaneous settlements play a positive or a negative role in the Third World urbanisation, suggests that

> "Age of squatter settlement and location are significant variables in understanding the role of a squatter settlement; the two variables are also related. It is hypothesised that the older a squatter settlement, the better its location relative to employment opportunities and urban amenities, and better its site characteristics. Furthermore, as a squatter settlement ages the potential for it to become a legal part of the city increases; once it occurs, squatter residents begin to feel more secure and more a part of the city. Ultimately, tenure may occur and the term 'squatter settlement' is then no longer appropriate." (Ulack, 1978, page 538)

Ulack (1978) misses the dynamics of the process of location of spontaneous settlements, which is their most interesting characteristic, their evolution in time. This evolution is a marriage of global and local factors. At the same time that the housing stock and services are improving, or being 'upgraded', the city grows as a whole, changing the relative location of the settlements, and 'recontextualising' them.

The term 'upgrading' is used to describe both the improvement of social and physical infrastructure within an existing settlement, and the 'self-build' improvements of homes by their occupants (Shankland Cox Partnership, 1977). As describe by Soto (1989, page 17), in spontaneous settlements the typical stages of traditional urban development are *reversed*: "First, the informals occupy the land, then they build on it, next they install infrastructures, and only at the end they acquire ownership". The result is that such settlements are in a constant process of *upgrading*, giving the impression of being permanently under construction.

In peripheral settlements, upgrading occurs simultaneously with the incorporation of the settlement into the inner city by urban growth. In the present study the term 'upgrading' will be used to designate improvements in the physical conditions of the settlement itself, while the term 'recontextualisation' will be used to refer to the process of change in the relative location of the spontaneous settlement.

The process can be generally described as follows.

> "With the passage of time and with the provision of minimal public services, squatter settlements can become stable and substantial neighbourhoods. Because the amount of filtered-down middle-class housing is limited, especially in cities undergoing rapid growth, people simply improve their present homes and neighbourhoods. The older the squatter settlement, the better and more substantial it is. It is not uncommon to find entire neighbourhoods of two-storied concrete houses that have evolved during a decade or two from one-room cardboard shacks. With the gradual addition of paved streets, street lights, and plantings, many early squatter settlements are now indistinguishable from neighbourhoods with less checkered histories. New squatter settlements, meanwhile, are being added to peripheries so that the poorest quality housing continues to be located at the edge of the city." (Griffin & Ford, 1980, page 404)

The process of upgrading can be either a natural process of improvement or it can be a result of an upgrading project, which is usually an intervention by the local government. Even as a natural process of evolution, there is always a partial plan, since the services are provided by the local government and not by the inhabitants themselves.

It was on the basis of this evolution, or upgrading process, that Turner suggested that spontaneous settlements were housing solutions rather than problems. He argued that housing conditions in low-income housing settlements improve over a period of time and, thus, there was no need to demolish spontaneous settlements since they were part of the housing solution. He also argued that slum dwellers preferred the opportunity to improve and consolidate their existing housing by self-help, as opposed to accepting an institutionalised government strategy of redevelopment (Turner, 1967). His ideas started a large discussion of the role of the urban poor and spontaneous settlements, more specifically about whether the settlements were 'slums of hope' or 'slums of despair'.

As the upgrading process can be viewed as a solution to the housing problem, the recontextualisation process as spontaneous settlements develop over time has a profound impact on the dynamics of growth of Latin American cities. This issue has been addressed in the model of Griffin and Ford (1980), discussed in the previous chapter.

Spontaneous settlements which developed on what was once the city's periphery[5] some decades (or years) ago are often on land that has become very valuable, as the city expands (UNCHS, 1996). While some peripheral settlements are upgraded and become embedded in the inner city, others take on the characteristics of slum areas as time passes (UNCHS, 1982). Desyllas, Greene and Hillier (1997) studied the reasons why some spontaneous settlements develop differently from others, being successfully incorporated to the city body or remaining 'slum-like'.

---

[5] The term 'periphery' in Latin-American cities refers to the space located on the borders of the city as to the poor suburbs, and carries a connotation for poverty.

They argue that the spatial configuration of the settlements, particularly the way in which the site is embedded in the larger spatial structure of the city, is an important variable in the consolidation process. Mukhija (2001) also studied the influence of the physical condition of settlements on the upgrading process. According to his study, the location of settlements, the land uses in them, the settlements' layouts and the size of the lots, can all impact on the success of the upgrading process (Mukhija, 2001). Yet there have been no conclusive studies of these questions.

The present study assumes that the relative location of peripheral spontaneous settlements is *dynamic* and *changes with the growth of the city*. The periphery is a place under constant mutation, always reproducing new extensions of land while the old peripheries are gradually incorporated into the city, and are occupied by new inhabitants (Mautner, 1999). Peripheral growth is a result of interconnected location choices, where low-income settlements play an important role in the process. Thus, in what follows the process of growth and consolidation of these settlements will be related to the overall process of urban growth.

3.2 The process of peripherisation

As a static phenomenon, spontaneous settlements are seen only as a housing problem. From a dynamic perspective, however, they present a problem of urban development. So far, static analysis by both the research community and in planning policies has been the major approach to understanding spontaneous settlements, and the connection between spontaneous settlements and urban growth has attracted little, if any, attention. The high rates of growth have been seen as the main cause of the formation of spontaneous settlements in Third World cities, together with the inability of the State to deal with rapid urbanisation.

Although the formation of spontaneous settlements has been considered a consequence of high rates of growth in recent years, its persistence, despite the recent slowdown of these rates in Latin American cities, suggests a consolidation of the process as the normal mode of urban growth of those cities. It must be noted that what is considered here is the consolidation of the *process* rather than of the *spatial structure*. This means that the topological structure of location (core-periphery) remains the same, while the spatial location of the periphery is modified in a constant movement towards the city's borders. Spontaneous settlements keep moving and expanding the urban frontiers as a consequence of the core's development. The spatial structure can be considered as 'a pattern in time' (Holland, 1995), since it is a dynamic phenomenon in which the spatial pattern is constantly being reproduced (see Figure 3.1).

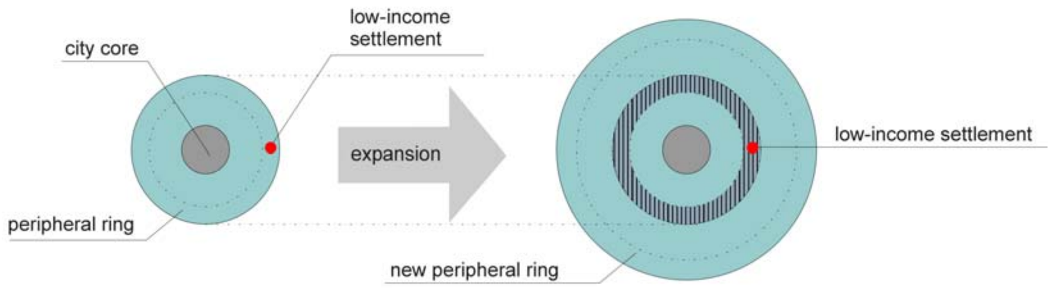


Figure 3.1 - Schema of recontextualisation of low-income settlements.

The processes of consolidation and recontextualisation are the underlying factors in Griffin and Ford's model (1980). Other authors also recognise the existence of such processes within the development of Third World cities. Dwyer (1975, page 33) suggests that "as with other categories of settlement, processes of invasion and succession may affect the location of spontaneous settlements as the city expands", in a clear connection with Burgess's ideas of invasion and succession.

An interesting feature of the recontextualisation process is its impact on land value. As mentioned before, low-income groups live on land unwanted by others. This land is expected to lose value over time because of the lack of public investment. However, as a consequence of physical expansion, many low-income housing settlements, once in peripheral and marginal locations, may come to occupy a more strategic and central location in the geography of the city. As a consequence, there is a potential differential in land values and an incentive for redevelopment (Mukhija, 2001).

This spatial pattern of Latin American cities, as suggested by Gilbert (1987), is mainly the result of market forces. Industrial and commercial companies and high-income residential groups can afford to bid for high-value land, while the low-income groups occupy residual land. According to Gilbert (1987), the dynamics of urban land use are mainly determined by the interaction of demand and supply, with only occasional interventions by government through their servicing, planning policies, and through taxation. The combination of market forces and servicing policy tends to accentuate the patterns of segregation of the Latin American city, as described by Gilbert (1987):

> "The service agencies first supply areas where the owners are politically powerful and/or can afford to pay the cost of services; industrial areas and higher income residential areas area always fully serviced. The location of service lines and roads affects the price of land and helps to determine neighbouring land uses. New high-income residential areas will develop close to existing elite areas both to gain access to services and to share the social cachet of the prestige locations. Thus, land values and the way that the land is divided by developers into large or small plots will determine the future patterns of land use. In low-income areas, the lack of adequate services will discourage most higher income groups from moving in." (Gilbert, 1987, pages 182-183)

In her study of São Paulo, Mautner (1991) describes this process and suggests that the production of the city happens in three stages of development (or labour). The first consist in the division of peripheral land and the construction of houses on that land (usually self-help housing). As soon as owners occupy these houses, demand for infrastructure is created. The second stage corresponds to the response of local government to the pressure for infrastructure. For that, however, the land has to be legalised. When the legalisation is complete, the infrastructure is provided and the periphery is made into a proper urban space. At this point the land is ready to be included in the market and the lots comprising the urban landscape start a process of change that can take up to 20 years, until the periphery is perfectly incorporated into the city (Mautner, 1991).

It must be noted that, although the processes of upgrading have been studied mainly as improvement processes and planning solutions for spontaneous settlements, these processes can also be observed for all low-income housing areas in Latin American cities, and together with the recontextualisation process play an important role in the process by which the Latin American city grows.

As the city grows, low-value residential areas, which are mainly low-income areas, gain value through the combination of improvements in the housing stock as a result of individual long-term investments and changes in the relative location caused by spatial change (urban expansion). Investments in infrastructure in these areas follow this rises in value, reinforcing the process. As a result of the increased value of the land, taxation also increases, making it too expensive for low-income residents to remain in the area. In addition, with the increase in value, competition for the area increases and it becomes profitable (interesting) for the low-income residents to consider selling their properties and moving further out to cheaper locations. In other words, as the growth of the city passes over low-income areas, many of their original inhabitants move further out, renting or selling their houses that are now located in more valuable areas (Dwyer, 1975). Hence, Latin-American suburbs (or peripheries) are pushed out to the border of the city by internal growth, in a process very similar to that of invasion-succession described by Burgess (1925). This invasion-succession process, produced by the combination of market forces and higher valuation of land as an effect of the expansion of the city, is one of the main dynamics that engender the process of peripherisation.

Peripherisation is defined here then as a kind of growth process characterised by the expansion of borders of the city through the massive formation of peripheral settlements, which are, in most cases, low-income residential areas. These areas are incorporated into the city by a long-term process of expansion in which some of the low-income areas are recontextualised within the urban system and occupied by *higher* economic groups while new low-income settlements keep emerging on the periphery.

In terms of urban planning policies, the peripherisation phenomenon is seen usually from a static point of view. The focus of government interventions is still on the local/housing scale, upgrading settlements and providing housing tracts for low-income groups. There has been little focus, either on the dynamics of the process or on the linkage between the local and global scales, that is, on the overall growth of cities, which has been seen as a mere result of a demographic phenomenon.

Peripherisation, like urban sprawl in Western cities, is a suburbanisation phenomenon. Whilst urban sprawl has been studied in detail and its main features seem to be broadly understood, in Latin America's case the need to understand the peripherisation process remains a central issue. In contrast to sprawl, which is an inherently spatial problem, urban peripherisation is essentially a social problem with a spatial character. From a social point of view, peripherisation is not a simple problem to solve, neither is in the hands of planners to attempt a solution. As a spatial problem, and more specifically as an urban development problem, the phenomenon requires considerably more investigation. According to Batty:

> "Interest remained alive [in urban morphology] because of the long-standing recognition that, although the form of the cities is the ultimate outcome of both subtle and convoluted social processes, it is form that we are most able to manipulate and design in the grander quest to improve the urban condition." (Batty, 1999, page 475)

For those reasons, the spatial features of peripherisation will be addressed here.

Like urban sprawl, peripherisation is a fragmented and discontinuous development. It also presents problems relating to urban sustainability. First, it is not environmentally sustainable since in many cases, spontaneous settlements are sited on environmental areas, which not only are unfit for urban use but also are potential environmental reserves. Also, the lack of basic urban infrastructure damages the environment. Secondly, peripherisation has profound impacts on the quality of life of inhabitants, not only due to the lack of infrastructure and other urban services, but also because their health and safety are at risk, since they often live under constant threat from 'natural' hazards such as earthquakes and floods. Finally, like urban sprawl, peripherisation has effects on transportation due to the resulting length of trips. As peripheral areas are usually low-income residential areas, however, the impact on transport is not so high as it might be, since transport services are mostly public and collective.

Peripherisation also has an impact on the costs of infrastructure and urban services. This is due to the irregular layout of spontaneous settlements, since their physical structure makes servicing difficult and costly to provide (Mukhija, 2001). By contrast with urban sprawl, low densities are not a problem of peripherisation. Studies from the 1970s suggest that the lowest densities in Latin American cities are those found in high-income residential areas, the highest densities are in middle-class areas and the densities of spontaneous settlements are somewhere between the two (Amato, 1970a).

Finally, an interesting difference between urban sprawl and peripherisation is that, while urban sprawl is directly related to people's preference for suburban settings, peripherisation is not a direct consequence of locational preference. On the contrary, people who move to the city's border do not wish to live there but are impelled to do so.

The peripherisation phenomenon is an increasingly significant issue, particularly in the larger cities of Latin America. In those cities, the demographic growth rate has slowed down, migration has taken second place to natural increase, and the bulk of the housing stock now consists of consolidated low-income residential areas (or areas in process of upgrading), with a large number of spontaneous settlements.

Peripherisation is an urban spatial problem with strong social and economic effects and one that is unlikely to be alleviated or reduced without informed planning action. The peripheral ring of Latin American cities consists mostly of low-income housing, including large spontaneous settlements, which usually lack urban services of any kind (see Figure 3.2). As such, peripherisation clearly constitutes a social problem. However, it is not simply a problem of extreme social inequalities being expressed in the city in a very concrete spatial form. Rather, the problem is the *perpetuation* of such a form in space and time and, in this sense, peripherisation is a social problem of spatial order.

Figure 3.2 - Aerial view of São Paulo, showing spontaneous settlement 'Favela Paraisópolis' and high-income neighbourhood 'Morumbi' in the background. Photo: Flavio Fatigati. August, 2000.

3.3 The spatial result: the morphology of Latin American cities

As discussed in Chapter 2, the spatial pattern of the Latin American city is still, to some extent, the core-periphery pattern, i.e. the high and middle-income groups are concentrated in the central areas of the city while low-income groups live on the outskirts. This is almost the reverse of the pattern of Western cities, where higher incomes are located in low-density areas in the outskirts and lower-income groups live in the urban centre. However, this is not the only difference in morphology between those cities.

As mentioned previously, the urbanization process in cities in developing countries is often insufficiently planned and poorly co-ordinated. The urban development is fragmented, that is, it creates overcrowding in certain parts of cities and leaves unused land in others. Spontaneous settlements fill some of the gaps in this erratic development, at the same time creating obstacles for any attempts to rationalise the development process and introduce effective land-use control measures (UNCHS, 1982). Hence, land occupation by spontaneous settlements not only adds to this haphazard growth, but also is partly a result of it.

The upgrading of spontaneous settlements results in changes not only in the quality of space, which is provided by urban services and infrastructure, but also brings about a morphological evolution. Erickson and Lloyd-Jones (1997) suggest that in the upgrading process, a regular street pattern is often imposed on an urban form that has grown organically, and incrementally. The morphological changes occur in the opposite direction from that suggested by Kostof and Tobias (1991) for the transformation of a Roman colony into an Islamic city (see Figure 3.3)(Erickson & Lloyd-Jones, 1997). This process, however, does not occur homogeneously in the peripheral rings.

Mautner (1991) suggests that the closer spontaneous settlements come to traditional standards, the better chance they have of being reached by urban services. Although spontaneous settlements must be connected to the city since obviously all new streets must be connected to existing ones, these settlements nevertheless have poor links to the rest of the city.

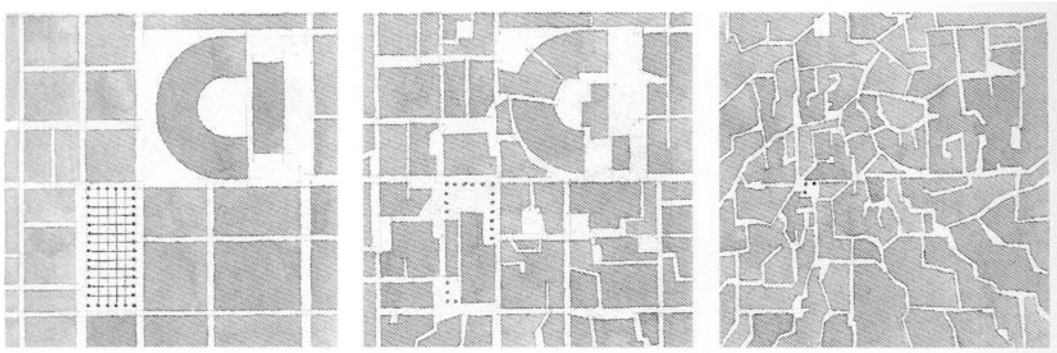


Figure 3.3 - Morphological transformation of a Roman colony into a Islamic city. Source: Kostof & Tobias (1991, page 49).

Although the peripheral areas of Latin American cities are predominantly areas of low-income settlement, other kinds of housing areas can be found on the urban periphery. As mentioned before, high-income suburban housing is often on the urban fringe, although this tends to be concentrated in specific parts of the city, away from the spontaneous settlement areas. Other kinds of low-income housing can be found mixed with spontaneous settlements, including developer-sponsored subdivisions and government-sponsored housing (UNCHS, 1982). A number of large-scale suburban developments have been erected as an attempt by the government to provide housing to low-income groups. These housing tracts and land subdivisions have been mostly located on the urban fringe, extending the urban infrastructure over large areas of unused land within the urban perimeter (Kowarick, 1993). This tendency has accentuated the problems of these cities, for example by increasing transport problems. Fragmentation also raises the cost of urbanisation since to supply the new planned residential areas with water and sanitation, this infrastructure have to pass through empty areas without servicing them (Balbao, 1993; Villa & Rodriguez, 1996). The presence of vacant areas within the urbanised area is a land market strategy for raising the value of land which is very common in Latin American cities. Speculators leave vacant areas in between those that have been occupied so that the former can be put onto the market later at higher prices. This contributes substantially to the fragmented spatial pattern of those cities (for a discussion on this issue, see Clichevsky, 2002).

The morphological result is a fragmented set of patches, in different patterns and often disconnected from each other. In a description of the spatial structure of the Third World city, Balbao (1993, page 24) suggests that it "shows a spatial city made of many different pieces drawn together in a rather accidental way. There are more of some kinds than of others. Those in the periphery are incomplete and more 'fragile', while older areas are well established with clearly-defined boundaries". This can be clearly observed in the aerial photograph of Belo Horizonte, Brazil shown in Figure 3.4, below. The image shows an area of the peripheral ring of the city where spontaneous settlements and vacant areas serve to disconnect pieces of the grid, fragmenting the urban fabric.

Balbao (1993) suggests that spatial fragmentation is one of the most distinctive features of the city of the Third World, and the one that most differentiates it from the Western city. However, little research has been carried out either on this subject or on the morphology of Latin American cities as a whole (Moudon, 1997).

Figure 3.4 - Aerial photo of Belo Horizonte, Brazil. Source: PRODABEL (2001).

Although fragmentation is a characteristic of the Latin American city, it tends to be more accentuated in peripheral areas and decreases towards the city centre. Thus, the more consolidated the urban area, the less fragmented is its spatial pattern. Fragmentation could also be recognised as related to the size of the grain, that is, to the size of building units. This is another distinction between core and periphery, since the sizes of building units tend to be smaller in the periphery.

3.4 Summary

This chapter has examined the dynamics of the peripherisation phenomenon. Since past research has been largely devoted to spontaneous settlements, studies of their evolution are a key component in understanding the process of peripherisation. Hence, the topic of spontaneous settlements has been reviewed, and aspects related to their evolution and location discussed. On the basis of these studies, the chapter has explored the relevant aspects of the peripheral growth process and their impact on inner city change in Latin American cities. Finally, a brief analysis of the resultant morphology has been developed.

The next chapter will examine the inner city processes that, together with the peripherisation phenomenon, shape Latin American cities.

Chapter 4

Inner City Dynamics

This chapter examines intra-urban mobility processes in Latin American cities, reviewing the literature on these processes and discussing their relevance in the Latin American context.

There is very little research on intra-urban mobility or inner city dynamics in Latin American cities. The most important study was developed by Turner (1968), based on his experiences of spontaneous settlements in Peru. Turner proposed a model of intra-urban mobility for low-income groups, more specifically, for migrants from rural areas. According to this model, migrants would follow a two-stage process of initial settlement and subsequent intra-urban relocation in which they would first live as renters in the inner city and later move as owners to peripheral settlements. His model provoked a large discussion within the research community, which discussed the validity of the model for different contexts based on the available evidence (see Brett, 1974; Conway, 1985; Conway & Brown, 1980; Gilbert & Ward, 1982; Kliest & Scheffer, 1981; Lindert, 1992).

Apart from these studies of the mobility of migrants, there has been a lack of research on the Latin American inner city, especially regarding decline and revitalisation processes, similar to those occurring in Anglo America and Europe. There are concerns as to whether these processes are fundamentally different in *kind* or only in *degree,* as pointed out by Ward (1993). In his studies, Ward (1993) analysed the processes of core decline, revitalisation and gentrification, making a comparison between these phenomena in Anglo-America and Europe and equivalent phenomena in Latin American cities. Amato (1968; 1970b) has developed some interesting work on the movement of elites in Bogotá, Colombia, which will be discussed below.

In the following sections each of the concepts for each of the inner city processes (filtering, gentrification, movement of elites to the outskirts and central-decay) will be presented in their original context (in Western developed cities) and then their applicability, and similarities and differences discussed in the Latin American context.

For the purposes of this study, the inner city change process will be referred to as 'residential mobility' only, focusing on the process of change by which housing stock passes from one social group to another.

4.1 Filtering

Filtering consists basically in a housing area moving down the income scale. Thus, as the former inhabitants move to better quality dwellings their previous homes become relatively cheaper and so accessible to lower-income groups.

Filtering is generally seen as part of the normal dynamic of urban growth, as a natural consequence of the deterioration of dwellings and income growth, in a process that would start with the highest income groups moving out to new homes, initiating a rippling process whereby all other homes would move down the income scale.

According to Mills and Hamilton (1994), the notion of filtering is based on the observation that most people, and particularly poor people, live in hand-me-down housing in Western cities. Filtering has been as a natural solution for the low-income housing problem, since the demand for new homes by higher classes initiates a process that should generate housing for the lower-income groups by the filtering-down process. When this same concept is applied in a Latin American context, however, the notion that low-income groups live in filtered-down housing in the centre of the city does not seem so appropriate. As discussed in the previous chapters, it is well known that low-income groups in Latin American cities live mostly in the peripheral ring of the city and usually in spontaneous or illegal settlements built by self-help construction (Gilbert, 1987).

There are, of course, decaying areas in the centres of large cities which house a small percentage of the low-income population. It is necessary to note, however, that these areas not only house a small percentage of the low-income population, but also are not found in all sizes of Latin American city, since many medium and small cities do not have this kind of housing in the city centre.

It is also necessary to note that there are very few studies of the processes of change in central areas of Latin American cities, as Ward (1993) points out, and no empirical study of the filtering process in particular. Although some authors affirm that filtering is a significant process for Latin American urban development (Johnston, 1973; Ward, 1993) there is no empirical evidence that this process has any major role in explaining urban growth in these cities.

Filtering theory is also based on the assumption that the housing stock is not demolished or replaced, but is re-used by a lower economic group when it loses value. The housing stock in Latin American cities, however, seems to be more easily demolished and replaced than in Western developed countries. In Latin America there is a growing concern about the preservation of surviving historical buildings in the centres of cities, the majority of which have already been. Frequently, these buildings are demolished and replaced by modern buildings with higher densities, or occupied by governmental or highly profitable commerce/financial uses[6]. Of course, this varies according to the country and in most large Latin American cities at least part of the historical centre is filtered down.

However, it is difficult to find evidence that filtering is a significant process to explain urban change in Latin America. Alonso (1964) in a short paper about the form of cities in developing countries suggests that the demand for housing for low-income groups far exceeds the supply of affordable housing, and that this surplus of demand spills into the nearest available space, resulting in the peripheral slums and rings of spontaneous settlements.

The fact is that even if a filtering process exists in Latin American cities, the demand for affordable housing is indeed much larger than the supply produced by the filtering process. This is due to the distribution of income in Latin American society which can be represented by a pyramidal model[7], where the high-income group form a minority of the top of the triangle, the middle-income group are in the middle part of the triangle and the low-income group is the large majority of the bottom of the triangle (see Figure 4.1).

---

[6] An example of this is Paulista Avenue in São Paulo, Brazil, which is one of the financial centres of the city. This avenue consisted mainly of large isolated houses in large plots, with front gardens, which used to house high-income groups. The majority of these houses have been demolished, and the very few that remain are either part of a building complex composed by high buildings, or are occupied by high-profitable commerce destined to upper-middle and high income groups.

[7] This notion originates from sociology of the 1960s in Latin America, and it is now considered a traditional common-sense within the research community. References to the pyramidal model are also found in the literature of the pre-industrial society (Sjoberg, 1960).

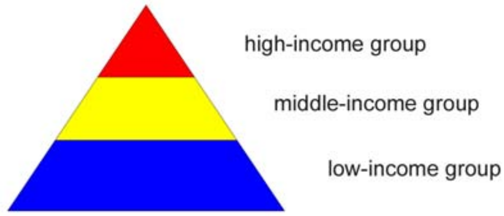


Figure 4.1 - Pyramid model of distribution of population into economic groups in Latin America.

So, even if there is a movement of the high-income groups towards the outskirts of the city and a consequent decay of the central area, it seems that first, the amount of available housing created by this process is far from sufficient to house the low-income group which is rather large; and, second, that most buildings abandoned by the high-income groups are not filtered-down to lower-income groups but are either demolished and replaced or house commercial, institutional or financial uses.

4.2 **Gentrification**

Gentrification has been seen as a deviation of the filtering process. It consists essentially of reinvestment in a central urban area, usually one which has lost value for occupation by a higher economic group (the *gentry*). The reinvestment serves both to revitalise the urban area and to renovate the housing stock, and might be a private investment, as in Britain, or planned and publically funded, as in the United States (Smith, 1979). This process usually involves some kind of displacement of the lower economic group which occupies the area.

According to Smith (1982, page 141) gentrification as a process is “rendered a chance, extraordinary event, the accidental outcome of a unique mix of exogenous factors”. Its roots are “the devaluation of much inner and central city residential property; the middle class migration to other neighbourhoods and the suburbs; the accelerated devaluation of suburban property with the influx of working class tenants and their landlords" (Smith, 1979, page 25).

Gentrification has been erroneously understood as a return of people from the suburbs, but it actually involves "migration within the inner city itself as small, youthful households, commonly in the prechildbearing or early child rearing stages of the family cycle have moved from rented accommodation to single-family owner occupation in the inner city, rather than seeking a suburban home" (Hamnett, 1984, page 285).

Although many studies of gentrification have been made, the most widely acknowledged hypothesis is Smith's 'rent gap' theory (Smith, 1979; 1982; 1986). Smith's theory is essentially focused on the 'consumer-side' of gentrification, rather than using the traditional demand-orientated economic approach. In short, Smith suggests that gentrification occurs in places where the value of the potential rent is higher than the actual rent (due to the age of the building stock and the physical condition of the neighbourhood). Gentrification consists of a process of revitalization of an urban area and its housing stock, in order to obtain the highest possible rent in the area. Smith argues that gentrification is part of a larger process of uneven development, in which investments in urban areas are always shifting location in search of higher profits. In Smith's view, there is a cycle of capital investment and disinvestment, which results in devaluation (or depreciation) in a specific urban area, while other areas are being revitalised and gentrified.

Ward (1993) argues that gentrification is not a very significant phenomenon in Latin American cities. According to him, the trends towards downtown reinvestment, inner city regeneration and gentrification led by both public and private sectors cannot be found in Latin America. Ward suggests that this is due, first to the lack of political ability and effective initiative by planning authorities; second, that the 'rent gap' is probably insufficient to stimulate reinvestment in the central areas, since devaluation of property and land has never taken place on the same scale as in Western cities; and, third, the nature of Latin American demand, that is, high-income groups tend to locate in segregated areas and not in the same areas than low-income groups.

Although Ward (1993) seems to be right about the lack of impact of gentrification in Latin American cities, there seem to be other strong reasons for this. The first is related to the ratio of income groups or, in other words, the small demand for gentrified areas due to the relatively small proportion of the high-income population. The second reason is that although high-income groups have moved towards the outskirts of the Latin American city, they are still located in relatively central areas, and the outer ring of the city remains the location of the low-income group's. Therefore, it is understandable that gentrification does not seem so necessary in an urban structure where the high-income groups are not so far from the centre and not as numerous as in the cities of Western developed countries.

In his study of gentrification in the United States, Clay (1980) argues that two kinds of private residential reinvestment have taken place in American cities. One is gentrification, which has been discussed above. The second is referred to as "incumbent upgrading" and consists of an upwards social change in the housing area, without spatial mobility. According to Clay (1980) this usually takes place in the United States in moderate-income neighbourhoods where reinvestment is primarily accomplished by long-term residents.

The differences between these two processes can be confusing, especially when dealing with Latin American urbanisation (see Kool, Verboom & Van Der Linden, 1989). Moreover, the process of "incumbent upgrading" described by Clay (1980) seems to be very similar to the upgrading process in low-income settlements discussed in Chapter 3, which is a common feature of Latin American urban development. For this reason, it will be discussed in a separate section.

4.3 Upgrading and succession

One of the main differences in the urbanisation process of Latin American cities when compared with the Western developed city is the process of upgrading, which is essential in understanding the dynamics of growth and change of Latin American cities.

By contrast with Western cities, where the majority of low-income groups are located in the decayed centre of the city in filtered down housing, most of the Latin American low-income groups lives in self-build houses either in 'pockets' within the city centre or on the urban periphery.

This 'self-built' housing stock is then slowly renovated by a process known as *upgrading*. As discussed in Chapter 3, the term 'upgrading' is usually used to describe both the improvement of social and physical infrastructure within an existing settlement, and the 'self-build' improvements of homes by their occupants. Although upgrading is usually a natural and individual process of housing improvement, it can be helped by institutional upgrading projects, which provide the necessary urban services and infra-structure and are mainly initiated by programmes for the renovation and legalization of spontaneous settlements executed by the local planning authorities.

As discussed in the previous chapter, the process of upgrading occurs simultaneously with the incorporation of the settlement into the inner city by urban growth. At the same time that the housing stock and services are improving, or being 'upgraded', the city grows as a whole, changing the relative locations of such settlements, and 'recontextualising' them.

It is interesting to see the similarities between the upgrading-succession process in Latin American cities and the gentrification process in Western cities, as noticed by Clay (1980). The first thing worth mentioning is that both processes involve renovation of an urban area and its housing stock. Whilst upgrading is usually part of the natural process of expansion of Latin American cities, the improvement in the area occurs due both to the upgrading process and the improvement of urban services, together with the process of recontextualisation, which results in an increase in the relative accessibility of the location. The second point is that the actual succession process in Latin America usually happens by replacement of the housing stock, rather than by a renovation process. Finally, upgrading does not seem to be subject of a cycle of investment and disinvestment, but is a continuing process towards the city fringe. However, Smith's theory of the rent gap could also apply if this process is viewed in terms of urban land rent rather than housing stock, given the logic that there is a gap between the rent obtained for the actual use and the highest rent possible in that location.

4.4 Movement of elites towards the suburbs

The movement of high-income groups to the urban outskirts is seen as a process that is interrelated with the decay of the city centres. It is originated by a change in the locational preferences of the high-income groups, who prefer to leave the city centre for suburban areas, provoking the decay of central areas, which then start being occupied by lower income groups through filtering.

In the Latin American city, the high-income groups were originally located close to the main square, which was the governmental, commercial and ecclesiastic centre of the city, while the lower income areas were further out at the edge of the city (Gilbert, 1987; Griffin & Ford, 1980). Amato (1970b) suggests that the breakdown of the colonial land use model was due to the movement of the upper class who, during the twentieth century, gradually moved away from the central city to occupy suburban housing. This movement towards new suburban locations was provoked by the overcrowding and deterioration of living conditions in the centre core, caused by the pressure of rural-urban migration into the inner cities (Payne, 1977). The high-income groups moved to low density areas, generally the ones best favoured by microclimate, urban services and transport facilities (Dwyer, 1975). It must be noted that as the cities continually expanded to absorb these new settlements, high-income groups continually moved further out (Payne, 1977).

Amato examines the role of the elites in the rearrangement of traditional land use patterns and development of new settlement configurations, and relates the pattern of elite residential location choice to environmental quality and, in turn, to city form and internal spatial structure (Amato, 1970b). He argues that the direction and shape this movement has taken may be considered the turning point in the spatial development of the Latin American city, and the key influence on residential locational choices of other groups. According to his studies, the possibilities for residential location open to the upper income group have been limited, in ways which he summarises as follows:

> "The first hypothesis relates the *accessibility* of particular sites to the centre city. This, of course, was a primary concern of the wealthy in their early move outward. Places of work, commerce, trade, and cultural and recreational activities remained in the centre of the city for many decades after the collapse of the colonial residential model. Accessibility criteria dictated that residential locations be near good avenues of transportation. If these did not exist, they had to be developed before wealthy groups would move. The second hypothesis relates to the particular *characteristics of the site* itself. Natural features such as mountainous terrain, areas subject to inundations, and riverbeds or large bodies of water necessarily constrained residential locational choices of upper income groups. A third hypothesis involves existing social and physical *characteristics of the immediate neighbourhood*. Clearly, upper class residents would not move into or very near areas already settled by less advantaged groups." (Amato, 1970b, pages 96-97)

Although his study focuses only on the transitional period, that is, during the late stages of the colonial city and early stages of the transitional city, it seems that his criteria could also be applied to the contemporary Latin American city. One could say the same about his study of the spatial directions of the outward movement of the elite. According to Amato, the direction and form of the movement of elites out from the city centre is not haphazard or scattered. Rather, upper income groups proceed out from the city core along specific lines and continue to move in the same direction over long periods of time (see Figure 4.2). This is in part due to the fact that new high-income residential areas tend to develop close to existing elite residential areas, both to gain access to services and to share the social cachet of the prestige locations (Gilbert, 1987).

It seems that, besides the restrictions suggested by Amato, the upper class in the Latin American city have the ability to maintain their location, as well as to gain the newest and most desirable facilities, and, in short, to live where they please (Amato, 1970b). Thus, the very wealthy also seek out the choicest land in the community in terms of environmental advantages and amenities. Another very interesting point about the issue of elite residential location is that the upper classes tend to occupy an inordinate amount of space in comparison to other urban contexts. In other words, the high-income groups, which are usually a very small percentage of the total population of a city, occupy a disproportionately large area, sometimes as much as one-quarter or one-third of the total urban space (Griffin & Ford, 1980).

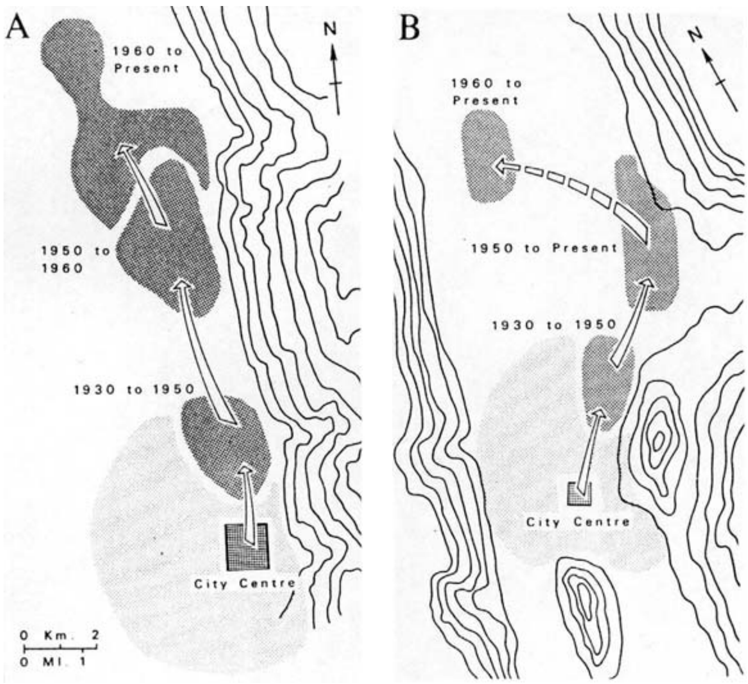


Figure 4.2 - Movement of elite residential locations in (A) Bogotá and (B) Quito. Source: Dwyer (1975, page 23).

The locational residential choice of the elite is a key element to an understanding of the internal structure of Latin American cities. This is due to the influence that their locations have on both residential location of other economic groups and the location of urban public services.

Usually, service agencies first supply areas where the owners are politically powerful and/or can afford to pay the cost of services. The result is that industrial and higher income residential areas are always fully serviced. The location of service lines and roads affects the price of land and helps to determine neighbouring land uses (Gilbert, 1987), which is another factor that contributes to the development of the high income residential areas in the same direction. The city centre also always has public services.

4.5 Inner city decay

In the typical Latin American city, the former colonial city is now the city core. In most cities, the city core plays the role of CBD, housing the financial and accommodating the commercial and governmental powers of the city[8]. Latin American cities are still characterised by vibrant, highly dynamic centres. In contrast to most Anglo-American cities, the city core in Latin America remains the prime employment, commercial and entertainment node (Griffin & Ford, 1980; Ward, 1993), although in most cities these are made use by the lower-income groups only.

According to Griffin and Ford, the expansion of the Latin American city core was rarely evident prior to the 1930s, although today it is the norm in most large cities (Griffin & Ford, 1980). The city core has suffered many transformations during recent decades. "Streets were widened, old mansions demolished, parking garages and lots created, skyscraper-office towers built, shopping malls developed, bus

[8] São Paulo, for example, is an exception to this rule. The CBD of São Paulo is separated from the commercial centre, and has moved location twice since the 1950s.

terminals constructed, and a variety of hotels, restaurants, and arenas erected in and around the city centre" (Griffin & Ford, 1980, page 400).

As in Anglo-America and Europe, some of the largest metropolitan areas have recently begun to show an absolute decline in population in their *central* areas associated with the combined loss of housing stock brought about physical decay and urban redevelopment (Ward, 1993). This has been partly caused by the movement of the elite, but in contrast to Anglo-America and Europe, the phenomenon has had neither the same impact on the centre's decline nor in the formation of low density settlements on the urban fringe. This is due to the fact that upper-income groups are a minority of the total urban population in Latin America, as mentioned previously, but also because the city centre continues to serve as a market district and low-income shopping zone. Another factor is that space in Latin American cities is too much in demand to be unused or underutilised (Griffin & Ford, 1980). The fact is that nowhere in Latin America has population loss from the inner city begun to resemble that experienced in the USA or the United Kingdom (Ward, 1993). Furthermore, in the smaller urban areas of Latin America, inner city districts continue to grow, albeit more slowly than the city as a whole (Ward, 1993).

According to Ward (1993), although the upper income group has left the city core, the centre remains an option for the lower income groups. He stresses that Latin American inner cities still fulfil important local service functions, especially insofar as they provide the location for many minor trades and services which offer cheap goods such as food, clothes, domestic services, cleaning, and repairs.

4.6 **Summary**

This chapter has presented an overview of inner city processes of change. Each of the most important intra-mobility dynamics were examined and discussed in the Latin American context.

In comparing these urban processes in both Western developed countries and Latin America, it seems that filtering occurs in both kind of cities, but is more significant in the Western countries than in Latin America. Also, it is important to stress that it cannot be seen as the normal process that drives urbanisation in Latin America, nor as the way housing is provided to low-income groups. On the other hand, upgrading seems to be a very important process in Latin American cities and together with succession seems to be the driving process of their urbanisation, whilst it has no significant importance in Western cities. Finally, movement of elites and inner city decay seem to be processes that occur in very similar ways in both kinds of cities. However, they happen to a different extent from that in Western developed countries. The reason seems to be, first, because of the ratio of the high-income group in relation to the total population, which causes the movement/decay not to happen on the same scale and, second, because the centre remains a very active and important commercial location for the low-income groups.

The next part of the thesis will explore the dynamic processes elaborated up to now through an agent-based simulation model.

Part II

Simulating Urban Dynamics in Latin American Cities

Chapter 5

Urban Modelling and Simulation

Urban systems have been traditionally studied using modelling techniques. Since the 1960s a number of models have been developed and, more recently, with advances and popularisation of computer tools, the possibilities for exploring urban systems from this viewpoint have increased considerably.

The computer became an important research environment in geography and urbanisation in the 1960s and techniques and tools have been developed ever since. In urban research, the use of automata-based models, more specifically cellular automata, is replacing traditional transport and land use models, shifting the paradigm of urban models towards a complexity approach. The idea of structure emerging from a bottom-up process where local actions and interactions produce the global pattern has been widely developed ever since automata-based models proved to be useful for a number of different urban applications. This chapter will present a discussion of the use of these two bottom-up approaches for urban simulation, starting with a brief overview of the complexity theory approach.

5.1 The complexity theory approach

Urban systems have been traditionally analysed using the systemic approach, which conceives the city as a system, i.e., a set of interconnected parts. What complexity theory introduces to this well known concept is the idea that, instead of studying the phenomenon as a *sum* of parts, the urban system, as well as other systems are seen as a result of the *interaction* of these parts. This means that not only is the whole more than a sum of parts, but the whole is different from the sum of parts (Batty, 2000).

From this point of view, complexity theory can be seen as a new systemic approach, which studies the relation between parts and whole in a different way, stressing the idea of a structure emerging from a bottom-up process where local actions and interactions produce the global pattern.

Because emergent phenomena show up in many different scientific disciplines, the exploration of the subject is perforce interdisciplinary. Many ideas developed in other fields, as in biology and physics, have been applied to the urban field as metaphors in order to explore the possible similarities between phenomena. This method provides interesting insights about the nature of urban phenomena themselves and also may feed back knowledge to complexity theory (Wilson, 2000).

The ultimate goal of the complexity theory approach is to find general laws driving all complex systems, in an interdisciplinary attempt to build a general theory of such systems.

One of the highlights of complexity theory is that it offers a new approach to the study of systems' *dynamics* (Wu, 2002). Dynamics can be more important than structure (Batty, 2000) since they permit the understanding of such systems to go beyond description (in static terms), towards capturing the internal essence of the phenomena of change. Hence, the identification and exploration of the various *surprise-generating mechanisms* governing the behaviour of complex systems is an essential issue, and moreover, a *sine qua non* for the development of a theory of complex systems (Casti, 1997).

The identification of the features of the dynamical behaviour of complex systems in several different fields, such as feedback, sensitive dependence on initial conditions, path dependence, discontinuities, phase transitions and emergence, has allowed a general study of such systems, distinguishing them from simple systems. These concepts provide a new basis for understanding change and evolution.

One of these important concepts is 'self-organization' which can be understood as "adaptive response to changing external conditions" (Allen, 1997, page 7). As Allen suggests, "the mechanism underlying self-organization is that of successive local instabilities, as fluctuations create new areas of growth and decline in the system, breaking symmetries, and creating structure and organisation" (Allen, 1997, page 16).

A second important concept is that of an 'emergent phenomenon', which is a very controversial definition within the complexity theory research community. For the purposes of the present study, 'emergence' will be used to denote "stable macroscopic patterns arising from the local interaction of agents" (Epstein & Axtell, 1996, page 35).

In short, complexity theory deals with collective behaviour in complex systems. The belief is that it is possible to understand how complex behaviour can emerge out of a simple cause, how small changes in the initial configuration of a system can give rise to dramatically different behaviours of the system, or even how small disturbances at one place in the system can give rise to large changes elsewhere (Casti, 1997).

This framework seems to fit social sciences very well, as many important social processes are complex (Epstein & Axtell, 1996) and a number of studies involving societies and their behaviour have being developed using this approach. One of the pioneer fields in complexity theory within social sciences is economics and the findings in this field throw light on the study of urban systems (Arthur, 1994; Krugman, 1996). Economic systems, like urban systems, are composed of large numbers of interacting agents, mutually adjusting to each other as time passes. The agents in an economy decide their actions consciously, with a view to the possible future actions and reactions of other agents. That is, they learn and adapt and, as this learning and mutual adaptation take place, new economic structures or patterns may emerge.

In urban studies as in economics, the agent is the human being. The city, however, presents an extra feature compared to economics: space. "[T]he spatial organization of a system does not result uniquely and necessarily from the 'economic and social laws' […] but also represents a 'memory' of particular specific, deviations from these average behaviours" (Allen & Sanglier, 1981, page 168).

Complexity theory is essentially about understanding collective behaviour in systems with a large number of elements. This seems to fit with what urban studies have been seeking for a long time, the relation between space and society; that is, to understand the collective behaviour of urban society and how the actions of individuals produce the spatial pattern we call a city.

Urban morphology can be understood as the result of a combination of particular antecedents or accidents – the initial conditions – with the generalities of the development process (Allen, 1997). Cities are systems where historical events also shape the evolutionary process and each event brings about change through system feedback, possibly driving the evolution in a different way.

Furthermore, the morphological structure of the city is built from the interplay of different dynamics, adding an extra level of complexity to these systems. As Holland (1995, page 1) suggests "a city's coherence is somehow imposed on a perpetual flux of *people and structures*". From Holland's words one can identify two different kinds of flux: the flux of people and the flux (or change) of structures. The ever-changing nature of cities, however, seems to require both interpretations for a better understanding. Not only is it necessary to understand the complex nature of each of these fluxes, but it also seems to be necessary to understand the connections (or interactions) between these complex layers, which together produce the emergent structure of urban space.

The study of complex systems is largely built upon the use of computer simulations, using the computer as a *silicon laboratory* for urban studies as well as other social systems. In what follows, two of the main approaches for simulating complex systems, Cellular Automata and Agent-Based Systems, are introduced and discussed.

5.2 Cellular automata models

From research developed in other fields, cellular automata (CA) models started to be used to represent and simulate urban phenomena. Although at first they were used essentially as metaphors, CA models have been adapted to simulate actual urban features and, thus, have been brought closer to urban reality. Indeed, the way in which the cellular automaton determines local change through vicinity rules matches perfectly with the way urban space is organised in neighbourhoods and this is certainly one of the reasons for the use of CA as a natural metaphor for urban systems. As mentioned in the foregoing, CA provide a bottom-up approach, that is based on the idea that complex global patterns emerge directly from the application of local rules. The fact that cities are complex systems that also present emergent properties makes CA such interesting tools to explore urban change.

CA-based models are inherently spatial models of complexity that deal directly with dynamic change. CA give "equal weight to the importance of space, time and systems attributes, thus imposing a frame which forces researchers to think very hard about representing any system where the importance of one of these elements becomes emphasised relative to the others" (Batty, Couclelis & Eichen, 1997, pages 160-161).

A cellular automata system is composed of a set of finite-state automata distributed over the nodes of a periodic network, which is generally a two-dimensional grid. Each automaton (or cell) is linked to the automata which surround it, and its inputs are linked to the states of its neighbours. Hence, the state of a cell at one instant depends not only on its preceding state, but also on the preceding states of all its neighbours. In a CA, all the automata are identical, have the same transition rules, and all transitions are synchronous (Ferber, 1999).

In other words, CA are models in which adjacent *cells*, as in a rectangular grid, change their *states* – their attributes or characteristics – through the repetitive application of simple rules. The *transition rules* drive changes of state in each cell as some function of states of the neighbour cells. The behaviour of a CA is, thus, determined wholly in terms of *local relations.*

Yet, the four principles that define CA (cells, states, neighbourhood, and transition rules) impose strictness on the representation of urban problems. These rules have been relaxed in order to adapt the formalisation to urban systems (see Batty et al, 1997) in different ways. The main relaxations are: neighbourhood relaxation to incorporate action-at-a-distance, cells with multiple states, and incorporation of external mechanisms in order to improve the realism of CA models.

The natural affinity with the data structures of raster GIS has helped in improving realism in some CA simulations by providing real data and, thus, offering new possibilities to approximate CA models to urban reality (Couclelis, 1997). In spite of this fact, the majority of CA-based models to date have not been linked to a GIS (White & Engelen, 1997).

Despite issues with realism and relaxations, cellular automata models have been successfully built as operational models, that is, for real-world applications. The focus of those models is diverse and include urban growth in a number of contexts like Brazil (Almeida et al., 2002), United States (Batty & Xie, 1994; Clarke, Hoppen & Gaydos, 1997; Xie & Batty, forthcoming), and Italy (Besussi, Cecchini & Rinaldi, 1998); and focusing on different aspects of urban development, like land-use change (White & Engelen, 1993; 1997), and inner city urban growth (Wu, 1998).

Similarly, CA have been successful as heuristic descriptive models, with more theoretical objectives like Batty's exploration of spontaneous urban growth (Batty, 1998), and Portugali's study of segregation and polarisation issues in Israeli cities (Portugali, 2000; Portugali, Benenson & Omer, 1997).

5.3 Agent-based models

Multi-agent systems (MAS) are systems composed of multiple interacting computer elements, known as *agents*. Therefore, the concept of agent-based models is intrinsically linked with the notion of emergence.

Agent-based systems were first studied in the field of computer science, specifically in studies of artificial intelligence. This is a relatively new field, as they have been studied only since the 1980s and have gained widespread recognition since about the mid-1990s (Ferber, 1999; Wooldridge, 2002).

Over the past few years, multi-agent systems have become more and more important not only in computer science but also in broadening their boundaries to other fields of research such as the cognitive and social sciences (psychology, ethnology, sociology, philosophy) and the natural sciences (David et al., 2004; Ferber, 1999). Agent-based simulation (ABS) introduced the possibility of modelling complex phenomena where structures emerge from interactions between individuals, opening up new avenues for theoretical and experimental research into self-organising mechanisms present in the real world.

An agent-based model consists basically of a number of agents and an environment. A simulation environment can be defined as "a medium separate from the agents, *on* which the agents operate and *with* which they interact" (Epstein & Axtell, 1996, page 5). Figure 5.1 shows a diagram of the simplest form of agent-based model, where the action output generated by the agent affects its environment, which in turn affects the agent's actions, in a feedback mechanism. Agent-based models allow modellers to explore not only agent-environment relationships, as shown in Figure 5.1, but three distinct layers or interactions: agent-agent, agent-environment and environment-environment (Barros & Alves Jr., 2003).

A multi-agent simulation includes more than one kind of agent within this modelling framework.

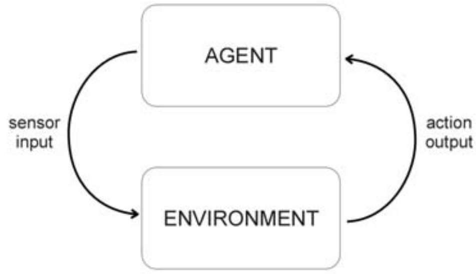


Figure 5.1 - Feedback mechanism between an agent and its environment. Source: Wooldridge (2002, page 16).

Because multi-agent approaches are used in a wide range of fields, an agent can be defined in a number of ways, according to the specificity of the problem in hand. In fact, the definition of an agent is arbitrary and depends on what an agent represents within a simulation as well as on the objective of the modeller (Anderies, 2002; Wooldridge, 2002).

In the most general terms, "an *agent* is a computer system that is *situated* in some *environment*, and that is capable of *autonomous action* in this environment in order to meet its design objectives" (Wooldridge, 2002, page 15). As Wooldridge points out, while there is a general consensus that *autonomy* is central to the definition of an agent, there is little agreement beyond this. According to him, part of the difficulty is that various attributes associated with an agent are of differing importance for different domains. While in some domains the ability of agents to learn is essential, in others it might be unimportant or even undesirable (Wooldridge, 2002). Other agents' attributes include intelligence, mobility, communication, perception, and vision.

Multi-agent simulation (MAS) offers the possibility of directly representing individuals, their behaviour and their interactions. Moreover, the idea of a system composed of individual and heterogeneous agents is a natural metaphor for many real-world systems, as in ecological systems where each being (animal, plant, etc) is a kind of agent cohabiting the same environment, or in social systems, where an agent is, obviously, a human being or, alternatively, may represent an organization or some similar entity.

The proximity of the notion of agent with an individual in a society or any kind of organization makes the construction of models of individuals and their behaviours in a computerised form a very intuitive process, facilitating the modelling process and making it more accessible to researchers with limited computer programming background. The development of user-friendly programming packages, such as RePast (University of Chicago, 2003) and StarLogo (MIT Media Laboratory, 2004), has also smoothed this process, stimulating the use of ABS as experimental tools in the social sciences.

Epstein and Axtell (1996) suggest that agent-based simulation is the basis of the 'generative social sciences', proposed by them as a new approach for social sciences, which has the simulation of 'artificial societies' as its principal scientific instrument. They argue that if a given set of initial agents, environments, and rules is sufficient to generate macrostructures of interest, then the latter can be considered 'explained' by the former (Epstein & Axtell, 1996).

In fact, agent-based simulation offers new possibilities for studying human society in many senses, including the relationship between space and society. Agent-based models with this aim focus on the interaction between agents and environment, which is usually the landscape to be studied. Within this context, it is possible to identify two main research streams according to their research focus. The first group focuses on the *agent's behaviour* or, in other words, how agents react to and within a given spatial configuration (landscape). These are mostly pedestrian movement models that investigate issues like crowd dynamics and shopping behaviour (Batty, Desyllas & Duxbury, 2002; Batty, Jiang & Thurstain-Goodwin, 1998; Moulin, Chaker & Gancet, 2004; Schelhorn, O'Sullivan & Thurstain-Goodwin, 1999; Turner & Penn, 2002). The second group focuses on the landscape's behaviour, that is, they simulate spatial change. The use of ABS for this kind of modelling came out of the understanding that human decision-making plays a major role in the process of spatial change and hence must be an explicit part of the model framework. Most of this research fits into the Agent-Based Models for Land Use and Land Cover (MAS/LUCC) category, discussed in the following section.

5.3.1 Agent-based models for land use and land cover

Multi-agent systems for land use and land cover (MAS/LUCC) can be defined as a specific class of agent-based model that

> "[…] combines two key components into an integrated model. The first component is a cellular model that represents the landscape over which actors make decisions. The second component is an agent-based model that describes the decision-making architecture of the key actors in the system under study. These two components are integrated through specification of interdependencies and feedbacks between the agents and their environment." (Parker et al., 2003, page 314)

The cellular module of a MAS/LUCC is commonly misunderstood as a cellular automaton model but, in fact, may draw on a number of spatial modelling techniques (including cellular automata), such as spatial diffusion models and Markov models (Parker, Berger & Manson, 2001). Moreover, it may not have any kind of autonomous dynamic behaviour and thus may simulate a static landscape modified by agent behaviour only.

Moreover, in LUCC models, the similarities between agent-based and CA models lead to confusion about the categorisation of some models. As discussed in section 5.2, CA models have incorporated a number of relaxations from their original formulation. In some cases, each cell is treated as an autonomous cell (or agent) with individually defined neighbourhoods and transition rules and, therefore, this model fits into the agent-based definition, although it can be considered – to some extent – a CA model as well.

As is clear from their current development, the category of agent-based models is a very broad one, since its modelling framework allows modellers to work with different layers of interactions, although their definition does not require that all these layers are taken into account in the model. From this point of view, perhaps, it is wise to differentiate CA models from MAS according to the kind of interactions they incorporate: while ABS include all agent-agent, agent-environment and environment-environment interactions, a CA model explores only environment-environment interactions through fixed neighbourhood relationships (transition rules).

The main advantage of MAS/LUCC is that they consider decision-making behaviour explicitly, while CA models may at best use transition rules as proxies for decision making. Another important advantage of ABM over CA models is that they offer a "high degree of flexibility that allows researchers to account for heterogeneity and interdependencies among agents and their environment. Further, when coupled with a cellular model representing the landscape on which agents act, these models are well suited for explicit representation of spatial processes, spatial interaction, and multi-scale phenomena" (Parker et al., 2001, page 1).

Yet, the choice for an MAS approach is not always a matter of its advantages over a CA model, but of the specific requirements for the system under study. Box (2002, page 60) points out that "there are a number of systems where population dynamics and environmental interaction are so fundamentally interrelated that a modeller cannot satisfactorily represent one without the other". In cases like these, a suitably dynamic representation of interactions between agents and environment interactions is necessary, that is, it is essential to study the population effects on their environment, and on the effects of changes in the environment on the population's actions (Box, 2002).

This is the case for many systems in the real world, and researchers have been using agent-based techniques in a number of different fields, including archaeology, ecology, agricultural (land) economics, and urban studies (Parker et al., 2001; Parker et al., 2003). In archaeology, ABS have been used to simulate population patterns in historical settlements from landscape properties like resources and climate records (Dean et al., 2000; Kohler et al., 2000). In ecology, they have been particularly used for the study of ecosystems management (Janssen, 2002). In agricultural economics, they have been used to examine the effects of new agricultural practices within a region, including the study of spill over effects (Balmann et al., 2002; Deffuant et al., 2002; Parker, 2000) as well as the adoption of new agricultural practices by farmers (Berger, 2001; Polhill, Gotts & Law, 2001). Two particularly interesting studies within land economics developed by Box (2003) and Turton (2003) recreate Von Thünen's location theory using agent-based simulations.

There have also been several studies applying ABS to urban studies, where at least two distinct research directions can be identified. The first focuses on the dynamics of land use, which emerges from a bottom-up process where agents are understood as individuals who locate according to their individual preferences. Otter and colleagues (Otter, van der Veen & de Vriend, 2001) developed a generic agent-based model which simulates the locational decisions of households and firms and explores the formation of spatial patterns. Research with this focus also includes investigations into specific urban land-use and morphological problems like urban sprawl, such as the conceptual model for sprawl in North America developed by Torrens, which simulates residential locational behaviour at the fine scale (Torrens, 2003); the model by Loibl and Toetzer (2003), which simulates urban sprawl in Vienna focusing on migration at a regional scale; and Brown and colleagues' (Brown et al., 2004a) model that examines the effect of green belts in residential developments in the urban fringe in North America.

Following the same research direction, Benenson has studied residential dynamics and spatial segregation in Israeli cities (Benenson, 1998; 2004; Benenson, Omer & Hatna, 2002), and Ducrot et al (2004) have developed a model of urbanization in peri-urban areas in Brazil to investigate the connection between urbanization, land-use change and hydrological processes.

The second research direction investigates the planning process itself, in an attempt to understand the conflict of interests among different actors involved in the process. In this case the emergent land use pattern is seen not as the result of a myriad of individuals' locational decisions, but as a mixture of bottom-up and top-down processes where rules and conflicts define the final outcomes. Within this framework, Ligtenberg et al developed a hybrid model of multi-actor decisions in land use planning combined with land use allocation processes (Ligtenberg, Bregt & van Lammeren, 2001). Semboloni et al (2004) developed another interesting model that simulates urban dynamics according to agents' behaviours and an economic system. Interestingly, their simulation model can be driven by virtual agents as well as by human users, and as a result it works as a simulation tool as well as an interactive decision support system.

In addition, hybrid projects involving different aspects of urban change have been developed recently. An example is ILUTE (Integrated Land Use, Transportation, Environment), an ambitious project that is being developed by a team of Canadian researchers (Miller et al., 2004). ILUTE is a hybrid agent-based system that includes all spatial processes affecting land use, locational choice, and transportation within the urban system.

Geographic Information System (GIS) integration with ABS has also been a target of research. So far, the most common form of integration is to use GIS to create a model's landscape from 'real world' data by importing data into an ABS system through a GIS (Box, 2002; Gimblett, 2002).

While MAS/LUCC models appear to be useful tools, it is essential to consider the kinds of information and knowledge that can be obtained from them. Two main kinds of objective can be found within ongoing research in the field: *exploratory* and *predictive* (or descriptive).

Exploratory research conceives simulation as a laboratory where theories can be explored and developed. Modellers start from a theoretical framework and formalize it in computer code in order to examine the ramifications of their framework and potentially generate new hypotheses to explore empirically. These models usually focus on particular processes or dynamics in order to achieve some fundamental understanding of specific aspects of a phenomenon (Parker et al., 2003).

Exploratory modelling can be seen as part of a theory-building process. It generally includes testing the theory (or hypothesis) by demonstrating that a set of rules can lead to the outcome of interest; it also allows the modeller to explore other possible causes that lead to the same outcome, formally exploring the robustness of the proposed causal explanations. In addition, this process might lead to the finding of outcomes not originally anticipated (Parker et al., 2003).

One of the limitations of this approach is the lack of an established method for evaluating the real-world validity of the simulations. Because these models are usually built upon abstract concepts and the outcomes are general patterns, it is difficult to determine what the models tell us about reality. While exploratory models can be excellent tools for provoking insights into general phenomena, they may provide less understanding of specific real-world systems (Parker et al., 2003).

Descriptive approaches, on the other hand, are more concerned with empirical validity and/or predictive capacity. These approaches attempt to reproduce specific real-world systems to facilitate direct empirical and policy scenario research (Parker et al., 2003).

Parker at al (2003) suggest that MAS modelling methods can be more effective than more conventional urban models for they allow modelling at a fine resolution and therefore make the best of available data; they account for heterogeneity and interdependencies, so that models can reflect important endogenous feedbacks between processes; and last, they are not constructed to meet a set of equilibrium criteria and can thus represent discontinuous and non-linear phenomena (Parker et al., 2003). Despite all the potential shown by MAS, the question as to whether a predictive role is an appropriate goal for ABM/LUCC remains open (Parker et al., 2001).

5.3.2 Evaluating ABS models

Model evaluation is an important part of any model's development processes and includes both comparisons of the model outputs with the modelled real-world system and understanding the sensitivity of the model to its internal parameters (Turner, Gardner & O'Neill, 2001).

These two evaluation steps are more commonly referred to as *verification* and *validation* and concern, respectively, the correctness of model construction and truthfulness of a model with respect to its problem domain. In other words, verification means building the system correctly, and validation means building the correct or most appropriate system (Parker et al., 2003; Sargent, 2001).

To perform verification, it is essential to conduct a sensitivity analysis of relationships between a model's parameters and its outputs. Validation, on the other hand, concerns how well the model outcomes represent the real system behaviour.

There are a number of approaches for the validation of simulation models (Sargent, 2001), including matching model output, in the case of LUCC models' spatial outcomes, to measured variables in the real-world system, and matching a model's components' structures and processes to structures and processes in the real-world system. In either case, validation depends largely on the model's objectives, and a critical issue is the decision as to how much detail the model is being designed to match. Validation usually involves performing a set of analyses that will demonstrate the relevance and accuracy of the model's results to understand or predict the real world, depending on the model's purpose.

For model validation purposes, it is important to identify clearly the objectives of a model. Where accurate predictions are the main goal, measures of the accuracy of spatial outcomes are necessary. Where the goal is to represent a process and explain general patterns that are observed across a variety of situations, validation might require evaluating how well a model reproduces critical system properties in terms of spatial and temporal dynamics (Brown et al., forthcoming-b; Rand et al., 2003). Brown et al. (2004b, page 2) stress that this process "involves judgments about how well a particular model meets the modeller's goals, which in turn depends on choices about what aspects of the real system to model and what aspects to ignore".

Validation is traditionally achieved by comparing the model outputs either with real-world data or observations or other model's output (Parker et al., 2003). This comparison is usually carried out using statistical methods, by establishing a reasonable correlation between a model's outputs and real data.

Validation seems to be a critical issue for any modelling approach applied to any system, but it can be especially difficult when using ABM to model complex systems. A number of authors stress these difficulties, which are summarised below:

- **The nature of complex systems.**

The main characteristics of a complex system, such as path dependence, feedbacks, and adaptation make these systems very uncertain and therefore, very difficult (or impossible) to predict. These issues can make it difficult or impossible to validate MAS models by replicating micro details of the system, let alone by making detailed predictions (Brown et al., 2004b; Rand et al., 2003).

- **Data availability**

In MAS/LUCC models, the data necessary to validate an agent's behaviour is hardly available as it involves privacy issues. Yet, the main objective of a LUCC model is, typically, to simulate a specific spatial pattern. Parker et al (2001, page 13) argue that "many of our research questions in LUCC are related to spatial structure of outcomes, in part because of the importance of spatial structure in other processes for which land use is an input". For this reason, a LUCC model is usually assessed through the spatial patterns that it produces, and their agreement with observed spatial patterns in the real world.

- **Validation of processes**

In some ABS models, it is necessary to validate the behaviour of the model or, in other words, to validate the individual dynamics that produce the final result analysed. The study of dynamics in a disaggregated form is one of the highlights of ABS models and, ideally, the dynamic behaviour of agents would be compared to real data; but data at the individual level, with the required detail for comparison, is hardly available in the real world.

In addition to these issues, it appears that if the purpose of the model is exploratory, that is, it is not intended to recreate a specific reality but rather to explore possible theoretical frameworks that could explain a general phenomenon, then a simple statistical comparison between outcomes and static data from a specific system might not be satisfactory to validate the model.

It is clear that model validation is one of the main pitfalls of this ABS modelling approach and there remains a need for new measures of fit between the model and data that go beyond spatial matching to focus on variability of outcomes and dynamics (Brown et al., forthcoming-b; Parker et al., 2001; Parker et al., 2003).

In fact few MAS/LUCC models demonstrate a consistent validation, or in other words, a validation in traditional terms, with proper statistical proof of consistency between real data and model outputs. Yet, in face of the issues discussed above, it seems that the concept of validation for MAS must be adapted to cope with dynamics and uncertainty issues.

In recent research, the most common form of validation found in MAS/LUCC models is through pattern analysis. Pattern analysis consists in the use of spatial configuration and composition metrics, mostly originating from landscape ecology. Landscape ecology studies the interaction between spatial patterns and ecological processes, with emphasis on the relationship between *dynamic processes* and *spatial patterns*.

Although the vast majority of recent publications on MAS/LUCC models do not make any reference to validation, and only a few mention some kind of statistical analysis with supposed satisfactory results, some significant research is being developed in this direction. Brown and colleagues (Brown et al., forthcoming-a; Brown et al., forthcoming-b; Rand et al., 2003) offer some interesting discussions of validation of ABS and propose alternative methods for validation of MAS/LUCC.

Rand et al (2003) propose validation of global patterns which are evident in empirical analysis of urban development processes: power law relationships between frequency and cluster size and a negative exponential relationship between density and distance from the centre.

Following the same assumption of the impossibility of validating ABS models in the traditional manner, Brown et al (2004b) propose validation through comparison of two different models (implementations) for residents' settlement choices in the presence of a green belt that generate the same fundamental results: a mathematical model and an agent-based model. They argue that agent-based models serve as minimally realistic models of real-world complex systems, but the fact that agent-based models cannot prove theorems give them a shaky foundation. They suggest that the scientific enterprise can be enriched if theorems of simplification can be proved through a mathematical model and the conclusions of those theorems explored in a more general context using agent-based models. Their paper demonstrates "docking" exercises using these two kinds of models, which at the very least seems to be an interesting simulation exercise, producing some discussion of real-world issues.

Finally, a third paper by Brown and colleagues discusses the impact of path dependence and stochastic uncertainty on the viability of validating MAS/LUCC. The authors argue that there are two contradictory impulses in the process of development of ABS models: the desire for *accuracy of prediction* and the recognition of *unpredictability in the process* due to path dependence and stochastic uncertainty in the models. They suggest that the predictability of such models is questionable and, therefore, the important issues concerning ABS models are whether the mechanisms and parameters of the model are correct, rather then the model outcomes. They then propose the *invariant-variant* method to assess the accuracy and variability of the multiple outcomes generated by MAS/LUCC models. This is built upon techniques for measuring spatial similarities.

From this discussion, it is possible to affirm that validation of ABS models in general, including MAS/LUCC models is still in its infancy and, as suggested by Brown and colleagues, that it is unlikely to be completely developed to the point that these kind of models will produce results with reliable predictive power. Validation of ABS models seems to be following a different path, which brings more understanding about the model, its mechanism, parameters, processes and behaviours. It seems that sensitivity analysis plays a very important role in this process, as well as a precise analysis of the outcomes, using pattern metrics and other similar techniques.

5.4 Summary

The present chapter presented a brief review of the literature on modelling and simulation approaches for urban systems taking a complexity theory approach. CA and ABS models are approaches used essentially to study a system's dynamics and have been largely used to explore aspects of urban change and growth.

Within the ABS class of models, special attention has been given to LUCC/MAS models, which are used to investigate land use and land cover using a disaggregated agent-based approach. Recent developments within the ABS approach were presented, and the main pitfalls of the approach discussed.

The following chapters introduce the Peripherisation Model and present its implementation and evaluation. Chapter 5 details the Peripherisation Model's logic and implementation, detailing how the conceptual model is implemented in the computer. Chapter 6 presents the model's evaluation through sensitivity analysis. Chapter 7 presents a set of simulation exercises which examine model outcomes in the light of urban reality in Latin America.

Chapter 6

The Peripherisation Model

The present chapter introduces the Peripherisation Model, an exploratory agent-based model for urban growth in Latin American cities. The model is built upon the theoretical framework for the dynamics of Latin American cities presented in Part I and belongs to the class of bottom-up models discussed in the previous chapter.

The main goal of this chapter is to introduce the logic and structure of each feature of this model which was developed, and will be presented, in four successive modules that add up to a model that simulates different aspects of urban growth and change in Latin American cities. Each of these features will be further explored through simulation exercises in Chapter 8.

This chapter is divided in two main parts. The first part describes the logic of the model, showing how different aspects of the peripherisation phenomenon were translated into a computerised form. The second part presents the implementation of this logic using the *JAVA* programming language within the software framework RePast (REcursive Porous Agent Simulation Toolkit*)*.

The model explores agent-landscape relationships only, and was elaborated in such a way that the behaviour rules were as simple as possible.

In the following sections each feature of the model will be thoroughly exposed and discussed.

6.1 The Peripherisation Model's logic

The simulation model was developed by adding features to a simple logic, or in other words, increasing the model's complexity step-by-step so that the understanding of its behaviour was not lost during the development process. Each of these developmental stages will be presented here as modules. Module one focuses on the peripherisation phenomenon. Module two consists of the Peripherisation module supplemented by a consolidation rule, and focuses on the formation of spontaneous settlements. Module three examines inner city processes, which are also added to the Peripherisation logic. Finally, Module four introduces spatial constraints on the simulation.

6.1.1 Module one: Peripherisation

As examined in Chapter 3, the main process behind the peripherisation phenomenon has similar dynamics to those proposed in Burgess' (1925) succession and expansion model. In Latin American cities, as the growth of the city passes over low-income areas, many of their original inhabitants move further out, while new peripheral rings are created on the border of the city. The process of peripherisation consists of the expansion of borders of the city thorough the formation of peripheral low-income settlements that are incorporated to the city by a long-term process of expansion in which some of the low-income areas are recontextualised within the urban system and occupied by *higher* economic group while new low-income settlements keep emerging on the periphery.

The main objective of the Peripherisation Module is to investigate the dynamics of formation and continuity of the core-periphery pattern. This module reproduces the process of expulsion and expansion by simulating the residential locational processes of distinct economic groups. In the model, the population is divided into three economic groups according to the pyramidal model of distribution of income in Latin American countries, as discussed in Chapter 4 (see Figure 4.1).

The simulation model is underlay by an economic logic, although it is not an economic model. It assumes that, despite the economic differences, all agents have the same locational preferences, which means they all want to locate close to the areas that are served by infrastructure, with nearby commerce, job opportunities and so on. Since in Third World cities these facilities are found mostly close to the high-income residential areas, as examined in Chapter 3, agents look for a place close to a high-income group residential area. The behaviours of the three income groups are differentiated by the restrictions imposed on their economic power. Thus, the high-income group (represented in the model in red) is able to locate in any place of its preference. The medium-income group (in yellow) can locate everywhere except where the high-income group is already located; and, in turn the low-income group (in blue) can locate only in otherwise vacant space. Figure 6.1 presents a detailed diagram of agents' rules of behaviour.

In short, agents are divided into three economic groups in a proportion based on the division of Latin American society by income. All the agents have the same objective, that is, to be as close as possible to the high-income places; but each income group suffers different restrictions on the place they can locate. Since some agents can occupy other agents' cells, this means that the latter are 'evicted' and must find other places to settle.

It is important to note that an agent's behaviour is simulated as a proxy for household behaviour. However, the peripherisation phenomenon is studied here independently of the spatial scale, and therefore the size of the cell might not correspond to the scale of a single plot.

The landscape is represented by a grid of empty cells, except from the initial seeds which are occupied by high-income agents. The resolution of the grid cell varies, as the phenomenon is studied here independently of scale.

In the Peripherisation Model, as well as in all experiments shown throughout this thesis, agents are represented by colours: the high-income agents are represented by *red*, the middle-income agents by *yellow*, and the low-income agents by *blue*. For a complete list of the colours used in the model and the cells states they represent, please refer to Appendix A.

6.1.1.1 Parameters

Two main parameters define the behaviour of the Peripherisation module: *steps* and *proportion of agents per income group*. *Steps* is the number of cells that the agent walks before trying to settle in a place (cell). This parameter represents how far people are willing to settle from their ideal location. The *proportion of agents per income group* is a percentage of the total number of agents that belong to each economic group. The total percentage must – obviously – sum to 100%, and the distribution must respect the pyramidal model presented previously.

It is important to note that the proportion of agents per economic group differs from country to country and even from city to city, and that the proportion in the model represents a relative proportion only, as there is no definition of 'economic group' implied in the model.

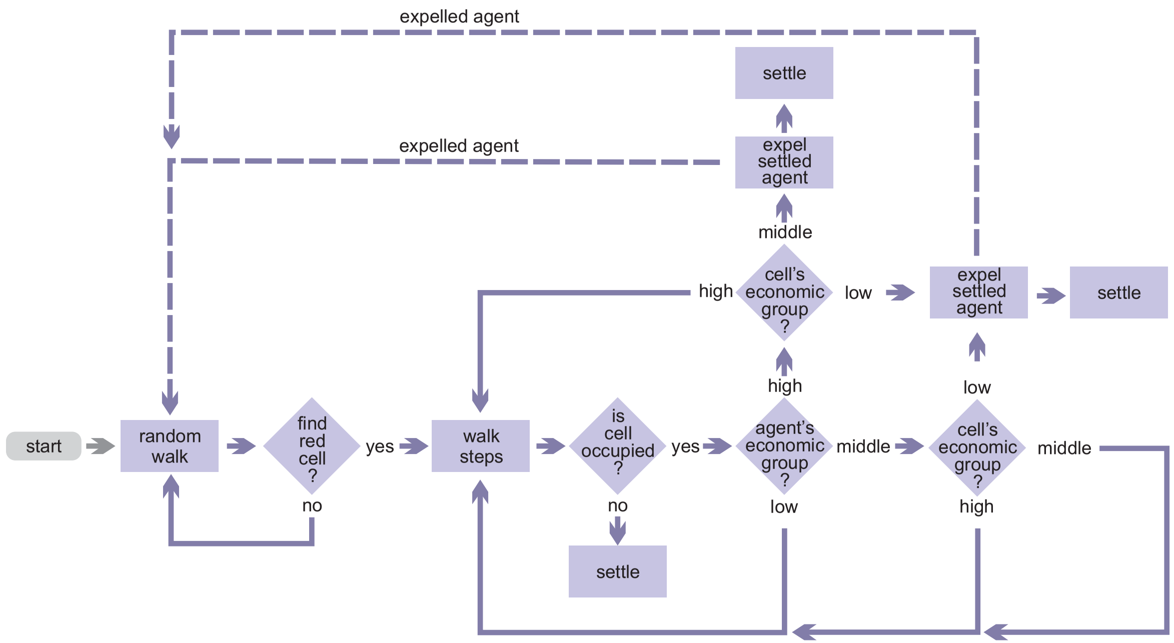


Figure 6.1 - Flowchart of the agent's rules - module 01.

6.1.2 Module two: spontaneous settlements

This module is intended to simulate the process of formation and consolidation of spontaneous settlements as part of the urban growth dynamics of the Latin American city.

The module is built upon the Peripherisation module by combining the original Peripherisation logic with a consolidation rule. This rule refers to a process in which spontaneous settlements are gradually upgraded, and, as time passes, turn into consolidated *favelas* or, in other words, spontaneous settlements that are immune from eviction, as detailed in Figure 6.2. As a result of the introduction of the consolidation logic, the spontaneous settlements module generates a more fragmented landscape than the homogeneous concentric-like spatial distribution of classes in which consolidated spontaneous settlements are spread all over the city.

The consolidation process is built into the model through a 'cons' variable. This cons variable has its value increased at each iteration of the model and, at a certain threshold (*consLimit*), the low-income cell turns into the consolidation state, represented in the model by a cyan colour. If a high-income or medium-income agent tries to settle on the low-income cell in a stage previous to the consolidation threshold, the low-income cell is replaced by the respective new occupant's economic group. Otherwise, consolidated cells are 'immune' from eviction.

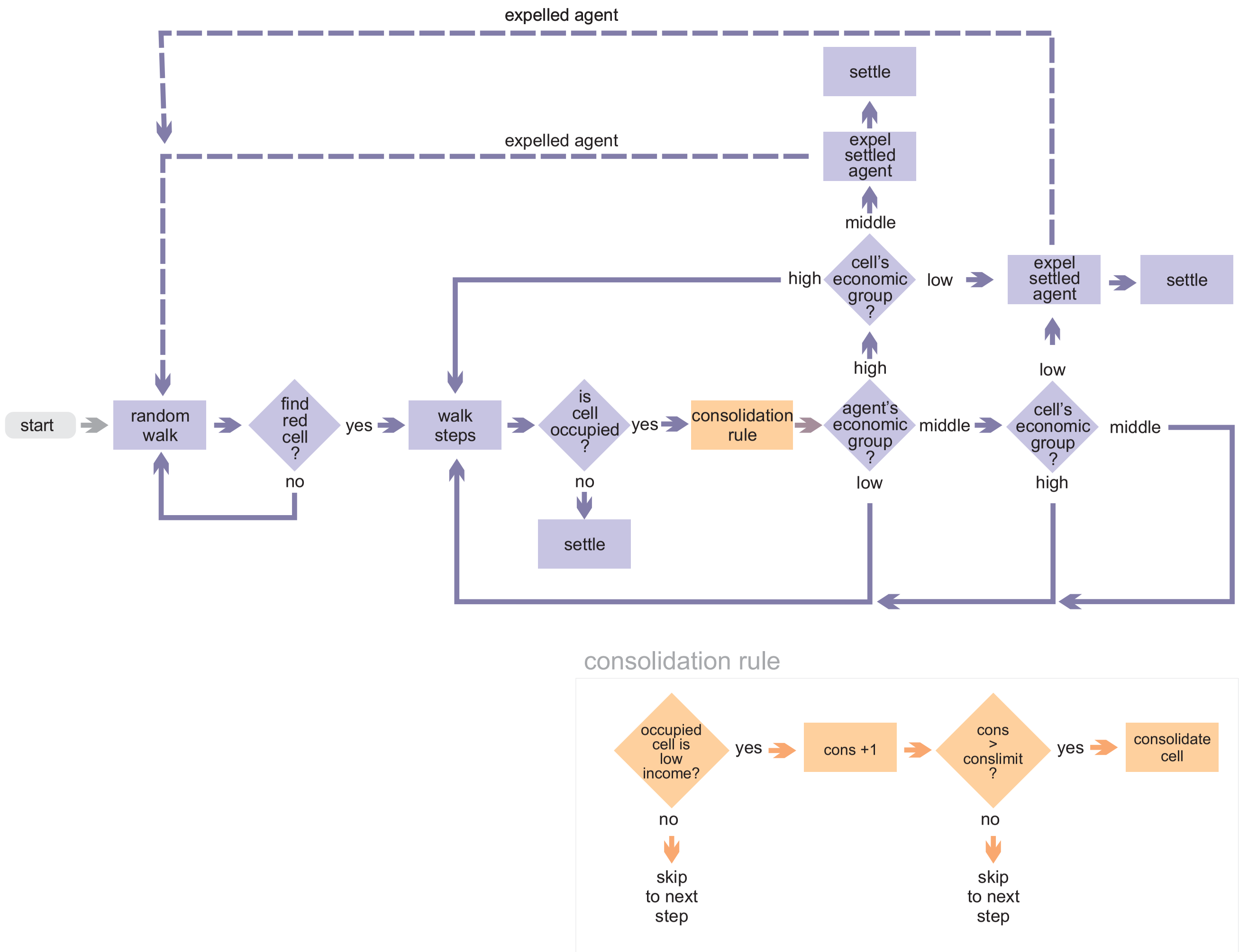


Figure 6.2 - Flowchart of the agent's rules - module 02.

6.1.3 Module three: inner city processes

In the Peripherisation module, the process of change for a higher economic group is part of the general process of growth, mainly characterised by an expansion and succession mechanism of growth rings as in the model proposed by Burgess (1925). The Peripherisation module does not consider re-occupation and regeneration of older housing in attractive inner city districts, which are common processes in Western countries and also important features of urban change in Latin American cities (as discussed in Chapter 4).

Hence, some features were added to the model in order to simulate these other aspects of Latin American urban development. These features attempt to reproduce some of the main dynamic processes in cities: inner city decay, movement of elites towards the city edge and gentrification by the process of location and relocation of individual agents from different income groups.

The model simplifies these dynamic processes using a set of very simple spatial interaction rules and allows the modeller to examine how these rules produce contrasting and complex spatial patterns in different kinds of cities. The aim of this module is to study the nature of inner city dynamic processes, and examine how these dynamics produce global spatial residential patterns.

Three sets of rules compose the inner city processes module: transition from higher to lower-income group; transition from lower to higher-income group, and movement of higher-income groups towards the suburbs.

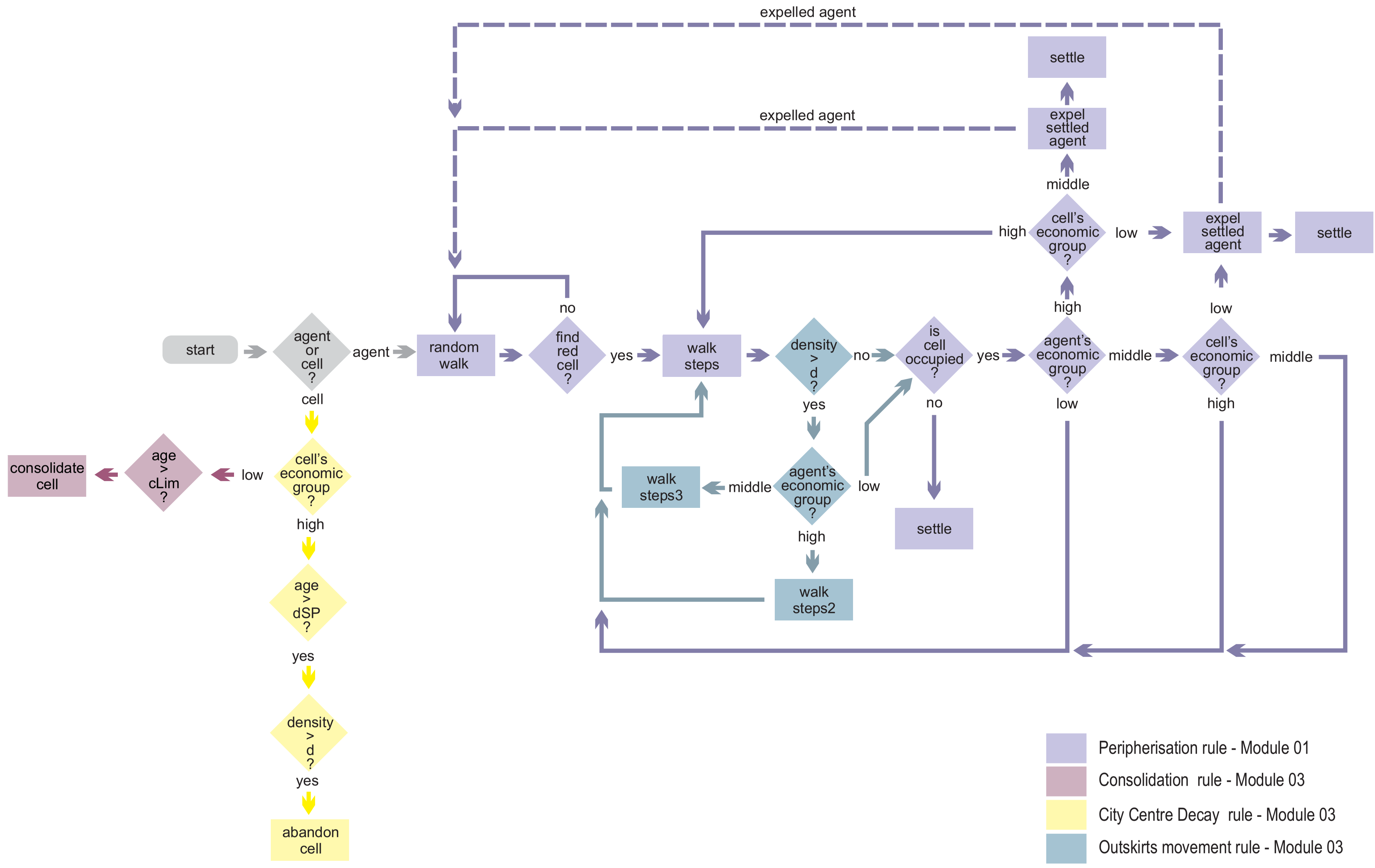


Figure 6.3 - Flowchart of the agent's rules - module 03.

Hence, the processes of filtering and gentrification are translated in the model into the change from one economic group to another. Simplifying the inner city change processes in this form, *filtering* is translated into the model as the occupation of housing stock by a lower income group than previously occupied it (moving down the social scale) while *gentrification* is the opposite, an occupation of a housing stock by a higher economic group. The detailed schema of the agent behaviour rules in these terms is illustrated in Figure 6.3.

In short, agents walk randomly through the grid in search of a place to settle. As in the Peripherisation module, all agents have the same spatial preferences but react according to different restrictions. Each agent settles on a place based on local knowledge only, such as neighbourhood density, their own income-group, and the income-group occupying their desired location.

The housing transition from higher to lower income groups is introduced in the model by adding two variables: 'age' which refers to occupied cells, and "density" which is the neighbourhood density. Two thresholds for these variables were also added: a parameter *decayStartPoint* which is the threshold value for age in which the decay is activated, and a parameter *d* which corresponds to the maximum neighbourhood density value.

At every iteration, age value is increased in the simulation for all cells occupied by high-income agents. When age gets to a certain limit (parameter *decayStartPoint*), a percentage of cells occupied by high-income agents may start to decrease through 'inoccupation' as these places become available to lower-income groups (their colour turns green, which corresponds to empty cells that can only be occupied by low-income groups). The percentage used in the model was 50% and a random number, which could be used as a parameter, regulates this in the code.

Also, if the age value of a high-income cell is higher than the parameter *decayStartPoint* and density is higher than *d,* the cell becomes available to the lower-income agents who wish to settle there, and the high-income agent must look for a place further out.

A second part of the rule determines that low-income cells with an age value higher than the parameter *consolidationLimit* will consolidate, that is, will no longer be subject to eviction.

6.1.3.1 Parameters

The simulation of the movement of the high-income groups towards the suburbs was defined using the existing variable density, the parameter *d* and *steps*, and adding two new parameters: *steps2* and *steps3*.

Thus, before settling on a patch, red agents check the number of red patches in the neighbourhood. If the density is higher than the established threshold (parameter *d*), the agents look for places further out. The amount of steps that the agent will make before trying to settle again is defined by the parameter *steps2* for high-income agents, and *steps3* for medium-income agents.

The parameter *decayStartPoint*, as mentioned in the foregoing, is the threshold value of the age of red patches, which determines when the decay rule starts acting in the system.

The parameter *consolidationLimit* is also a threshold for the age value, but it is valid for the low-income cells only. This value determines the age of a patch on which an agent cannot be evicted by other agents any more, and becomes consolidated (turns into cyan). This parameter was created in order to avoid high and middle-income agents evicting low-income agents from the centre, meaning that if a cell is consolidated, it is no longer an interesting location for the other agents.

It is important to note that the parameters *decayStartPoint* and *consolidationLimit* are both thresholds for the variable "age", and therefore are based on time. The time scale in the simulation changes according to the settings of the other parameters in the model, as well as system size and total number of agents.

The parameter *steps* is an original parameter from the Peripherisation module and is defined as the number of steps each agent walks when searching for a place to settle. *Steps2* is the number of steps that the high-income group make in their movement to the suburban areas and *steps3* is the number that the medium-income agents make in their movement.

The parameter *d* is the neighbourhood density threshold that red and yellow agents check before they settle on a place. If the density is higher than *d*, the agents look for another place to settle, otherwise they settle there.

The density threshold also plays a role in the rule for the movement of high-income groups, in which it defines – together with the parameter *decayStartPoint* and a random variable – whether a red agent will abandon the location where it is settled and migrate to a further location or not.

6.1.4 Module four: spatial constraints

The objective of module four is to introduce spatial constraints to the simulation model. These spatial constraints represent bodies of water, steep slopes, or any other area where urbanisation is not possible.

Spatial constraints are implemented by the introduction of "grey" areas as initial conditions. Agents do not settle or even walk on grey areas, which means that the area for spatial development is restricted.

In the code, for every movement agents make towards new cells, they check if the new position is a grey cell or not and, if it is, they return to their previous position and change direction in order to avoid returning to the same cell.

Spatial constraints are input to the model through matrices read by the program from text-formatted files. These matrices contain the exact number of rows and columns that correspond to the model's space size, where grey cells are represented by value 1 while zero represents empty cells (see Appendix B).

6.2 Program implementation

The model was initially built on a StarLogo platform, StarLogo being a user-friendly parallel programming tool developed by the Epistemology and Learning Group of the Massachusetts Institute of Technology (Resnick, 1994) (see code in Appendix C). The same logic was also implemented in FORTRAN (see code in Appendix D).

The version presented here was written in the JAVA Programming Language, using RePast (REcursive Porous Agent Simulation Toolkit*)*. This is a software framework for creating agent-based simulations using the JAVA language developed by the University of Chicago's Social Science Research Computing (University of Chicago, 2003).

RePast provides a library of JAVA classes for creating, running, displaying and collecting data from an agent-based simulation. In addition, RePast allows the user to customise a simulation by taking snapshots of running simulations, and creating movies of simulations.

JAVA simulation programs that use RePast libraries typically have at least two classes: model class and agent class. The agent class describes the behaviours and characteristics (states, capabilities) of agents and it is largely simulation-specific. The model class sets up and controls both the representational and infrastructure parts of a RePast simulation. This class must implement the *SimModel* interface, which is included in abstract model classes that can be specialised to suit the modeller's purposes. A typical model class should contain specific methods and variables, which are also contained in the abstract classes provided (for more details of how to build a RePast model, see section 'How To Docs' on http://repast.sourceforge.net).

The Peripherisation Model was built on this basis, and is composed of three classes: agent class, model class, and conditions class. The conditions class was created for organisational reasons only and consists of methods that check conditions used throughout the simulation run.

In the Peripherisation Model, all agents and cells are *agent* objects, that is, belong to the same class *agent*. Inactive agents represent cells, while active agents represent agents belonging to one of the three economic groups. Agent objects have a number of attributes such as active/inactive, colour and code, economic group, 'age', and 'cons'. These variables define many of the states and conditions that set the model's behaviour (see code in Appendix E).

6.2.1 Program interface

The graphical user interface (GUI) of the simulation model is provided by RePast. Three main windows compose RePast's interface: the RePast toolbar window (Figure 6.5), the model's display window (Figure 6.6), and the model's settings window, which is subdivided into parameters tab, custom actions tab and RePast actions tab (Figure 6.7).

Figure 6.4 shows a screenshot of the Peripherisation program in use. From this figure, one can get a sense of the various capabilities provided by the program. In the image, the three standard RePast windows are visible, together with three chart windows.

In the top right of Figure 6.4, the RePast toolbar window is visible. This toolbar allows the user to start, stop, pause, set up, and exit a simulation (see below), and is standard for all models that use RePast's GUI.

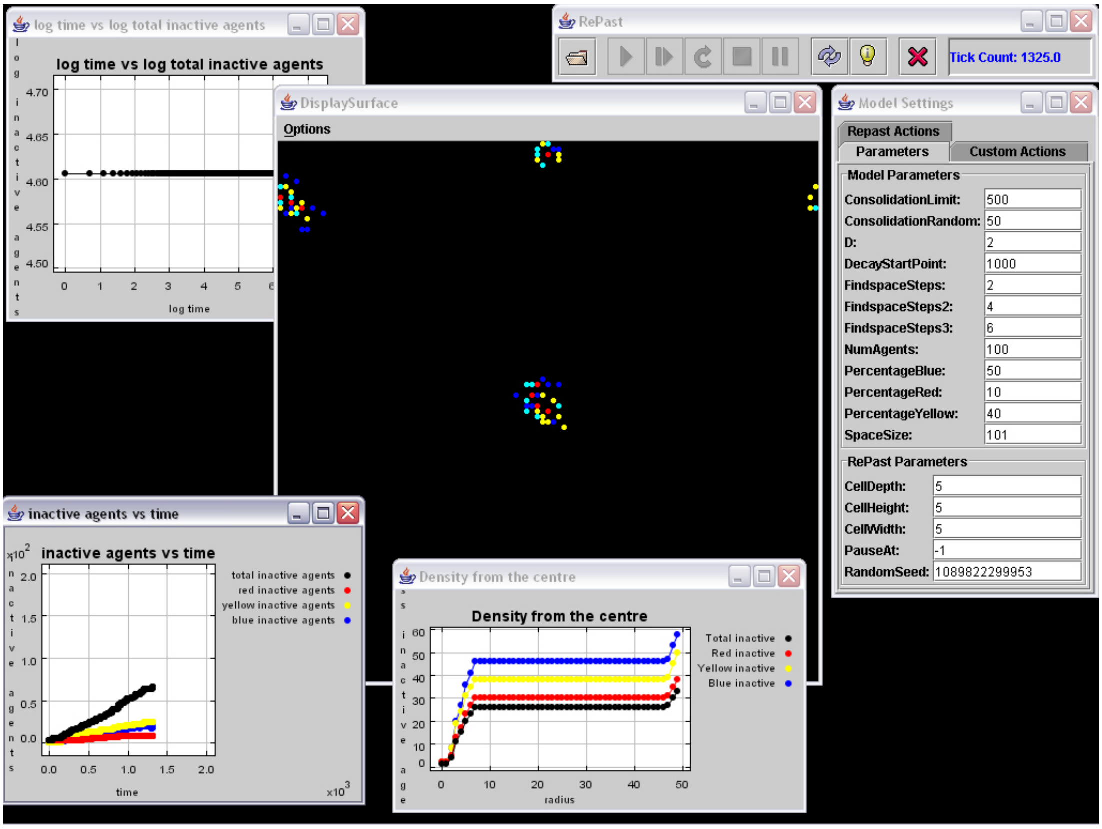


Figure 6.4 - Screenshot of Peripherisation Model.

Figure 6.5 - RePast toolbar window.

Figure 6.6 shows the display window, also visible at the centre of Figure 6.4. This window provides dynamic updates of current states of the simulation at a frequency defined by the modeller and allows the main model's behaviour to be observed, where cells states are dynamically updated and redisplayed in new colours reflecting changes in their state and occupation.

This window exhibits the space defined in the model, which can be a grid as well as a network and may present a number of different configurations according to the chosen kind of space. This choice available in RePast includes network spaces as well as a number of types of grid space like a simple two-dimensional grid, a toroidal grid, a torus, a hexagonal grid or a geographical raster space. These may contain one or multiple objects per cell (for more information see RePast documentation on http://repast.sourceforge.net).

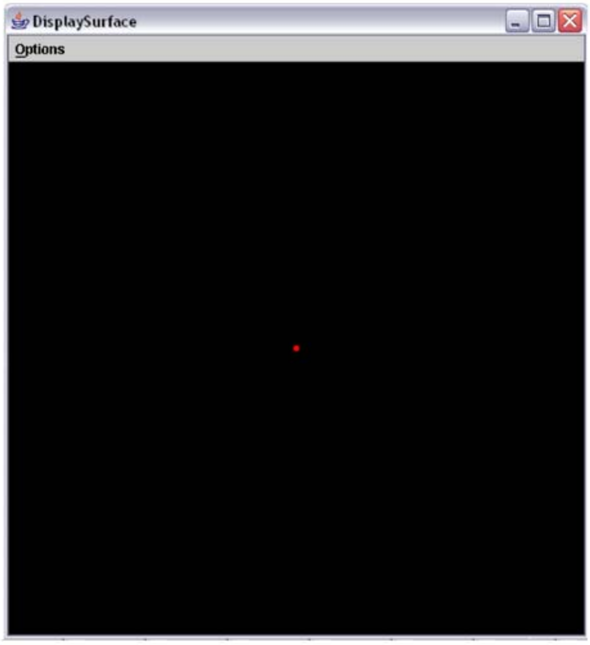


Figure 6.6 - Display window.

For the Peripherisation Model a two-dimensional torus (Multi2DTorus object) was used. This object consists of a two dimensional grid with periodic boundary conditions whose cells may contain more than one object in an undefined order. The dimensions of this grid are defined as one of the model's parameters and the cell size (grid resolution) is automatically adapted as a result of changes in the window's size.

The parameters tab shows all parameters set by the modeller as initial conditions for the model. The custom actions tab contains any sliders, buttons, or checkboxes that the simulation author has defined, which allow the user to interact with the model while it is running. The RePast Actions tab allows the user to control a series of different settings, which include making a movie and taking snapshots of the simulation run, creating a sequence chart and setting a display of custom charts, controlling the order of the parameters in the parameters tab, determining settings of the random number seed and for probing of objects, setting default parameter values, and saving current parameters to a file.

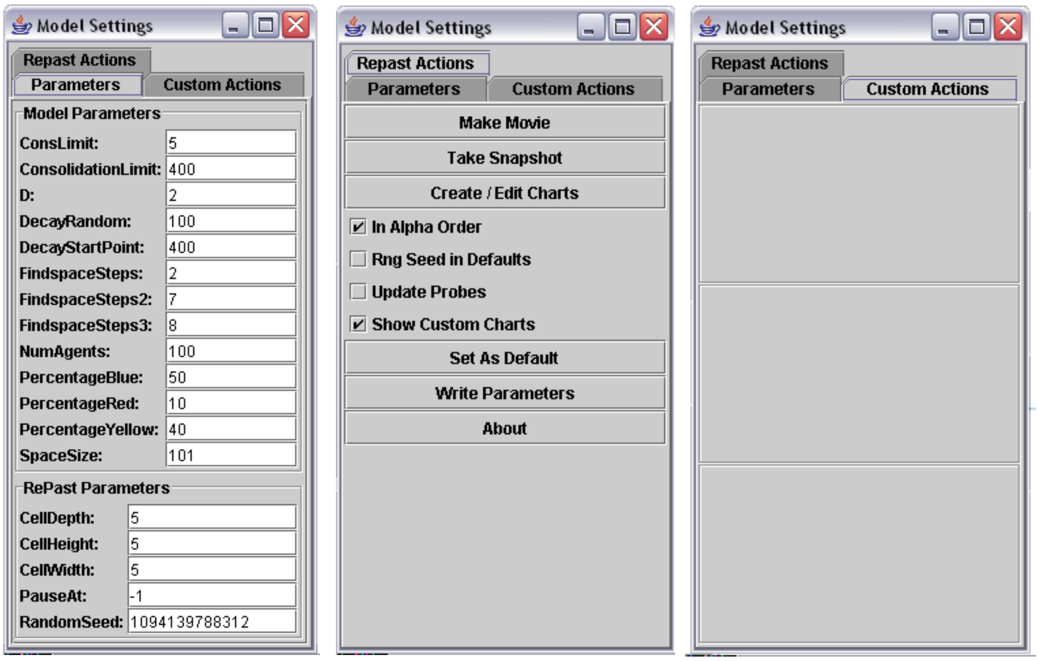


Figure 6.7 - Model settings window.

Figure 6.7 shows the model settings window for the Peripherisation Model. The first tab shows the model's parameters (detailed in section 6.1) in alphabetic order: *consLimit*, *consolidationLimit*, *d*, *decayStartPoint*, findspaceSteps (*steps*), findspaceSteps2 (*steps2*), findspaceSteps3 (*steps3*), *numAgents*, *percentageBlue*, *percentageRed*, *percentageYellow*, and *spaceSize*. The RePast actions tab is not used in the Peripherisation Model, since settings for snapshots and charts are made within the code (see Appendix E). In the same manner, the custom actions tab was not set for the Peripherisation Model, and all modules and initial conditions options are activated using Boolean variables within the code.

RePast also provides sequence graphs, histograms, and plot windows, which are optional and must be set by the modeller either in the RePast Actions window or within the code. A number of charts can be seen in Figure 6.4. Figure 6.8 shows in detail an example of a sequence graph created in RePast.

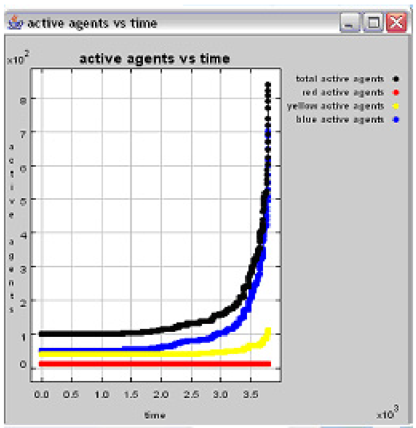


Figure 6.8 - Sequence graph produced with RePast for Peripherisation Model.

Although RePast provides means for producing charts from its GUI, RePast's development group advises that robustness of RePast statistics are still under development and charts should be used only as guides. Therefore, the plots presented in the following chapters were produced in Microsoft Excel using data exported from the simulation runs.

6.2.2 Running the Peripherisation Model

Before initiating a simulation run of the Peripherisation Model, it is necessary to set initial conditions, which include the model's parameters, features to be simulated (activate simulation modules), spatial constraints and initial seeds, as well as the desired output files, and display charts for the current simulation exercise. This can be done by setting Boolean variables in the model as true or false. It must be noted that RePast allows these actions to be set as custom actions, but they have not yet been implemented as such in the current version of the Peripherisation Model.

The initial conditions options can be seen in the figure below, and include: one seed in the centre, two seeds in random locations, two pre-positioned seeds, four seeds in random locations, four pre-positioned seeds, grid-like seeds, and path-like seeds.

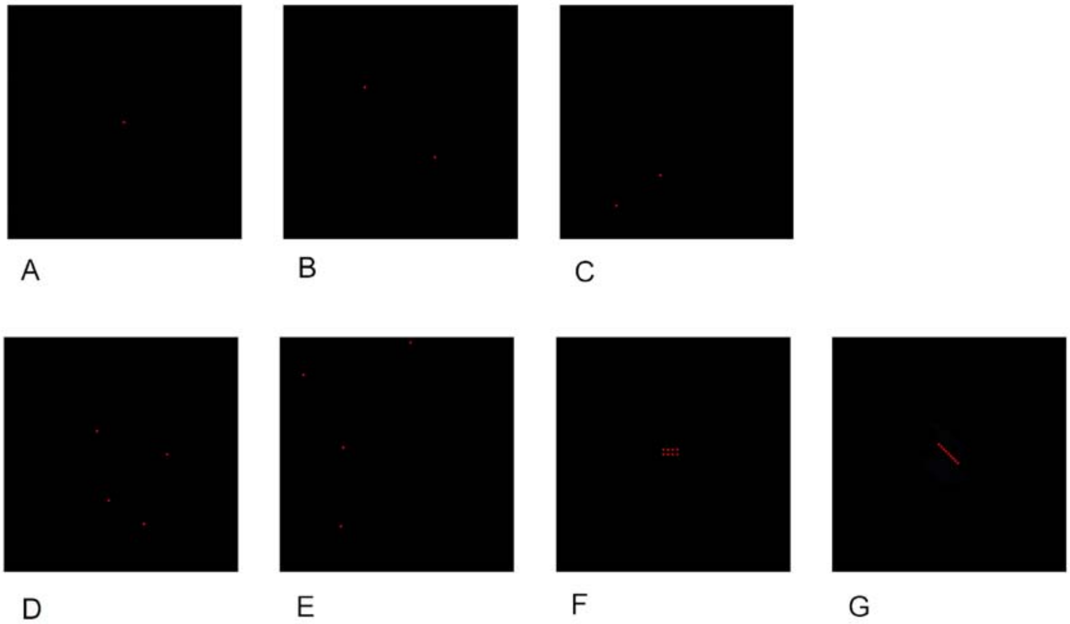


Figure 6.9 - Possible initial conditions in the Peripherisation Model.

To activate the spatial constraints module it is necessary to prepare the desired landscape condition for the simulation in advance and save it as a text-formatted file (Appendix B). For the simulation exercises presented in the next chapters, three landscape configurations were used (see Figure 7.39 and Figure 8.10 in the following chapters) and can be automatically loaded by activating the model's spatial constraints module.

The charts built into the Peripherisation program are: the number of active agents versus time, the number of occupied cells versus time, a double logarithm plot of number of occupied cells versus time, and the density from the centre. As mentioned, the robustness of the inbuilt charts is still uncertain. For that reason, to enable quantitative analysis of the model behaviour, a set of data outputs was provided (see Appendix B).

6.3 Summary

This chapter has presented the Peripherisation Model, which has been used as the basis of the work reported in the remainder of the thesis. The logic of each of the model's modules was discussed and their implementation reported and detailed. The RePast software framework was discussed and the functionalities of the Peripherisation program explored. The next chapter will explore the behaviour of the model and test each of the parameters introduced here.

Chapter 7

Sensitivity Analysis

This chapter describes a set of analyses that have been conducted to test the behaviour and outcomes of the model with different parameters. A sensitivity analysis consists of the study of the relationships between input and output of a model or, in other words, the study of the effects that changes in the parameters have on the output (Saltelli, Chan & Scott, 2000) which is, in this case, the spatial pattern.

As discussed in Chapter 5, an ABS model evaluation is not a trivial task. Grasping a whole set of aspects of an ABS model's behaviour usually requires a series of different analyses. Thus, a typical practice is to establish confidence in the results of a model through a mix of techniques, most of which contribute to both verifying and validating the model. Sensitivity analysis can provide support for the modeller's confidence in computer program correctness and model plausibility, by improving the understanding of the behaviour of a model under a range of plausible conditions (Brown et al., 2004b).

The following sections will present a series of analyses that help to improve understanding of the model's behaviour, identifying the role of each parameter and the typical behaviour of the model.

This chapter should serve to demonstrate the typical behaviour of the model and establish the relationship between inputs (parameters and initial conditions) and outputs (spatial pattern). In the next section the metrics employed to analyse each experiment will be presented. The succeeding sections describe tests conducted with each of the model's module parameters and initial conditions.

As in agent-based simulation models with stochastic processes where the results of each run are different from every other, it is necessary to conduct multiple runs in order to provide a sample from which a representative run can be selected. For each experiment shown in the next sections, a set of 10 simulation runs was conducted and a representative run was selected on the basis of the typical behaviour of the set, typical spatial results, and the average time of the simulations.

All the tests presented here are *Monte Carlo runs* as the model makes use of methods that utilize sequences of random numbers (stochastic processes) to perform the simulation, as described in the previous chapter.

7.1 Quantitative analysis

A set of simple measures was used to analyse the behaviour of the simulation model. The first set is measured along the simulation run, and plotted against time. The measures include basic metrics like the total number of active agents per economic group, number of occupied cells per economic group total, a double logarithmic plot of number of occupied cells, and the derivative in the number of occupied cells. The spatial pattern was analysed by the density of each cell class (economic group) from the centre as well as using landscape metrics. All of these metrics will be detailed below.

Number of active agents per economic group (*N*)

This number is kept constant during the simulation and the sequence chart produced shows this behaviour and indicates whether the simulation is performed correctly:

$$N_b = P_b * NumAg \qquad (01)$$

where:

N_b = Number of agents in an economic group (*b* for low-income group);

P_b = percentage of *b* or blue agents (model parameter);

NumAg = total number of agents within the simulation.

Number of occupied cells (*M*)

This metric is calculated by counting the number of cells occupied by each economic group as well as the total. The number of occupied cells metric (or mass) represents the amount of urbanization produced in each simulation:

$$M(t) = \sum_k M_k = M_b + M_y + M_r + M_c + M_g \quad (02)$$

where:

$k = b, y, r, c, g$;

M_b = Number of low-income cells;

M_y = Number of middle-income cells;

M_r = Number of high-income cells;

M_c = Number of consolidated cells;

M_g = Number of abandoned cells.

Double logarithmic plot of number of occupied cells versus time (*Log-log*)

This plots demonstrates the growth curve produced by each simulation, calculated as follows:

$$\text{Logt} = \log(\text{time}) \quad (03)$$

$$\text{Log}M = \log[M(t)]$$

Derivative of the number of occupied cells ($\frac{d}{dt}M(t)$)

This metric represents the growth rate of the simulation. The growth slope of spatial development is calculated for each time step as follows:

$$\frac{d}{dt}M(t) = \frac{M(t+\Delta t) - M(t)}{\Delta t} \quad (04)$$

where:

M = mass of occupied cells

t = time

Density from the centre (D_{rad})

This metric calculates the number of occupied cells for each value of R (radius) at the final time of the simulation run.

$$D_{rad} = \frac{M(R)}{\Pi R^2} \quad (05)$$

where:

M = number of occupied cells

R = radius within which the mass is measured

7.1.1 Landscape metrics

Most of the metrics used here were developed in *landscape ecology* (which is defined as the study of landscape patterns) including interactions among the elements of patterns, change of patterns and interactions over time, and application of these principles in the formulation and solving of real-world problems (McGarigal & Marks, 1995).

All measurements were performed using FRAGSTATS, a computer program designed to compute landscape metrics for categorical map patterns (McGarigal & Marks, 1995). FRAGSTATS computes several statistics for each cell or cell class (based here on the kind of occupation) in the landscape and for the landscape as a whole. Two main kinds of metrics are used: *landscape composition* and *landscape configuration*. Landscape composition metrics refer to features associated with the presence and amount of cell type within a given landscape, but not with the placement or location of these cells within the landscape (McGarigal & Marks, 1995). Landscape configuration metrics refer to the physical distribution or spatial character of cells within the landscape. They include measures of the placement of a cell relative to other cell types, as well as measures of the spatial character of the cells themselves (McGarigal & Marks, 1995).

For the purposes of the present investigation, a brief discussion of each of the metrics used is presented below. Three metrics were selected: perimeter-area Fractal dimension (*PAFRAC*), contagion (*CONTAG*), and interspersion and juxtaposition index (*IJI*), all of which quantify the landscape configuration. All selected metrics measure aggregate properties of the entire landscape, and not properties of each class (type) or patch. In the landscape metric calculations, all cells are considered to be 100m^2. A complete detailed explanation of landscape metrics and their use can be found in McGarigal & Marks (1995) and Turner (2001).

Perimeter-Area Fractal Dimension (*PAFRAC*)

PAFRAC measures the extent to which patches fill a landscape. Its value ranges between 1 and 2. As such, a fractal dimension greater than 1 indicates an increase in patch shape complexity. *PAFRAC* values approach 1 for shapes with very simple perimeters such as squares, and approach 2 for shapes with highly convoluted, plane-filling perimeters.

PAFRAC is calculated according to the following formula:

$$PAFRAC = \frac{2}{\dfrac{\left[N\sum_{i=1}^{m}\sum_{j=1}^{n}\left(\ln p_{ij} \bullet \log a_{ij}\right)\right] - \left[\left(\sum_{i=1}^{m}\sum_{j=1}^{n}\log p_{ij}\right)\left(\sum_{i=1}^{m}\sum_{j=1}^{n}\log a_{ij}\right)\right]}{\left(N\sum_{i=1}^{m}\sum_{j=1}^{n}\log p_{ij}^{2}\right) - \left(\sum_{i=1}^{m}\sum_{j=1}^{n}\log p_{ij}\right)^{2}}} \qquad (06)$$

where:

a_{ij} = area (m^2) of patch ij.

p_{ij} = perimeter (m) of patch ij.

N = total number of patches in the landscape.

Contagion (*CONTAG*)

The contagion index consists in the probability that two randomly chosen adjacent cells belong to two given cell types. The index thus measures both patch type interspersion (i.e., the intermixing of units of different patch types) as well as patch dispersion (i.e., the spatial distribution of a patch type). It measures dispersion in addition to patch type interspersion because cells, not patches, are evaluated for adjacency. It measures the extent to which landscape elements (patch types) are aggregated or clumped (i.e., dispersion). The calculation of this index is according to the formula:

$$CONTAG = \left[1 + \frac{\sum_{i=1}^{m} \sum_{k=1}^{m} \left[(P_i) \left(\frac{g_{ik}}{\sum_{k=1}^{m} g_{ik}} \right) \right] \bullet \left[\log (P_i) \left(\frac{g_{ik}}{\sum_{k=1}^{m} g_{ik}} \right) \right]}{2 \log (m)} \right] (100) \qquad (07)$$

where:

P_i = proportion of the landscape occupied by patch type (class) i

g_{ik} = number of adjacencies (joins) between pixels of patch types (classes) i and k based on the *double-count* method

m = number of patch types (classes) present in the landscape.

Higher values of contagion may result from landscapes with a few large, contiguous patches, whereas lower values generally characterize landscapes with many small and dispersed patches. The contagion index represents the observed level of contagion as a percentage of the maximum possible, given the total number of patch types.

Interspersion and Juxtaposition Index (*IJI*)

Each patch is evaluated for adjacency with all other patch types. Because *IJI* is based on patch adjacencies (it measures the adjacency of a group of cells with the same type and not isolated cells), its results differ from the contagion index. It measures the extent to which patch types are interspersed (not necessarily dispersed), according to the following formula:

$$IJI = \frac{-\sum_{i=1}^{m}\sum_{k=i+1}^{m}\left[\left(\frac{e_{ik}}{E}\right)\bullet \log\left(\frac{e_{ik}}{E}\right)\right]}{\log\left(0.5\left[m(m-1)\right]\right)}(100) \qquad (08)$$

where:

e_{ik} = total length (m) of edge in landscape between patch types (classes) i and k

E = total length (m) of edge in landscape, excluding background

m = number of patch types (classes) present in the landscape.

For *IJI*, higher values correspond to landscapes in which patch types are well interspersed (i.e. equally adjacent to each other), and low values occur in landscapes where patch types are poorly interspersed (i.e., disproportionate distribution of patch type adjacencies). Like the contagion index, the interspersion index is a relative index that represents the observed level of interspersion as a percentage of the maximum possible, given the total number of patch types. It approaches zero when the corresponding patch type is adjacent to only one other patch type, and the number of patch types increases. *IJI* = 100 when the corresponding patch type is equally adjacent to all other patch types (i.e. maximally interspersed and juxtaposed to other patch types).

7.2 Analysing the model's typical behaviour

A set of 10 simulation exercises was conducted to capture the typical behaviour of the model. For these exercises, the initial conditions were a single seed in the centre of the grid, and the parameters were fixed as follows: *steps* = 2, *proportion of agents per economic group* 10% high-income, 40% middle-income, 60% low-income.

The ten runs were conducted in sequence in order to guarantee that all behavioural differences were illustrated. In what follows these runs will be presented and their results analysed.

7.2.1 Sequence of snapshots

Figure 7.1 shows each of the ten simulation runs (A to J) in its development process over time. It is interesting to note that the same set of parameters produce different paths of development, and the distinction between them is due to operation of randomness in the model. The model's rules (described in section 6.1) determine that the speed of development is greatly dependent on the probability of high-income agents becoming settled. In other words, the more high-income agents settle at the beginning of the simulation, the faster the overall spatial development will be. This is because the settling rule of all agents is only initiated when agents 'find' a high-income cell. Section 7.3 discusses other factors which impact on the simulation time-scale.

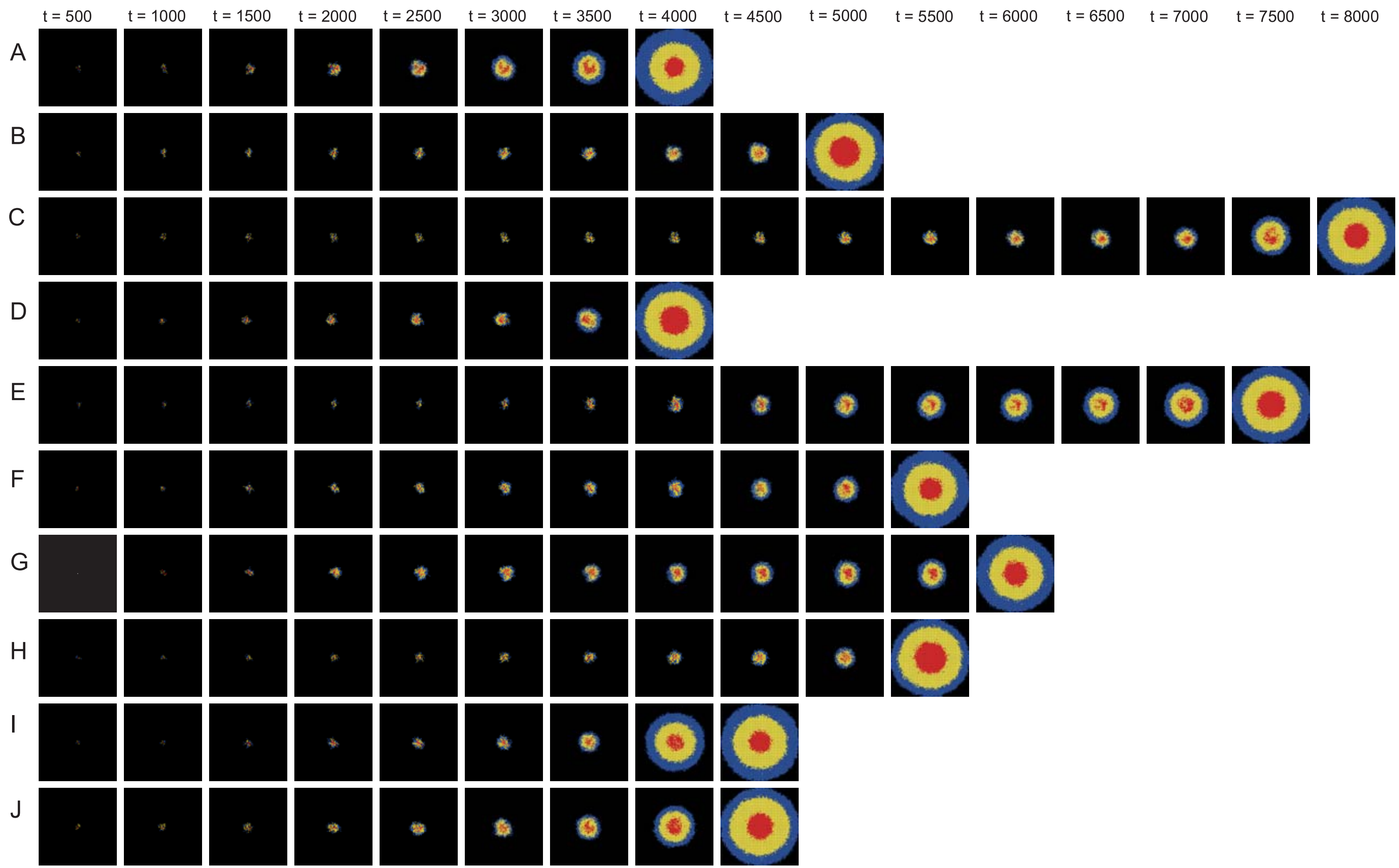


Figure 7.1 - Sequences of snapshots, set of representative runs (A to J).

7.2.2 Number of active agents

The number of active agents is kept constant during the simulation. At each time step, the program checks how many agents have settled and creates new agents to replace them. As all tests here were run with the same initial number of agents, they present a constant number of active agents, as can be observed in Figure 7.2 below.

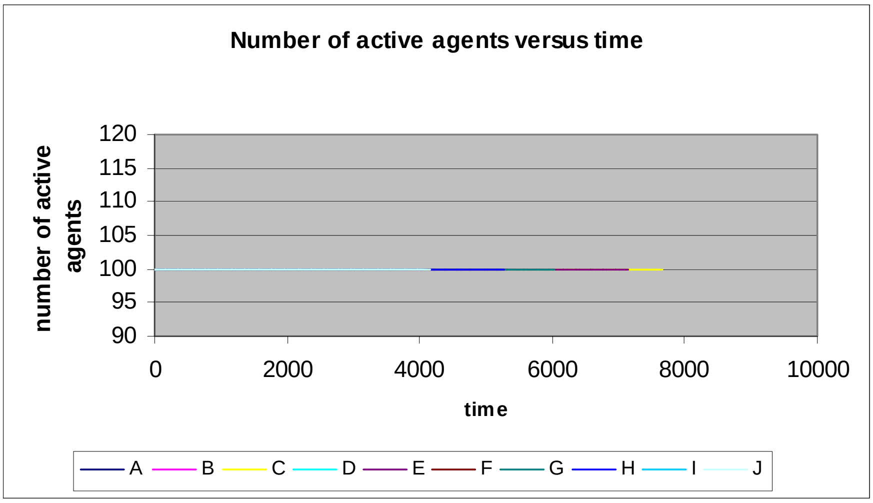


Figure 7.2 – Graph of total number of active agents versus time for a set of representative runs (A to J).

7.2.3 Number of occupied cells

As the simulation runs, the number of occupied cells increases, up to the point where all cells are occupied. At this stage of the simulation, cells occupied by lower-income groups start being replaced by a higher-income group, and if the simulation is kept running, there is a tendency for all of them to be replaced. In other words, if the simulation was run indefinitely, the final spatial pattern would be mainly composed of high-income group cells. This is because the rules of the Peripherisation Model allow upper income groups to settle on cells occupied by lower income groups. This fact, together with a limited grid space, produces a spatial pattern where low-income cells tend to disappear first, and then middle-income tend to disappear as well. In all the simulation exercises presented here, this was not allowed to happen. All simulation exercises were stopped as the spatial pattern reached the grid borders, whatever the grid size in question. In all tests presented in the next sections, the final number of occupied cells was fixed as 2000 cells for grid size 51 x 51 cells; 8000 cells for grid size 101 x 101 cells; and 30500 cells for grid size 201 x 201 cells.

The chart below shows the number of occupied cells versus time, essentially demonstrating differences in the time-scale of simulation runs performed with the exact same set of initial conditions and parameters.

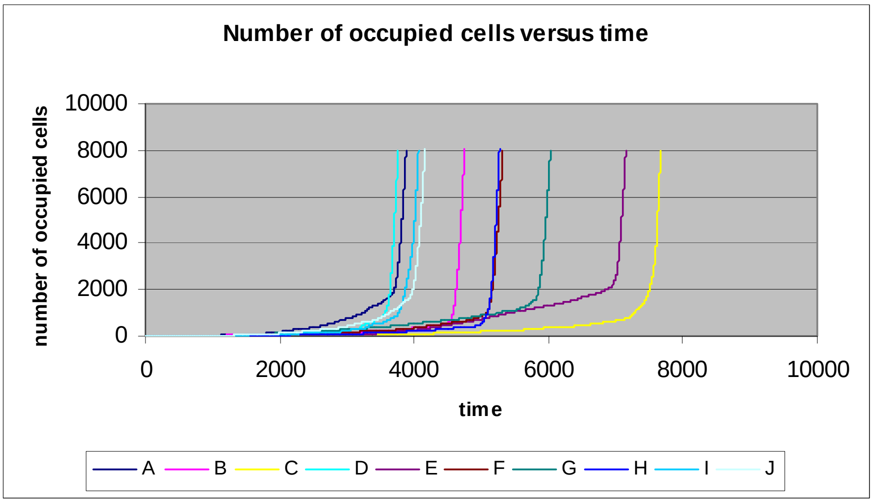


Figure 7.3 - Graph of the total number of occupied cells versus time for a set of representative runs (A to J).

It is clear from Figure 7.3 that the same number of occupied cells is achieved at very different points in time for each of these simulation runs. It is also interesting to observe that the points at which the number of occupied cells takes off, that is, the start points of each curve in the graph, are also on different time scales. The period of time between zero and the start point of the curve consists in the time taken for agents to walk randomly until they find high-income cells. This is part of the stochastic process of the model and it is therefore comprehensible in that it fluctuates considerably.

The next figures show the number of occupied cells by economic group. It is interesting to note that although the final number of occupied cells is constant, the number of occupied cells for each economic group varies significantly.

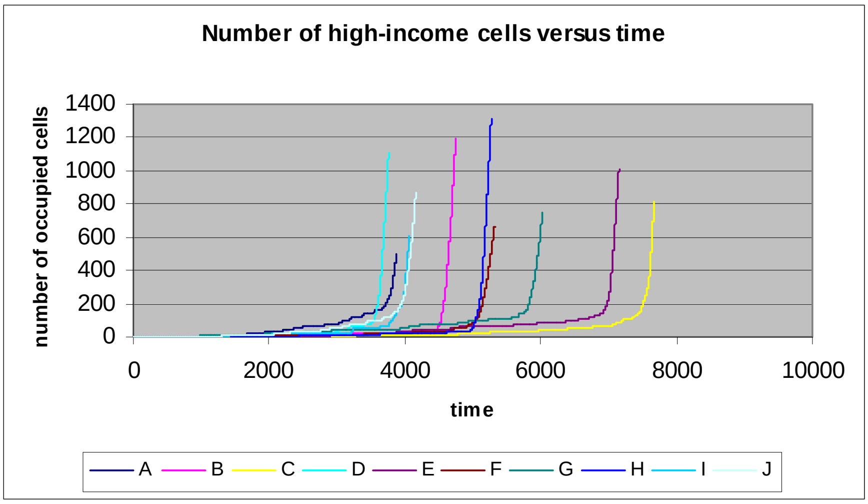


Figure 7.4 - Graph of the number of high-income cells versus time for a set of representative runs (A to J).

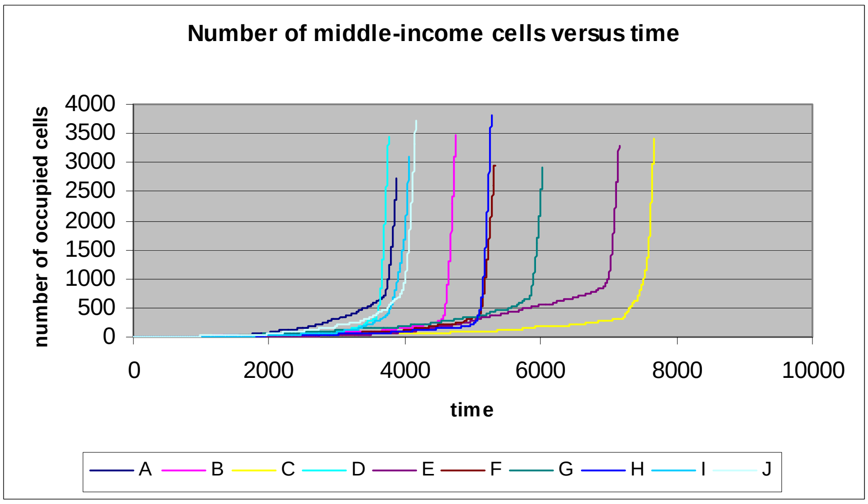


Figure 7.5 - Graph of the number of medium-income cells versus time for a set of representative runs (A to J).

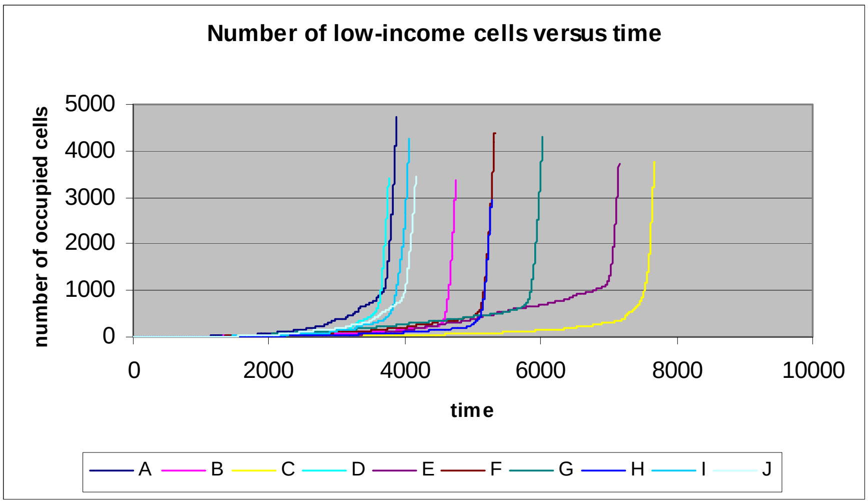


Figure 7.6 - Graph of the number of low-income cells versus time for a set of representative runs (A to J).

7.2.4 Log-log occupied cells versus time (growth curve)

The log-log graph is presented here to demonstrate that, despite the fluctuation in the time-scale of simulations observed in the charts above, the behaviour of the simulation runs is generally consistent. This can be clearly observed in Figure 7.7, where, for each simulation run, the logarithm of the total number of occupied cells is plotted against the logarithm of time. The chart shows ten similar curves, which demonstrates that the growth curves for all simulation runs present the same characteristics and are statistically the same. An even more effective way of showing this would be to collapse the time intervals for each growth curve onto an identical overall time period, although Figure 7.7 is sufficient to indicate the statistical similarity.

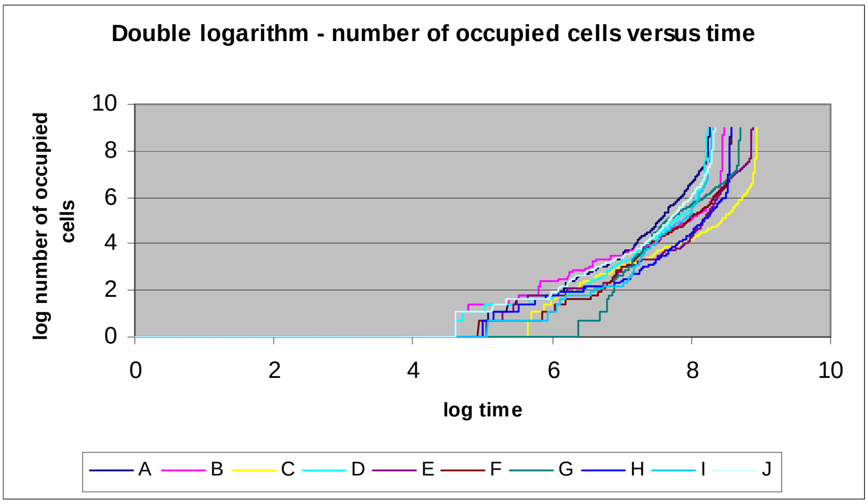


Figure 7.7 - Log-log graph of number of occupied cells versus time for a set of representative runs (A to J).

7.2.5 Derivative (growth rate curve)

While the double logarithm plot demonstrates the growth curve, the derivative shows the rate of growth for the simulation runs. Two charts are presented here: a simple chart and a double logarithm graph for the same values, where both axes are on a logarithmic scale.

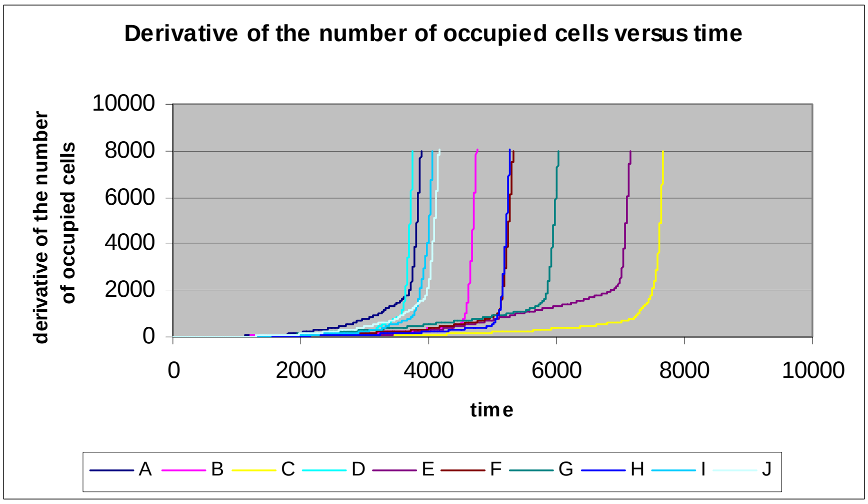


Figure 7.8 - Derivative of the number of occupied cells versus time for a set of representative runs (A to J).

The chart in Figure 7.8 shows that growth starts very slowly and at a certain point it increases significantly and remains so until the end of simulation. The same trend can be observed in Figure 7.9, where all curves present similar changing points.

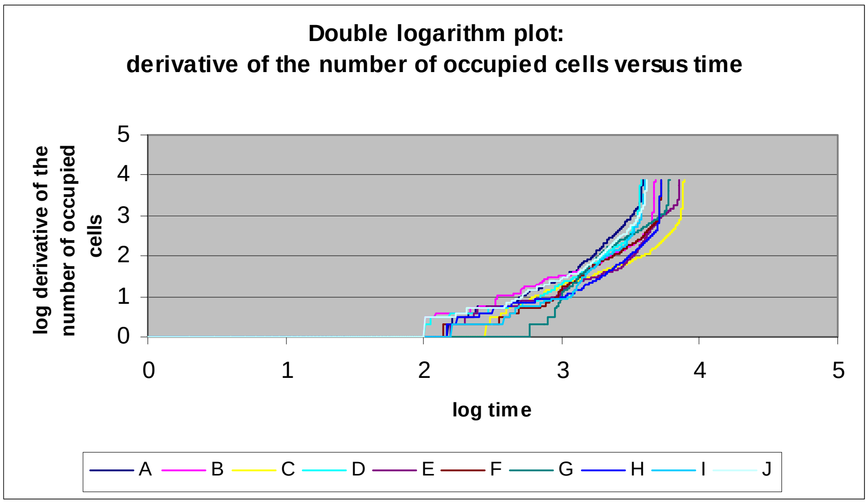


Figure 7.9 - Double logarithm plot of derivatives of the number of occupied cells versus time for a set of representative runs (A to J).

7.2.6 Density from the centre

As discussed in the foregoing, this measure is directly related to the spatial pattern produced by the simulation model. It is interesting to note that although the simulation runs present many differences in time-scale, the density graph is precisely the same for all runs, as can be observed in Figure 7.10. This means that the final results for the spatial configuration are statistically identical.

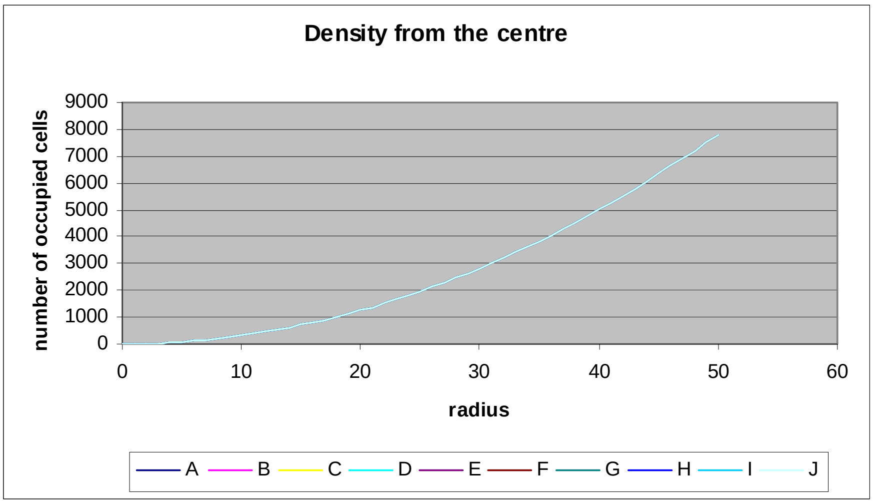


Figure 7.10 - Total density of inactive agents per radius for a set of representative runs (A to J).

The charts in the following figures show that the number of occupied cells differs from one simulation run to another. Thus, the graphs display variations in the curves related to this aspect only.

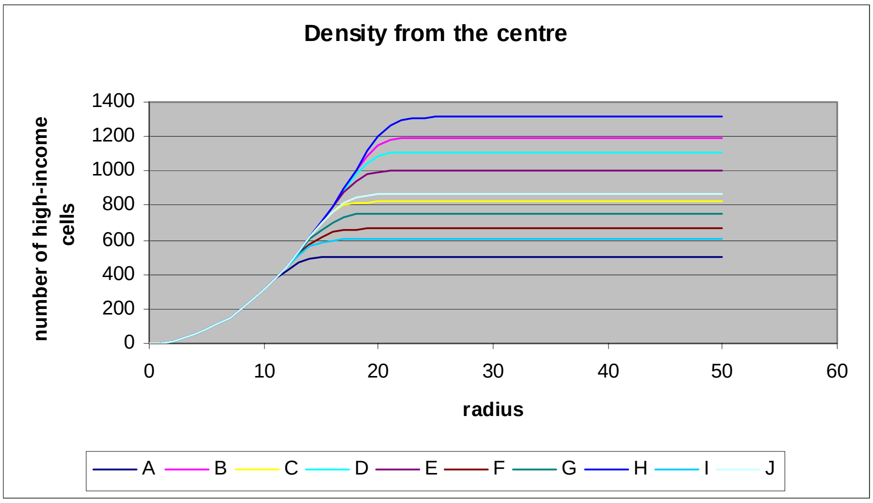


Figure 7.11 - Density of high-income cells per radius for a set of representative runs (A to J).

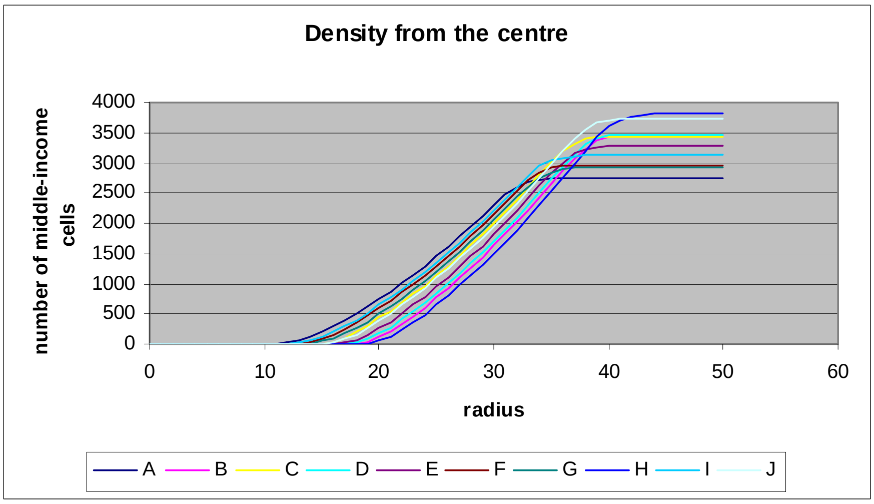


Figure 7.12 - Density of medium-income cells per radius for a set of representative runs (A to J).

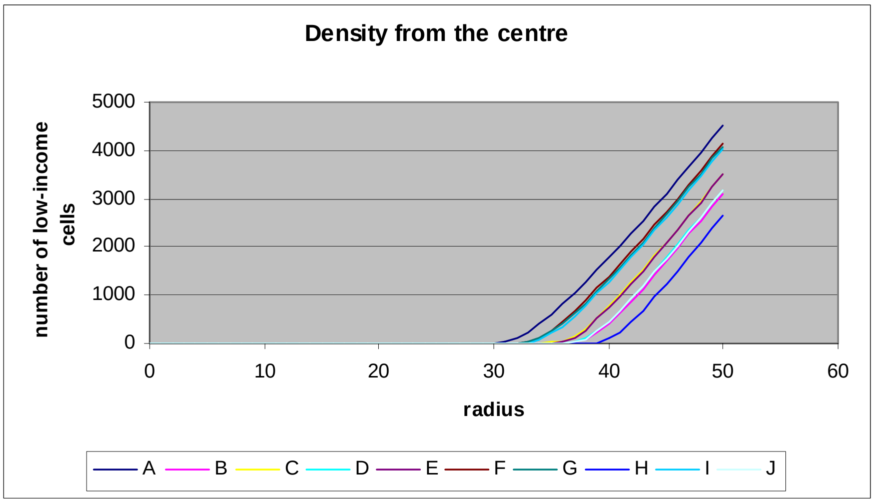


Figure 7.13 - Density of low-income cells per radius for a set of representative runs (A to J).

7.3 Impact of change in grid size and number of agents

The main impact of variations of grid size and number of agents is on simulation speed (time-scale). To show simulation performance with different grid sizes and number of agents, a set of tests was conducted keeping parameters fixed and using three different grid sizes: 51 by 51 cells, 101 by 101 cells, and 201 by 201 cells. The same kinds of tests were carried out for each number of agents, examining the behaviour of the model with three different values for a total number of agents of 50, 100 and 200.

All tests were conducted using a single seed as initial condition, *steps* equal to 2, *proportion of agents per economic group* 10% high-income, 40% middle-income, and 50% low-income. For tests with grid size, the number of agents was fixed to 100; and for tests with number of agents, grid size was fixed at 101 by 101 cells.

Figure 7.14 shows three sequences of snapshots, where sequence A represents tests with grid size 51 by 51 cells, sequence B grid size 101 by 101 cells, and sequence C grid size 201 by 201 cells. It must be noted that while snapshots in sequences A and B have intervals equal to 500 iterations, sequence C snapshots represent the state of the simulation every 2000 iterations. This means that the time-scale is quite different for all three sequences and that grid size has a major impact on the simulation time-scale.

Figure 7.15 presents representative tests with the total number of agents. Test D was conducted using 50 agents, test E using 100 agents, and test F using 200 agents. In this figure, sequence D presents snapshots for every 1000 iterations while sequences E and F show snapshots for every 500 iterations. As for tests with grid size, the number of agents has a significant impact on simulation time-scale. This is confirmed by the chart presented in Figure 7.16, which displays the total number of occupied cells for each of the six sequences (A-F).

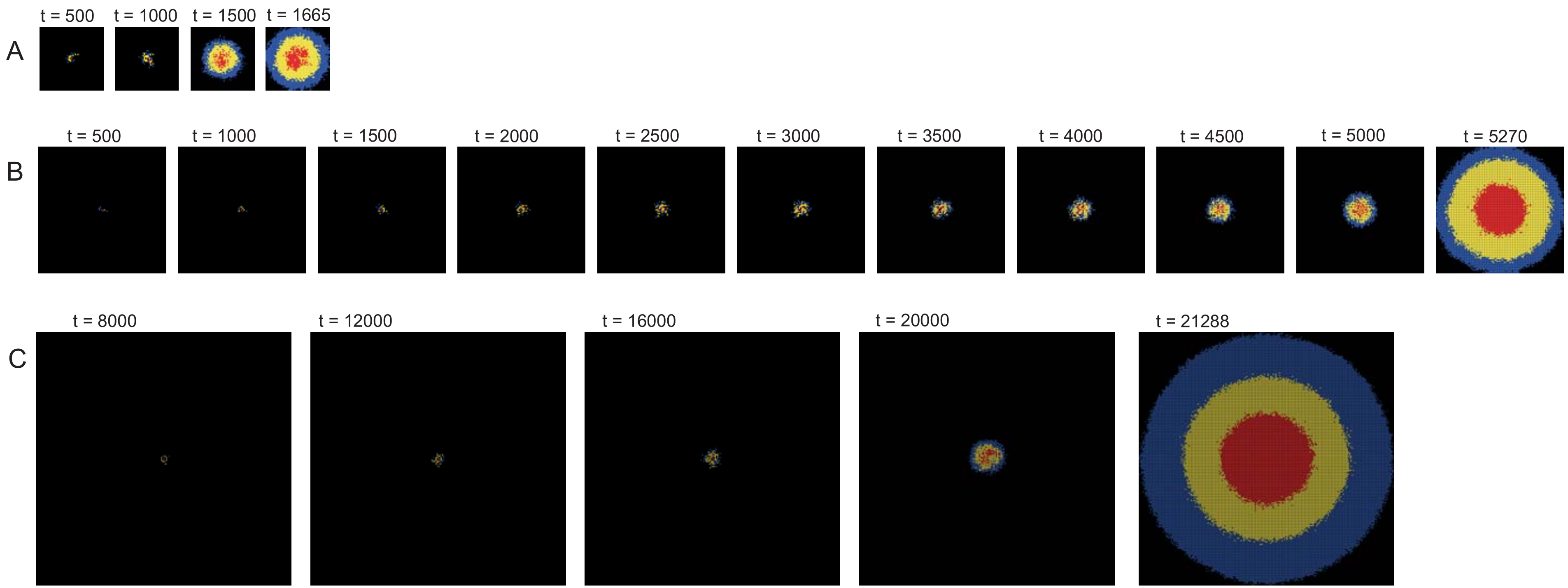


Figure 7.14 - Sequences of snapshots testing grid size, where sequence A has a grid size 51 by 51 cells, sequence B has a grid size 101 by 101 cells, and sequence C has a grid size 201 by 201 cells.

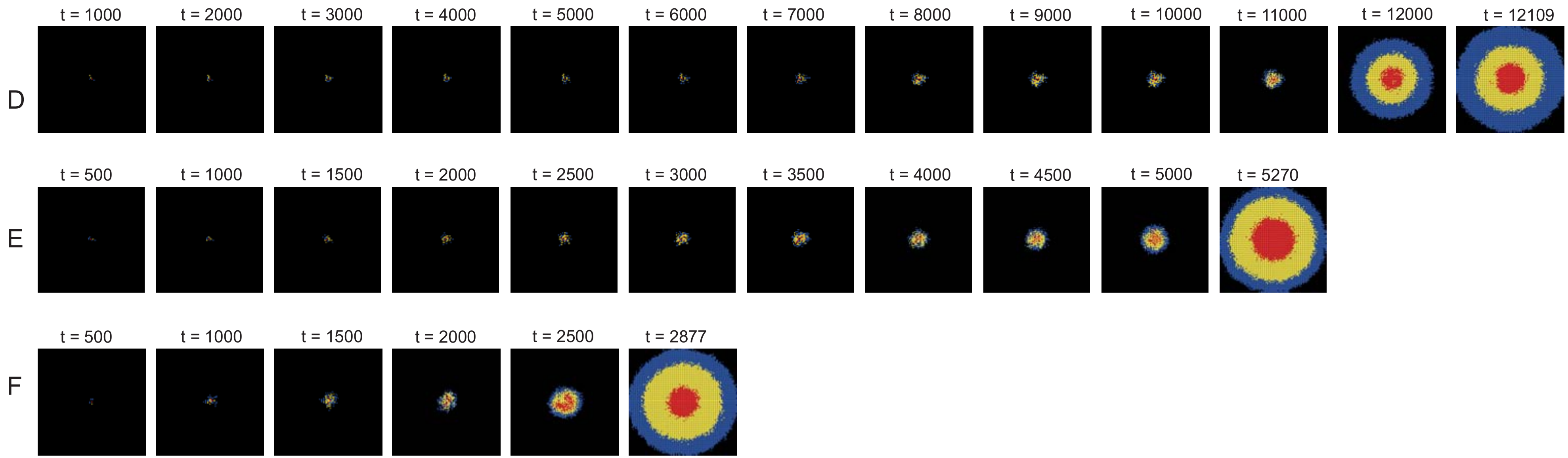


Figure 7.15 - Sequences of snapshots testing total number of agents, where sequence D has 50 agents, sequence E as 100 agents, and sequence F has 200 agents.

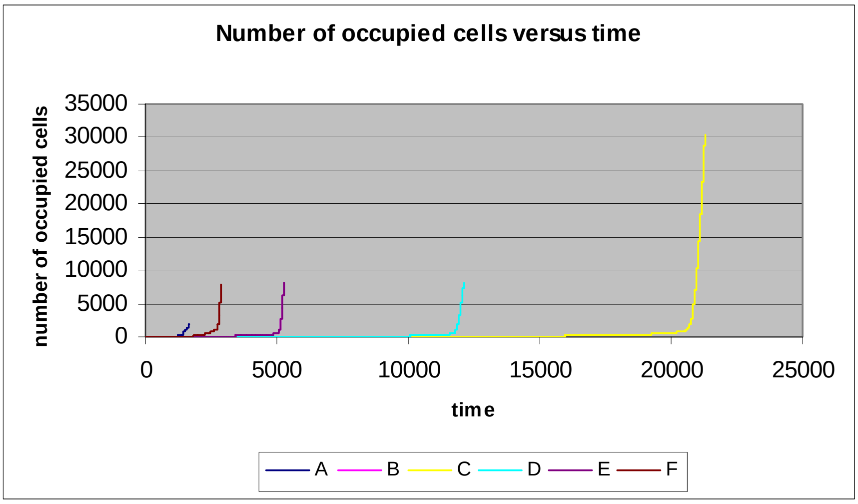


Figure 7.16 - Chart showing tests with grid size and number of agents, where sequences A to C have a fixed number of agents equal to 100 and change in grid size (L x L), respectively L = 51, 101 and 201 cells. Sequences D to E have fixed grid size L = 101 cells, and changed number of agents, respectively 50, 100 and 200 agents.

The result of the tests with grid size and total number of agents is a striking one. It was known that the simulation speed depends largely on two aspects: time that agents take to start the settling process (until they reach high-income seeds); and the probability that high-income agents will reach the seeds and will settle. The faster high-income agents initiate the settlement process, the faster the whole simulation will run.

On this basis, it seemed, intuitively, that this probability was directly related to the density of agents in the grid space, i.e., the more agents per cell space, the larger the probability of an agent finding the seed cell. The tests revealed that this is only partially true and demonstrate that, although the density of agents plays a central role in determining the simulation speed, the time-scale is greatly affected by the stochastic process and path dependence effects.

7.4 Tests with parameters and initial conditions

The next sections will present tests with parameters and initial conditions for each of the model's modules. It is important to note that each parameter is analysed presenting only charts relevant for understanding the general behaviour of each parameter, and that the aspects discussed for each parameter are those that differ from the model's typical behaviour presented in Section 7.2.

7.4.1 Module one: Peripherisation

As seen in the foregoing, the Peripherisation module simulates the main mechanism of the peripherisation process. Two main parameters will be analysed in the following sections: *steps* and *proportion of agents per economic group*. The third set of tests considers the impact of changes in initial conditions, in terms of the initial seeds used for the simulation.

The parameter *steps* is responsible for the degree of fragmentation in the final spatial pattern. As demonstrated in Chapter 6, this parameter determines the number of steps (cells) an agent will walk when searching for a place to settle. The *proportion of agents per economic group* is one of the key parameters in the Peripherisation Model and defines the percentage of agents in each economic group. Initial conditions can vary according to the number and position of 'seeds', that is, cells that allow the settling process to be initiated. The impact of changes in initial settings and parameters in the behaviour and spatial result of the simulation will be tested and analysed in the following.

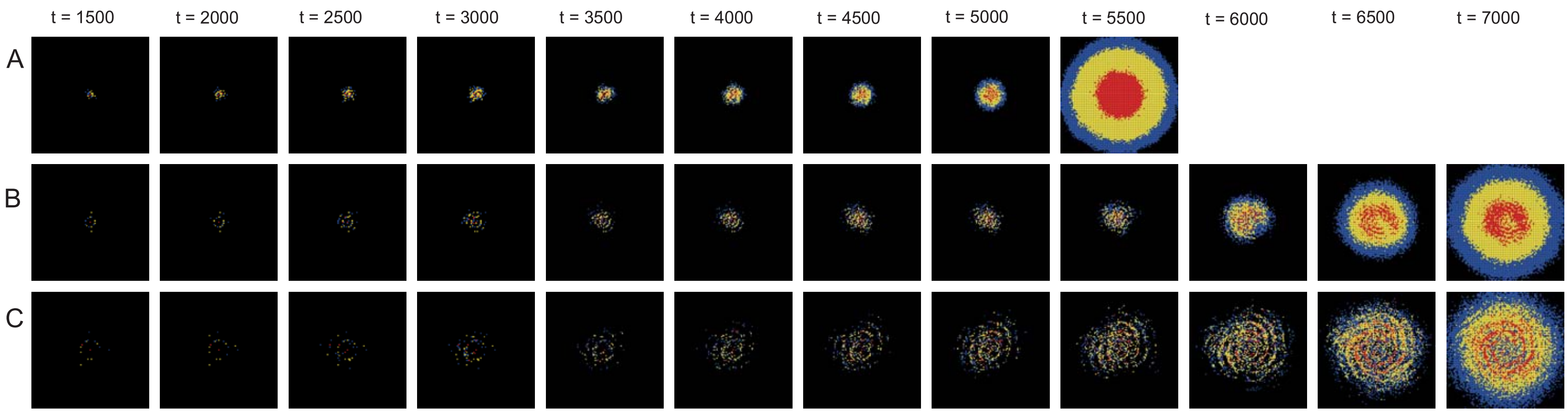


Figure 7.17 - Sequences of snapshots testing parameter *steps*, where sequence A uses *steps* = 2, sequence B uses *steps* = 4, and sequence C uses *steps* = 8 .

7.4.1.1 *steps*

Because *steps* is a parameter that affects directly the configuration of the resultant spatial pattern, it is important not only to analyse its impacts on the general behaviour of the model but also to evaluate the impact it has on the spatial pattern itself. This can be easily observed in the set of snapshots shown in Figure 7.17, where it is clear that the higher the value of *steps* is, the more fragmented the landscape looks. High values for the parameter *steps* also produce regularities in the landscape, as can be observed in sequence C (Figure 7.17). In this sequence of snapshots, the space interval between cells of the same income group can be visually identified, and the spatial pattern seems slightly 'spiral-shaped'. The impact of parameter *steps* in the spatial pattern is quantitatively demonstrated in Table 7.1, which shows the landscape metric values calculated for each value of the parameter tested.

	PAFRAC	*CONTAG*	*IJI* (%)
steps = 2	1.3950	39.3278	59.4799
steps = 4	1.5579	37.9115	63.9479
steps = 8	1.6809	20.5978	77.4128

Table 7.1 – Landscape metrics results for tests with *steps*.

The resultant values of landscape metrics presented in Table 7.1 are better understood if analysed together with the spatial patterns presented in Figure 7.17. It is possible to observe that as the value of *steps* increase, the dispersion and heterogeneity in the spatial pattern are also greater. This is consistent with the results of landscape metrics that show that as the value for *steps* increase, so do the values for *PAFRAC* and *IJI*. The increase in *PAFRAC* values indicates an increase in patch shape complexity in the spatial pattern. The increase of *IJI* values means that the distribution of patch type adjacencies becomes more even, as the values of *steps* increase. Finally, the decrease of *CONTAG* values is also consistent with the observed spatial results, as it indicates that patches become more fragmented, i.e. smaller and more dispersed.

The values of *steps* seem to have some impact on the time-scale of the simulation. The time-scale for tests performed with *steps* equal to 2 ranged from 3760 to 7670 time-ticks, with a mean value of 5204.4 and a median equal to 5009. Tests performed with *steps* equal to 4 had a time-scale from 3602 to 9325, and values of mean and median were slightly higher, 6772.5 and 6716.5 respectively. Tests with *steps* equal to 8 do not differ much from previous values, presenting time-scales ranging from 4782 to 8259, a mean equal to 6702.3, and a median equal to 6875. Figure 7.18 shows two charts that demonstrate the behaviour of the simulation in time using representative runs.

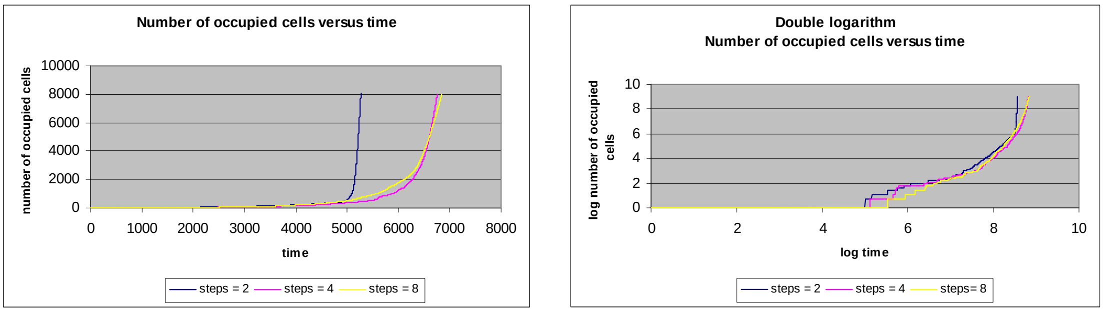


Figure 7.18 - Charts testing the *steps* parameter, where sequence A uses *steps* = 2, sequence B *steps* = 4, and sequence C *steps* = 8.

There seems to be a slight tendency for larger *steps* values to result in a faster simulation run. This is due to the fact that the larger the step, the more empty spaces are left between patches, making the search easier, that is, the agents find an appropriate place to settle faster. This does not seem to affect the time-scale starting from a certain threshold, which could explain the similarities in time-scales from simulation runs with *steps* equal to 4 to runs where *steps* is equal to 8, for instance.

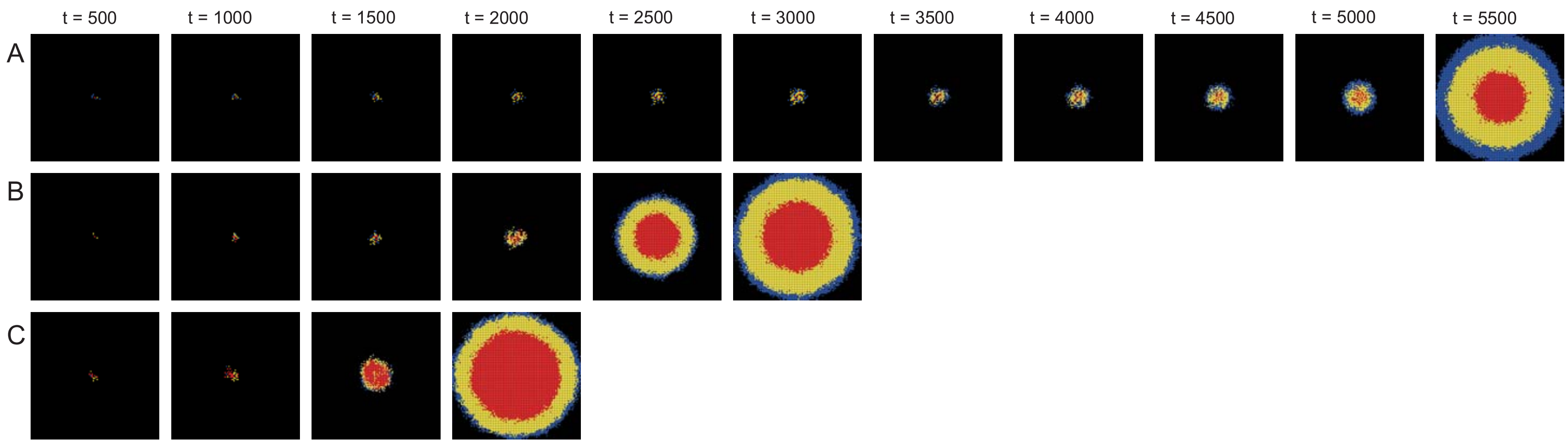


Figure 7.19 - Sequences of snapshots testing parameter *proportion of agents per economic group,* where sequence A uses ratio 10, 40, 50, sequence B uses 25, 50, 25, and sequence C 50, 40, 10.

7.4.1.2 Proportion of agents per economic group

While *steps* impacts mainly on the spatial pattern results of the simulation, *proportion of agent per economic group* has a major impact on the behaviour of the simulation.

Figure 7.19 shows the sequence of snapshots for three different sets of values for the *proportion of agents per economic group* parameter, respectively, 10% high-income, 40% middle-income, and 50% high-income (sequence A in Figure 7.19), 25% high-income, 50% middle-income, and 25% high-income (sequence B), and 50% high-income, 40% middle-income, and 10% high-income (sequence C). The tests presented here use representative runs.

From the snapshots in Figure 7.19, it is possible to observe the difference in the time-scale for each set of parameter settings. Sequence A clearly presents a longer time-scale than the other two, and sequence B has a longer time-scale than sequence C. This can be explained by the fact that different economic groups develop spatially at different speeds. Because there are fewer high-income agents in test A, high-income agents tend to settle faster than the other two groups since they can settle anywhere except in cells already occupied by the high-income group. In addition, the more cells occupied by the high-income group there are, the faster the other agents will settle.

Figure 7.20 shows the number of occupied cells in time, both for the total and per economic group. The number of occupied cells per economic group reveals that, despite the final number of occupied cells being the same for all three tests, the simulation runs generate great variations in the number of occupied cells per economic group.

It might be expected that the number of occupied cells per economic group would be somehow proportional to the proportion set by the parameter. Tests with representative runs shown in charts in Figure 7.20 demonstrate that this is not always the case. Although the tendency of the model is to follow the same proportion defined by the parameters, the exact number of cells occupied by each economic group will also be affected by the stochastic process and path dependence present in the model.

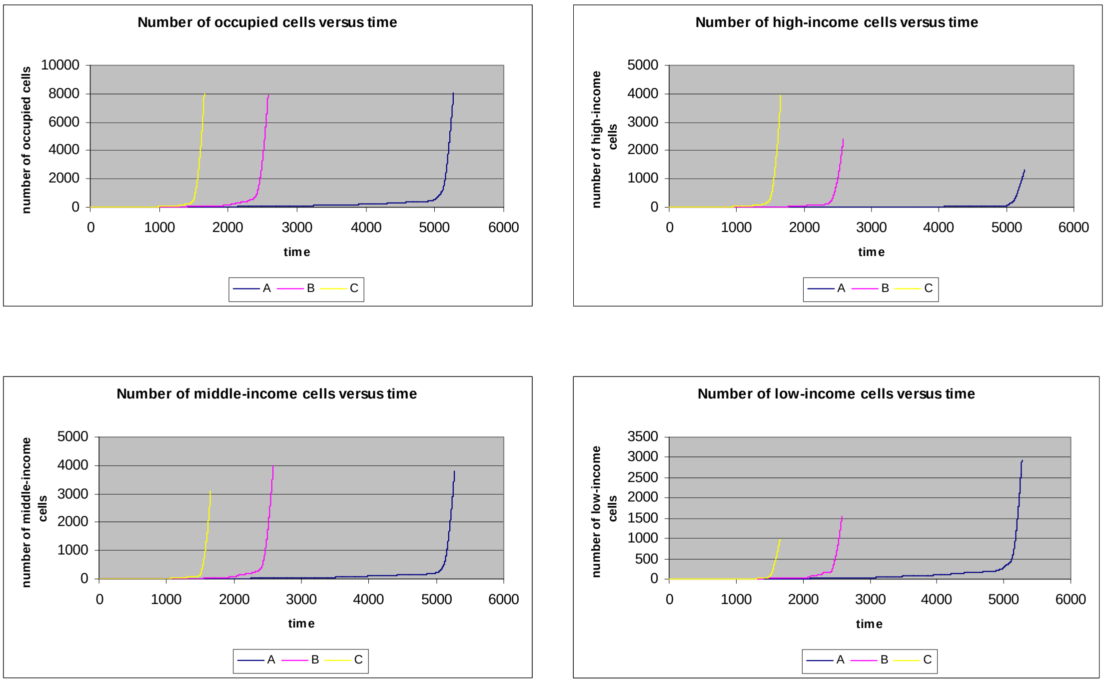


Figure 7.20 - Charts testing the proportion of agents per economic group, where sequence A uses ratio 10, 40, 50, sequence B 25, 50, 25, and sequence C 50, 40, 10.

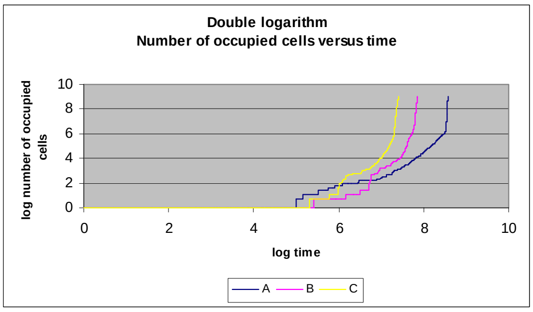


Figure 7.21 - Double logarithmic plot testing the proportion of agents per economic group, where sequence A uses ratio 10, 40, 50, sequence B 25, 50, 25, and sequence C 50, 40, 10.

7.4.1.3 Initial conditions

Tests with different initial conditions were carried out to explore their impact on the behaviour and spatial results of the model. All experiments use fixed parameters (*steps* = 2 and *proportion of agents per economic group* = 10 40 50). Figure 7.23 shows some of these experiments. Experiment A is the classical case with a central seed as the initial condition. Experiments B and C explore the idea of multiple seeds. This is a proxy for the case of a metropolitan area which results from the combination of several cities or villages that end up as a single spatial area because of their proximity. Sequence D presents as an initial condition an attempt to replicate a typical regular grid of colonial Portuguese and Spanish cities in Latin America. Finally, the initial condition of the sequence E represents development along a path or a road, which is also a very common situation in real urban development.

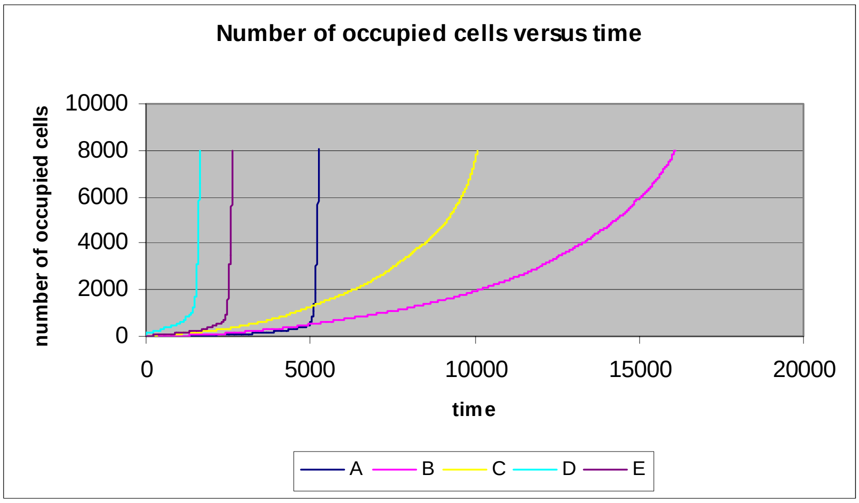


Figure 7.22 - Chart of number of occupied cells versus time testing initial conditions, where sequence A uses a single central seed, sequence B and sequence C use multiple seeds, sequence D uses proxy for a regular grid, and sequence E a proxy for a path or road.

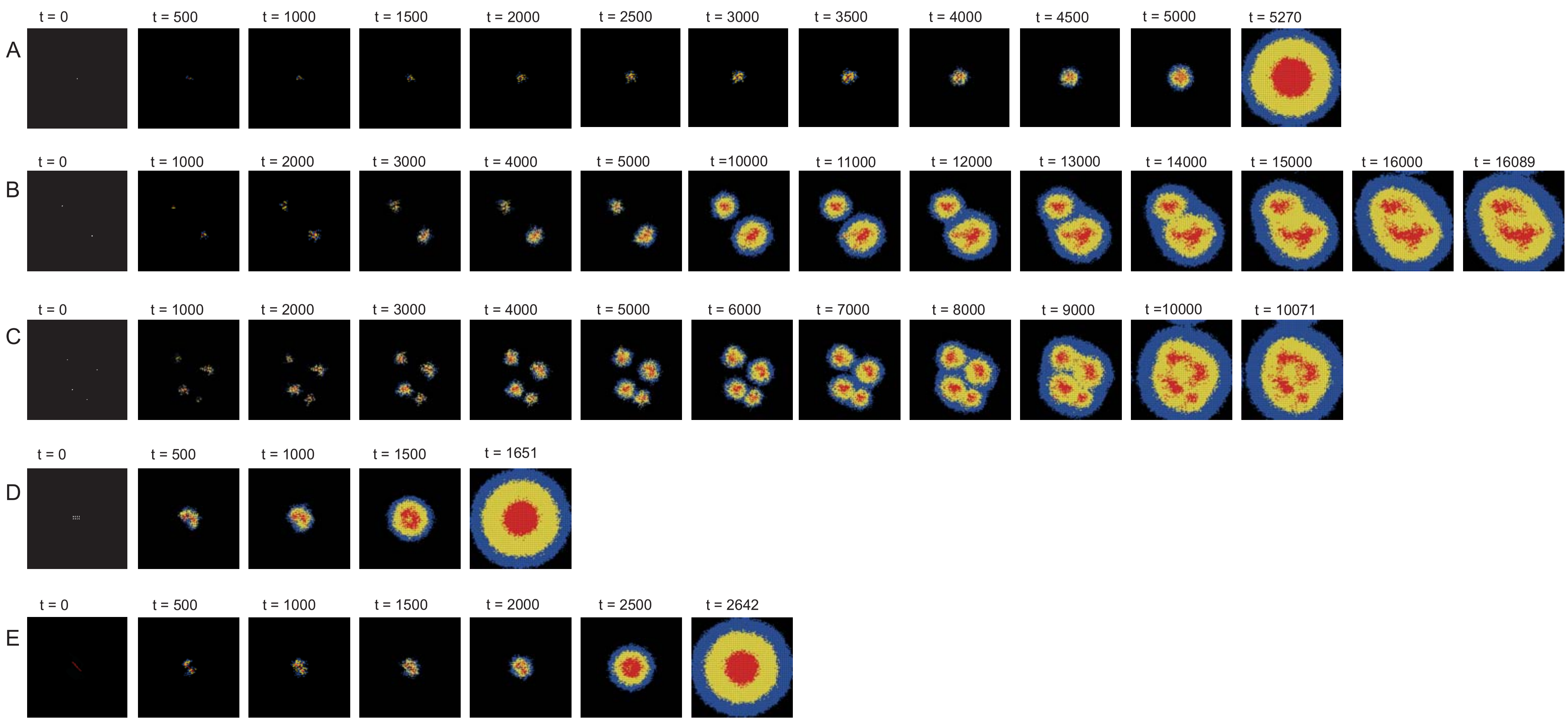


Figure 7.23 - Sequences of snapshots testing initial conditions, where sequence A uses a single central seed, sequence B and sequence C use multiple seeds, sequence D uses a proxy for a regular grid, and sequence E uses a proxy for a path or road..

From Figure 7.22 it is evident that tests with multiple seeds around the centre, which is the case for the grid and path, presents an identical behaviour to the single seed, only slightly faster. It is also clear from this chart that there is a difference in the speed of growth when the number of seeds scattered in the simulation space increases, which is the case for test B (two fixed seeds) and test C (four fixed seeds).

This can also be observed in the snapshots presented in Figure 7.23, and is consistent with the charts of density from the centre, presented in Figure 7.24, where it is clear that tests B and C present rather different spatial results from the other three tests.

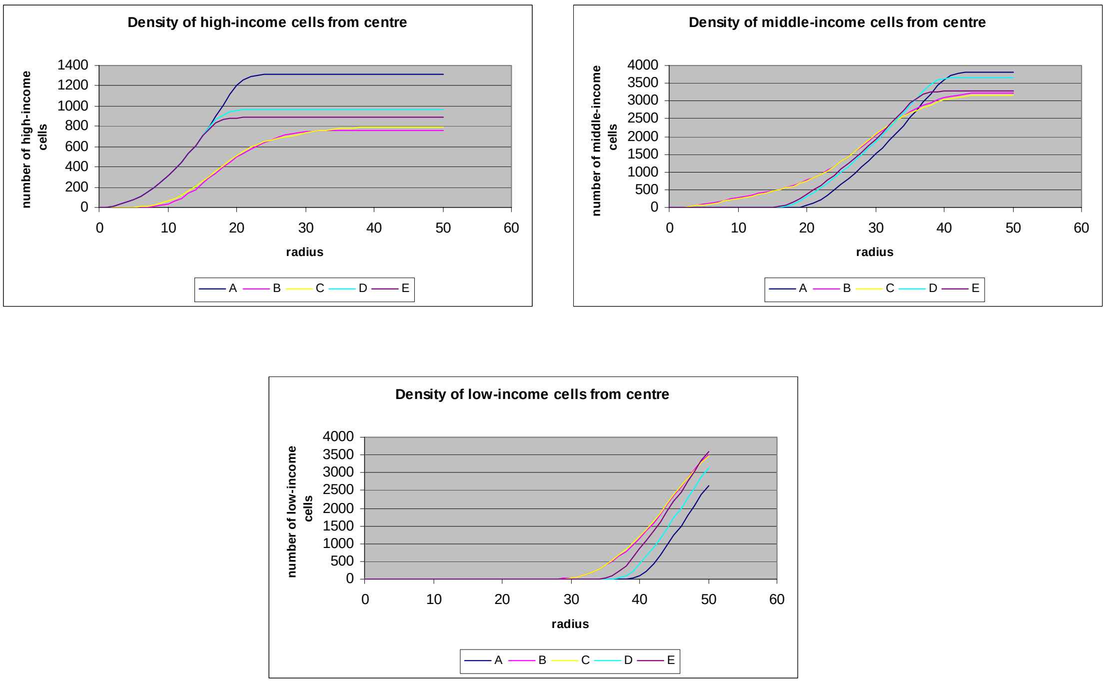


Figure 7.24 - Charts of density from the centre testing initial conditions, where sequence A uses a single central seed, sequence B and sequence C use multiple seeds, sequence D uses proxy for a regular grid, and sequence E a proxy for a path or road.

7.4.2 Module two: spontaneous settlements

As seen in Chapter 6, module two consists in the study of the consolidation of low-income cells into a spontaneous settlement. The number of consolidated cells is regulated by the parameter *consLimit* that defines the threshold of the variable 'cons' in which the consolidation process will happen.

In what follows the behaviour of this parameter will be presented and discussed. All tests below were carried out using a single central seed as initial condition, *proportion of agent per economic group* 10% high-income, 40% middle-income, and 50% income, and *steps* equal to 2.

7.4.2.1 consLimit

The *consLimit* parameter does not affect the overall behaviour of the model, impacting only on the number of consolidated cells, as can be observed from the sequences of snapshots in Figure 7.25, where sequence A presents representative runs conducted using *consLimit* equal to 2, sequence B equal to 4, and sequence C equal to 8. In the image, consolidated cells are represented by the cyan colour.

It is interesting that this parameter namely affects the number of non-consolidated low-income cells, despite the fact that the number of low-income active agents is kept constant along the simulation run. This effect can be clearly observed in the snapshots in Figure 7.25. This seems to be due to the fact that low-income agents take longer than other agents to settle because it is more difficult for them to find empty locations. In a normal simulation run, there is a fixed number of active low-income agents plus a number of evicted agents. When the *consLimit* parameter value is very low as in sequence A, more cells are consolidated and less agents are evicted, resulting in a reduction of the total number of low-income agents settling and, therefore, in fewer low-income cells.

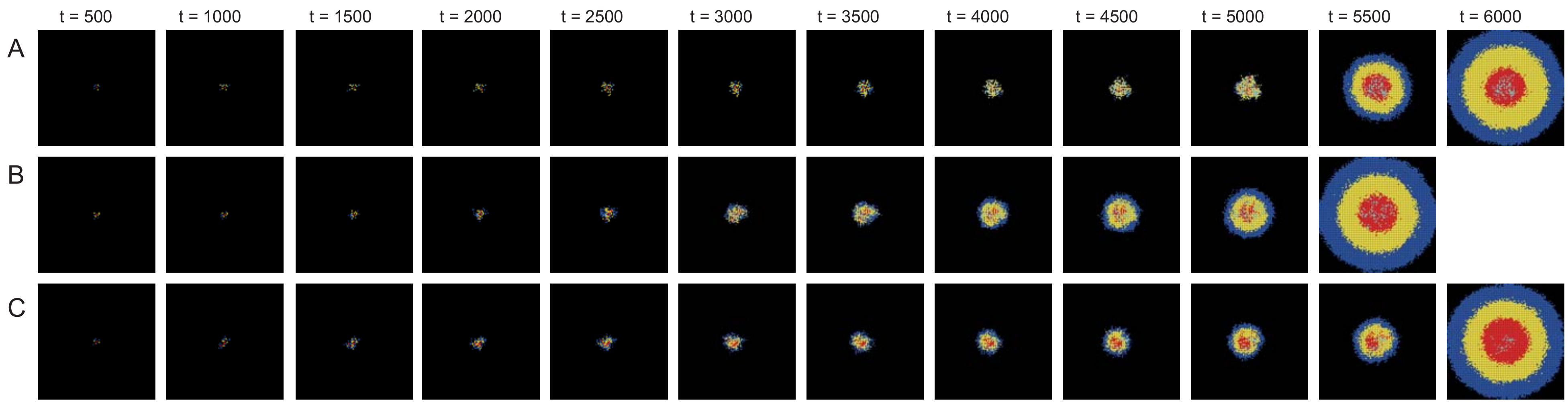
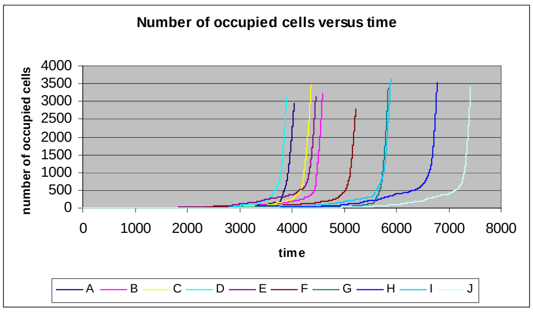
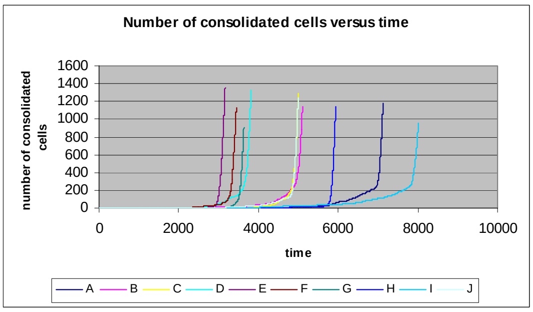


Figure 7.25- Sequences of snapshots testing the parameter *consLimit*, where sequence A uses *consLimit* = 2, sequence B uses *consLimit* = 4, and sequence C uses *consLimit* = 8 .

Figure 7.26 shows 3 sequence charts where the number of consolidated cells was plotted against time, testing different values for the *consLimit* parameter. Because tests with the *consLimit* parameter present considerable variance, the charts in Figure 7.26 present all the conducted tests, and not just the representative runs. Each chart displays the behaviour of the number of consolidation cells in time for each of the 10 sequential runs carried out with the simulation model for each value of the *consLimit* parameter.





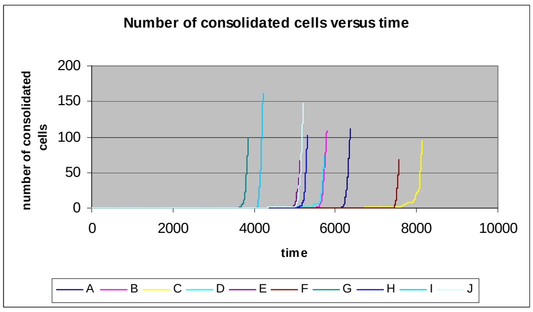


Figure 7.26 - Tests with the *consLimit* parameter, where sequence A uses *consLimit*= 2, sequence B *consLimit* = 4, and sequence C *consLimit* = 8.

From the charts in Figure 7.26, it is possible to notice that the number of consolidated cells tends to decrease as the *consLimit* value increases.

Results of landscape metrics are shown in Table 7.2, where it can be observed that an increase in *consLimit* values results in a decrease in the complexity of shapes, as indicated by the decrease in *PAFRAC* values. Values for *CONTAG* and *IJI* display an odd behaviour, since the metric results do not show a consistent trend as the value for *consLimit* increases.

The *CONTAG* value decreases when *consLimit* is raised from 500 to 1000 but increases significantly when *consLimit* changes from 1000 to 2000. As higher *CONTAG* values indicate a more aggregated landscape, the metrics suggest that when *consLimit* changes from 500 to 1000, the landscape as a whole becomes less aggregated; but when this value is further increased to 2000, the landscape as whole becomes more aggregated. It is important to keep in mind that these metrics are global metrics, i.e. they translate the state of the entire landscape and not only the state of a single class. In this case, it seems that a change in the number of consolidated cells only, which decreases as *consLimit* value increases, impacts on the landscape's attributes in such a way that it becomes slightly more disaggregated. The first change in *consLimit* values presents a significantly more aggregated characteristic when the number of consolidated cells decreases further, as would be expected.

The values for *IJI* present the same kind of behaviour. As the value for *consLimit* is increased, *IJI* values show an increase followed by a significant decrease. *IJI* values indicate how mixed the landscape is. It seems that the first rise in the *consLimit* values makes the entire landscape more mixed, but as the value is further increased the landscape becomes less mixed again.

	PAFRAC	*CONTAG*	*IJI* (%)
consLimit = 2	1.7381	32.6425	42.9889
consLimit = 4	1.6891	24.2799	77.3730
consLimit = 8	1.4100	50.8724	17.8397

Table 7.2 – Landscape metrics results for tests with *consLimit*.

This behaviour in results for *CONTAG* and *IJI* is found for all tests where the dispersion of a single class is changed, as will be examined in the following sections.

7.4.3 Module three: inner city processes

Module three comprises 5 parameters. *steps2* and *steps3* define the number of steps that an agent has to move once it has found the area to settle. Parameter *d* is a threshold value for neighbourhood density. *decayStartPoint* is the threshold value of the age of high-income cells and determines when the decay rule starts to act in the system. Finally, the parameter *consolidationLimit* is also a threshold for the age variable, which is valid for the low-income cells only, determining the value from which cells can consolidate.

The next sections will present tests with each of these parameters and discuss their roles in the model's behaviour. All experiments have parameters fixed as follows: *proportion of agents per breed* = 10, 40, 50, *steps* = 2, *steps2* = 2, *steps3* = 2, *d* = 4, *decayStartPoint* = 2000, *consolidationLimit* = 1000, *consolidationRandom* = 100. For each of the experiments presented, these values vary for the parameter under investigation only.

7.4.3.1 steps2 and *steps3*

Like *steps*, presented in section 7.4.1.1, *steps2* and *steps3* are parameters that affect directly the configuration of the resultant spatial pattern. While the parameter *steps2* determines the distance a high-income agent will walk seeking a low-density location, *steps3* determines the same distance for the middle-income.

Figure 7.27 shows snapshots of representative runs testing different values for *steps2*. Sequence A uses *steps2* equal to 2, sequence B equal to 4, and sequence C equal to 8. It is possible to observe that the higher the value for *steps2*, the more dispersed are high-income cells in the images. The same can be seen in tests with *steps3* shown in Figure 7.28, where sequence A uses *steps3* equal to 2, sequence B equal to 4, and sequence C equal to 8.

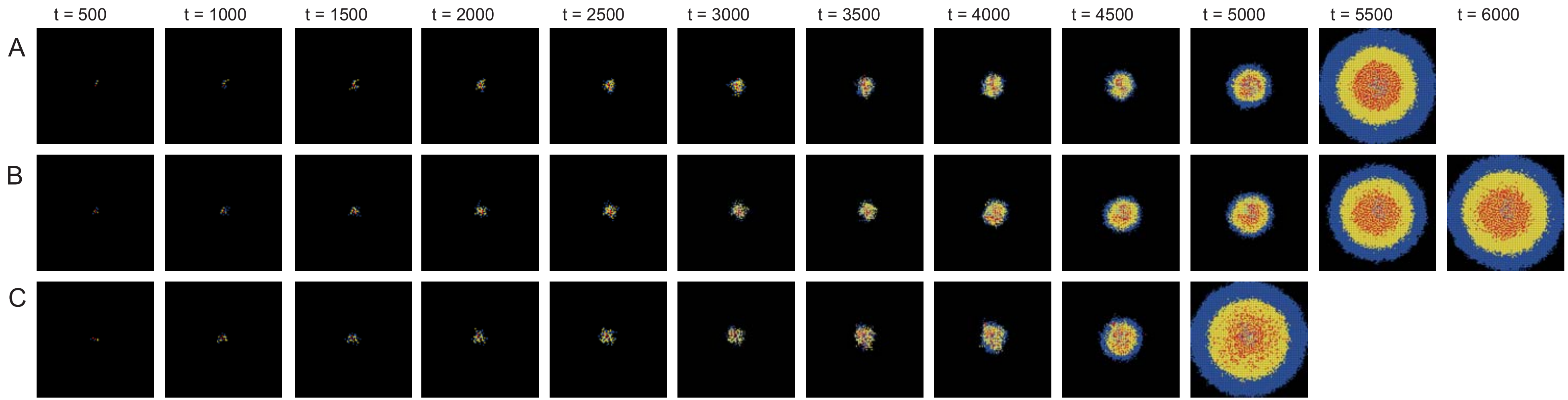


Figure 7.27 - Sequences of snapshots testing the parameter *steps2*, where sequence A uses *steps2* = 2, sequence B uses *steps2* = 4, and sequence C uses *steps2* = 8 .

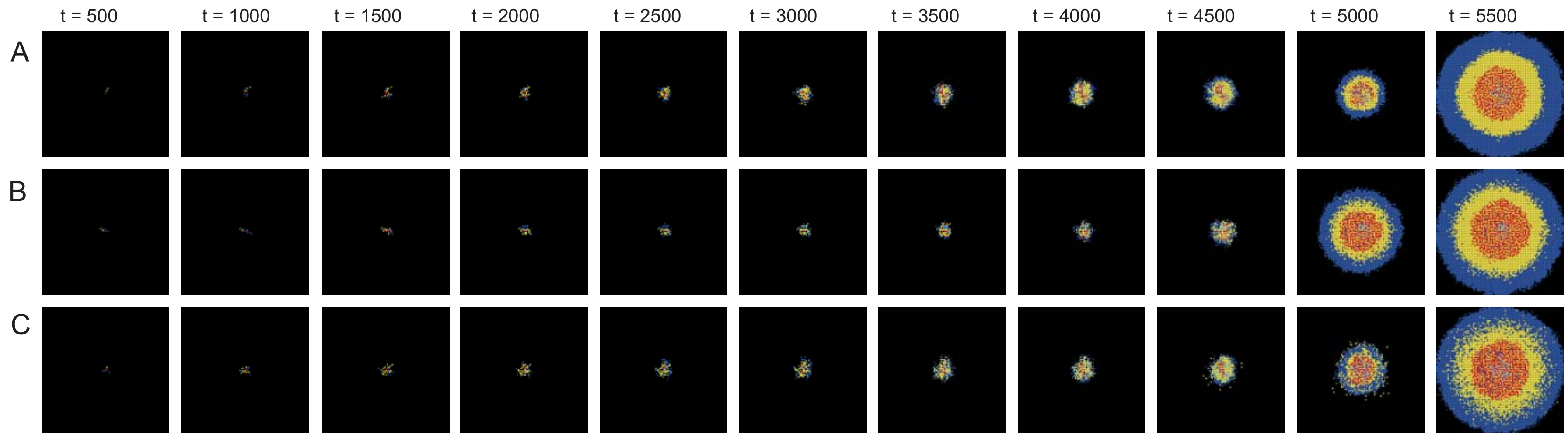


Figure 7.28 - Sequences of snapshots testing the parameter *steps3*, where sequence A uses *steps3* = 2, sequence B uses *steps3* = 4, and sequence C uses *steps3* = 8 .

Landscape metrics are presented in Table 7.3 for *steps2* and Table 7.4 for *steps3*. The results in Table 7.3 are consistent with the spatial pattern observed in the sequence of snapshots and demonstrate that *PAFRAC* results tend to be higher as the value for *steps2* increases, which means an increase in patch shape complexity in the resultant spatial pattern. The same trend is found for tests of *steps3* in Table 7.4.

The *CONTAG* results for tests with *steps2* and *steps3* show an opposite trend from the results presented for *steps* in section 7.4.1.1. While the *CONTAG* results increase with the rise in *steps* values, they decrease with rises in *steps2* and *steps3* values. As seen in the foregoing, the increase of *CONTAG* values indicates that patches become more fragmented (smaller and more dispersed). As *CONTAG* is a value for the entire landscape and not for each class, this might be explained by the fact that both *steps2* and *steps3* concern a single class of cells, and therefore the impact of their fragmentation alone presents a different effect from the landscape's fragmentation as a whole.

The results in both Table 7.3 and Table 7.4 show that *CONTAG* values decrease when values for *steps2* and *steps3* change from 2 to 4, but slightly increase when the value for *steps2* and *steps3* is raised from 4 to 8. The same trend can also be observed for *IJI* results in both tables below. It seems that the settling distance of high-income cells initially causes the fragmentation of the entire system to increase, and interspersion to decrease, but does not continue to have the same impact after a certain point.

	PAFRAC	*CONTAG*	*IJI* (%)
steps2 = 2	1.6096	50.0481	54.0238
steps2 = 4	1.6183	47.9831	47.6832
steps2 = 8	1.6424	49.3302	52.8204

Table 7.3 – Landscape metrics results for tests with *steps2*.

	PAFRAC	*CONTAG*	*IJI* (%)
steps3 = 2	1.6096	50.0481	54.0238
steps3 = 4	1.6439	43.0841	51.5369
steps3 = 8	1.6783	36.1299	57.1188

Table 7.4 – Landscape metrics results for tests with *steps3*.

In terms of time-scale, the charts in Figure 7.29 and Figure 7.30 indicate that the behaviour of the model through time is not affected by changes in the parameters *steps2* and *steps3*.

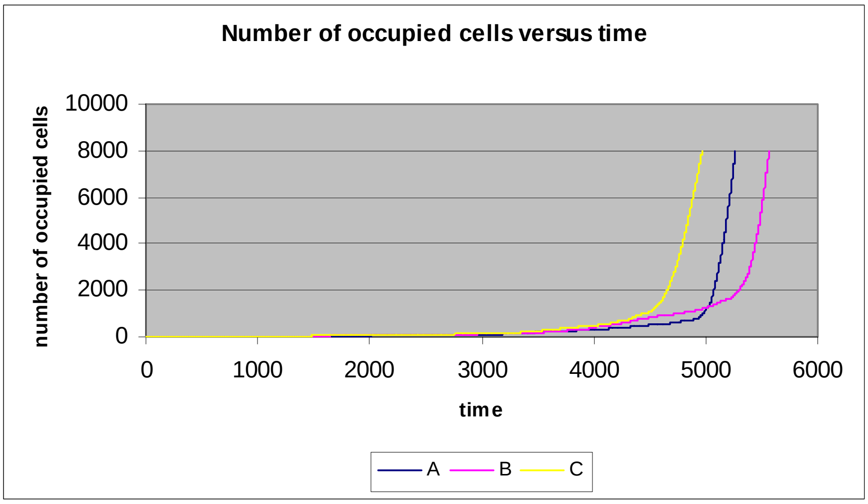


Figure 7.29 - Chart testing the parameter *steps2*, where sequence A uses *steps2* = 2, sequence B *steps2* = 4, and sequence C *steps2* = 8.

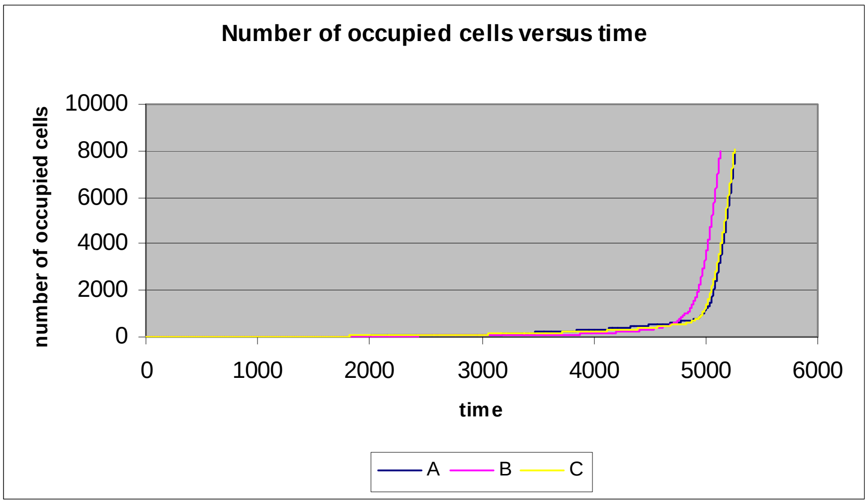


Figure 7.30 - Chart testing the parameter *steps3*, where sequence A uses *steps3* = 2, sequence B *steps3* = 4, and sequence C *steps3* = 8.

7.4.3.2 *Density threshold d*

As mentioned in the foregoing, the parameter *d* is a threshold for neighbourhood density, that is, it determines the maximum density accepted by an agent for a given location. In other words, if the neighbourhood density is higher than the established threshold (parameter *d*), the agents look for a place to settle further out.

Figure 7.31 shows sequences of snapshots of representative runs using different values for the parameter *d*. Sequence A was tested using *d* equal to 2, sequence B equal to 4, and sequence C equal to 8.

Observing the snapshots in Figure 7.31, it seems clear that lower densities produce a more fragmented landscape, with high-income agents producing a more spread out spatial pattern. However, the results of landscape metrics shown in Table 7.5, demonstrate that the overall fragmentation is not affected as such.

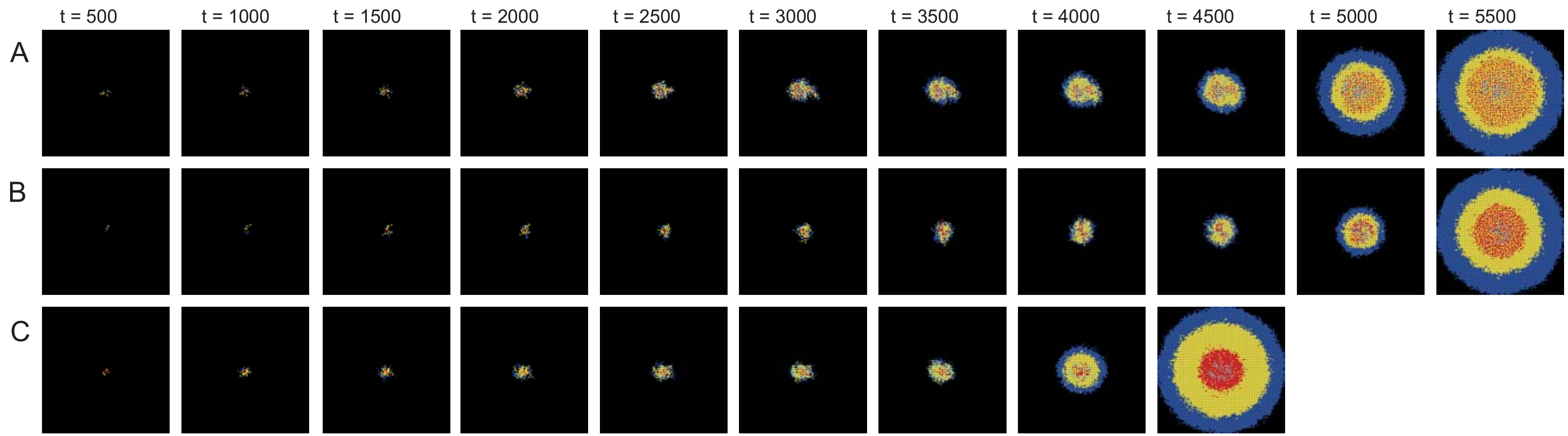


Figure 7.31 - Sequences of snapshots testing the parameter d, where sequence A uses $d = 2$, sequence B uses $d = 4$, and sequence C uses $d = 8$.

Results in Table 7.5 show that an increase in *d* values results in a decrease in the complexity of shapes, indicated by the decrease in *PAFRAC* values. As in the results presented for *steps2* and *steps3*, values of *CONTAG* are first increased but drop as the value for *d* changes from 4 to 8. A similar trend is found for *IJI* values, that is, when values for d increase from 2 to 4, there is also an increase in the interspersion of the landscape, but when values are increased from 4 to 8, the interspersion and juxtaposition index drops again.

	PAFRAC	*CONTAG*	*IJI* (%)
d = 2	1.6438	46.7814	55.0447
d = 4	1.6096	50.0481	54.0238
d = 8	1.4720	52.2558	55.6638

Table 7.5 – Landscape metrics results for tests with *d*.

Changes in values for *d* do not impact on the simulation's time-scale, as is clear from the chart in Figure 7.32.

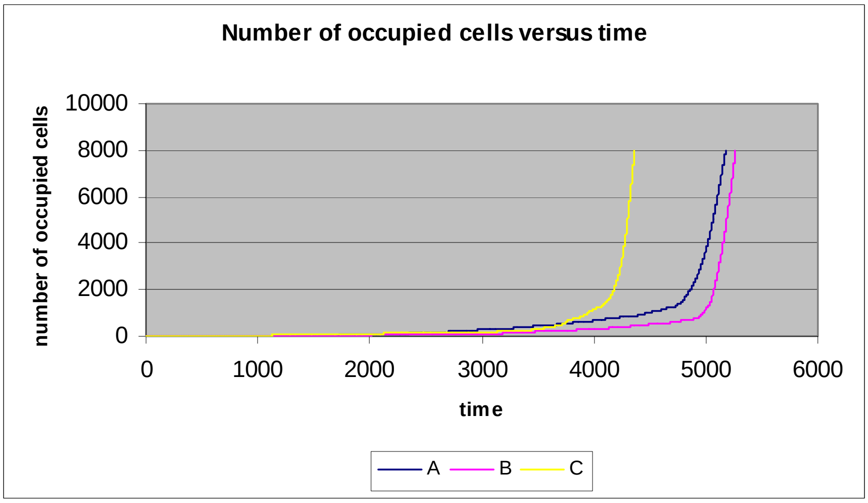


Figure 7.32 - Chart showing tests with parameter *d*, where sequence A uses *d* = 2, sequence B *d* = 4, and sequence C *d* = 8.

7.4.3.3 *decayStartPoint and decayRandom*

The *decayStartPoint* parameter determines the point in time that cells in the centre will start being abandoned. Abandoned cells are likely to be occupied by low-income groups and to consolidate. *decayRandom* is a random variable that determines how the percentage of cells suitable for being abandoned actually leads to cells being abandoned by agents.

Figure 7.33 shows three sequences of snapshots testing different values for *decayStartPoint*. Sequence A was run using a value equal to 500, sequence B using a value equal to 1000, and sequence C a value equal to 2000.

To better demonstrate the behaviour of *decayStartPoint* and *decayRandom*, the charts in Figure 7.33 and Figure 7.34 do not use a representative run for each value tested, but the whole sequence of 10 runs conducted to test each parameter value. In Figure 7.33, the three sequences charts use respectively *decayStartPoint* values equal to 500, 1000 and 2000. In Figure 7.34 the values for *decayRandom* were respectively, 20, 40 and 80%.

Both the sequences of snapshots in Figure 7.33 and the chart in Figure 7.34 show that the higher the value for *decayStartPoint*, the fewer the number of abandoned cells there will be in the final result. It is important to note that this is true for simulation runs with similar time-scales only, because *decayStartPoint* is a time-dependent parameter.

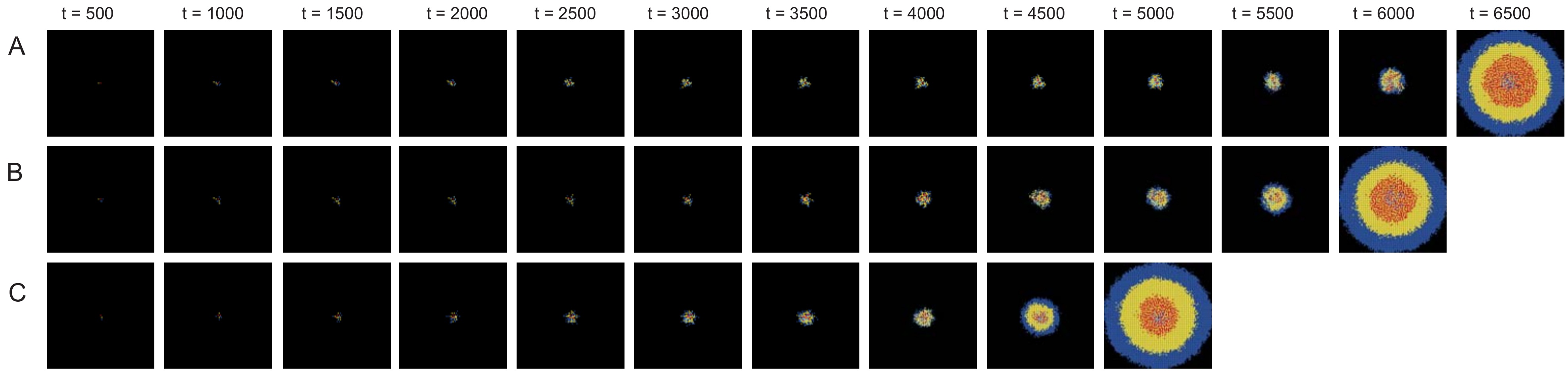


Figure 7.33 - Sequences of snapshots testing the parameter *decayStartPoint*, where sequence A uses *decayStartPoint* equal to 500, sequence B equal to 1000, and sequence C equal to 2000.

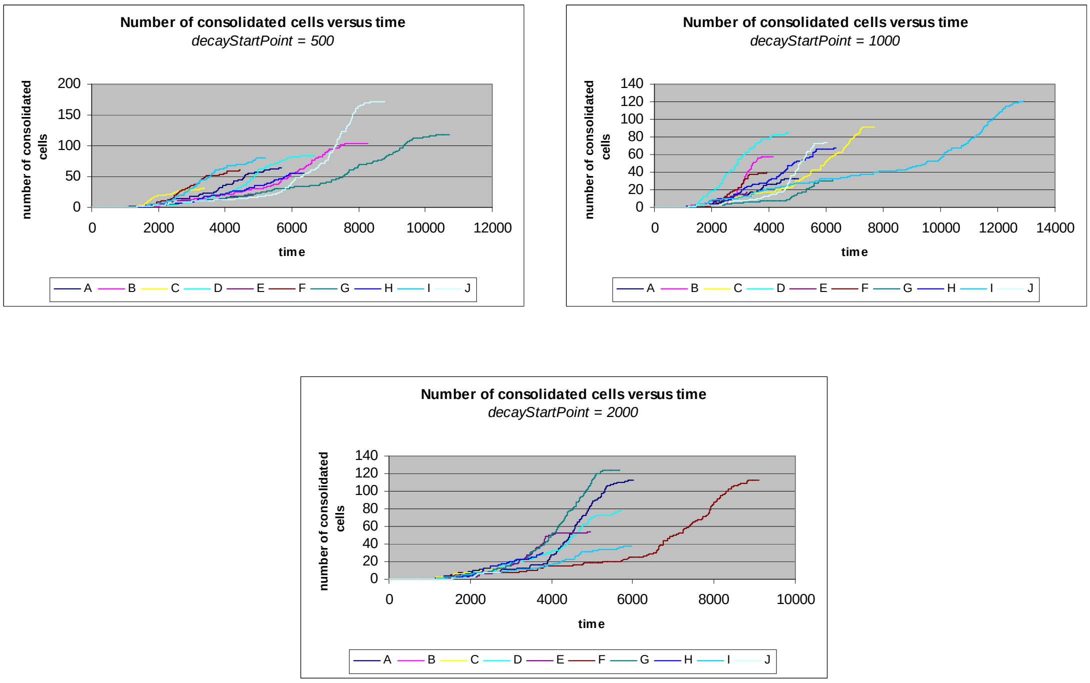


Figure 7.34 - Chart of the number of abandoned and consolidated cells versus time for a set of representative runs (A to J) testing the parameter *decayStartPoint*. Graph on the top-left of the image uses *decayStartPoint* equal to 500, graph on the top-right equal to 1000, and graph on the bottom equal to 2000.

Similarly, as the *decayRandom* value increases, there is a tendency for the number of consolidated cells to increase, as shown in Figure 7.35 and Figure 7.36.

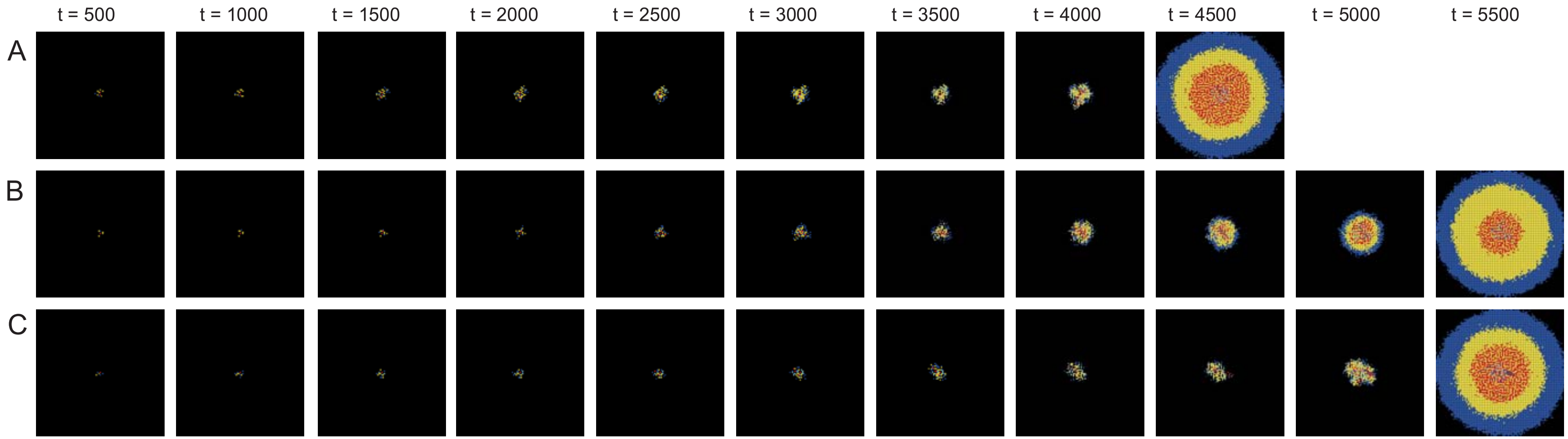
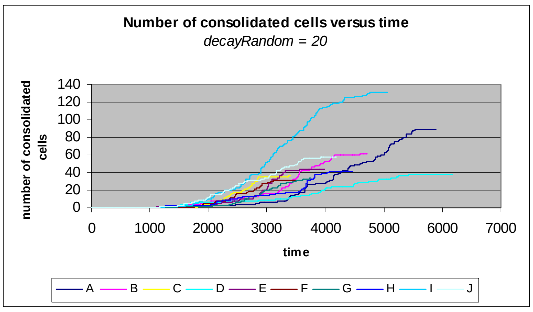
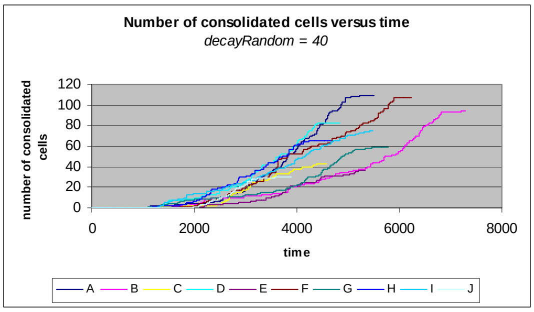


Figure 7.35 - Sequences of snapshots testing the parameter *decayRandom,* where sequence A uses *decayRandom equal to* 20, sequence B equal to 40, and sequence C equal to 80.





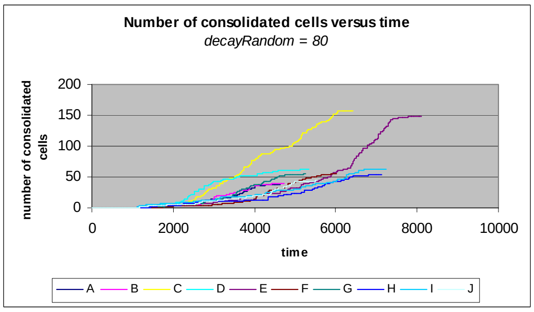


Figure 7.36 - Chart of the number of abandoned and consolidated cells versus time for a set of representative runs (A to J) testing the parameter *decayRandom*. Graph on the top-left of the image uses *decayRandom* equal to 20, graph on the top-right equal to 40, and graph on the bottom equal to 80.

7.4.3.4 *consolidationLimit*

The *consolidationLimit* operates together with the *decayStartPoint* parameter, playing the role of assuring that abandoned cells in the centre of the spatial pattern, once occupied by low-income groups, will not be re-occupied by higher income groups.

The change in values for this parameter produces more or fewer consolidated cells in the centre of the model space, as can be observed in the snapshots in Figure 7.37, where sequence A shows tests run with *consolidationLimit* equal to 500, sequence B equal to 1000, and sequence C equal to 2000.

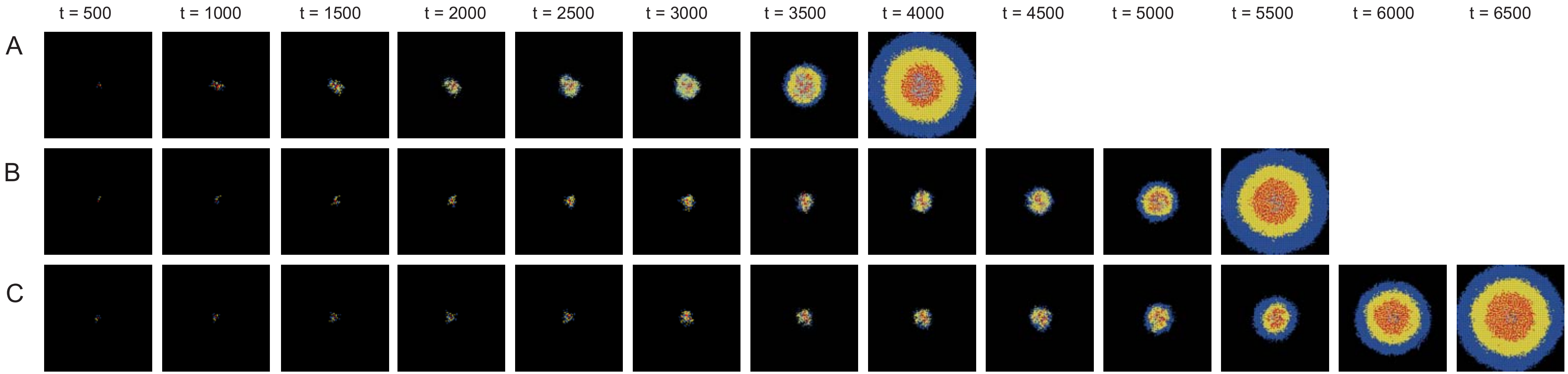


Figure 7.37 - Sequences of snapshots testing the parameter *consolidationLimit*, where sequence A uses *consolidationLimit* equal to 500, sequence B equal to 1000, and sequence C equal to 2000.

The same behaviour can be observed in the charts in Figure 7.38, below. The chart shows the number of abandoned and consolidated cells through time. Consolidated cells in module 03 of the Peripherisation Model are actually abandoned cells that have been re-occupied.

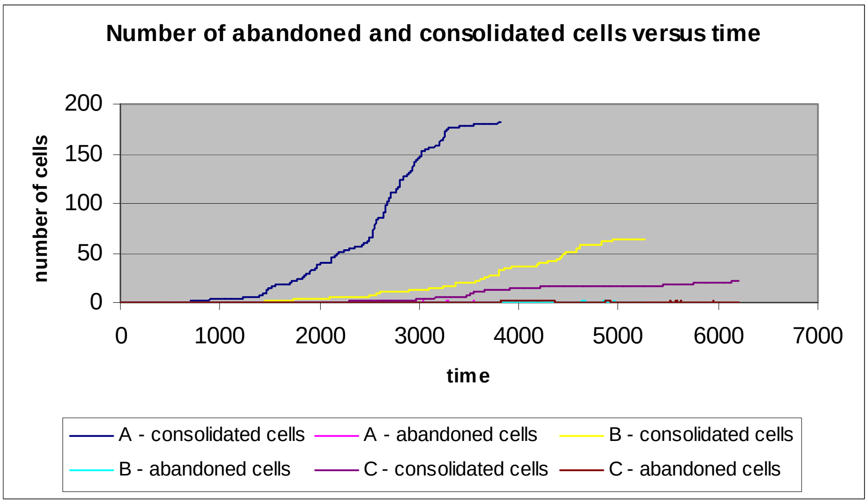


Figure 7.38 - Numbers of abandoned and consolidated cells versus time, where sequence A uses *consolidationLimit* equal to 500, sequence B equal to 1000, and sequence C equal to 2000.

From the charts of tests with *decayStartPoint*, *decayRandom* and *consolidationLimit* (Figure 7.34, Figure 7.36 and Figure 7.38) it is possible to notice that the number of consolidated cells tends to decrease as, for example, the *consolidationLimit* value increases. It also can be observed that longer simulation runs generate a higher number of consolidation cells, which is perfectly understandable as the parameter is time-dependent. However, the number of consolidated cells does not vary proportionally with the time-scale of the simulation, as would be expected. The charts presented show that there is a considerable amount of variation within the values shown in each chart, which are tests run with the exact same set of parameters. This suggests that path dependence and stochastic processes play important roles in the behaviour of these parameters.

7.4.4 Module four: spatial constraints

Three sets of runs were carried out to show the impact of spatial constraints. Sequence A in Figure 7.40 does not use any spatial constraints, while sequences B and C use two different configurations of constraint areas, presented in detail in Figure 7.39.

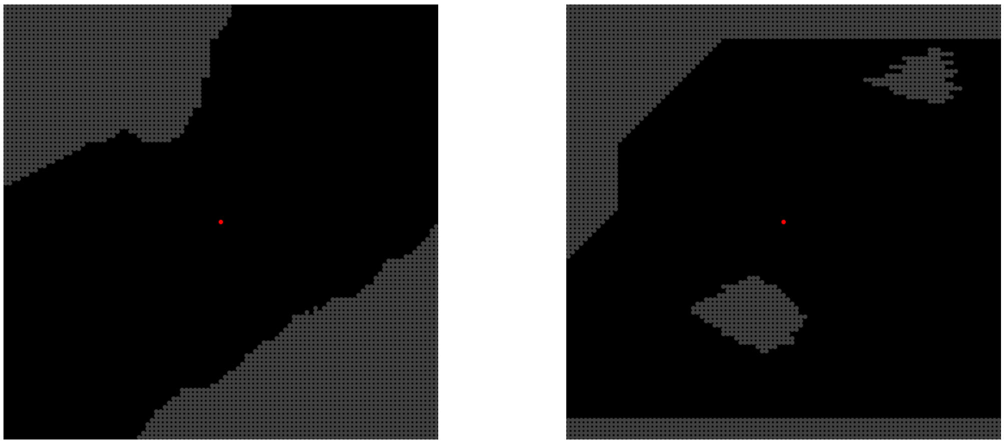


Figure 7.39 - Spatial constraints incorporated in the initial conditions used for sequences A (on the right) and B (on the left).

Representative results for these three sets of tests are compared in the charts below. The chart in Figure 7.41 shows the behaviour of each of the tests through time. Spatial constraints impact on the time-scale of the model because the available grid area is reduced. In sequence B, 3385 cells are occupied by grey areas while in sequence C 2593 cells are occupied by spatial constraints. The changes in behaviour follows the same trend as the impact of grid size discussed in section 7.3 of the present chapter.

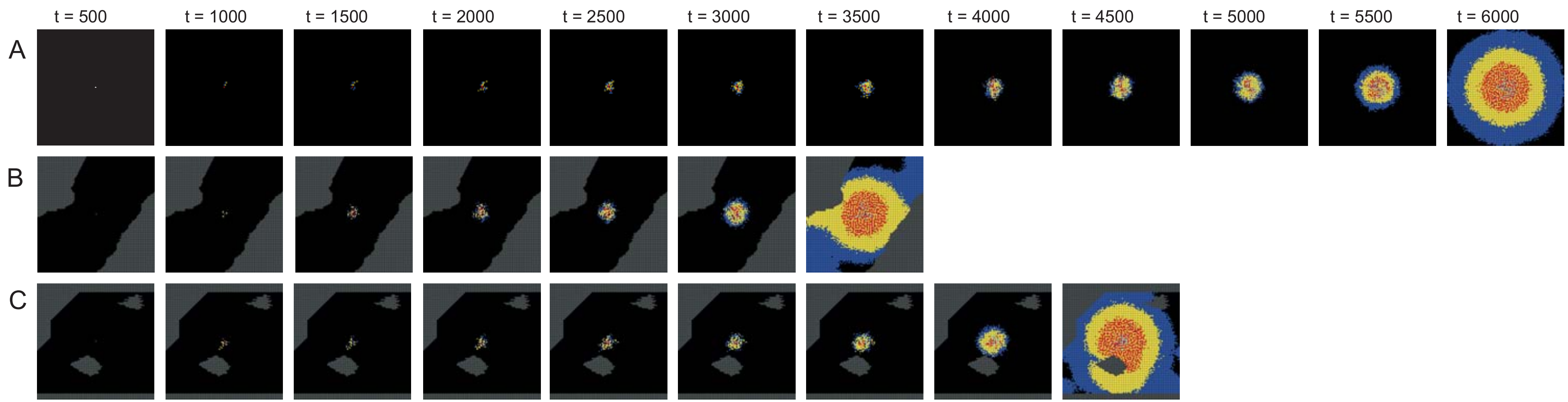


Figure 7.40 - Sequences of snapshots testing spatial constraints.

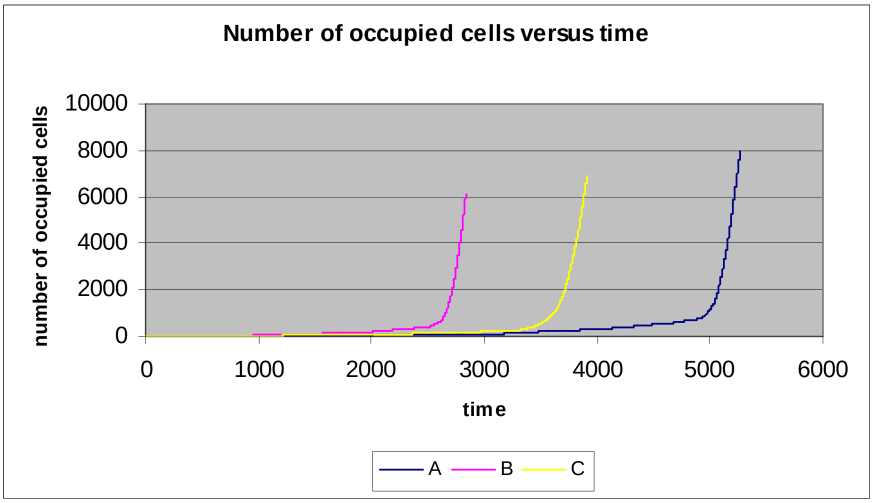


Figure 7.41 - Impact of spatial constraints on the model's time-scale, where sequence A uses a single central seed a initial conditions, and sequences B and C use spatial constraints shown in Figure 7.39.

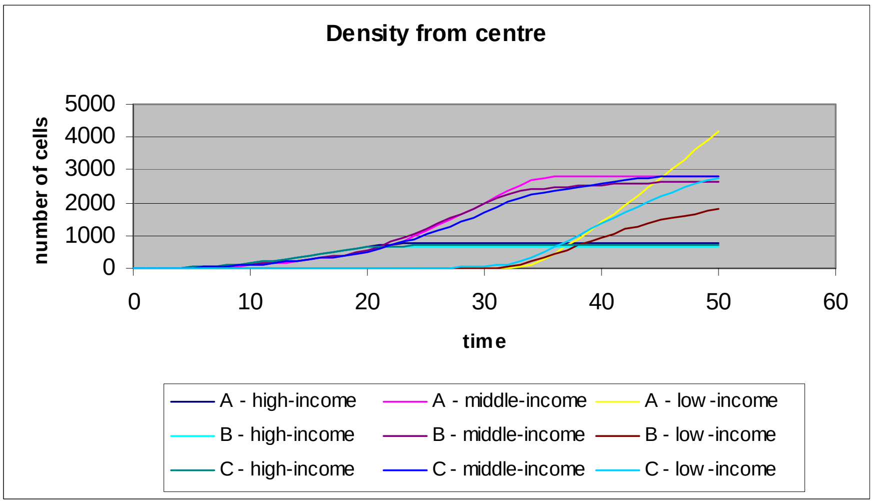


Figure 7.42 - Impact of spatial constraints on the density from the centre, where sequence A uses a single central seed a initial conditions, and sequences B and C use spatial constraints shown in Figure 7.39.

In fact, the impact of spatial constraints on the model's behaviour can be best seen in the sequence of snapshots in Figure 7.40, where is clear how the spatial development clearly moulds itself to the spatial constraints, breaking the circular ring pattern displayed in previous tests. Figure 7.42 illustrates that, despite the changes seen in the snapshots, the density from the centre remains very similar in all three sequences. This is due to the fact that all three tests define a central area of the grid free of spatial constraints.

7.5 Summary

This chapter reported experiments to evaluate the Peripherisation Model. Each of the parameters of the model was tested and its impact on the model's behaviour analysed. It is interesting that some of the analysed parameters do not have a great impact when isolated from changes in other parameters. Their impact can be seen with greater effect on the experimental exercises shown in the next chapter.

One of the main findings of this chapter is that, despite the specific roles played by each parameter within the model, the final outcomes are strongly defined by path dependence and random process effects. Tests reveal that although changes in parameters have an impact on general tendencies in the final outcomes, deterministic behaviour cannot be found for any of the model's parameters. This is characteristic of complex systems and is an expected consequence of rules used to model behaviour in the model.

One of the striking features of the model behaviour is related to the time-scale. Tests revealed that a number of different factors impact on simulation time-scale. The most significant ones are changes in grid size and number of active agents. Few parameters as such have a significant impact on time-scale, but the *proportion of agents per economic group* seems to be the most important of them. Tests suggest that the time-scale is mainly defined by the probability of settlement of high-income agents, and therefore is highly vulnerable to stochastic processes and path dependence effects.

One of the important outcomes of this chapter is the acknowledgement of the typical behaviour of the model and the effect of changes in the parameter values in the simulation outcomes. This lead to a reasonable set of parameter settings for the model, which are used when it is applied to explore aspects of the real world.

Furthermore, the sensitivity tests allow the modeller to understand the typical behaviour of the model and to distinguish the features that are result of the behaviour of the parameters within the model from those that can be observed as proxies for the real behaviour of the system modelled. This is particularly important when the model is built as an instrument to help in thinking and questioning about aspects of the real system, as the next chapter will discuss.

Chapter 8

Simulation Exercises

This chapter will present simulation exercises that explore a number of aspects of dynamic change in Latin American cities. These exercises consist in analyses of the model's outcomes and from these results, the chapter attempts to provoke discussions and draw conclusions about the reality of Latin American cities. The contents of this chapter bring together the theoretical discussion in Part I and the modelling process developed in the last two chapters.

The objective is to discuss the main issues of urban growth and change presented in Part I in the light of the simulation exercises. Simulation brings to light aspects of the real system by allowing tests of different hypotheses through changes in the initial conditions and parameter values of the model.

In what follows, four simulation exercises will be presented, each of these exploring one of the Peripherisation Model simulation modules. Exercise one is based on the Peripherisation module and discusses general issues of urban growth in Latin American cities. Exercise two investigates the role of spontaneous settlements within the global process of urban growth. Exercise three explores inner city dynamics in an experiment that investigates the similarities and differences of inner city dynamics in two different kinds of city: Western cities and Latin American cities. Finally, exercise four explores the impacts of spatial constraints in creating more realistic simulation outcomes.

8.1 **Urban growth in Latin American cities**

Although the Peripherisation Model is composed of four modules, as presented in Chapter 6, the Peripherisation module is the model's basis on which all other features were built. The Peripherisation module's agents rule base is a very simplistic one. Agents are divided into three economic groups which have the same locational preferences but different restrictions. Restrictions are representations of economic power, so that high-income agents can 'afford' to locate anywhere, middle-income agents can locate anywhere except where high-income agents have already located and low-income agents can only locate on empty land. On top of these restrictions, there is a rule that allows agents with higher-incomes to evict lower-incomes agents from places where they are already located.

This set of very simple rules produces the spatial pattern presented in Figure 8.1 below. The parameters used for this simulation run were *steps* equal to 2 and *proportion of agents per economic group* 10% high-income, 40% middle-income, and 50% low-income. Figure 8.1 shows the spatial development of the simulation run through a sequence of snapshots. As time in the simulation is fictitious, i.e. is not related to time in reality, time values were omitted in the figure.

The final spatial pattern shown in the sequence of snapshots in Figure 8.1 consists of three concentric rings where red represents high-income settled agents, yellow represents middle-income agents, and blue represents low-income agents. Although the simulation starts with a mixed set of patches, the final outcome is a clearly segregated pattern where each of the three economic groups is settled forming a single large patch and each is completely separated from the other economic groups.

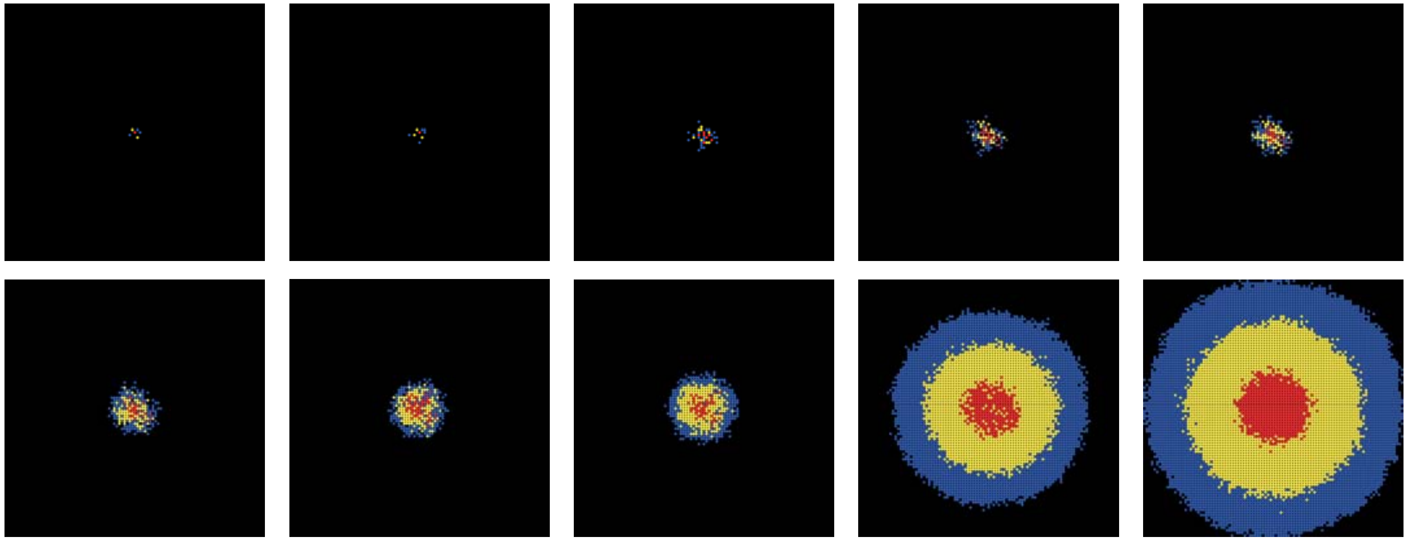


Figure 8.1 - Spatial pattern produced with the Peripherisation module.

This outcome is not 'unpredictable' and makes sense within the rules of the model. However, there is nothing in the rule base that suggests that the spatial outcome of the model would be a segregated pattern, and nothing that suggests that high-income groups should be located in the centre surround by buffering rings of middle and low-income cells. It is possible to say, therefore, that the pattern is emergent, since it is the product of the local interactions of agents only.

Although very simplistic, the pattern produced by the model might one say approximates the spatial structure found in the residential locational pattern of Latin American cities, described previously in Chapter 2. Similarly, the pattern produced by the model when using multiple initial seeds, resembles certain characteristics of metropolitan areas. Figure 8.2 presents a sequence of snapshots using four seeds, all equidistant from the centre of the grid. The simulation was conducted using the same set of parameter values used for the previous experiment.

The sequence shown in Figure 8.2 approximates to the development of metropolitan areas, which are the result of the combination of several cities or villages that end up as a single spatial area because of their proximity. It is interesting to note that the spatial development starts with a very mixed structure, and as time passes, the core-periphery structure emerges. As in reality, this spatially segregated pattern is consolidated in the model, and as the simulation runs, the spatial development expands, maintaining the core-periphery structure.

It is particularly striking to see how the high-income areas of the spatial patterns become slowly linked to each other, and the yellow and blue rings simply follow the shape imposed by high-income areas, acting as buffering zones for high-income areas, following the pattern described by Amato (1970b) for Bogotá, Colombia. This suggests that the model reproduces not only a final spatial pattern that is consistent with reality, but also that the evolutionary process shaping this pattern is relevant too.

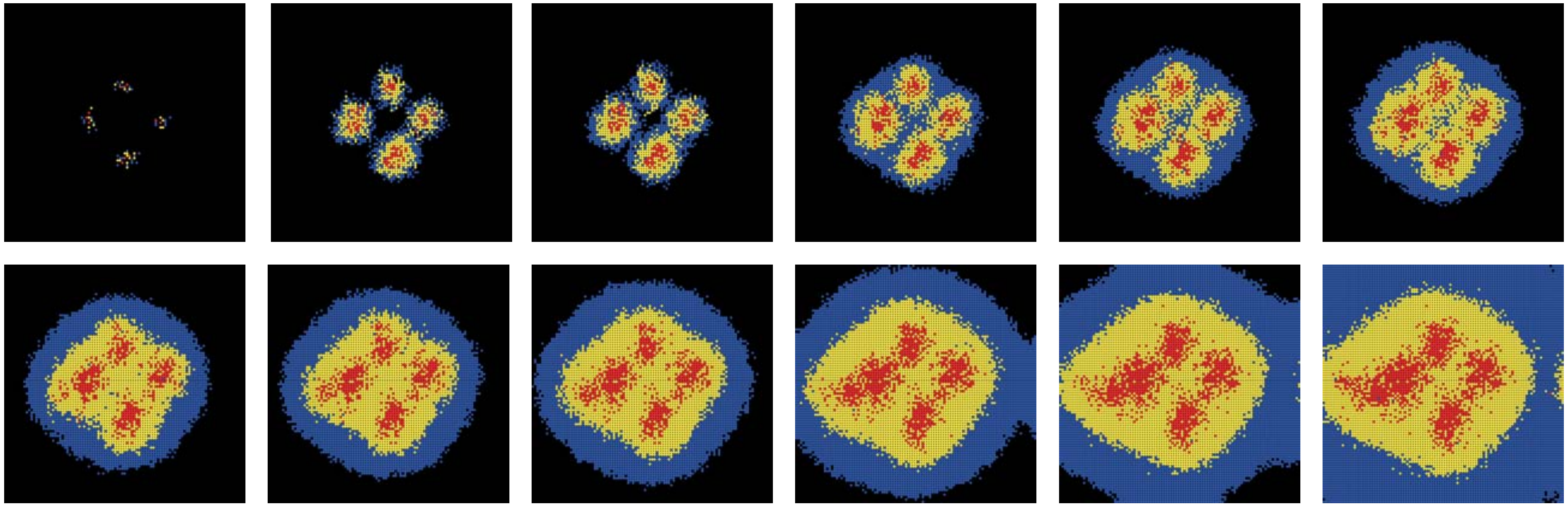


Figure 8.2 - Spatial pattern produced with the Peripherisation module using multiple seeds as initial condition.

The simple spatial pattern produced by the Peripherisation Model resembles in essence the spatial pattern and dynamics of urban growth in Latin American cities described in Part I. As discussed previously, Latin American cities are characterised by their high rates of growth, and their spatial patterns are the result of this fast process of urban development. It is understood that high rates of urban growth have overwhelmed the capacity of urban governments to provide either adequate services or infrastructure and, therefore, are the main cause of the urban inequalities found in Latin American cities.

The simulation experiments provide material with which to discuss this assumption using the model as a *tool to think with,* and to examine the context of the speed of development within the simulation. In the Peripherisation Model, the presence of a great percentage of high-income agents, for instance, results in larger and faster development. This is implicit in the model's rules since high-income cells act as a catalyst for urban development. For very different reasons, it seems that in reality the presence of high-income groups has a similar effect. The richer a city is, the more attractive it is and, therefore, more people migrate to it causing a higher speed of growth.

Whenever urban growth in Latin American cities is encountered, the first factor to be mentioned is the high speed of growth. In the literature, this speed is seen as an essential cause of the spatial patterns that results. The present simulation exercises make clear that if the rules of the model are related in any sense to the way locational decision takes place in reality, then speed has little, if any, influence on the generation of the core-periphery spatial pattern. In the model, speed can be manipulated by, for example, increasing the number of agents within the simulation. This does not affect the spatial pattern at all, as can be observed in Figure 8.3 (which is a reproduction of Figure 7.15), which shows snapshots of simulation runs using different numbers of agents.

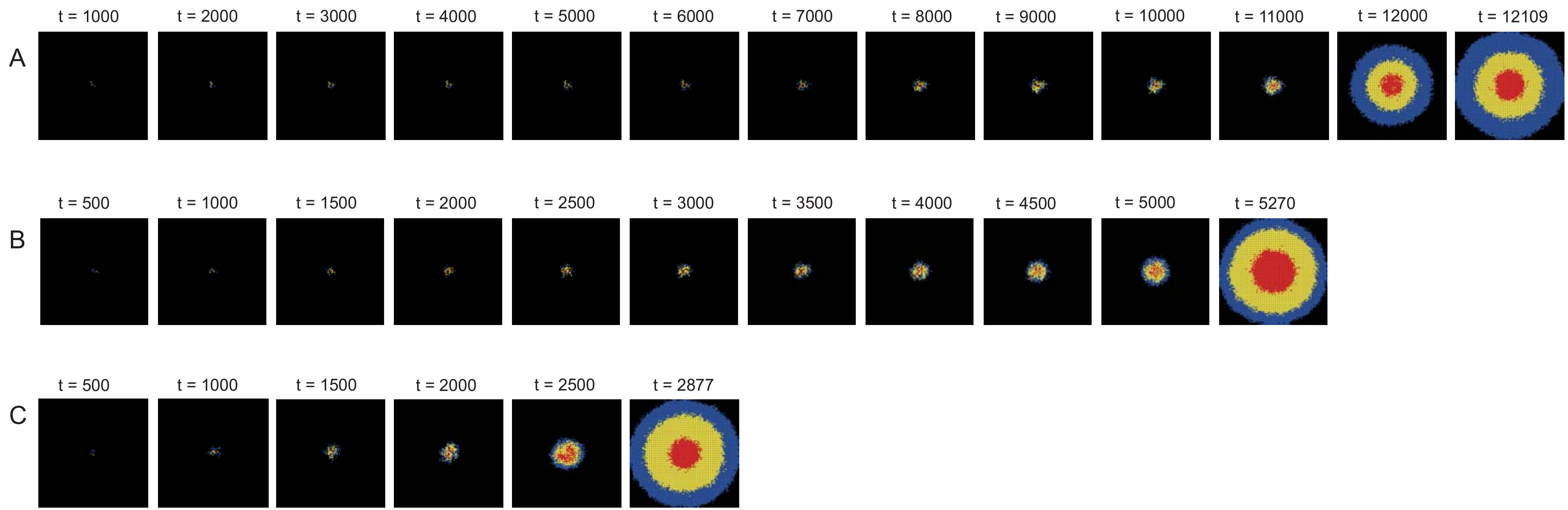


Figure 8.3 - Sequences of snapshots testing different number of agents.

In this light, a question arises as to what in fact generates the segregated spatial pattern found in most Latin American cities: is it the speed of growth that was so uncontrollable that planners could not contain or control it, or is it perhaps a simple product of the inequalities of a segregated urban society? If it is assumed that the roots of a segregated spatial pattern can be largely explained by the unequal division of urban society and its economic power, then the role played by speed in the formation of the spatial pattern must be questioned.

8.1.1 Comparison with reality

This section presents simple maps built from the Census 2000 dataset for São Paulo. Although these maps are static representations of patterns of income concentration, together with the simulation model they help us to demonstrate the locational pattern generated by peripherisation in Latin American cities.

Figure 8.4 shows maps of income distribution in the metropolitan area of São Paulo, Brazil. The city of São Paulo has a population of over 10.4 million inhabitants and occupies an area of 1,509 km^2, out of which 900 km^2 are urbanized. Its metropolitan region is comprised of 39 autonomous cities with a resident population of more than 17.8 million inhabitants occupying an area of 8,501 km^2.

The maps show the distribution of income per census sector in the urbanized area of São Paulo (Figure 8.4). The limits of the maps are the administrative boundaries of greater São Paulo, which is composed by the 39 autonomous cities or municipalities.

The data used here are the average of the head of household monthly income per census sector (enumeration district / census block) which are part of the Census 2000 dataset provided by the Brazilian Institute of Geography and Statistics (IBGE). This variable was chosen because of its similarities to the rules of the Peripherisation Model, which is based on the division of agents into economic groups.

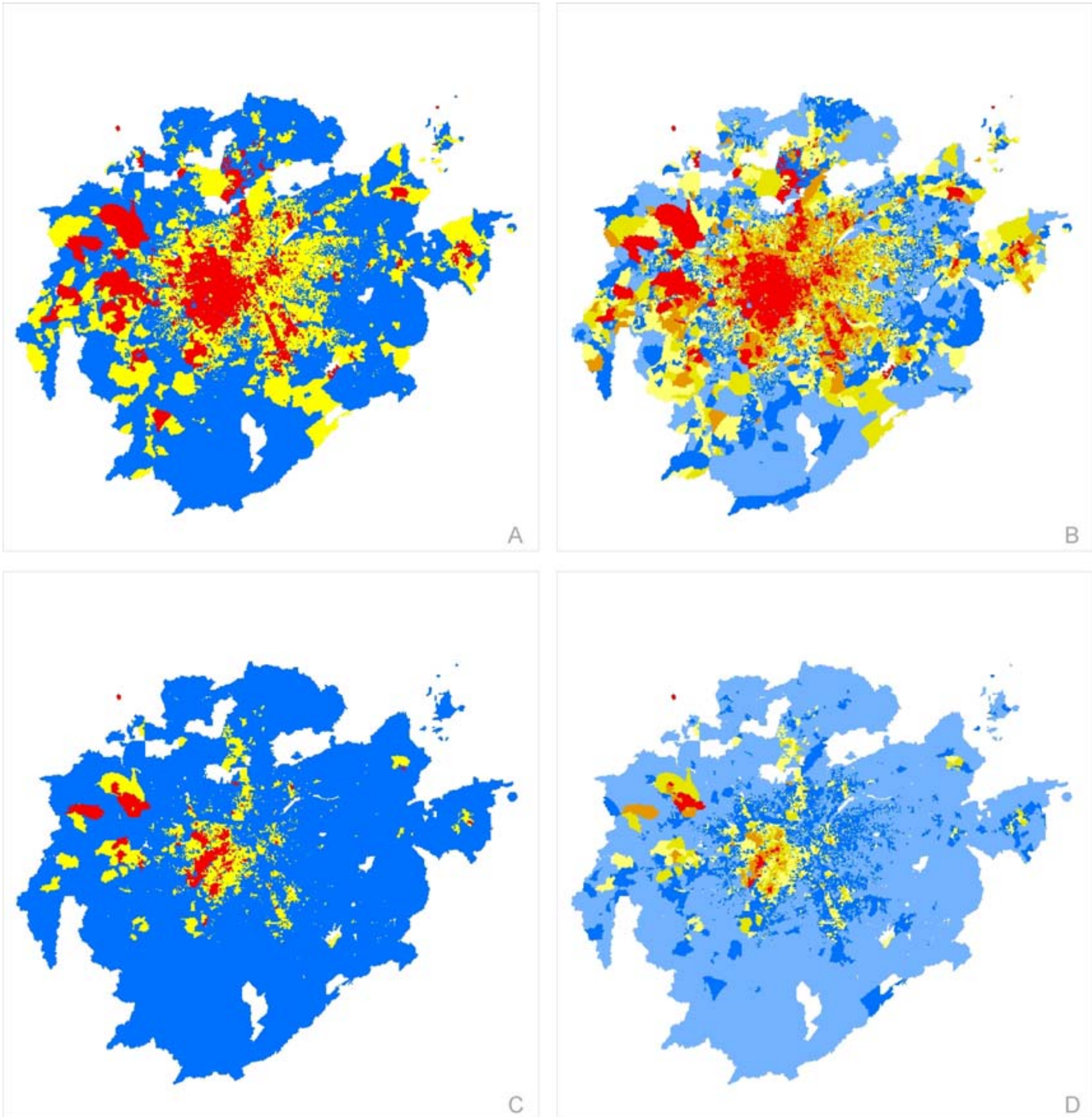


Figure 8.4 - Maps of São Paulo showing distributions of income in the urban area. Maps A and B were built, respectively, using 3 and 6 quantile breaks and maps C and D were built using, respectively, 3 and 6 natural breaks.

The aggregated data per urban census sector were normalised by the number of householders in each sector and then classified into three ranges (maps A and C in Figure 8.4) or six ranges (maps B and D). The maps use red for the higher income groups, yellow for middle-income groups, and blue for the lower income groups as in the simulation model, to aid comparison. As in the images produced by the model, one can easily identify a concentric pattern in map A in Figure 8.4, in which the high-income groups are concentrated towards the centre of the urban area and thus the concentration decreases towards the urban periphery. The graduation is more easily observed in map B in Figure 8.4 where the same data were graduated into six classes, showing again a decrease in the level of income towards the edge of the city.

Maps C and D in Figure 8.4 show a different classification of income, where the extent of high-income areas is smaller in comparison with the two other maps. In these maps, the classification was made according to the natural groupings of the data on (income) values. What the map actually shows is that there are very few people belonging to the high-income group and a lot of people belonging to the low-income group. It should be noted, however, that we have not used established definitions of income groups either in the simulation model or in the maps shown above, and our focus is only on the relative locational pattern of these groups within the city. As such, the actual number in each income group is not relevant for the present study.

When comparing the spatial pattern produced by the Peripherisation Model to the maps in Figure 8.4, the first noticeable conclusion is that the spatial pattern in reality is not as concentric as are the patterns produced by the simulation model. This is due to various factors such as initial conditions, topography, the presence of bodies of water, etc. In particular, the topography of these areas has strong influences on the spatial development of these cities.

A second very clear difference is that high-income groups are not all concentrated in the (historical) centre of the city, but may reach towards the city's outskirts. Similarly, middle-income groups are at times located on the city edge and in more central areas surrounded by low-income areas. These suggest that there are more dynamic processes in action than those simulated in the model.

8.1.2 Discussion

These experiments were part of the first phase of this research project. The shortcomings of these exercises pointed towards the next steps in the development of the model. The fact that real cities do not have the shape of circular concentric rings is evidence that urban development is shaped by inner city processes as well as by spatial constraints on development such as topographic slopes and bodies of water. Adding these features to the model were the next steps in the research and the following exercises explore these features.

8.2 Spontaneous settlements in the urban growth context

This section consists of a brief theoretical analysis of the process of formation of inner city squatter settlements within the global process of urban growth. As presented in Chapter 6, the spontaneous settlements module of the model extends the basic Peripherisation logic with a *consolidation rule*. This rule refers to a process in which spontaneous settlements are gradually upgraded, and, as time passes, turn into consolidated *favelas* or, in other words, spontaneous settlements that are harder to evict. As a result of the introduction of the consolidation logic, the spontaneous settlements module generates a more fragmented landscape than the homogeneous concentric-like spatial distribution of classes in which consolidated spontaneous settlements are spread all over the city.

The experiments show that a combination of empty spaces and a more mixed pattern produce a fragmented spatial result. This process resembles what actually can be observed in Third World cities, where, despite the general tendency for economic segregation, there are 'fragments' of low and middle-income residential areas within high-income zones, and vice-versa.

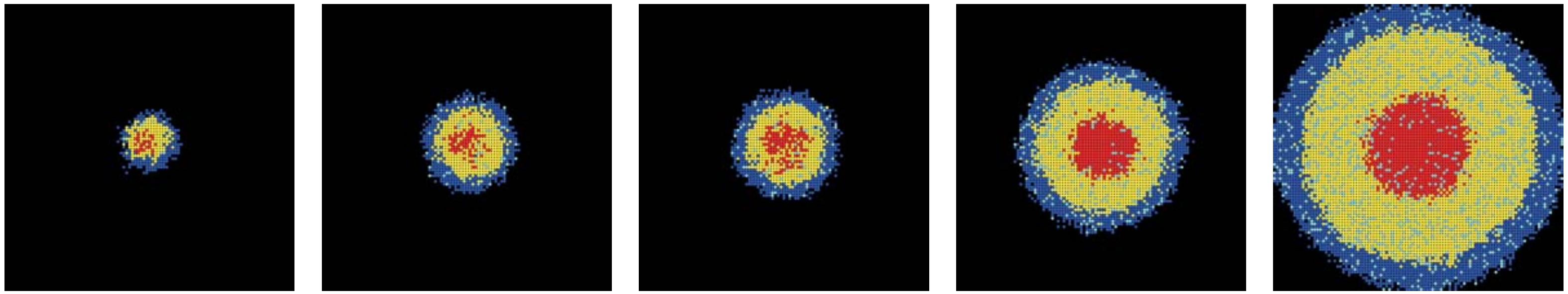


Figure 8.5 - Sequence of snapshots showing the consolidation of spontaneous settlements.

At the beginning of the simulation there are no cyan patches (consolidated spontaneous settlements) in the virtual urban landscape. After some iterations, cyan cells appear in all three social-economic zones, resembling what occurs in actual cities. In the last snapshots one can identify a very particular pattern, which seems quite similar to the typical distribution of spontaneous settlements in Third World, especially Latin American cities.

8.2.1 Discussion

The process simulated by the model reveals some interesting points in relation to real cities. First, it makes evident that central spontaneous settlements are the same as those produced in what was once the urban periphery. This is an obvious, but nonetheless a very important point, as most of the literature does not seem to acknowledge the dynamic process of location in the formation of spontaneous settlements. Spontaneous settlements, in this context, are sub-systems that mutate and evolve within a global dynamic process of change and growth. Of course, the internal changes in these settlements, discussed in Chapter 3, also play a role in the absorption of the settlement to the urban fabric.

A second important point concerns the location of spontaneous settlements. In the model this process was represented by a stochastic mechanism only. Although in reality some randomness might exist, there are many factors that contribute to the consolidation of a spontaneous settlement. The morphological structure of a settlement, for instance, might affect the likehood that this settlement will be incorporated into the urban fabric or will remain as an isolated patch. A second and very important factor is the value and nature of the land where the spontaneous settlement is located. As discussed in Chapter 3, spontaneous settlements tend to be located in unwanted areas. If there is no competition for the areas where spontaneous settlements are located, as in hazardous places like the hills of Rio de

Janeiro, for example, the likelihood that low-income groups are not going to be evicted increases substantially.

These factors are not included in the model, and would actually make interesting possibilities for future work. The implementation of a land value and suitability map, for example, would have substantial impacts in making the model more realistic.

Even so, the study of spontaneous settlements as it stands in the present version of the model raises some interesting questions. Spontaneous settlements have traditionally been approached from a local point of view, as anomalies rather than as inherent global features of the urban system. This exercise shows the need for a global approach, considering spontaneous settlements as a dynamic element of the system, influencing the whole process of urban growth, as key parts of the spatial pattern of Latin American cities. The present exercises offer a change of perspective, focusing on the role that spontaneous settlements play in the global dynamics of development, as they are upgraded internally and develop as entities.

From a socio-spatial point of view, the existence of spontaneous settlements can be understood as 'instability pockets', which are necessary for the structural stability of the global system (Portugali, 2000). If it is considered that spontaneous settlements actually absorb part of the existing social instability - translated here as housing deficit - in unstable pockets within the city, one could say that they are necessary for the structural stability of the global system. Viewed as such, spontaneous settlements are fragments that keep the system away from what otherwise would be a breakdown of the already fragile and unstable equilibrium of the socio-spatial structure of Third World cities. This idea comes to reinforce Turner's (1988) argument that spontaneous settlements can be seen as an alternative solution, rather than a problem for the housing deficit. In the Third World urban context, spontaneous settlements play a paramount role within a system in which the parts do explain the whole, but only when seen in the light of a self-organised process.

8.3 Inner city processes: Latin American and Western cities

As discussed in Chapter 2, Latin American cities present a specific mode of urban growth, which differs in many aspects from urban growth in Western countries. Yet, most theories about urban growth, morphological patterns and inner city change have treated the city as a generic entity and, therefore, knowledge of the specificities of the dynamics of different kind of cities has not so far been fully explored.

Despite a number of researchers' work with a general concept of the 'generic city', urban theories are built upon observations and data from real cities which not only are not generic but also belong to specific countries, cultures, and have a specific history. Of course, this does not mean that there is no such thing as a general theory of the city, or that there are not general principles that drive urban development across cultures and countries. On the contrary, only by recognising and studying these differences in depth might be possible to understand what a generic city and its universal urban processes are.

The present investigation does not intend to bridge this gap, but to initiate a discussion about the subject by presenting a comparative study of two different patterns of urban development: the phenomenon of urban sprawl in Western cities and the pattern of rapid urban growth in the developing world.

This simulation exercise tests hypotheses (or theories) about inner city processes of residential change and their applicability to cities across cultures presenting different spatial patterns of residential location. The idea is to relate urban growth and residential spatial patterns to inner city change at the neighbourhood scale, understanding how residential land uses behave in a dynamic way.

By examining the dynamic processes in the inner city in both Western and developing countries it is possible to identify similarities in the essence of residential locational processes, such as filtering, inner city decay, movement of elites towards the city edge, and gentrification. Although the spatial patterns in those two kinds of cities differ completely, the dynamic processes that shape these patterns might be similar in nature.

The aim of this simulation experiment is to examine whether the inner city dynamics present similarities or differences, and how these dynamics produce different global spatial residential patterns. It is necessary to note that the present investigation is not concerned with the driving forces of these processes, nor with the economic or social theories that attempt to explain them, but just with the occurrence and intensity of such processes and the production of different global spatial patterns of residential location.

8.3.1 Western versus Latin American cities

As discussed in Chapter 2, while the problem of urban growth has been recently stressed in Europe and America in terms of sprawl, in the Third World – and more specifically in Latin America – the main focus has been on the rapid growth of cities, as well as the social inequalities produced by this process of urban development.

It must be clear from the foregoing that, in Latin American cities, the urban growth process is characterised by an expansion of the borders of the city through the massive formation of peripheral settlements, which are, in most cases, low-income residential areas including spontaneous settlements. In Western countries, on the contrary, the low-income groups generally live in the centre and high-income groups in the suburban areas. Another important difference is the urban form that these kinds of growth take. Third world cities present a fragmented urban development, which presents overcrowding in certain parts of cities and leaves unused land in others. Urban sprawl in a Western context, on the other hand, takes a completely different urban form, which is always characterised by low-density but may vary between contiguous suburban growth, ribbon or strip development, and scattered or leapfrog development.

The differences between these two kinds of growth lie not only in terms of form and impacts, but also and most importantly in terms of the resultant residential locational pattern. Whilst in Western developed countries higher income groups live on or towards the periphery of the urban area, in Latin American countries, the periphery is essentially the place of the low-income groups. Moreover, in Western countries there is a positive correlation between distance from the city centre and higher social status. In Latin America, on the contrary, the location of socio-economic groups show a higher degree of correlation with land rents (Amato, 1968). Another interesting difference between urban sprawl and peripherisation is that urban sprawl is directly related to the preference of people for a suburban location. The peripherisation phenomenon, however, is not a direct consequence of this locational preference. On the contrary, people who move to the city's border do not wish to live there but are impelled to.

In the Peripherisation module, the process of change for a higher economic group is part of the general process of growth (succession). The general mechanism of this process resembles the expansion and succession of growth rings proposed by Burgess (1925) and does not consist of re-occupation and regeneration of older housing in attractive inner city districts, as is the case in Western cities.

Features were added to the Peripherisation module in order to add different behaviours to the simulation, as explained in Chapter 6. These features are attempts to reproduce some of the main dynamic processes in cities: inner city decay, movement of elites towards the city edge and gentrification by the process of location and relocation of individual agents from different income groups. The model simplifies these dynamic processes using a set of very simple spatial interaction rules and looks at how these rules produce contrasting and complex spatial patterns in different kinds of cities.

8.3.2 Simulating Western and Latin American spatial patterns

Figure 8.6 shows a sequence of snapshots of two simulations where the parameters were set in an attempt to simulate two different spatial patterns, the Latin American city pattern, on the first line; and the Western pattern on the second line.

For the Western city pattern, the parameters were set as follows: *d* = 2, *steps* = 2, *steps2* = 7, *steps3* = 8, *decayStartPoint* = 400, *decayRandom* = 40, and *consolidationLimit* = 400. The income-groups ratio used was 40% of high-income agents, 50% of medium-income agents, and 10% of low-income agents.

For the Latin American city, the ratio used was 10% of high-income agents, 40% of medium-income agents, and 50% of low-income agents. The parameters were set as follows: *d* = 3, *steps* = 2, *steps2* = 4, *steps3* = 2, *decayStartPoint* = 800 and *consolidationLimit* = 600.

Each snapshot in Figure 8.6 has a fixed number of patches to aid comparison between the spatial developments of the two simulations. This is necessary because the time scales of the two simulations are very different. The final snapshot of the Western city simulation shows 7000 patches and its final time was 1973 periods. The final snapshot of the Latin American city simulation, on the other hand, has the same number of patches but its final time is 6208 periods.

The difference between the two patterns can be clearly observed in the Figure 8.6. The first sequence presents a much more mixed and scattered spatial development, while the second one presents a very defined and segregated spatial pattern. It is interesting to note that these results were obtained with very little variation in the model's parameters, but with a nearly reversed proportion of high-income and low-income groups.

From the set of simulations presented above, one can get an idea of how well the model reproduces the proposed patterns. From these analyses it is possible to hint at some of the main differences and similarities between the two processes of urbanisation and growth.

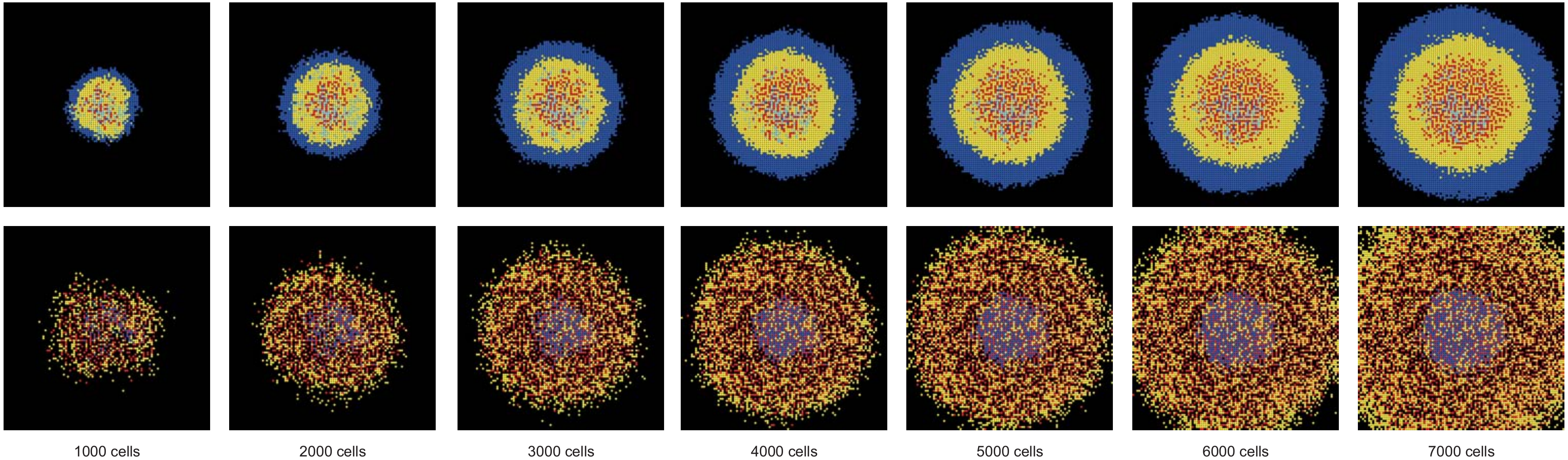


Figure 8.6 - Attempt to simulate the spatial patterns of Latin American (top sequence) and Western cities (bottom sequence).

In terms of the decay rule, one can observe that in both cities the 'abandoned' cells in the central places were completely occupied by the low-income agents (colours blue and cyan). It is also very interesting to see that, in the Western city simulation low-income agents, despite being restricted to empty cells only, are not located in the city fringe at all. Instead, they are concentrated in the central areas, occupying abandoned cells only. This is understandable since the ratio of abandoned cells in Western cities is enough to house the number of low-income agents active in the simulation exercise.

Despite these results, the simulation of Western cities does not seem so clear as the Latin American pattern. This might be caused by the absence of many important dynamic locational processes that are not represented in the simulation, such as the gentrification process. This is because of the complex mechanisms of the gentrification process, which cannot be approximated in simple spatial interaction rules, like those in the model. The second reason for the unclear pattern seems to be that Western cities, by contrast with Latin American, present other patterns of segregation not based solely on income-groups as in Latin American cities. Racial issues, for example, seem to play an important role in the formation of spatial clusters within the city, while public housing also contributes to complicate the picture.

8.3.3 Comparison with reality

In what follows, some simple maps built from the Census datasets will be presented for two Latin American cities: São Paulo and Belo Horizonte, Brazil; and for two Western cities: Boston and Buffalo, United States.

The maps show the distribution of income per census sector in São Paulo (Figure 8.7– A to C) and Belo Horizonte (Figure 8.7– D to F). The limits of the maps are the administrative boundaries of the cities and therefore the outer metropolitan areas (Great São Paulo and Great Belo Horizonte) are not included. As in the maps presented in section 8.1.1, the data used for these maps was the monthly household income per census sector (enumeration district / census block), which was normalised by the number of householders in each sector and then classified into three classes (quantiles) in maps A and D (Figure 8.7 and Figure 8.8), or six classes in maps B and E (Figure 8.7 and Figure 8.8).

For the maps for the cities in the United States, the attribute data used was median household income 1989 (Table 80A) from the 1990 US Census at the census tract level. For the Buffalo maps (A to C, Figure 8.8) the area covered was Erie County, NY, and the boundary data was for New York State; and for the Boston maps (D to F, Figure 8.8) the area covered was Middlesex County, Norfolk County, and Suffolk County, MA; and the boundary data was for Massachusetts.

The maps of Latin American cities (Belo Horizonte and São Paulo - maps A and D in Figure 8.7) reveal a concentric pattern similar to the images produced by the model. In maps B and E in Figure 8.7, the same data was graduated into six classes. Maps C and F in both Figure 8.7 and Figure 8.8 show a different classification of income, where the classification was made according to the natural groupings of the data (income) values.

It is also important to mention that the maps shown here do not encompass the metropolitan area of those two cities, but are restricted to their administrative boundaries. This means that São Paulo and Belo Horizonte are actually part of polycentric urban areas like those shown in Figure 8.4 and, therefore, the analysis of their urban form is restricted.

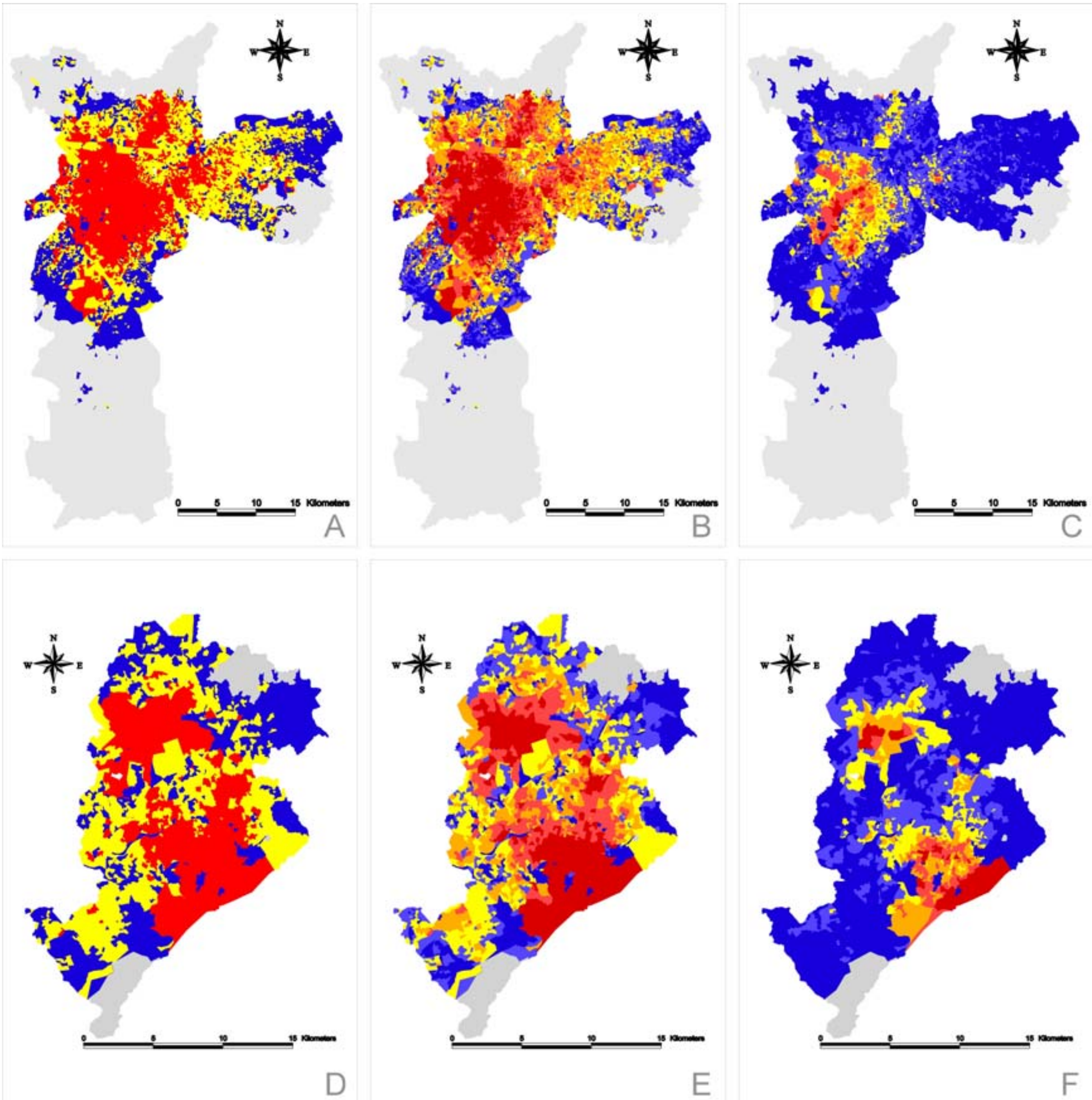


Figure 8.7 - Maps of Sao Paulo (A, B and C) and Belo Horizonte (D, E and F) showing distributions of income. Maps A, B, D, and E use quantile breaks; and maps C and F use natural breaks.

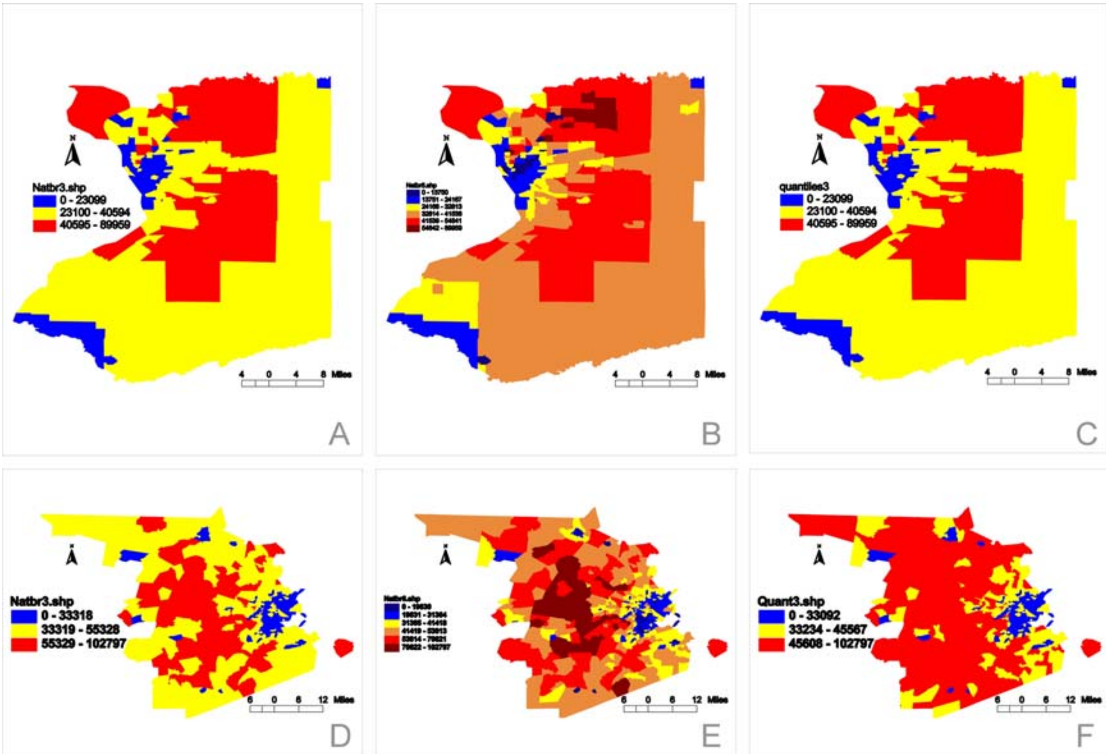


Figure 8.8 - Maps of Buffalo (A, B and C) and Boston (maps D, E and F) showing distributions of income. Maps A, B, D, and E use quantile breaks; and maps C, and F use natural breaks.

Looking at the maps of United States cities in Figure 8.8, one notices that the spatial pattern is not as clear as in Latin American cities, especially when income is classified in six classes rather than three. This is probably due to the nature of the income distribution in United States, which is not as uneven as in Latin America, where the difference in income between high and low-income groups is extreme. Nevertheless, the maps show clearly that income distribution in space is almost the reverse of the Latin American case, with the low-income groups at the centre of the city and high-income groups in an intermediate position surrounded by middle-income groups. As in the maps shown in previous sections, no established definitions of income groups were used in the maps shown above.

8.3.4 Discussion

The simulation exercises for Latin American and Western cities, despite making oversimplifications of complex urban realities, allowed the analysis of a set of dynamic processes, and made possible to draw hypotheses about the differences and similarities between two distinct types of urban reality. On the basis of these exercises, it is possible to suggest that the processes of filtering, core decay, and movement of high-income groups towards the city's outskirts are of *similar nature* in both Latin American and Western cities, and that they differ mainly in degree. However, as seen in Chapter 4, there are other processes going on in Latin American cities that are of paramount importance to an understanding of their dynamics of urbanisation and growth.

The 'reversed' spatial pattern of location seen in Latin American cities and Western cities seems to be caused by a combination of differences in degree in processes of similar nature. There are different processes which are not so significant in Western developed countries, such as upgrading and succession. Most importantly there are strong differences in the composition of the urban societies of these countries, which change the actual impact of these dynamics on the urban spatial pattern.

Finally, it is important to stress the need for more empirical studies of the inner urban change processes in Latin America and on the rest of the developing world. Empirical evidence is necessary to study these dynamic processes effectively, as a better understanding of the cities in the developing world would contribute to the field of urban studies as a whole.

8.4 Spatial constraints

For this exercise, the simulation will attempt to reproduce the spatial constraints of a real Latin American city: Porto Alegre, Brazil. The city of Porto Alegre has a population over 1.360 thousand inhabitants and occupies an area of 497 km^2. It is located in the state of Rio Grande do Sul in the South of Brazil.

Figure 8.9 - Aerial photograph of Porto Alegre. Source: Menegat et al (1998, page 10).

The city has developed along the River Guaíba, and has grown inland as can be seen in the aerial-photograph in Figure 8.9. In the photograph, one can observe the topography of the site where the city developed. In the south part of the site there are several hills, labelled 'morros' (meaning 'hills' in Portuguese) in the photograph.

The city is constrained by the river on the west and the hills on the south and these are the spatial constraints reproduced in the model, as shown in Figure 8.10.

As seen in Chapter 6, grey cells represent areas where agents cannot 'walk' or settle. The snapshot shown in Figure 8.10 presents grey areas in the west representing the river, and patches in the south-central area representing hills. The initial seed is located in the place where the city started, close to where the port was located.

Figure 8.11 shows the urban evolution of the city, displaying the location of its first settlements. In this exercise, the initial seed is also the start-point for all agents, i.e., all agents throughout the simulation run are created from the coordinates of the initial seed.

Figure 8.10 - Spatial constraints used for Peripherisation Model.

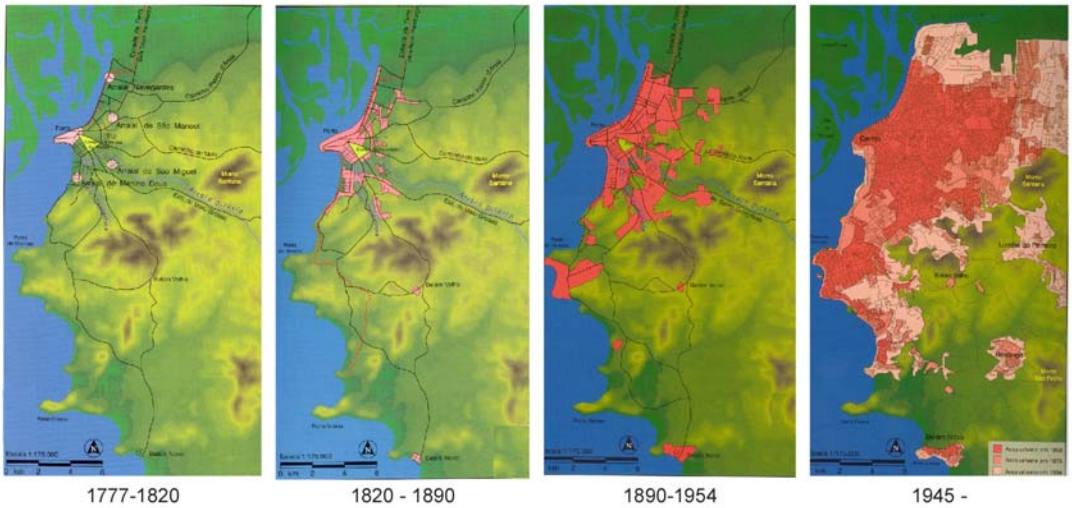


Figure 8.11 - Urban evolution of Porto Alegre. Source: Menegat et al (1998, page 100-101).

8.4.1 Exercises

Using the snapshot presented in Figure 8.10 as the initial condition for the simulation, two sets of run were carried out using the settings for the parameters given below, one using module one of the Peripherisation Model and the second using the inner city process module.

It is important to keep in mind that the objective here is not to fit the model precisely to reality, but rather to check what are the impacts of spatial constraints on the tendencies shown by the model in previous experiments, and how they can be related to reality.

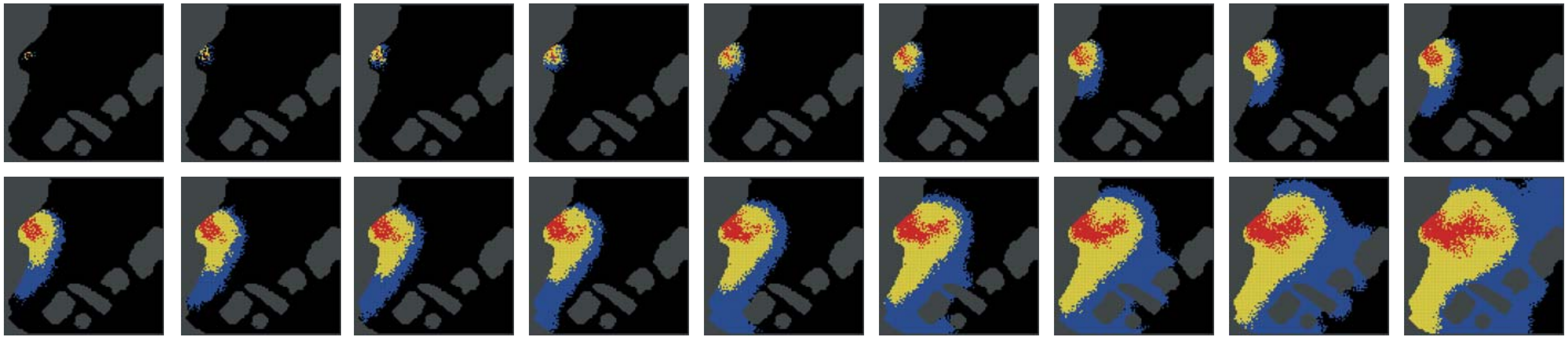


Figure 8.12 - Exercise A, sequence of snapshots for Porto Alegre.

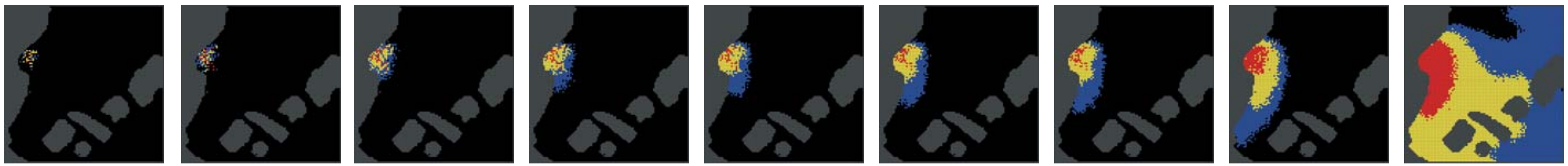


Figure 8.13 - Exercise B, sequence of snapshots for Porto Alegre.

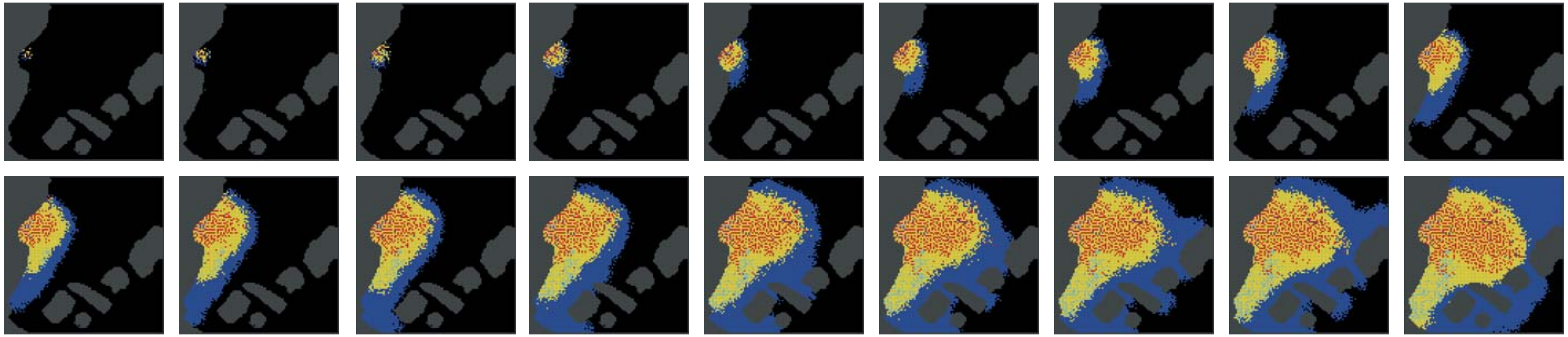


Figure 8.14 - Exercise C, sequence of snapshots for Porto Alegre.

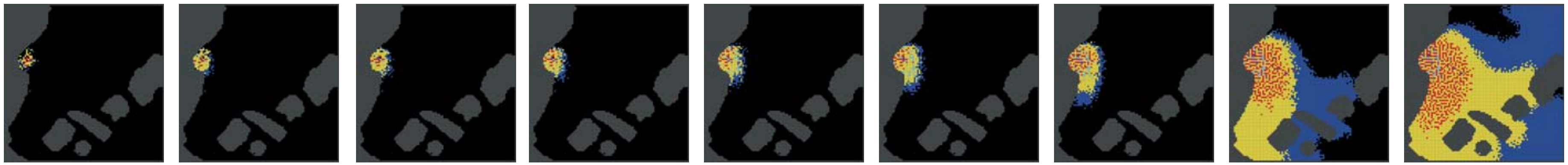


Figure 8.15 - Exercise D, sequence of snapshots for Porto Alegre.

The figures above (Figure 8.12, Figure 8.13, Figure 8.14, and Figure 8.15) show some of the exercises whose outcomes seem to be relevant to the real Porto Alegre. The first sequence of snapshots was run with the Peripherisation Model's module one, using parameter *steps* equal to 2 and the *proportion of agents per economic group* as 10% high-income, 40% middle-income, 50% low-income. Two sets of exercises were conducted using the same settings: exercise A (Figure 8.12) using the initial seed at coordinates (33, 15) and agents' initial location at the grid centre - coordinates (50, 50); and exercise B (Figure 8.13) using coordinates (33, 15) for both initial seed and all agents' initial location.

The results of these two exercises differ from each other, but it is interesting that each of them generates patterns of development that broadly resemble the settlement in reality.

The same two exercises were conducted with the inner city process module activated. For these sequences, the parameters used were: *proportion of agents per economic group* = 10, 40, 50; *steps* = 2; *steps2* = 4; *steps3* = 2; *d* = 4. The time-dependent parameters, *decayStartPoint* and *consolidationLimit* had to be different for each simulation because of the different time-scales of the exercises. As the final time for exercise C was 24708, the value used for *decayStartPoint* was 8000 and for *consolidationLimit* 6000, while for exercise D, with a final time 2627, *decayStartPoint* was 800 and *consolidationLimit* was 600.

Two clear general tendencies can be found in the spatial outcomes of these exercises. In exercises A and C, where the agents' initial location is the centre of the grid, the spatial location of the high-income group moves towards the right of the grid, differing from the concentric rings pattern. In exercises B and D, where the initial location of agents was the same as the initial seed, the pattern tends to spread around the spatial constraint boundaries. The interesting point of these two developments is that both of them show similarities to reality, as demonstrated in the next section.

8.4.2 Comparison with reality

The maps in Figure 8.16 show the distribution of income per census sector in the urbanized area of Porto Alegre. Like the maps presented in the previous sections, the data used are the average of the head of household's monthly income per census sector (enumeration district / census block) normalised by the number of householders in each sector and then classified using quantiles (maps A and B) or natural breaks (maps C and D) into three ranges (maps A and C) and six ranges (maps B and D). Maps shown in Figure 8.16 do not encompass the metropolitan area of Porto Alegre, but are restricted to the administrative boundaries of the city.

The maps in Figure 8.16 show different distributions of income for Porto Alegre. As in the maps presented in section 8.1.1, no strict definitions for income groups are given.

Analysing the set of four maps together, it is possible to observe that high-income areas tend to move from the initial seed location (also the historical point of origin of the city, as can be seen in Figure 8.11) towards the right as well as in a scattered pattern along the river, towards the south.

8.4.3 Discussion

The exercises with spatial constraints make clear the importance of those elements to a more realistic simulation outcome. They also show that real cities are not composed in perfect concentric rings, but can be understood to some extent according to these kinds of theories, while obeying spatial constraints. The role of spatial constraints in shaping urban morphology has been explored using simulation exercises using established theories of locational development such as von Thünen's model of agricultural change (see Steadman, 1999).

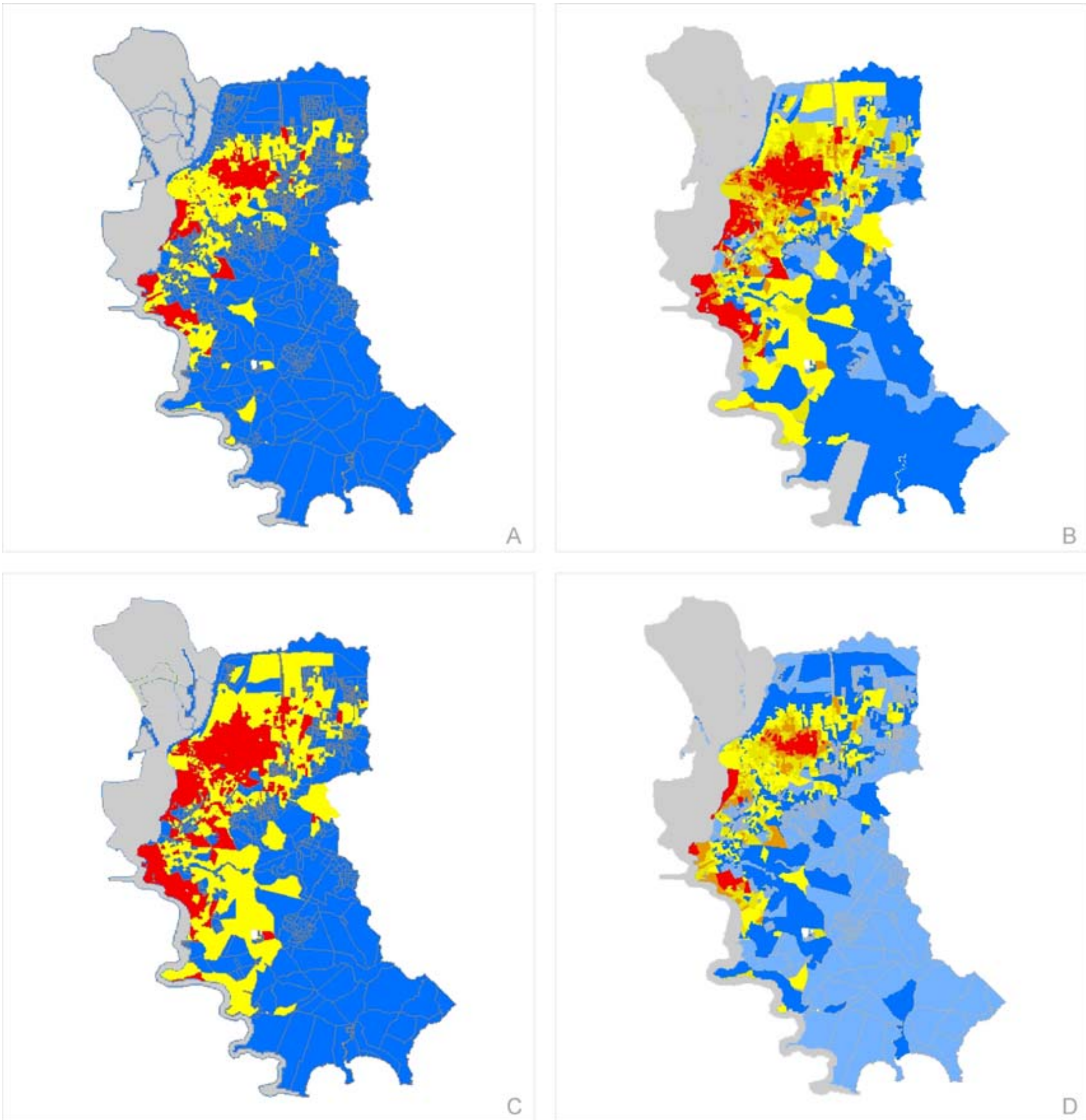


Figure 8.16 - Maps of Porto Alegre showing distributions of income in the urban area. Maps A and B were built, respectively, using 3 and 6 quantile breaks and maps C and D were built using, respectively, 3 and 6 natural breaks.

The exercises presented in this section showed that a change in the initial position of agents has an impact on the spatial outcomes of simulations. Interestingly, both results were relevant when compared to maps of reality. This shows that there are more features of the model worth exploring. The initial position of agents is simply a way of initiating the random walk that generates agents' behaviour in the simulation exercises. The random walk is a technique to introduce stochastic processes within the model and is not intended to be entirely realistic.

The striking point demonstrated by the spatial constraints exercises is how the simple introduction of unreachable areas within the grid can shape the spatial development in such different ways. This suggests that this module of the model should be further developed. The introduction of a number of different constraints such as slopes, natural reserves, and hills, together with changes in the rules could produce even more realistic results for the simulation. As mentioned in section 8.2, the location of spontaneous settlements, for example, could benefit from this kind of feature.

8.5 Discussion and summary

This chapter investigated the dynamics of Latin American cities using simulation exercises. The outcomes of each of the four modules of the Peripherisation model were discussed and related to typical urban problems in those cities.

Exercises with the Peripherisation module, which is the basis for the model, showed that some assumptions about urban growth in Latin American cities have to be questioned and further investigated. The simulation exercises made evident that the research community must review the causes driving the spatial patterns of those cities and this knowledge must be fed back to urban planning practice.

The spontaneous settlements exercise explored the impact of a consolidation of the spontaneous settlement process within the global urban growth process of Latin American cities. It shows that spontaneous settlements are part of the dynamics of those cities and suggests that their locational process must be further investigated.

The exercise with the inner city processes module examined the nature of inner city processes through a comparative analysis of their impact in Western and Latin American cities. The outcomes of this exercise suggested that inner city processes in those two kinds of cities are essentially of the same nature and differ only in degree. It also made evident that these differences are mostly due to the economic composition of the society in each case, which determines the magnitude of the impact of a given inner city process on the general urban spatial pattern.

Finally, the exercises with spatial constraints demonstrated the importance of those elements in order to understand urban morphology. The outcomes of the exercises make it clear that simple processes can form complex spatial structures if constraint by spatial elements that shape the form of the urban development.

The main idea of the Peripherisation Model is concentrated on module one, which, therefore, forms the main stream of this thesis. This is because the module is based on the rather simple idea that residential locational patterns in Latin American cities can be explained by essentially two concepts: the first being the idea that the composition of society, or how society is divided in groups, has a great impact on spatial development; and the second being that restrictions rather than preferences generate the spatial pattern. Once these two factors are established, urban development seem to lock itself into a vicious circle in which high-income groups get always the best locations while low-income groups are pushed away from all urban facilities.

Examining the kind of government housing typically provided in Latin American cities, which consists mainly of housing tracts for low-income groups located in the vast majority of cases in the urban fringe, it is evident that governments have acted without a knowledge of the global dynamics of the urban system, and therefore, reinforced the current dynamics, attracting more low-income groups to the outskirts of the city. As such, the need for centrally located housing for low-income groups was not understood. Similarly, government policies for spontaneous settlements have disregarded the dynamics of the global process of which their location and evolution is part. Because of the absence of available housing in central areas, together with the lack of good transport systems, spontaneous settlements continue to be a reasonably good alternative for low-income citizens. Yet urban interventions continue to approach the problem from a static and local point of view, and although upgrading interventions by urban governance have supported the natural upgrading process, those intervention programmes have not dealt with the problem as a whole, and new spontaneous settlements continue to mushroom on the urban fringes and in other unoccupied areas within the city. From this point of view, it seems that rather than not being able to cope with the housing demand, urban governments do not have the appropriate knowledge to deal with the situation.

It seems that the major planning problem in Latin American cities is of how to stop such a process, once it has been initiated. The role of the present investigation is not to answer this question, but to attempt to raise alternative points of view and speculate about urban development in those cities on the basis of the simulation experiments. It is important to note that the simulation exercises provide insights provoking debate, not only when the simulation results accord with reality, but even when they are not and the modeller is obliged to look for further explanations. Hence, the findings from this chapter are neither conclusive nor proven. Rather, they draw attention to gaps in our knowledge of urban development of Latin American cities that deserve further investigation.

The dynamic modelling exercises presented in this chapter have helped to further develop an understanding of the rapid urbanisation process and its dynamics, changing the perspective on the problem from a demographic and static viewpoint to a dynamic and morphological one. The findings of this chapter have thus taken a step in the direction of bringing a new perspective to an old problem.

Chapter 9

Conclusions

This is the concluding chapter of the thesis, in which the possibilities for future work are discussed and the main contributions are summarised.

9.1 Future work

From the research developed in this thesis, a number of options for future work emerged. These are discussed in the next sections.

9.1.1 Extensions of the Peripherisation Model

Once the main framework of a simulation model has been developed and implemented, as is the case of the Peripherisation Model, it is relatively easy to introduce new features. The Peripherisation Model can be extended in a number of ways, but during the development of the model some features emerged as of special relevance. These are described below.

9.1.1.1 Introduction of an economic framework

As discussed in Chapter 6, although the model is not formally an economic model, the logic underlying the Peripherisation Model is, to some extent, an economic one. However, this is one of the avenues of research where the further development of the Peripherisation Model could lead. To build a formal economic logic would not only be an option to further develop the simulation model, but would also be a way of formalising the theoretical framework developed in this thesis. The next two items on this list could also fit as part of this economic model framework: the introduction of the housing ladder concept, and the development of a matrix of change in land value.

9.1.1.2 Introduction of the housing ladder concept

The introduction of the concept of housing ladder into the model, as part of the agent's rules would improve the inner city module of the Peripherisation Model. The concept of housing ladder is related to the idea of social mobility, and refers to the way people live in different types of housing as their lifestyle changes and as they move up in the social ladder. The idea is that neighbourhoods and the types of housing in them shape the way we organize our society and its social structure (Husock, 1996) .

9.1.1.3 Development of a matrix of change in land value

This would consist of incorporating environment-environment relationships into the model, by adding a neighbourhood change effect through a CA or other diffusion process. This would be part of a set of major changes in the model's rules towards a formal economic conceptual framework. Instead of a simply locational preference for a certain location, the agent's preferences and restrictions would also be guided by a matrix of land values. Unwanted areas like hills and slopes, for instance, would start off with low impact values, even if located in central areas. This would allow a more realistic simulation of the locational process of spontaneous settlements, for instance. Once a cell was occupied by different income-groups, its neighbourhood would suffer the effects of a gain or loss in value, depending on the impact of the economic group of the agent which settled in the area. This would allow a more refined locational simulation, and would permit a more realistic approach to the peripherisation phenomenon. This could account for some of the complexities of the land use change ignored so far in the model.

9.1.1.4 Introduction of the effects of historical change

The impact of social and technological changes, such as those in transport technology, for example, is another aspect which could be incorporated into the model. It is well known that these kinds of changes have major impacts in the spatial development of cities, but it is not always simple to introduce them into models. The simplest way to implement these kinds of changes would be to treat them as exogenous factors, implemented, for example, as pre-established spatial conditions. However, it would be more desirable to implement them as internal mechanisms of change. A number of issues would be involved in the implementation of historical changes in the Peripherisation Model, including questions regarding the spatial and time scales. This topic deserves further investigation, although the benefits that such extensions would bring would have to be evaluated.

9.1.1.5 Use of real data

The use of real data as input is usually seen as a step towards a more realistic model, and therefore this topic deserves further investigation. A number of data sources could provide input for the Peripherisation Model, including data on roads, topographic information, historical data, and income data. However, in models with exploratory objectives like the Peripherisation Model, the outcomes obtained with addition of real data might not result in greater realism. It seems that the use of real data as initial conditions for spatial constraints, for instance, would certainly be useful. In any other case, the kind of data should be considered carefully, and the benefits evaluated.

It must be noted that extensions to a simulation model like the Peripherisation Model mean that each new factor added will have associated with it parameters that need to be set. Since these parameters are set according to the values that produce outputs that best fit the data, there is a risk of losing control of the process-based understanding that models of this sort help us to grapple with. The same kind of concerns should be kept in mind regarding the use of real data. Extra information given by real data not always contribute to a more realistic model, nor to improve the understanding of the process. Hence, it must be kept in mind whether the main interest is in fitting the data or understanding the process.

9.1.2 Variations in the evaluation method

As discussed in Chapter 5, the evaluation and validation of agent-based models are still much-debated issues, and their improvement depends largely on the further development of the category of ABS/LUCC. However, some variations on the evaluation methods used in the present research are possible and could result in interesting outcomes for the further development of this investigation.

9.1.2.1 Use of time-series data

In regard to the evaluation of the Peripherisation Model and given the present stage of its development, a possible evaluation would be to compare the outcomes of the model to real urban time-series data, such as series of remote sensed data from different time periods. This was not possible during the development of the present thesis due to difficulties in obtaining such data. However, it seems in general to be possible to obtain this kind of data, and its use would greatly improve the confidence on the model. Moreover, this analysis would provide substantial feedback to the further development of the model, bringing up a number of new features to be incorporated, and therefore helping in the process of understanding the dynamics of Latin American cities.

9.1.2.2 *Use of landscape metrics to compare results with reality*

The model's results could be evaluated using quantitative methods to compare characteristics of the simulated landscape with maps built from real data. The kind of data to use for this purpose, however, should be carefully selected. Census data such as those used for the maps presented in Chapter 8, for instance, are not suitable for this purpose, since the spatial representation has a pre-defined shape determined by the design of census tracts. Landscape metrics would not provide accurate results.

Once the appropriate kind of data is selected, other concerns should be carefully considered. The spatial similarity between the model results and real maps might not be close because of quite other reasons besides the lack of success of the model. The Peripherisation Model, like any model, is a simplification of a phenomenon and thus fails to consider several important aspects of reality which might have large influences on the spatial characteristics of the real system. On the other hand, even if the outcomes of the model do match the map in meaningful ways, this does not necessarily mean that the processes contained in the model are correct.

Considering these aspects, one should keep in mind that the further development of evaluation methods should contribute to an understanding of new aspects of reality, and should provide feedback for further development of the model. In other words, evaluation is part of the development process of the model and should contribute to the development of the understanding of reality, rather than serving just as a validation method.

9.1.3 Applicability for other Third World cities

In developing this thesis, similarities were evident between dynamic processes and spatial patterns in Latin American cities and other Third World cities. Although the scope of the thesis was limited to Latin American cities, an interesting development of the research framework would be to study the applicability of the model to other kinds of cities, and extend and adapt the model to investigate similarities and differences between various Third World cities.

9.2 Limitations of the present study

This thesis has attempted to draw together two approaches, a more descriptive approach to urban growth and change in Latin American cities presented in Part I, and the simulation of these processes described in Part II.

The line that divides these two approaches is a fine one, and the limitations found during the course of the research were, at least in part, those concerned with the limits of simulation, or in other words, with how to learn about reality through simulation exercises. This question is much broader than the scope of this thesis. However, it is worth discussing briefly how some of these limitations impacted on the development of the study, as these can also be considered as a contribution of this investigation.

Simple models like the one described here are not intended to give a detailed and precise description of urban systems, but only to capture something of the conceptual dynamics. However, it is difficult to determine whether a simulation model is realistic and robust enough to provide useful information about reality. The simulation of complex systems using bottom-up approaches presents several difficulties, such as variations due to stochastic processes and feedbacks, as discussed in Chapter 7, which are difficult to handle. The research reported in Chapter 7 represents one way of addressing these issues, that is, by improving the knowledge of the model's typical behaviour in order to be able to address issues concerning reality with some confidence.

Besides these general issues, there are at least three specific limitations of the present study. First, this is clearly an experimental study, which intends to provoke further enquiry rather than be definitive about the nature of dynamics and spatial patterns in Latin America. Second, this thesis attempts to generalise and, as such, sacrifices the ambition to reproduce the nature of the process going on in any specific city in the cause of trying to understand them all better. Third, this investigation is based on a simulation model which reproduces aspects of reality through very simple spatial interaction rules. It therefore ignores a whole set of theories and socio-economic implications, and much of the complexity that characterises the real urban system.

Despite making an oversimplification of a complex reality, the Peripherisation Model allowed the analysis of dynamic processes and made it possible to draw hypotheses about the dynamics of urban growth and change in Latin American cities. These are part of the contributions of this thesis and are discussed in the next section.

9.3 Contributions of the thesis

The thesis has offered contributions of two kinds: in the simulation of urban dynamics, where an agent-based model was proved to be a suitable tool to explore urban dynamics in Latin American cities, and in the understanding of the urban dynamics and spatial patterns of Latin American cities themselves.

9.3.1 On the simulation of urban dynamics

In terms of the contribution to simulation techniques, the Peripherisation Model is a good example of an exploratory simulation model and the simulation exercises seem to be an effective way to explore aspects of reality. In addition, agent-based simulation proved to be a suitable technique to explore urbanisation issues at the conceptual level, and allowed spatial patterns, dynamics and social issues to be handled within the same conceptual and modelling framework. Furthermore, the simulation exercises brought to light some important aspects of reality, which are further discussed in the next section.

9.3.2 On Latin American cities

The development of a theoretical framework for Latin American urban dynamics, which relates processes to resultant global spatial patterns and urban morphology, presented in Part I, is itself a contribution. The literature of dynamics, spatial patterns, and urban morphology in these cities is not abundant. So to glean this material and present it together in an organised form is an useful achievement, since the assembled material can be used as a reference for future research in this field.

In addition, the results obtained from the experiments in Chapter 8 brought up important insights related to the spatial development of Latin American cities. The simulation exercises allowed the investigation of their dynamics, changing perspective on the problem from a demographic viewpoint to a dynamic and morphological one. Moreover, the experiments with the model made clear that the actual process of development of Latin American cities is determined by socio-economic inequalities that are reproduced in space by the locational process. The peripherisation process was initially caused by the high rates of urban growth in these countries but it is now consolidated as the normal process of development of these cities. The result is an emergent pattern of spatial segregation characterised by stark differences between core and periphery, which have become consolidated as the residential spatial pattern of Latin American cities. The perpetuations of both process and spatial pattern reinforce the social inequality, which was their cause in the first place, working in a vicious cycle.

The exercises with the model made clear that the actual development process in Latin American cities is a manifestation of socio-economic. The composition of the urban society is a key factor for the understanding of spatial patterns and processes in cities, and seems to play a major role in producing differences between different kinds of cities, as demonstrated in Chapter 8. The exercises also allowed the identification of similarities and differences in real-world inner city processes when comparing Western and Latin American cities.

The research identified issues and gaps in our knowledge of Latin American cities dynamics and spatial patterns. However, the findings of this research should be further investigated using evidence from reality.

The study of cities as complex systems provides a different perspective, from which is possible to study how new planning policies could drive the urbanisation of Latin American cities along a different path, rather than simply trying to control it. The objective of the study of the dynamics of complex systems is to locate points that offer opportunities for intervention. Brown and colleagues (forthcoming-a), in their study on path dependence using agent-based simulation for land-use change, argue that path dependence, at least in theory, creates the possibility of policy leverage, or in other words, intervention. They suggest that if it is known that different paths of development pattern are possible, then we might be able to influence the process, through policy and the use of what Holland (Holland, 1995) calls 'lever points', so that the most desirable path emerges. This is an interesting point that can help in making studies of urban systems, using the complexity approach, more applicable for planning actions.

9.4 **Summary**

The need for an increased understanding of urban spatial phenomena in cities of the Third World is essential in order to provide a basis for future planning actions and policies. The approach outlined in this study has taken a step in this direction, bringing a new perspective to an old problem. This study provides evidence that urban modelling tools can provide an appropriate basis for research on Latin American urban processes, and makes clear the need to approach the problem by relating morphology and dynamics, for which dynamic modelling provides the appropriate means.

References

Allen, P. & Sanglier, M. (1981), 'Urban Evolution, Self-Organization, and Decisionmaking'. *Environment and Planning A*, 13, 167-183.

Allen, P. M. (1997), *Cities and Regions as Self-Organizing Systems: Models of Complexity,* Gordon and Breach Science Publishers, Amsterdam.

Almeida, C. M., Monteiro, A. M. V., Camara, G., Soares-Filho, B. S., Cerqueira, G. C., Pennachin, C. L. & Batty, M. (2002), Empiricism and Stochastics in Cellular Automaton Modeling of Urban Land Use Dynamics. *CASA Working Papers*, available online at http://www.casa.ucl.ac.uk/working_papers/Paper42.pdf.

Alonso, W. (1964), 'The Form of Cities in Developing Countries'. *Papers Regional Science Association*, 13, 165-173.

Amato, P. (1968), *An Analysis of the Changing Patterns of Elite Residential Areas in Bogota, Colombia*, unpublished PhD Thesis, Cornell University, Ithaca.

Amato, P. (1970a), 'A Comparison: Population Densities, Land Values and Socioeconomic Class in Four Latin American Cities'. *Land Economics*, 46, 447-455.

Amato, P. (1970b), 'Elitism and Settlement Patterns in the Latin American City'. *Journal of American Institute of Planners*, 36, 96-105.

Anderies, J. M. (2002), The Transition from Local to Global Dynamics: A Proposed Framework for Agent-Based Thinking in Social-Ecological Systems, *in* Janssen, M., ed, *Complexity and Ecosystem Management: The Theory and Practice of Multi-Agent Systems,* Elgar, Cheltenham.

Arthur, W. B. (1994), Urban Systems and Historical Path Dependence, *in* Arthur, W. B., ed, *Increasing Returns and Path Dependence in the Economy,* The University of Michigan Press, Michigan.

Balbao, M. (1993), 'Urban Planning and the Fragmented City of Developing Countries'. *Third World Planning Review*, 15 (1), 23-35.

Balmann, A., Happe, K., Kellermann, K. & Kleingarn, A. (2002), Adjustment Costs of Agri-Environmental Policy-Switchings: An Agent-Based Analysis of the German Region Hohenlohe, *in* Janssen, M., ed, *Complexity and Ecosystem Management: The Theory and Practice of Multi-Agent Systems,* Elgar, Cheltenham.

Barros, J. & Alves Jr., S. (2003), Simulating Rapid Urbanisation in Latin American Cities, *in* Longley, P. and Batty, M., eds, *Advanced Spatial Analysis: The CASA Book of GIS,* ESRI Press, London.

Batty, M. (1998), 'Urban Evolution on the Desktop: Simulation Using Extended Cellular Automata'. *Environment and Planning A*, 30, 1943-1967.

Batty, M. (1999), 'A Research Programme for Urban Morphology'. *Environment and Planning B: Planning and Design*, 26, 475-476.

Batty, M. (2000), 'Less Is More, More Is Different: Complexity, Cities, and Emergence'. *Environment and Planning B: Planning and Design*, 27, 167-168.

Batty, M., Couclelis, H. & Eichen, M. (1997), 'Urban Systems as Cellular Automata'. *Environment and Planning B: Planning and Design*, 24, 59-164.

Batty, M., Desyllas, J. & Duxbury, E. (2002), The Discrete Dynamics of Small-Scale Spatial Events: Agent-Based Models of Mobility in Carnivals and Street Parades. *CASA Working Papers*, available online at http://www.casa.ucl.ac.uk/working_papers/Paper56.pdf.

Batty, M., Jiang, B. & Thurstain-Goodwin, M. (1998), Local Movement: Agent-Based Models of Pedestrian Flow. *CASA Working Papers*, available online at http:/ www.casa.ucl.ac.uk/working_papers.htm.

Batty, M. & Xie, Y. (1994), 'From Cells to Cities'. *Environment and Planning B: Planning and Design*, 21, s31-s48.

Batty, M., Xie, Y. & Sun, Z. (1999), Dynamics of Urban Sprawl. *CASA Working Papers Series*, available online at http://www.casa.ucl.ac.uk/sprawl.pdf.

Benenson, I. (1998), 'Multi-Agent Simulations of Residential Dynamics in the City'. *Computers, Environment and Urban Systems*, 22 (1), 25-42.

Benenson, I. (2004), Agent-Based Modeling: From Individual Residential Choice to Urban Residential Dynamics, *in* Goodchild, M. F. and Janelle, D. G., eds, *Spatially Integrated Social Science: Examples in Best Practice,* Oxford University Press, Oxford.

Benenson, I., Omer, I. & Hatna, E. (2002), 'Entity-Based Modeling of Urban Residential Dynamics: The Case of Yaffo, Tel Aviv'. *Environment and Planning B: Planning and Design*, 29, 491-512.

Berger, T. (2001), 'Agent-Based Spatial Models Applied to Agriculture: A Simulation Tool for Technology Diffusion, Resource Use Changes, and Policy Analysis'. *Agricultural Economics*, 25 (2-3), 245-260.

Besussi, E., Cecchini, A. & Rinaldi, E. (1998), 'The Diffused City of the Italian North-East: Identification of Urban Dynamics Using Cellular Automata Urban Models'. *Computers, Environment and Urban Systems*, 22, 497-523.

Box, P. (2002), Spatial Units as Agents: Making the Landscape an Equal Player in Agent-Based Simulations, *in* Gimblett, H. R., ed, *Integrating Geographic Information Systems and Agent-Based Modeling Techniques for Simulating Social and Ecological Processes,* Oxford University Press, New York and Oxford.

Box, P. & Sasaki, Y. (2003), 'Agent-Based Verification of Von Thünen's Location Theory'. *Journal of Artificial Societies and Social Simulation*, 6 (2), online at http://jasss.soc.surrey.ac.uk/6/2/9.html.

Brett, S. (1974), Low-Income Urban Settlements in Latin America: The Turner Model, *in* De Kadt, E. and Williams, G., eds, *Sociology and Development,* Tavistock Publications, London.

Brown, D. G., Page, S. E., Riolo, R. & Rand, W. (2004a), 'Agent Based and Analytical Modeling to Evaluate the Effectiveness of Greenbelts'. *Environmental Modelling and Software*, 19 (12), 1097-1109.

Brown, D. G., Page, S. E., Riolo, R. & Rand, W. (2004b), 'Agent Based and Analytical Modeling to Evaluate the Effectiveness of Greenbelts'. *Environmental Modelling and Software*, in press,

Brown, D. G., Page, S. E., Riolo, R., Zellner, M. & Rand, W. (forthcoming-a), 'Path Dependence and the Validation of Agent-Based Spatial Models of Land Use'. *International Journal of Geographical Information Science*, in press, online at http://www.cscs.umich.edu/sluce/publications/ijgis-sluce-final.pdf.

Brown, D. G., Walker, R., Manson, S. M. & Seto, K. (forthcoming-b), Modeling Land-Use and Land-Cover Change, *in* Turner, B. L. and Gutmann, G., eds, *Land Change Science: Observing, Monitoring, and Understanding Trajectories of Change on the Land Surface,* Kluwer, Dordrecht.

Burgess, E. W. (1925), The Growth of the City: An Introduction to a Research Project, *in* Park, R., et al., eds, *The City,* University of Chicago Press, Chicago.

Caldeira, T. (2000), *City of Walls: Crime, Segregation, and Citizenship in São Paulo,* University of California Press, Berkeley ; London.

Casti, J. L. (1997), *Would-Be Worlds: How Simulation Is Changing the Frontiers of Science,* Wiley, New York.

Clarke, K. C., Hoppen, S. & Gaydos, L. (1997), 'A Self-Modifying Cellular Automaton Model of Historical Urbanization in the San Francisco Bay Area'. *Environment and Planning B: Planning and Design*, 24, 247-261.

Clay, P. (1980), The Rediscovery of City Neighbourhoods: Reinvestment by Long-Time Residents and Newcomers, *in* Laska, S. and Spain, D., eds, *Back to the City: Issues in Neighbourhood Renovation,* Pergamon Press, New York.

Clichevsky, N. (2002), *Tierra Vacante en Ciudades Latinoamericanas,* Lincon Institute of Land Policy, Cambridge, MA.

Conway, D. (1985), 'Changing Perspectives on Squatter Settlements, Intraurban Mobility, and Constraints on Housing Choice of the Third World Urban Poor'. *Urban Geography*, 6 (2), 170-192.

Conway, D. & Brown, J. (1980), 'Intraurban Relocation and Structure: Low-Income Migrants in Latin American and Caribbean'. *Latin American Research Review*, 15 (3), 95-125.

Couclelis, H. (1997), 'From Cellular Automata to Urban Models: New Principles for Model Development and Implementation'. *Environment and Planning B: Planning and Design*, 24, 165-174.

David, N., Marietto, M. B., Sichman, J. S. & Coelho, H. (2004), 'The Structure and Logic of Interdisciplinary Research in Agent-Based Social Simulation'. *Journal of Artificial Societies and Social Simulation*, 7 (3), online at http://jasss.soc.surrey.ac.uk/7/3/4.html.

Dean, J. S., Gumerman, G. J., Epstein, J. M., Axtell, R., Swedlund, A. C., Parker, M. T. & McCarrol, S. (2000), Understanding Anazi Culture Change through Agent-Based Modeling, *in* Kohler, T. A. and Gumerman, G. J., eds, *Dynamics in Human and Primate Societies: Agent-Based Modeling of Social and Spatial Processes,* Oxford University Press, New York.

Deffuant, G., Huet, S., Bousset, J. P., Amon, G. & Weisbuch, G. (2002), Agent-Based Simulation of Organic Farming Conversion in Allier Département, *in* Janssen, M., ed, *Complexity and Ecosystem Management: The Theory and Practice of Multi-Agent Systems,* Elgar, Cheltenham.

Desyllas, J., Greene, M. & Hillier, B. (1997), Santiago de Chile: The Spatial Configuration of a Rapidly Growing City - Implications on the Quality of Urban Life, *in* '5th Conference on Computers in Urban Planning and Urban Management - CUPUM 97', Mumbai, India.

Ducrot, R., Le Page, C., Bommel, P. & Kuper, M. (2004), 'Articulating Land and Water Dynamics with Urbanization: An Attempt to Model Natural Resources Management at the Urban Edge'. *Computers, Environment and Urban Systems*, 28, 85-106.

Dwyer, D. J. (1975), *People and Housing in Third World Cities: Perspectives on the Problem of Spontaneous Settlements,* Longman, London.

Epstein, J. M. & Axtell, R. (1996), *Growing Artificial Societies: Social Science from the Bottom Up,* Brookings Institution Press, Washington, D.C.

Erickson, B. & Lloyd-Jones, T. (1997), 'Experiments with Settlement Aggregation Models'. *Environment and Planning B: Planning and Design*, 24, 903-928.

Ewing, R. (1997), 'Is Los Angeles-Style Sprawl Desirable?' *Journal of American Planning Association*, 63 (01), 107-126.

Eyre, L. A. (1972), 'The Shantytowns of Montego Bay, Jamaica'. *Geographical Review*, 62, 394-413.

Ferber, J. (1999), *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence,* Addison-Wesley, Harlow.

Ford, L. (1996), 'A New and Improved Model of Latin American City Structure'. *Geographical Review*, 86 (3), 437-440.

Garner, B. J. (1967), Models of Urban Geography and Settlement Location, *in* Chorley, R. J. and Haggett, P., eds, *Models in Geography,* Methuen, London.

Gilbert, A. (1987), Urban Growth, Employment and Housing, *in* Preston, D. A., ed, *Latin American Development: Geographical Perspectives,* Longman, Harlow, Essex, England.

Gilbert, A. & Ward, P. M. (1982), 'Residential Movement among the Poor: The Constraints on Housing Choice in Latin American Cities'. *Transactions of the Institute of British Geographers*, 7, 129-149.

Gimblett, H. R. (2002), *Integrating Geographic Information Systems and Agent-Based Modeling Techniques for Simulating Social and Ecological Processes,* Oxford University Press, New York and Oxford.

Griffin, E. & Ford, L. (1980), 'A Model of Latin American City Structure'. *Geographical Review*, 70, 397-422.

Gugler, J. (1996), *The Urban Transformation of the Developing World,* Oxford University Press, Oxford.

Hall, P. (1983), Decentralization without End? A Re-Evaluation, *in* Patten, J., ed, *The Expanding City: Essays in Honour of Professor Jean Gottmann,* Academic Press, London.

Hamnett, C. (1984), Gentrification and Residential Location Theory: A Review and Assessment, *in* Herbert, D. and Johnston, R. J., eds, *Geography and the Urban Environment,* John Wiley & Sons Ltd.,

Holland, J. H. (1995), *Hidden Order: How Adaptation Builds Complexity,* Addison-Wesley, Reading, Mass.

Hoyt, H. (1939), *The Structure and Growth of Residential Neighborhoods in American Cities,* U.S. Govt. print. off., Washington.

Husock, H. (1996), *Repairing the Ladder: Toward a New Housing Policy Paradigm,* Reason Public Policy Institute, available online at: http://www.rppi.org/ps207.pdf.

Janssen, M. (2002), *Complexity and Ecosystem Management: The Theory and Practice of Multi-Agent Systems,* Elgar, Cheltenham.

Johnston, R. J. (1973), 'Towards a General Model of Intra-Urban Residential Patterns: Some Cross-Cultural Observations'. *Progress in Geography*, 4, 84-124.

Klak, T. & Lawson, V. (1993a), 'An Argument for Critical and Comparative Reseach on the Urban Economic Geography of the Americas'. *Environment and Planning A*, 25 (8), 1071-1083.

Klak, T. & Lawson, V. (1993b), 'Dialogue on the Americas: Issues of Representation, Comparison, and Integration in Americanist Geography'. *Environment and Planning A*, 25 (8), 1068-1070.

Kliest, T. J. & Scheffer, H. R. (1981), 'John Turner's Theory of Intra-Urban Mobility and the African Reality'. *Tijdschrift voor Economic en Social Geografie*, 72 (5), 258-265.

Kohler, T. A., Kresl, J., Van West, C., Carr, E. & Wilshusen, R. H. (2000), Be There Then: A Modeling Approach to Settlement Determinants and Spatial Efficiency among Late Ancentral Pueblo Populations of the Mesa Verde Region, U.S. Southwest, *in* Kohler, T. A. and Gumerman, G. J., eds, *Dynamics in Human and Primate Societies: Agent-Based Modeling of Social and Spatial Processes,* Oxford University Press, New York.

Kool, M., Verboom, D. & Van Der Linden, J. (1989), 'Squatter Settlement Improvement and Displacement: A Review of Concepts, Theory and Comparative Evidence'. *Habitat International*, 13 (3), 187-199.

Kostof, S. & Tobias, R. (1991), *The City Shaped: Urban Patterns and Meanings through History,* Thames and Hudson, London.

Kowarick, L. (1993), *A Espoliação Urbana,* Paz e Terra, São Paulo.

Krugman, P. R. (1996), *The Self-Organizing Economy,* Blackwell Publishers, Cambridge, MA.

Ligtenberg, A., Bregt, A. K. & van Lammeren, R. (2001), 'Multi-Actor-Based Land Use Modelling: Spatial Planning Using Agents'. *Landscape and Urban Planning*, 56, 21-33.

Lindert, v. (1992), Social Mobility as a Vehicle for Housing Advancement? Some Evidence from La Paz, Bolivia, *in* Mathéy, K., ed, *Beyond Self-Help Housing,* Profil Verlag, London.

Loibl, W. & Toetzer, T. (2003), 'Modeling Growth and Densification Processes in Suburban Regions - Simulation of Landscape Transition with Spatial Agents'. *Environmental Modelling & Software*, 18, 553-563.

London, B. & Flanagan, W. G. (1976), Comparative Urban Ecology: A Summary of the Field, *in* Walton, J. and Masotti, L. H., eds, *The City in Comparative Perspective: Cross-National Research and New Directions in Theory,* Sage Publication, New York.

Mautner, Y. (1991), *The Periphery as a Frontier for the Expansion of Capital*, unpublished PhD Thesis, University College London, London.

Mautner, Y. (1999), A Periferia Como Fronteira de Expansão Do Capital, *in* Deák, C. and Schiffer, S., eds, *O Processo de Urbanização no Brasil,* EDUSP, São Paulo.

McGarigal, K. & Marks, B. J. (1995), *Fragstats: Spatial Pattern Analysis Program for Quantifying Landscape Structure,* General Technical Report PNW-351, USDA Forest Service.

Menegat, R., Porto, M. L., Carraro, C. C. & Fernandes, L. A. D., eds (1998), *Atlas Ambiental de Porto Alegre,* Ed. Universidade / UFRGS, Porto Alegre.

Miller, E. J., Hunt, J. D., Abraham, J. E. & Salvini, P. A. (2004), 'Microsimulating Urban Systems'. *Computers, Environment and Urban Systems*, 28, 9-44.

Mills, E. & Hamilton, B. (1994), *Urban Economics,* HarperCollins College Publishers, New York.

MIT Media Laboratory (2004), StarLogo 2.1 (Software). *Media Laboratory and Teacher Education Program - MIT*, available online at http://education.mit.edu/starlogo/.

Moudon, A. V. (1997), 'Urban Morphology as an Emerging Interdisciplinary Field'. *Urban Morphology*, 1 (1), 3-10.

Moulin, B., Chaker, W. & Gancet, J. (2004), 'Padi-Simul: An Agent-Based Geosimulation Software Supporting the Design of Geographic Spaces'. *Computers, Environment and Urban Systems*, 28 (4), 387-420.

Mukhija, V. (2001), 'Upgrading Housing Settlements in Developing Countries: The Impact of Existing Physical Conditions'. *Cities*, 18 (4), 213-222.

Otter, H. S., van der Veen, A. & de Vriend, H. J. (2001), 'ABLOoM: Locational Behaviour, Spatial Patterns, and Agent-Based Modelling'. *Journal of Artificial Societies and Social Simulation*, 4 (4), online at http://www.soc.surrey.ac.uk/JASSS/4/4/2.html.

Pacione, M. (2001), 'Models of Urban Land Use Structure in Cities of the Developed World'. *Geography*, 86 (2), 97-119.

Parker, D., Berger, T. & Manson, S. M. (2001), *Agent-Based Models of Land-Use and Land-Cover Change,* LUCC, Irvine, California, USA.

Parker, D., Manson, S. M., Janssen, M., Hoffmann, M. J. & Deadman, P. (2003), 'Multi-Agent Systems for the Simulation of Land-Use and Land-Cover Change: A Review'. *Annals of the Association of American Geographers*, 93 (2), 314-337.

Parker, D. C. (2000), *Edge-Effects Externalities: Theoretical and Empirical Implications of Spatial Heterogeneity*, unpublished PhD, University of California, Davis.

Payne, G. K. (1977), *Urban Housing in the Third World,* L. Hill, London.

Polhill, J. G., Gotts, N. M. & Law, A. N. R. (2001), 'Imitative Versus Non-Imitative Strategies in a Land Use Simulation'. *Cybernetics and Systems*, 32 (1-2), 285-307.

Portugali, J. (2000), *Self-Organization and the City,* Springer, Berlin.

Portugali, J., Benenson, I. & Omer, I. (1997), 'Social Cognitive Dissonance and Spatial Emergence in a Self-Organizing City'. *Environment and Planning B: Planning and Design*, 24, 263-285.

PRODABEL (2001), available online at www.belohorizonte.com.br.

Rand, W., Brown, D. G., Page, S. E., Riolo, R., Fernandez, L. E. & Zellner, M. (2003), Statistical Validation of Spatial Patterns in Agent-Based Models, *in* 'Proceedings of Agent Based Simulation 2003', Montpellier, France. Online at http://www.pscs.umich.edu/research/projects/sluce/publications/sluce-abs.pdf.

Resnick, M. (1994), *Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds,* MIT Press, Cambridge, Mass.

Saltelli, A., Chan, K. & Scott, E. M. (2000), *Sensitivity Analysis,* Wiley, New York.

Sargent, R. G. (2001), Some Approaches and Paradigms for Verifying and Validating Simulation Models, *in* 'Proceedings of the 2001 Winter Simulation Conference', Online at http://www.informs-cs.org/wsc01papers/013.PDF.

Schelhorn, T., O'Sullivan, D. & Thurstain-Goodwin, M. (1999), Streets: An Agent-Based Pedestrian Model. *CASA Working Paper Series*, available online at http://www.casa.ucl.ac.uk/working_papers.htm.

Schnore, L. F. (1965), On the Spatial Structure of Cities in the Two Americas, *in* Hauser, P. M. and Schnore, L. F., eds, *The Study of Urbanization,* Wiley, New York.

Secco, A. & Squeff, L. (2000), 'A Explosão Da Periferia'. *Veja*, (85), 86-93.

Semboloni, F., Assfalg, J., Severio, A., Gianassi, R. & Marsoni, F. (2004), 'Citydev, an Interactive Multi-Agents Urban Model on the Web'. *Computers, Environment and Urban Systems*, 28 (1-2), 45-64.

Shankland Cox Partnership (1977), *Third World Urban Housing: Aspirations, Resources, Programmes, Projects,* Overseas Division, Building Research Establishment, UK, London.

Sjoberg, G. (1960), *The Preindustrial City: Past and Present,* Free Press, New York.

Sjoberg, G. (1980), The Pre-Industrial City: A Backward Glance, a Forward Look, *in* Press, I. and Smith, M. E., eds, *Urban Place and Process: Readings in the Anthropology of Cities,* Macmillan, New York.

Smith, N. (1979), 'Gentrification and Capital: Practice and Ideology in Society Hill'. *Antipode*, 11 (3), 24-35.

Smith, N. (1982), 'Gentrification and Uneven Development'. *Economic Geography*, 58, 139-155.

Smith, N. & Williams, P., eds (1986), *Gentrification of the City,* Allen & Unwin, Boston.

Soto, H. d. (1989), *The Other Path: The Invisible Revolution in the Third World,* I. B. Tauris & Co Ltd, London.

Steadman, P. (1999), *Feeding the City: Von Thünen's Model of Agricultural Location,* Open University, Milton Keynes.

Torrens, P. (2003), *Simulating Sprawl: A Dynamic Entity-Based Approach to Modelling North American Suburban Sprawl Using Cellular Automata and Multi-Agent Systems*, unpublished PhD Thesis, University College London, London.

Turner, A. & Penn, A. (2002), 'Encoding Natural Movement as an Agent-Based System: An Investigation into Human Pedestrian Behaviour in the Built Environment'. *Environment and Planning B: Planning and Design*, 29 (4), 473-490.

Turner, J. (1968), 'Housing Priorities, Settlement Patterns, and Urban Development in Modernizing Countries'. *Journal of American Institute of Planners*, 34, 354-363.

Turner, J. (1988), An Introductory Perspective, *in* Turner, B., ed, *A Third World Case Book,* BCB/HFB, London.

Turner, J. F. C. (1967), 'Barriers and Channels for Housing Development in Modernizing Countries'. *Journal of American Institute of Planners*, 33 (3), 167-181.

Turner, M. G., Gardner, R. H. & O'Neill, R. V. (2001), *Landscape Ecology in Theory and Practice: Pattern and Process,* Springer, New York.

Turton, I. (2003), Modelling Land-Use Development Using Multi-Agent Systems, *in* 'Geocomputation 2003', Southampton. Online at http://www.geocomputation.org/2003/Abstracts/Turton_1_Abs.pdf.

Ulack, R. (1978), 'The Role of Urban Squatter Settlements'. *Annals of the Association of American Geographers*, 68 (4), 535-550.

UNCHS (1982), *Survey of Slum and Squatter Settlements,* Published for UNCHS (Habitat) by Tycooly International Pub., Dublin.

UNCHS (1995), *Los Asentamientos Humanos en America Latina y el Caribe,* CEPAL, Santiago de Chile.

UNCHS (1996), *An Urbanizing World: Global Report on Human Settlements,* Oxford University Press for the United Nations Centre for Human Settlements (Habitat), Oxford.

UNCHS (2003), *The Challenge of Slums: Global Report on Human Settlements,* Earthscan, London.

University of Chicago (2003), RePast 2.0 (Software). *Social Science Research Computing Program*, available online at http://repast.sourceforge.net/.

Valladares, L. & Coelho, M. P. (1995), Urban Research in Brazil and Venezuela: Towards an Agenda for the 1990s, *in* Stren, R., ed, *Urban Research in the Developing World,* University of Toronto, Toronto.

Villa, M. & Rodriguez, J. (1996), Demographic Trends in Latin America's Metropolises 1950-1190, *in* Gilbert, A. and United, N. U., eds, *The Mega-City in Latin America,* United Nations University Press, Tokyo.

Ward, P. M. (1993), 'The Latin American Inner City: Differences of Degree or of Kind?' *Environment and Planning A*, 25 (8), 1131-1160.

White, R. & Engelen, G. (1993), 'Cellular Automata and Fractal Form: A Cellular Modelling Approach to the Evolution of Urban Land-Use Patterns'. *Environment and Planning A*, 25, 1175-1199.

White, R. & Engelen, G. (1997), 'Cellular Automata as the Basis of Integrated Dynamic Regional Modelling'. *Environment and Planning B: Planning and Design*, 24, 235-246.

Wilson, A. (2000), *Complex Spatial Systems,* Prentice Hall, Singapore.

Wooldridge, M. J. (2002), *An Introduction to Multiagent Systems,* J. Wiley, Chichester.

Wu, F. (1998), 'An Experiment on the Generic Polycentric of Urban Growth in Cellular Automatic City'. *Environment and Planning B: Planning and Design*, 25, 731-752.

Wu, F. (2002), 'Complexity and Urban Simulation: Towards a Computational Laboratory'. *Geography Research Forum*, 22, 22-40.

Xie, Y. & Batty, M. (forthcoming), Integrated Urban Evolutionary Modeling, *in* Atkinson, P. M., et al., eds, *Geodynamics,* CRC Press, Boca Raton, FL.

Appendix A
Legend of colours used in the Peripherisation Model

The table below describes each of the colours used in the Peripherisation Model and their respective representation.

Colour used in the model	Representation
Red	Cells occupied by high-income agents
Yellow	Cells occupied by middle-income agents
Blue	Cells occupied by low-income agents
Cyan	Consolidated low-income cells
Green	Empty cells that were abandoned by high-income groups
Grey	Spatial constraints cells

Appendix B
Input and Output Files Formats

B.1 Input file

The only input file used in the Peripherisation Model is the one that contains the landscape information for spatial constraints. This consists of a text-formatted file (*spatialconstraints02.txt*) which contains a matrix with the number of rows and collumns equal to the model's grid space size in use, so that each value will correspond to the state of a cell in the model's grid. In the matrix, empty cells are represented by value 0 (zero) while spatial constraint cells (grey cells) are represented by value 1.

B.2 Output files

A number of data files can be exported from the Peripherisation Model. They are basically text files that contain information that can be read using Excel. The following describes each of these files.

- **Number of active agents per time**. This is saved in a text-formatted file (*active.txt*) at the end of the simulation, composed of five columns, the first being the time; the second the total number of active agents, followed by the number of high-income agents, middle-income agents, and low-income agents.
- **Number of occupied cells (inactive agents) per time**. This is saved in a text-formatted file (*inactive.txt*) at the end of the simulation, composed of seven columns, the first being the time; the second the total number of inactive agents,

followed by the number of inactive high-income agents, inactive middle-income agents, inactive low-income agents, abandoned cells (green), and consolidated cells. Spatial constraint cells (grey cells) are not counted here, as they have no statistical purpose.

- **Logarithm of time and occupied cells.** This is saved in a text-formatted file (*log-log.txt*) at the end of the simulation, composed of two columns, the first being the logarithm of time; the second the logarithm of the total number of inactive agents.
- **Derivative of occupied cells**. This is saved in a text-formatted file (*derivative.txt*) at the end of the simulation, composed of four columns, the first being the derivative of the total number of inactive agents, followed, respectively, by the derivative of number of high-income, middle-income and low-income inactive agents.
- **Density from the centre**. This is saved in a text-formatted file (*density.txt*) at the end of the simulation, composed of six columns, the first being the radius from centre (the number of rows will be equal to half the model's space size), followed by the number of occupied cells for each state, for each radius; that is, each column will correspond to one of the five possible occupied cell states.
- **Matrix of occupied cells**. This consists of a matrix the same size as the grid space used in the model, where each pair of rows and columns corresponds to a cell in the model's space. A value equal to 1, 2, 3, 4, 5 and 9 will be given to a corresponding cell with the following states, respectively, high-income, middle-income, low-income, abandoned cells, consolidated cells, and empty cells. Grey cells (spatial constraints) are not considered here, as they have no statistical value for the simulation result. The values are presented in a matrix format and saved as a text-formatted file (*fragstats.txt*).

Appendix C
The Peripherisation Model code in StarLogo

The code in StarLogo includes all four modules of the Peripherisation Model. The code was written in StarLogo version 2.0.2 for Windows. The code is divided into two set of procedures: *observer* and *turtle* procedures, presented below.

OBSERVER PROCEDURES

```
globals [time r]
patches-own [density cons age]

to setup
ca
switch-scheduler
create-vermelho a
ask-vermelho [setc red]
create-amarelo b
ask-amarelo [setc yellow]
create-azul c
ask-azul [setc blue]
ask-turtles [setup]
ask-patch-at 0 0 [setpc red]
set time 0
ask-patches [set age 0]
end

to setup&keep
ask-patches [if (pc = red) or (pc = yellow) or (pc = blue) or (pc = brown) or (pc =
   purple) or (pc = green) [setpc black]]
ask-turtles [die]
switch-scheduler
create-vermelho a
ask-vermelho [setc red]
create-amarelo b
ask-amarelo [setc yellow]
create-azul c
ask-azul [setc blue]
ask-turtles [setup]
```

```
ask-patch-at 0 0 [setpc red]
set time 0
ask-patches [set age 0]
end

to a
output ( number * .01 * %red )
end

to b
output ( number * .01 * %yel)
end

to c
output ( number * .01 * %blu )
end

to go
ask-turtles [walk]
ask-turtles [if (pc = grey) [bk steps seth random 360]]
ask-vermelho [if (pc = red) [settle_red show "estou indo para o settle_red"]]
ask-amarelo [if (pc = red) [settle_yellow show "estou indo para o settle_yellow"]]
ask-azul [if (pc = red) or (pc = green) [settle_blue show "estou indo para o
    settle_blue"]]
ask-turtles [ifelse (hide = 0) [st] [ht]]
settime (time + 1)
if (der = 0) [derivative]
if (decay = 0) [
    ask-patches [  if pc = red [
            set age (age + 1)
            if (age > i)  and (random 100) < 50 [sprout          [
                                    set breed red
                                    setc red
                                    checkdensity
                                    ifelse density > d [ stamp green die] [die]
                                    ]
                          ]
            ]
    ]
]; end of if decay = 0
if (consolidation = 0) [ask-patches [if pc = blue [set age (age + 1) if (age > j) [setpc
    purple]]]]
if totalpatches >= 5587 [stopall]; makes the entire simulation stop when total patches
    reaches 4000
;if time >= 2000 [stopall]; makes the entire simulation stop when global time reaches
    1000
end



to calculate_mass;
print "red"
set r 0
loop [
    set r (r + 1)
```

```
    print count-patches-with [((distance 0 0) < r) and (pc = red)]
    ;show r
    if r > 53 [mass_yellow]; 53 is the maximum size of the grid (it varies!)
]; closing loop
end

to mass_yellow
print "yellow"
set r 0
loop [
    set r (r + 1)
    print count-patches-with [((distance 0 0) < r) and (pc = yellow)]
    ;show r
    if r > 53 [mass_blue]
]; closing loop
end

to mass_blue
print "blue"
set r 0
loop [
    set r (r + 1)
    print count-patches-with [((distance 0 0) < r) and (pc = blue)]
    ;show r
    if r > 53 [mass_brown]
]; closing loop
end

to mass_brown
print "brown"
set r 0
loop [
    set r (r + 1)
    print count-patches-with [((distance 0 0) < r) and (pc = brown)]
    ;show r
    if r > 53 [stopall]
]; closing loop
end


to totalpatches
output ((count-pc blue) + (count-pc red) + (count-pc yellow) + (count-pc purple) +
    (count-pc green))
end

to logtotalpatches
output (log totalpatches 10)
end

to logtime
output (log time 10)
end
```

TURTLES PROCEDURES

```
breeds [vermelho amarelo azul]

to setup
setxy 0 0
seth random 360
end

to walk
rt random 20 lt random 20 fd 1
end

to checkdensity
ifelse pc = red [set density 1] [set density 0]
if (pc-towards 0 1) = red [set density (density + 1)]
if (pc-towards 90 1) = red [set density (density + 1)]
if (pc-towards 270 1) = red [set density (density + 1)]
if (pc-towards 180 1) = red [set density (density + 1)]
neighbourhood ==> maximum d = 4
if (pc-towards 45 1) = red [set density (density + 1)]
if (pc-towards 135 1) = red [set density (density + 1)]
if (pc-towards 225 1) = red [set density (density + 1)]
if (pc-towards 315 1) = red [set density (density + 1)]
neighbourhood ==> maximum d = 8
end

to forward_steps
loop [
  seth random 360
  rt random 20 lt random 20 fd steps
  ifelse (pc != grey) [stop] [bk steps seth random 360]
    ]
end

to forward_steps2
loop [
  rt random 20 lt random 20 fd steps2
  ifelse (pc != grey) [stop] [bk steps2 seth random 360]
   ]
end

to forward_steps3
loop [
  rt random 20 lt random 20 fd steps3
  ifelse (pc != grey) [stop] [bk steps2 seth random 360]
   ]
end

to settle_red
forward_steps
switch_density_red
  case pc
           [
```

```
            green [settle_red]
            purple [settle_red]
            red [settle_red]
            blue [hatch [stamp color setbreed azul setc blue settle_blue die] go_zero]
            yellow [hatch [stamp color setbreed amarelo setc yellow settle_yellow die]
            go_zero]
            black [stamp color go_zero]
            grey [show "passou 1"]
            ]
end

to settle_red2
forward_steps2
case pc
            [
            green [settle_red2]
            purple [settle_red2]
            red [settle_red2]
            blue [hatch [stamp color setbreed azul setc blue settle_blue die] go_zero]
            yellow [hatch [stamp color setbreed amarelo setc yellow settle_yellow die]
            go_zero]
            black [stamp color go_zero]
            grey [show "passou 1"]
            ]
end

to settle_yellow
forward_steps
switch_density_yellow
  case pc
            [
            green [settle_yellow]
            purple [settle_yellow]
            red [ifelse age >= i [hatch [setbreed vermelho setc red settle_red die]
            go_zero] [settle_yellow]]
            yellow [settle_yellow]
            blue [hatch [stamp color setbreed azul setc blue settle_blue die] go_zero]
            black [stamp color go_zero]
            grey  [show "passou 2" stop]
            ]
end


to switch_density_red
ifelse (sdensity = 0) [
  checkdensity
  ifelse (density > d) [settle_red2] [stop]
  ]
  [stop]
end

to switch_density_yellow
ifelse (sdensity = 0) [
  checkdensity
  ifelse (density > d) [settle_yellow2] [stop]
```

```
  ]
  [stop]
end

to settle_yellow2
forward_steps3
  case pc
        [
        green [settle_yellow2]
        purple [settle_yellow2]
        red [ifelse age >= i [hatch [setbreed vermelho setc red settle_red die]
        go_zero] [settle_yellow]]
        yellow [settle_yellow]
        blue [hatch [stamp color setbreed azul setc blue settle_blue die] go_zero]
        black [stamp color go_zero]
        grey [show "passou 2" stop]
        ]
end

to settle_blue
forward_steps
if (consold = 0) [consolidate]
case pc
        [
        red [ifelse age >= i [hatch [stamp purple setbreed vermelho setc red
        settle_red die] go_zero] [settle_blue]]
        yellow [settle_blue]
        blue [settle_blue]
        black [stamp color go_zero]
        green [stamp blue go_zero]
        grey [show "passou 3" stop]
        purple [settle_blue]
        ]
end


to consolidate
if (pc = blue) [setcons (cons + 1)]
if ((pc = blue) and (cons > conslimit)) [stamp brown]
end

to go_zero
ifelse (zero = 0) [
  home
  seth random 360
        ]
[stop]
end
```

Appendix D
The Peripherisation Model code in FORTRAN

The code in FORTRAN 77 encompasses only the Peripherisation module and the spontaneous settlements module of the Peripherisation Model.

```
      programme cityofslum
      implicit none
      integer*4 totalnoturtles
      parameter (totalnoturtles = 5000)
      integer*4 turtle(1:totalnoturtles,1:4),globaltime
      common // turtle,globaltime
      integer*4 empty,red,yellow,blue,brown
      parameter (empty = 0, red = 1, yellow = 2, blue = 3, brown = 4)
      double precision probred,probyellow
      parameter (probred = 0.1D0, probyellow = 0.40D0)
      logical sequential
      parameter (sequential = .true.)
      integer*4 x,y,xc,yc,i
      integer*4 redpatches,yellowpatches,bluepatches,brownpatches
      integer*4 redturtles,yellowturtles,blueturtles
c
c     There are totalnoturtles number of turtles. The position of the i'th turtle
c     is x = turtle(i,1) and y = turtle(i,2), its direction is d = turtle(i,3) and
c     its breed (red, blue, or yellow) turtle(i,4).
c     The direction has to be one of 16 possible directions (in degrees
c     0,22.5,45,67.5,90, ... 337.5, 360 = 0).
c     The direction is specified by a number
c
c          1516 1 2 3
c          14       4
c          13   x   5
c          12       6
c          1110 9 8 7
c
      integer*4 L,conslimit
      parameter (L = 501, conslimit = 20)
      integer*4 patch(1:L,1:L),cons(1:L,1:L)
      common /land/ patch
c
c     The turtles are moving around on a patch of size LxL.
```

```
c     For computational reasons, L is odd so the center is
c     well defined.
c     The patch can be "empty", or occupied by a "red", "yellow"
c     a "blue" turtle or a consolidated blue turtle = "brown" turtle.
c
      character*25 fname1
      parameter (fname1 = 'patch.d')
      double precision ran1,rr,prob
      integer*4 iseed
c
c     Random number generator
c
      open(1,file=fname1,form='formatted',status='unknown')
      rewind(1)
      iseed = -46572131
      rr = ran1(iseed)
c
c     Initialisation of random number generator.
c
      do 10 y = 1,L
        do 15 x = 1,L
          patch(x,y) = empty
          cons(x,y) = 0
15      continue
10    continue
      xc = (L-1)/2
      yc = (L-1)/2
      patch(xc,yc) = red


c
c     The patch is initialized to be empty with a red turtle setteled
c     at the center (seed).
c
      redpatches = 1
      yellowpatches = 0
      bluepatches = 0
      brownpatches = 0
      redturtles = 0
      yellowturtles = 0
      blueturtles = 0
c
c     Helping variables to count the number of patches/turtles which/who
c     are red, yellow, blue, or (for patches only) which have consolidated
c     into brown. There are a total no. of "totalnoturtles" turtles.
c
      do 20 i = 1,totalnoturtles
        turtle(i,1) = xc
        turtle(i,2) = yc
        turtle(i,3) = INT(16.0D0*ran1(iseed))+1
c
c       Random integer number between 1 and 16 to determine direction.
c
        if (sequential) then
          if (i.le.int(totalnoturtles*probred)) then
```

```
            turtle(i,4) = red
            redturtles = redturtles + 1
          else if (i.le.int(totalnoturtles*(probred+probyellow))) then
            turtle(i,4) = yellow
            yellowturtles = yellowturtles + 1
          else
            turtle(i,4) = blue
            blueturtles = blueturtles + 1
          end if
        else
          prob = ran1(iseed)
          if (prob.le.probred) then
            turtle(i,4) = red
            redturtles = redturtles + 1
          else if (prob.le.(probred+probyellow))  then
            turtle(i,4) = yellow
            yellowturtles = yellowturtles + 1
          else
            turtle(i,4) = blue
            blueturtles = blueturtles + 1
          end if
        end if
c
c       With probability probred, the turtle is red.
c       With probability probyellow, the turtle is yellow.
c       With probability (1-probred-probyellow), the turtle is blue.
c
20    continue
      write(6,*) 'Redturtles = ',redturtles
      write(6,*) 'Yellowturtles = ',yellowturtles
      write(6,*) 'Blueturtles = ',blueturtles


c
c     Create turtles at (xc,yc) with a random direction.
c
      globaltime = 0
50    if (globaltime.ge.10000) goto 999
      if (MOD(globaltime,2000).eq.0) write(6,*) globaltime
      do 120 i = 1,totalnoturtles
        call walk(i)
120   continue
c
c     Ask all turtles i = 1, totalnoturtles to walk. Eqv. to ask-turtles [walk]
c
      do 220 i = 1,totalnoturtles
        if (turtle(i,4).eq.0) goto 220
        x = turtle(i,1)
        y = turtle(i,2)
        if (patch(x,y).eq.red) then
          call findspace(i)
        else
          goto 220
        end if
220   continue
```

```
c
c     Eqv. to ask-turtles [if (pc=red) [findspace]]
c
      globaltime = globaltime + 1
      do y = 1,L
        do x = 1,L
          if (patch(x,y).eq.blue) then
            cons(x,y) = cons(x,y)+1
            if (cons(x,y).ge.conslimit) patch(x,y) = brown
          end if
        end do
      end do
      goto 50
999   continue
      redpatches = 0
      yellowpatches = 0
      bluepatches = 0
      brownpatches = 0
      do y = 1,L
        do x = 1,L
          if (patch(x,y).eq.red) then
            write(1,*) x,y
            redpatches = redpatches + 1
          end if
        end do
      end do
      write(1,*) '  '
      do y = 1,L
        do x = 1,L
          if (patch(x,y).eq.yellow) then
            write(1,*) x,y
            yellowpatches = yellowpatches + 1
          end if
        end do
      end do
      write(1,*) '  '
      do y = 1,L
        do x = 1,L
          if (patch(x,y).eq.blue) then
            write(1,*) x,y
            bluepatches = bluepatches + 1
          end if
        end do
      end do
      write(1,*) '  '
      do y = 1,L
        do x = 1,L
         if (patch(x,y).eq.brown) then
           write(1,*) x,y
           brownpatches = brownpatches + 1
         end if
        end do
      end do
      write(6,*) 'Globaltime = ',globaltime
      write(6,*) 'Redpatches = ',redpatches
```

```
      write(6,*) 'Yellowpatches = ',yellowpatches
      write(6,*) 'Bluepatches = ',bluepatches
      write(6,*) 'Brownpatches = ',brownpatches
      close(1)
89     continue
      stop
      end

      function ran1(idum)
      implicit none
      double precision ran1,am,eps,rnmx
      integer*4 idum,ia,im,iq,ir,ntab,ndiv
      parameter (ntab = 32)
      integer*4 j,k,iv(ntab),iy
      save iv,iy
      data iv /ntab*0/, iy /0/
      ia = 16807
      im = 2147483647
      am = 1.0D0/dfloat(im)
      iq = 127773
      ir = 2836
      ndiv = 1 + (im-1)/ntab
      eps = 1.2D-7
      rnmx = 1.0D0 - eps
      if ((idum.le.0).or.(iy.eq.0)) then
        idum = max(-idum,1)
        do 803 j = ntab+8,1,-1
            k = idum/iq
            idum = ia*(idum-k*iq) - ir*k
            if (idum.lt.0) idum = idum + im
            if (j.le.ntab) iv(j) = idum
803       continue
        if (j.eq.0) j = 1
        iy = iv(j)
      end if
      k = idum/iq
      idum = ia*(idum-k*iq) - ir*k
      if (idum.lt.0) idum = idum + im
      j = 1 + iy/ndiv
      iy = iv(j)
      iv(j) = idum
      ran1 = min(am*iy,rnmx)
      return
      end


      subroutine walk(j)
c
c     Routine perform step of size 2 in turtle j's direction.
c
      integer*4 totalnoturtles
      parameter (totalnoturtles = 5000)
      integer*4 turtle(1:totalnoturtles,1:4),globaltime
      common // turtle,globaltime
      integer*4 j,dir
```

```
integer*4 L
parameter (L = 501)
dir = turtle(j,3)
select case (dir)
case (1)
  turtle(j,2) = turtle(j,2) + 2
  if (turtle(j,2).ge.(L+1)) turtle(j,2)=turtle(j,2)-L
case (2)
  turtle(j,1) = turtle(j,1) + 1
  turtle(j,2) = turtle(j,2) + 2
  if (turtle(j,1).eq.(L+1)) turtle(j,1)=turtle(j,1)-L
  if (turtle(j,2).ge.(L+1)) turtle(j,2)=turtle(j,2)-L
case (3)
  turtle(j,1) = turtle(j,1) + 2
  turtle(j,2) = turtle(j,2) + 2
  if (turtle(j,1).ge.(L+1)) turtle(j,1)=turtle(j,1)-L
  if (turtle(j,2).ge.(L+1)) turtle(j,2)=turtle(j,2)-L
case (4)
  turtle(j,1) = turtle(j,1) + 2
  turtle(j,2) = turtle(j,2) + 1
  if (turtle(j,1).ge.(L+1)) turtle(j,1)=turtle(j,1)-L
  if (turtle(j,2).eq.(L+1)) turtle(j,2)=turtle(j,2)-L
case(5)
  turtle(j,1) = turtle(j,1) + 2
  if (turtle(j,1).ge.(L+1)) turtle(j,1)=turtle(j,1)-L
case (6)
  turtle(j,1) = turtle(j,1) + 2
  turtle(j,2) = turtle(j,2) - 1
  if (turtle(j,1).ge.(L+1)) turtle(j,1)=turtle(j,1)-L
  if (turtle(j,2).eq.0) turtle(j,2)=turtle(j,2)+L
case (7)
  turtle(j,1) = turtle(j,1) + 2
  turtle(j,2) = turtle(j,2) - 2
  if (turtle(j,1).ge.(L+1)) turtle(j,1)=turtle(j,1)-L
  if (turtle(j,2).le.0) turtle(j,2)=turtle(j,2)+L
case (8)
  turtle(j,1) = turtle(j,1) + 1
  turtle(j,2) = turtle(j,2) - 2
  if (turtle(j,1).eq.(L+1)) turtle(j,1)=turtle(j,1)-L
  if (turtle(j,2).le.0) turtle(j,2)=turtle(j,2)+L
case(9)
  turtle(j,2) = turtle(j,2) - 2
  if (turtle(j,2).le.0) turtle(j,2)=turtle(j,2)+L
case(10)
  turtle(j,1) = turtle(j,1) - 1
  turtle(j,2) = turtle(j,2) - 2
  if (turtle(j,1).eq.0) turtle(j,1)=turtle(j,1)+L
  if (turtle(j,2).le.0) turtle(j,2)=turtle(j,2)+L
case(11)
  turtle(j,1) = turtle(j,1) - 2
  turtle(j,2) = turtle(j,2) - 2
  if (turtle(j,1).le.0) turtle(j,1)=turtle(j,1)+L
  if (turtle(j,2).le.0) turtle(j,2)=turtle(j,2)+L
case(12)
  turtle(j,1) = turtle(j,1) - 2
```

```
          turtle(j,2) = turtle(j,2) - 1
          if (turtle(j,1).le.0) turtle(j,1)=turtle(j,1)+L
          if (turtle(j,2).eq.0) turtle(j,2)=turtle(j,2)+L
        case(13)
          turtle(j,1) = turtle(j,1) - 2
          if (turtle(j,1).le.0) turtle(j,1)=turtle(j,1)+L
        case(14)
          turtle(j,1) = turtle(j,1) - 2
          turtle(j,2) = turtle(j,2) + 1
          if (turtle(j,1).le.0) turtle(j,1)=turtle(j,1)+L
          if (turtle(j,2).eq.(L+1)) turtle(j,2)=turtle(j,2)-L
        case(15)
          turtle(j,1) = turtle(j,1) - 2
          turtle(j,2) = turtle(j,2) + 2
          if (turtle(j,1).le.0) turtle(j,1)=turtle(j,1)+L
          if (turtle(j,2).ge.(L+1)) turtle(j,2)=turtle(j,2)-L
        case(16)
          turtle(j,1) = turtle(j,1) - 1
          turtle(j,2) = turtle(j,2) + 2
          if (turtle(j,1).eq.0) turtle(j,1)=turtle(j,1)+L
          if (turtle(j,2).ge.(L+1)) turtle(j,2)=turtle(j,2)-L
        case default
          Write(6,*) 'I am disorientated'
        end select
        turtle(j,3) = turtle(j,3) + INT(3.0D0*ran1(iseed))-1
        if (turtle(j,3).eq.17) turtle(j,3) = 1
        if (turtle(j,3).eq.0) turtle(j,3) = 16
c
c     Changes direction (0, +22.5, or -22.5 degrees)
c
        return
        end

        subroutine findspace(j)
c
c     Routine to find space.
c
        integer*4 totalnoturtles
        parameter (totalnoturtles = 5000)
        integer*4 turtle(1:totalnoturtles,1:4),globaltime
        common // turtle,globaltime
        integer*4 x,y,j,breed,dir
        integer*4 L
        parameter (L = 501)
        integer*4 patch(1:L,1:L)
        integer*4 empty,red,yellow,blue,brown
        parameter (empty = 0, red = 1, yellow = 2, blue = 3, brown = 4)
        common /land/ patch
        breed = turtle(j,4)
500     call walk(j)
        x = turtle(j,1)
        y = turtle(j,2)
        if (breed.eq.red) then
          if (patch(x,y.ne.red)) then
            call build(j)
```

```
            return
          else
            goto 500
          end if
        end if
        if (breed.eq.yellow) then
          if ((patch(x,y).ne.red).and.(patch(x,y).ne.yellow)) then
            call build(j)
            return
          else
            goto 500
          end if
        end if
        if (breed.eq.blue) then
          if (patch(x,y).eq.empty) then
            call build(j)
            return
          else
            goto 500
          end if
        end if
        return
        end

        subroutine build(j)
c
c      Routine to build.
c
        integer*4 totalnoturtles
        parameter (totalnoturtles = 5000)
        integer*4 turtle(1:totalnoturtles,1:4),globaltime
        common // turtle,globaltime
        integer*4 x,y,xc,yc,j,breed,dir
        integer*4 L
        parameter (L = 501)
        integer*4 patch(1:L,1:L)
        integer*4 empty,red,yellow,blue,brown
        parameter (empty = 0, red = 1, yellow = 2, blue = 3, brown = 4)
        common /land/ patch
        breed = turtle(j,4)
        xc = (L-1)/2
        yc = (L-1)/2
        x = turtle(j,1)
        y = turtle(j,2)
        if (patch(x,y).eq.empty) then
          patch(x,y) = breed
          turtle(j,4) = 0
        end if
        if ((breed.eq.red).and.(patch(x,y).eq.blue)) then
          patch(x,y) = breed
          turtle(j,4) = blue
          turtle(j,1) = xc
          turtle(j,2) = yc
        end if
        if ((breed.eq.red).and.(patch(x,y).eq.yellow)) then
```

```
            patch(x,y) = breed
            turtle(j,4) = yellow
            turtle(j,1) = xc
            turtle(j,2) = yc
          end if
          if ((breed.eq.yellow).and.(patch(x,y).eq.blue)) then
            patch(x,y) = breed
            turtle(j,4) = blue
            turtle(j,1) = xc
            turtle(j,2) = yc
          end if
          return
end
```

Appendix E
The Peripherisation Model code in JAVA – RePast

The *JAVA* code is divided in three main classes: model class, agent class and conditions class. The Model class contains all interface definitions, the main method of the program, the creation of all agents, and defines the basic actions of the model through the *execute* method. The Model class also contains a number of calculations for charts and data collection. The Agent class defines the agents, their attributes and actions. Finally, the Conditions class simply contains some of the conditions used throughout the simulation run, and was created to separate such methods from the methods of the Model class.

Model Class

```
package periproject;

import uchicago.src.sim.engine.SimInit;
import uchicago.src.sim.gui.DisplaySurface;
import uchicago.src.sim.space.Multi2DTorus;
import uchicago.src.sim.gui.MultiObject2DDisplay;
import uchicago.src.sim.analysis.OpenSequenceGraph;
import uchicago.src.sim.analysis.Sequence;
import uchicago.src.sim.analysis.*;
import uchicago.src.sim.util.SimUtilities;
import uchicago.src.sim.engine.*;
import uchicago.src.sim.util.Random;
```

```
import java.awt.*;
import java.lang.*;
import java.util.List;
import java.awt.Color.*;
import java.util.Hashtable;
import java.util.ArrayList;
import java.io.*;
import java.util.*;

public class Model extends SimModelImpl  {

  private Schedule schedule;
  private ArrayList agentList = new ArrayList();
  private Multi2DTorus space;

  ///Model parameters:
  private int spaceSize = 101;
  private int numAgents = 100;
  private int percentageBlue = 50;
  private int percentageYellow = 40;
  private int percentageRed = 10;
  public int findspaceSteps = 2;
  public int findspaceSteps2 = 4;
  public int findspaceSteps3 = 2;
  public int decayStartPoint = 800;
  public int decayRandom = 40;
  public int consolidationLimit = 600;
  public int consolidationRandom = 100;
  public int d = 3;
  public int consLimit = 4;

  //Activate / Desactivate inner city dynamics features
  boolean activateDecay = false;
  boolean activateConsolidation = false;
  boolean activateConsolidationOld = false;
  boolean activateOutskirtsRule = false;
```

```
//Activate / Desactivate graphs
boolean showLogLogPlot = false;
boolean showDensityPlot = false;
boolean showInactAgVsTime = false;
boolean showActAgVsTime = false;

//initial conditions
// one single seed at the centre
boolean centralSeed = false;
// 2 fixed seeds
boolean twoSeeds = false;
// 2 random seeds
boolean twoRandomSeeds = false;
//  4 random seeds
boolean fourRandomSeeds = false;
// 4 fixed seeds
boolean fourSeeds = true;
//grid colonial city
boolean gridSeeds = false;
// path
boolean pathSeeds = false;

//initial conditions with spatial constraints:
boolean spatialConstraints = false;
boolean spatialConstraintsPoa = true;

//Activate shuffle
boolean activateShuffle = true;

// The surface on which the agents and their environment are displayed
private DisplaySurface dsurf;

//GRAPH
// A graph that plots sequence
private OpenSequenceGraph actAgVsTime;
```

```
private OpenSequenceGraph inactAgVsTime;
private Plot logTimeVsLogInacAg;
private Plot densityPlot;
private OpenSequenceGraph derivVsTime;

//DATA RECORDERS
private DataRecorder recNumActAg;
private DataRecorder recNumInactAg;
private DataRecorder recLog;
private DataRecorder recDerivative;
static DataOutputStream outputDensity;
static DataOutputStream output;
static DataOutputStream outputASCIIGRID;

 //Activate / Desactivate output data files
 boolean activateRecNumActAg = true;
 boolean activateRecNumInactAg = true;
 boolean activateRecLog = true;
 boolean activateRecDerivative = true;
 boolean activateOutputDensity = true;
 boolean OutputMatrices = true;
 boolean outputASCIIfile = true;

 //Activate / desctivate snapshots
 boolean activateSnapshots = true;
 boolean activateSnapshotsByNumInactive = false;// takes snapshots every 1000 patches

 private Conditions aCondition;
 public Agent newAgent;
 Hashtable colorTable = new Hashtable();

public Model() {  }

//calculate the number of agent per group from the percentage given by parameters
int numBlue = (numAgents * percentageBlue / 100);
int numYellow = (numAgents * percentageYellow / 100);
int numRed = (numAgents * percentageRed / 100);
```

```
private void buildModel() {

  space = new Multi2DTorus(spaceSize, spaceSize, true);
  aCondition = new Conditions();

  // create the agents
  int a, b;
  // create the agents
  if (spatialConstraintsPoa){
  a = 15;
  b = 33;
  }
  else{
  a = (spaceSize / 2);
  b = (spaceSize / 2);
 }

  int index = 0;

  for (int i = 0; i <  numBlue; i++) {
    index = agentList.size()+ 1;
    Random.createUniform();
    int r = Random.uniform.nextIntFromTo(0, 360);
    Agent blue = new Agent(space, index, this);
    blue.setXY(a, b);
    blue.setActive(true);
    blue.setColor(Color.blue);
    blue.setAgentDirection(r);
    blue.setCodeColor('b');
    blue.setToOutskirts(false);
    agentList.add(blue);
  }

  for (int i = 0; i < numYellow; i++) {
    index = agentList.size()+ 1;
```

```
      Random.createUniform();
      int r = Random.uniform.nextIntFromTo(0, 360);
      Agent yellow = new Agent(space, index, this);
      yellow.setXY(a, b);
      yellow.setActive(true);
      yellow.setColor(Color.yellow);
      yellow.setAgentDirection(r);
      yellow.setCodeColor('y');
      yellow.setToOutskirts(false);
      agentList.add(yellow);
       }

    for (int i = 0; i < numRed; i++) {
      index = agentList.size()+ 1;
      Agent red = new Agent(space, index, this);
      Random.createUniform();
      int r = Random.uniform.nextIntFromTo(0, 360);
      red.setXY(a, b);
      red.setActive(true);
      red.setColor(Color.red);
      red.setCodeColor('r');
      red.setAgentDirection(r);
      red.setToOutskirts(false);
      agentList.add(red);
    }

//initial conditions
    if (centralSeed) {
      Agent meio = new Agent(space, index, this);
      meio.setXY(a, b);
      meio.setColor(Color.red);
      meio.setActive(false);
      meio.setCodeColor('r');
      meio.setToOutskirts(false);
      agentList.add(meio);
    }
```

```
if (twoRandomSeeds) {
  for (int i = 0; i < 2; i++) {
    int rx = Random.uniform.nextIntFromTo(0, spaceSize);
    int ry = Random.uniform.nextIntFromTo(0, spaceSize);
    Agent meio = new Agent(space, getAgentList().size() + 1, this);
    meio.setXY(rx, ry);
    meio.setColor(Color.red);
    meio.setActive(false);
    meio.setCodeColor('r');
    meio.setToOutskirts(false);
    agentList.add(meio);
  }
}

if (twoSeeds) {
  //seed one
  Agent firstAg = new Agent(space, getAgentList().size() + 1, this);
  //firstAg.setXY(35, 35);
  firstAg.setXY(35, 50);
  firstAg.setColor(Color.red);
  firstAg.setActive(false);
  firstAg.setCodeColor('r');
  firstAg.setToOutskirts(false);
  agentList.add(firstAg);

  //seed two
  Agent secondAg = new Agent(space, getAgentList().size() + 1, this);
  //secondAg.setXY(65, 65);
  secondAg.setXY(65, 50);
  secondAg.setColor(Color.red);
  secondAg.setActive(false);
  secondAg.setCodeColor('r');
  secondAg.setToOutskirts(false);
  agentList.add(secondAg);
}

if (fourRandomSeeds) {
```

```
  for (int i = 0; i < 4; i++) {
    int rx = Random.uniform.nextIntFromTo(0, spaceSize);
    int ry = Random.uniform.nextIntFromTo(0, spaceSize);
    Agent meio = new Agent(space, getAgentList().size() + 1, this);
    meio.setXY(rx, ry);
    meio.setColor(Color.red);
    meio.setActive(false);
    meio.setCodeColor('r');
    meio.setToOutskirts(false);
    agentList.add(meio);
  }
}

if (fourSeeds) {
//seed one
Agent firstAg = new Agent(space, getAgentList().size() + 1, this);
//firstAg.setXY(40, 40);
firstAg.setXY(50, 35);
firstAg.setColor(Color.red);
firstAg.setActive(false);
firstAg.setCodeColor('r');
firstAg.setToOutskirts(false);
agentList.add(firstAg);

//seed two
Agent secondAg = new Agent(space, getAgentList().size() + 1, this);
//secondAg.setXY(70, 50);
secondAg.setXY(35, 50);
secondAg.setColor(Color.red);
secondAg.setActive(false);
secondAg.setCodeColor('r');
secondAg.setToOutskirts(false);
agentList.add(secondAg);

//seed one
Agent thirdAg = new Agent(space, getAgentList().size() + 1, this);
//thirdAg.setXY(60, 80);
```

```
    thirdAg.setXY(65, 50);
    thirdAg.setColor(Color.red);
    thirdAg.setActive(false);
    thirdAg.setCodeColor('r');
    thirdAg.setToOutskirts(false);
    agentList.add(thirdAg);

    //seed four
    Agent fourthAg = new Agent(space, getAgentList().size() + 1, this);
    //fourthAg.setXY(45, 70);
    fourthAg.setXY(50, 65);
    fourthAg.setColor(Color.red);
    fourthAg.setActive(false);
    fourthAg.setCodeColor('r');
    fourthAg.setToOutskirts(false);
    agentList.add(fourthAg);
  }

    if (gridSeeds){
      for (int sx = 46; sx < 54; sx++){
        for (int sy = 48; sy < 52; sy++){
          if (sx == 46 || sx == 48 || sx == 50 || sx ==52 || sx == 54) {
            if (sy == 48 || sy == 50 || sy == 52) {
              Agent meio = new Agent(space, getAgentList().size() + 1, this);
              meio.setXY(sx, sy);
              meio.setColor(Color.red);
              meio.setActive(false);
              meio.setCodeColor('r');
              meio.setToOutskirts(false);
              agentList.add(meio);
            }
          }
        }
      }
    }//grid seeds

    if (pathSeeds){
```

```
    for (int i = 46; i < 55; i++){
      Agent meio = new Agent(space, getAgentList().size() + 1, this);
      meio.setXY(i, i);
      meio.setColor(Color.red);
      meio.setActive(false);
      meio.setCodeColor('r');
      meio.setToOutskirts(false);
      agentList.add(meio);
    }
  }

  if (spatialConstraints){
    //spatial constraints POA exercise
    readMatrix("spatialconstraint-poa4.txt");
    //spatial contraint spatialconstraint02:
    //readMatrix("spatialconstraint02.txt");
    //spatial contraint spatialconstraint04:
    //readMatrix("spatialconstraint04.txt");
  }

}

  private void buildDisplay() {

  MultiObject2DDisplay display = new MultiObject2DDisplay(space);

  display.setObjectList(agentList);
  //in case of spaceSize > 101:
  display.reSize(505,505);

  colorTable.put(Color.blue, "b");//low-income
  colorTable.put(Color.red, "r");// high-income
  colorTable.put(Color.yellow, "y"); // middle-income
  colorTable.put(Color.cyan, "c");// consolidated cells
  colorTable.put(Color.green, "g"); // abandoned central cells
  colorTable.put(Color.darkGray, "d");//spatial constraints
```

```
    dsurf.addDisplayableProbeable(display, "space");
    //set name of snapshot
    dsurf.setSnapshotFileName("snapshot");

 //GRAPHs
 //Active agents x time graph
if (showActAgVsTime){
   actAgVsTime.addSequence("total active agents", new CountNumActiveAgents(),
                        Color.BLACK, OpenSequenceGraph.FILLED_CIRCLE);
   actAgVsTime.addSequence("red active agents", new CountNumActiveRedAgents(),
                        Color.red, OpenSequenceGraph.FILLED_CIRCLE);
   actAgVsTime.addSequence("yellow active agents",
                        newCountNumActiveYellowAgents(),Color.yellow,
OpenSequenceGraph.FILLED_CIRCLE);
   actAgVsTime.addSequence("blue active agents",
                        new CountNumActiveBlueAgents(), Color.blue,
OpenSequenceGraph.FILLED_CIRCLE);

   actAgVsTime.setAxisTitles("time", "active agents");
   actAgVsTime.setXRange(0, 2000);
   actAgVsTime.setYRange(0, 200);
}

 //Inactive agents x time graph
if (showInactAgVsTime){

   inactAgVsTime.addSequence("total inactive agents",
                         new CountNumInactiveAgents(), Color.BLACK,
OpenSequenceGraph.FILLED_CIRCLE);
   inactAgVsTime.addSequence("red inactive agents",
                         new CountNumInactiveRedAgents(), Color.red,
OpenSequenceGraph.FILLED_CIRCLE);
   inactAgVsTime.addSequence("yellow inactive agents",
                         new CountNumInactiveYellowAgents(),
                         Color.yellow, OpenSequenceGraph.FILLED_CIRCLE);
   inactAgVsTime.addSequence("blue inactive agents",
```

```
                    new CountNumInactiveBlueAgents(), Color.blue,
OpenSequenceGraph.FILLED_CIRCLE);


   inactAgVsTime.setAxisTitles("time", "inactive agents");
   inactAgVsTime.setXRange(0, 2000);
   inactAgVsTime.setYRange(0, 200);
}
 //----------------------------------------end GRAPH

 ////RECORDERS
 //Recorder Number of Active Agents
 recNumActAg = new DataRecorder ("./activeag.txt", this, "Recorder Number of active
agents");
 recNumActAg.addNumericDataSource("total active agents", new
NumActiveAgentsSource());
 recNumActAg.addNumericDataSource("red active agents", new
NumActiveRedAgentsSource());
 recNumActAg.addNumericDataSource("yellow active agents", new
NumActiveYellowAgentsSource());
 recNumActAg.addNumericDataSource("blue active agents", new
NumActiveBlueAgentsSource());

//Recorder Number of Inactive Agents
recNumInactAg = new DataRecorder ("./inactiveag.txt", this, "Recorder Number of inactive
agents");
recNumInactAg.addNumericDataSource("total inactive agents", new
NumInactiveAgentsSource());
recNumInactAg.addNumericDataSource("red inactive agents", new
NumInactiveRedAgentsSource());
recNumInactAg.addNumericDataSource("yellow inactive agents", new
NumInactiveYellowAgentsSource());
recNumInactAg.addNumericDataSource("blue inactive agents", new
NumInactiveBlueAgentsSource());
recNumInactAg.addNumericDataSource("cyan inactive agents", new
NumInactiveCyanAgentsSource());
```

```
recNumInactAg.addNumericDataSource("green inactive agents", new
NumInactiveGreenAgentsSource());

//Recorder Number of Log time vs Log inactive agents
recLog = new DataRecorder ("./log.txt", this, "Recorder log-log");
recLog.addNumericDataSource("log Time", new LogTimeSource());
recLog.addNumericDataSource("log Inactive Agents", new LogTotalInactiveAgentsSource());

 //calculateMassPerTime
 //Recorder Derivative
 recDerivative = new DataRecorder ("./derivative.txt", this, "Recorder derivative");
 recDerivative.createNumericDataSource("derivative", this, "calculateMassPerTime");

   addSimEventListener(dsurf);
  }

  private void buildSchedule() {
    class ModelRunner extends BasicAction {
      Agent ag;

      public void execute() {
        // call the shuffleAgents method on this model
        if (activateShuffle) shuffleAgents();

       // on each Agent in the agentList
        for (int i = 0; i < agentList.size(); i++) {
          ag = (Agent) agentList.get(i);

          if (!ag.getActive()){
              int ran = Random.uniform.nextIntFromTo(0, 100);
              char agColor = ag.getCodeColor();
              //all agents age
              int agAge = ag.getAge();
              ag.setAge(agAge + 1);
              if (activateDecay) {
              if (agColor == 'r') {
                //if age value is higher than parameter i, decay rule is activated (also random):
```

```
          if ( (agAge >= decayStartPoint) && (ran <= decayRandom)) {
            //if age and density are higher than parameters i and j, then red agent leaves the
location
            int densityHere = ag.checkDensity();
            if (densityHere >= d) {
              System.out.println(ag.getWho() + " leaving from centre");
              ag.setActive(true);
              ag.setAge(0);// added 10 august
              ag.setToOutskirts(true);
              createGreenAgentHere(ag.getX(), ag.getY());
            }
          }
        }
      }//end of activate decay

      if (agColor == 'b') {
        //if consolidation boolean is true
        if (activateConsolidation){
          int ran2 = Random.uniform.nextIntFromTo(0, 100);
          //if age of agent is higher than consolidationLimit, consolidate agent!
          if ((agAge >= consolidationLimit)&& (ran2 <= consolidationRandom)) {
            ag.setCodeColor('c');
            ag.setColor(Color.cyan);
            //ag.setActive(false);
          }
        }
      }
    }// end - if inactive

      // call the walk method on all active agents
      boolean s = ag.getActive();
      if (s == true) {
        ag.walk();
        if (aCondition.isThereRedInactive(space, ag.getX(), ag.getY())) {
          ag.findSpace(findspaceSteps);
        }
      } //if
```

```
} //for

// updates the number of active agents
double numActBlue = countNumActiveBlueAgents();
double numActYell = countNumActiveYellowAgents();
double numActRed = countNumActiveRedAgents();

if (numActBlue < numBlue) {
  double numAgToCreate = numBlue - numActBlue;
  int i;
  for (i = 0; i < numAgToCreate; i++) {
    createNewAgent('b');
  }
}

if (numActYell < numYellow) {
  double numAgToCreate = numYellow - numActYell;
  int i;
  for (i = 0; i < numAgToCreate; i++) {
    createNewAgent('y');
  }
}

if (numActRed < numRed) {
  double numAgToCreate = numRed - numActRed;
  int i;
  for (i = 0; i < numAgToCreate; i++) {
    createNewAgent('r');
  }
}

dsurf.updateDisplay(); //display every tick

//update graphs
if (showActAgVsTime)
  actAgVsTime.step();
```

```
if (showInactAgVsTime)
  inactAgVsTime.step();

if (showLogLogPlot)
    plotLogLog();

//stops the simulation automatically if reaches maximum number of occupied cells
//it still has to be tested and final values corrected
if (spatialConstraints){
  int totalOccupiedCells = countNumInactiveAgents();
    if (totalOccupiedCells >= 9500) stop();//9000 for POA!
}
else{
  if (spaceSize == 51) {
    int totalOccupiedCells = countNumInactiveAgents();
    if (totalOccupiedCells >= 2000) stop();
  }
  if (spaceSize == 101) {
    //int totalOccupiedCells = countNumInactiveAgents();
    //if (totalOccupiedCells >= 8000) stop();
    //if western city then total occupied cells = 7000:
    //if (totalOccupiedCells >= 7000) stop();
  }
  if (spaceSize == 201) {
    int totalOccupiedCells = countNumInactiveAgents();
    if (totalOccupiedCells >= 30500) stop();
  }
  if (spaceSize == 301) {
    int totalOccupiedCells = countNumInactiveAgents();
    if (totalOccupiedCells >= 88500) stop();
  }
}//else

//take snapshots every 1000 patches
if (activateSnapshotsByNumInactive){
  int numInactiveNow = countNumInactiveAgents();
  double timeNow = getTickCountDouble();
```

```
if ((numInactiveNow >= 1000) & (numInactiveNow <=1100)) {
  dsurf.takeSnapshot();
  System.out.println(numInactiveNow + " time = " + timeNow);
}
if ((numInactiveNow >= 2000) & (numInactiveNow <=2100)) {
  dsurf.takeSnapshot();
  System.out.println(numInactiveNow + " time = " + timeNow);
}
if ((numInactiveNow >= 3000) & (numInactiveNow <=3100)) {
  dsurf.takeSnapshot();
  System.out.println(numInactiveNow + " time = " + timeNow);
}
if ((numInactiveNow >= 4000) & (numInactiveNow <=4100)) {
  dsurf.takeSnapshot();
  System.out.println(numInactiveNow + " time = " + timeNow);
}
if ((numInactiveNow >= 5000) & (numInactiveNow <=5100)) {
  dsurf.takeSnapshot();
  System.out.println(numInactiveNow + " time = " + timeNow);
}
if ((numInactiveNow >= 6000) & (numInactiveNow <=6100)) {
  dsurf.takeSnapshot();
  System.out.println(numInactiveNow + " time = " + timeNow);
}
if ((numInactiveNow >= 7000) & (numInactiveNow <=7100)) {
  dsurf.takeSnapshot();
  System.out.println(numInactiveNow + " time = " + timeNow);
}
if ((numInactiveNow >= 8000) & (numInactiveNow <=8100)) {
  dsurf.takeSnapshot();
  System.out.println(numInactiveNow + " time = " + timeNow);
}
if ((numInactiveNow >= 9000) & (numInactiveNow <=9100)){
  dsurf.takeSnapshot();
  System.out.println(numInactiveNow + " time = " + timeNow);
}
if ((numInactiveNow >= 10000) & (numInactiveNow <=10100)){
```

```
        dsurf.takeSnapshot();
        System.out.println(numInactiveNow + " time = " + timeNow);
      }
      if ((numInactiveNow >= 11000) & (numInactiveNow <=11100)) {
        dsurf.takeSnapshot();
        System.out.println(numInactiveNow + " time = " + timeNow);
      }
      if ((numInactiveNow >= 12000) & (numInactiveNow <=12100)) {
        dsurf.takeSnapshot();
        System.out.println(numInactiveNow + " time = " + timeNow);
      }
    }
  } //execute
} //basic action

// the schedule has been created in setup()
schedule.scheduleActionBeginning(0, new ModelRunner());
//display every 10 ticks - (need to take off the dsurf.updatedisplay method above)
//schedule.scheduleActionAtInterval(10, dsurf, "updateDisplay");//display every 10 ticks

if (showDensityPlot) schedule.scheduleActionAtEnd(this, "calculateDensity");
if (OutputMatrices)
schedule.scheduleActionAtEnd(this, "exportMatricesClusterIndentification");
//export files for analysis every 1000 ticks
schedule.scheduleActionAtInterval(1000, this,"exportMatricesClusterIndentification");
if (outputASCIIfile)
schedule.scheduleActionAtEnd(this, "exportASCIIfile");
//export files for analysis every 1000 ticks
//schedule.scheduleActionAtInterval(1000, this,"exportASCIIfile");

//call recorders
schedule.scheduleActionBeginning(0, new BasicAction(){
  public void execute(){
    recNumActAg.record();
    recNumInactAg.record();
    recDerivative.record();
    recLog.record();
```

```
    }
  });

//compute density and write to file if boolean is true (kind of recorder)
if (activateOutputDensity){
    schedule.scheduleActionAtEnd(this, "computeDensityTotal");
  }

  //write the recorder information to file at the end of simulation if the boolean is true
  if (activateRecNumActAg)
  schedule.scheduleActionAtEnd(recNumActAg, "writeToFile");
  if (activateRecNumInactAg)
  schedule.scheduleActionAtEnd(recNumInactAg, "writeToFile");
  if (activateRecDerivative)
  schedule.scheduleActionAtEnd(recDerivative, "writeToFile");
  if (activateRecLog)
  schedule.scheduleActionAtEnd(recLog, "writeToFile");

  //set snapshots every 250 ticks and at end of simulation
  if (activateSnapshots){
    schedule.scheduleActionAtInterval(100, dsurf, "takeSnapshot");
    schedule.scheduleActionAtEnd(dsurf, "takeSnapshot");
    schedule.scheduleActionAt(0, dsurf,"takeSnapshot");
    //schedule.scheduleActionBeginning(0, dsurf, "takeSnapshot");
  }

  if (activateSnapshotsByNumInactive){
    schedule.scheduleActionAtEnd(dsurf, "takeSnapshot");
    schedule.scheduleActionAt(0, dsurf,"takeSnapshot");
  }

  } //build schedule

  // Randomize the order of the object (the SugarAgents) in the agentList
  public void shuffleAgents() {
    SimUtilities.shuffle(agentList);
  }
```

```
public void begin() {
  buildModel();
  buildDisplay();
  buildSchedule();
  dsurf.display();

//---------------------------GRAPHS
  if (showActAgVsTime)
    actAgVsTime.display();

  if (showInactAgVsTime)
    inactAgVsTime.display();

  if (showLogLogPlot)
      logTimeVsLogInacAg.display();
}

public void setup() {
  schedule = null;
  agentList = new ArrayList();
  space = null;

  if (dsurf != null)
    dsurf.dispose();
  dsurf = null;

//-------------GRAPHS
if (showActAgVsTime){
    if (actAgVsTime != null) actAgVsTime.dispose();
    actAgVsTime = null;
  }

if (showInactAgVsTime){
    if (inactAgVsTime != null) inactAgVsTime.dispose();
    inactAgVsTime = null;
  }
```

```
if (showLogLogPlot) {
  if (logTimeVsLogInacAg != null) logTimeVsLogInacAg.dispose();
  logTimeVsLogInacAg = null;
}

if (densityPlot != null) densityPlot.dispose();
  densityPlot = null;

  System.gc(); //o que 'e isso? (explicacao acima)

  // create a schedule with an interval of one.
  schedule = new Schedule(1);
  dsurf = new DisplaySurface(this, "DisplaySurface");

  registerDisplaySurface("Display", dsurf);

//-----------GRAPHS
  if (showActAgVsTime)
    actAgVsTime = new OpenSequenceGraph("active agents vs time", this);

  if (showInactAgVsTime)
    inactAgVsTime = new OpenSequenceGraph("inactive agents vs time", this);

  if (showLogLogPlot)
      logTimeVsLogInacAg = new Plot("log time vs log total inactive agents");

  if (showDensityPlot)
    densityPlot = new Plot ("Density from the centre");

    derivVsTime = new OpenSequenceGraph("derivative (mass) vs time", this);
}

  // GETTERS AND SETTERS
  public int getNumAgents() {
    return numAgents;
  }
```

```
public void setNumAgents(int numAgents) {
  this.numAgents = numAgents;
}

public int getSpaceSize() {
  return spaceSize;
}

public void setSpaceSize(int spaceSize) {
  this.spaceSize = spaceSize;
}

public Multi2DTorus getSpace() {
  return space;
}

public int getPercentageBlue() {
  return percentageBlue;
}

public void setPercentageBlue(int perc) {
  this.percentageBlue = perc;
}

public int getPercentageYellow() {
  return percentageYellow;
}

public void setPercentageYellow(int perc) {
  this.percentageYellow = perc;
}

public int getPercentageRed() {
  return percentageRed;
}
```

```
  public void setPercentageRed(int perc) {
    this.percentageRed = perc;
  }

  public ArrayList getAgentList() {
    return agentList;
  }

  public int getFindspaceSteps() {
    return findspaceSteps;
  }

  public void setFindspaceSteps(int s) {
    this.findspaceSteps = s;
  }

//parameters modules 2 and 3
  public int getFindspaceSteps2() {
  return findspaceSteps2;
}

public void setFindspaceSteps2(int s) {
  this.findspaceSteps2 = s;
}

public int getFindspaceSteps3() {
  return findspaceSteps3;
}

public void setFindspaceSteps3(int s) {
  this.findspaceSteps3 = s;
}

public int getDecayStartPoint() {
  return decayStartPoint;
}
```

```
public void setDecayStartPoint(int s) {
  this.decayStartPoint = s;
}

public int getConsolidationLimit() {
  return consolidationLimit;
}

public void setConsolidationLimit(int s) {
  this.consolidationLimit = s;
}

public int getConsolidationRandom() {
  return consolidationRandom;
}

public void setConsolidationRandom(int s) {
  this.consolidationRandom = s;
}

public int getDecayRandom() {
  return decayRandom;
}

public void setDecayRandom(int s) {
  this.decayRandom = s;
}


public int getD() {
  return d;
}

public void setD(int s) {
  this.d = s;
}
```

```
public int getConsLimit() {
  return consLimit;
}

public void setConsLimit(int s) {
  this.consLimit = s;
}

  public String[] getInitParam() {
    String[] params = {
        "numAgents", "SpaceSize", "percentageBlue", "percentageYellow", "consLimit",
        "percentageRed", "findspaceSteps", "findspaceSteps2", "findspaceSteps3",
"decayStartPoint", "consolidationLimit", "d", "consolidationRandom", "decayRandom"};
    return params;
  }

  //////////END OF GETTERS AND SETTERS

  public void addNumActiveAgents(Agent ag) {
    int count = 0;
    Agent ag1;
    for (int i = 0; i < agentList.size(); i++) {
      ag1 = (Agent) agentList.get(i);
      if (ag1.getActive())
        count++;
    }
  }

  public void createNewAgent (char c) {
    int a, b;
  //create the agents
  if (spatialConstraintsPoa){
    a = 15;
    b = 33;
  }
  else{
    a = (spaceSize / 2);
```

```
    b = (spaceSize / 2);
  }

    int index = getAgentList().size() + 1;
    int r = Random.uniform.nextIntFromTo(0, 360);
    Agent extraAg = new Agent (space, index, this);
    extraAg.setXY(a, b);
    extraAg.setActive(true);
    extraAg.setAgentDirection(r);
    extraAg.setToOutskirts(false);

    if (c == 'r') {
      extraAg.setCodeColor('r');
      extraAg.setColor(Color.red);
    }

    if (c == 'y') {
      extraAg.setCodeColor('y');
      extraAg.setColor(Color.yellow);
    }

    if (c == 'b') {
      extraAg.setCodeColor('b');
      extraAg.setColor(Color.blue);
    }
    agentList.add(extraAg);
  }

  public void createGreenAgentHere(int a, int b){
    int index = agentList.size()+ 1;
    Agent greenAg = new Agent (space, index, this);
    greenAg.setXY(a, b);
    greenAg.setActive(false);
    greenAg.setColor(Color.green);
    greenAg.setCodeColor('g');
    agentList.add(greenAg);
    //System.out.println("green agent created at "+ a + "," + b );
```

```
 }

////MEASUREMENTS --- GRAPHS

///RECORDERS
  class NumActiveAgentsSource implements NumericDataSource {
  public double execute() {
    double count = 0;
    Agent ag1;
    for (int i = 0; i < agentList.size(); i++) {
      ag1 = (Agent) agentList.get(i);
      if (ag1.getActive())
      count++;
    }
   return count;
  }
}

//number of red active agents
class NumActiveRedAgentsSource implements NumericDataSource {
  public double execute() {
        double count = 0;
        Agent ag1;
        for (int i = 0; i < agentList.size(); i++) {
          ag1 = (Agent) agentList.get(i);
          if (ag1.getActive() && (ag1.getCodeColor()== 'r'))
            count++;
        }
        //System.out.println("number of active agents: "+ count);
        return count;
  }
}

//number of yellow active agents
class NumActiveYellowAgentsSource implements NumericDataSource {
  public double execute() {
    double count = 0;
```

```
    Agent ag1;
    for (int i = 0; i < agentList.size(); i++) {
      ag1 = (Agent) agentList.get(i);
      if (ag1.getActive() && (ag1.getCodeColor() == 'y'))
        count++;
    }
    return count;
  }
}

//number of blue active agents
class NumActiveBlueAgentsSource implements NumericDataSource {
  public double execute() {
    double count = 0;
    Agent ag1;
    for (int i = 0; i < agentList.size(); i++) {
      ag1 = (Agent) agentList.get(i);
      if (ag1.getActive() && (ag1.getCodeColor() == 'b'))
        count++;
    }
    return count;
  }
}

//number of inactive agents
    class NumInactiveAgentsSource implements NumericDataSource {
      public double execute() {
        double count = 0;
        Agent ag1;
        for (int i = 0; i < agentList.size(); i++) {
          ag1 = (Agent) agentList.get(i);
          if (ag1.getActive() == false)
            count++;
        }
        return count;
      }
    }
```

```
//number of red inactive agents
  class NumInactiveRedAgentsSource implements NumericDataSource {
   public double execute() {
    double count = 0;
    Agent ag1;
    for (int i = 0; i < agentList.size(); i++) {
     ag1 = (Agent) agentList.get(i);
     if ( (ag1.getActive() == false) && (ag1.getCodeColor() == 'r'))
      count++;
    }
    return count;
   }
  }

//number of yellow inactive agents
  class NumInactiveYellowAgentsSource implements NumericDataSource {
   public double execute() {
    double count = 0;
    Agent ag1;
    for (int i = 0; i < agentList.size(); i++) {
     ag1 = (Agent) agentList.get(i);
     if ( (ag1.getActive() == false) && (ag1.getCodeColor() == 'y'))
      count++;
    }
    return count;
   }
  }

//number of blue inactive agents
  class NumInactiveBlueAgentsSource implements NumericDataSource {
   public double execute() {
    double count = 0;
    Agent ag1;
    for (int i = 0; i < agentList.size(); i++) {
     ag1 = (Agent) agentList.get(i);
     if ( (ag1.getActive() == false) && (ag1.getCodeColor() == 'b'))
```

```
        count++;
      }
      return count;
    }
  }

  //number of cyan inactive agents
  class NumInactiveCyanAgentsSource implements NumericDataSource {
    public double execute() {
      double count = 0;
      Agent ag1;
      for (int i = 0; i < agentList.size(); i++) {
        ag1 = (Agent) agentList.get(i);
        if ( (ag1.getActive() == false) && (ag1.getCodeColor() == 'c'))
          count++;
      }
      return count;
    }
  }

//number of green inactive agents
    class NumInactiveGreenAgentsSource implements NumericDataSource {
      public double execute() {
        double count = 0;
        Agent ag1;
        for (int i = 0; i < agentList.size(); i++) {
          ag1 = (Agent) agentList.get(i);
          if ( (ag1.getActive() == false) && (ag1.getCodeColor() == 'g'))
            count++;
        }
        return count;
      }
    }

  class LogTimeSource implements NumericDataSource {
    private Model myModel;
    public double execute() {
```

```
        myModel = new Model();
        double logTime;
        double time;
        time = getTickCountDouble();
        logTime = Math.log(time);
        return logTime;
  }
}

    class LogTotalInactiveAgentsSource implements NumericDataSource {
      public double execute() {
        double logInactiveAgents;
        logInactiveAgents = Math.log(countNumInactiveAgents());
        return logInactiveAgents;
      }
    }
/////////END RECORDERS

     class CountNumActiveAgents implements Sequence {
      public double getSValue() {
        double count = 0;
        Agent ag1;
        for (int i = 0; i < agentList.size(); i++) {
          ag1 = (Agent) agentList.get(i);
          if (ag1.getActive())
          count++;
        }
      return count;
      }
    }

    //number of red active agents
    class CountNumActiveRedAgents implements Sequence {
      public double getSValue() {
            double count = 0;
            Agent ag1;
            for (int i = 0; i < agentList.size(); i++) {
```

```
        ag1 = (Agent) agentList.get(i);
        if (ag1.getActive() && (ag1.getCodeColor()== 'r'))
          count++;
      }
      return count;
  }
}

//number of yellow active agents
class CountNumActiveYellowAgents implements Sequence {
  public double getSValue() {
    double count = 0;
    Agent ag1;
    for (int i = 0; i < agentList.size(); i++) {
      ag1 = (Agent) agentList.get(i);
      if (ag1.getActive() && (ag1.getCodeColor() == 'y'))
        count++;
    }
    return count;
  }
}

//number of blue active agents
class CountNumActiveBlueAgents implements Sequence {
  public double getSValue() {
    double count = 0;
    Agent ag1;
    for (int i = 0; i < agentList.size(); i++) {
      ag1 = (Agent) agentList.get(i);
      if (ag1.getActive() && (ag1.getCodeColor() == 'b'))
        count++;
    }
    return count;
  }
}

//number of inactive agents
```

```
  class CountNumInactiveAgents implements Sequence {
    public double getSValue() {
      double count = 0;
      Agent ag1;
      for (int i = 0; i < agentList.size(); i++) {
        ag1 = (Agent) agentList.get(i);
        if (ag1.getActive() == false)
          count++;
      }
      return count;
    }
  }

//number of red inactive agents
  class CountNumInactiveRedAgents implements Sequence {
    public double getSValue() {
      double count = 0;
      Agent ag1;
      for (int i = 0; i < agentList.size(); i++) {
        ag1 = (Agent) agentList.get(i);
        if ( (ag1.getActive() == false) && (ag1.getCodeColor() == 'r'))
          count++;
      }
      return count;
    }
  }

//number of yellow inactive agents
  class CountNumInactiveYellowAgents implements Sequence {
    public double getSValue() {
      double count = 0;
      Agent ag1;
      for (int i = 0; i < agentList.size(); i++) {
        ag1 = (Agent) agentList.get(i);
        if ( (ag1.getActive() == false) && (ag1.getCodeColor() == 'y'))
          count++;
      }
```

```
        return count;
      }
    }

//number of blue inactive agents
    class CountNumInactiveBlueAgents implements Sequence {
      public double getSValue() {
        double count = 0;
        Agent ag1;
        for (int i = 0; i < agentList.size(); i++) {
          ag1 = (Agent) agentList.get(i);
          if ( (ag1.getActive() == false) && (ag1.getCodeColor() == 'b'))
            count++;
        }
        return count;
      }
    }

    class LogTotalActiveAgents implements Sequence {
      public double getSValue() {
        double logActiveAgents;
        logActiveAgents = Math.log(countNumActiveAgents());
         return logActiveAgents;
      }
    }

    class LogTime implements Sequence {
      private Model myModel;

      public double getSValue() {
        myModel = new Model();
        double logTime;
        double time;
        time = myModel.getTickCountDouble();
        logTime = Math.log(time);
        return logTime;
  }
```

```
}

  class LogTotalInactiveAgents implements Sequence {
    public double getSValue() {
      double logInactiveAgents;
      logInactiveAgents = Math.log(countNumInactiveAgents());
      return logInactiveAgents;
    }
  }

//number of active agents
  public int countNumActiveAgents() {
    int count = 0;
    Agent ag1;
    for (int i = 0; i < agentList.size(); i++) {
      ag1 = (Agent) agentList.get(i);
      if (ag1.getActive())
        count++;
    }
    return count;
  }

  //number of inactive agents
    public int countNumInactiveAgents() {
      int count = 0;
      Agent ag1;
      for (int i = 0; i < agentList.size(); i++) {
        ag1 = (Agent) agentList.get(i);
        if (ag1.getActive()== false)
        count++;
     }
      return count;
    }

   //number of inactive red agents
     public int countNumInactiveRedAgents() {
       int count = 0;
```

```
    Agent ag1;
    for (int i = 0; i < agentList.size(); i++) {
      ag1 = (Agent) agentList.get(i);
      if ((ag1.getActive()== false) && (ag1.getCodeColor()== 'r'))
      count++;
    }
    return count;
  }
```

//method to update log-log graph

```
  private void plotLogLog (){
    double logActiveAgents = Math.log(countNumActiveAgents());
    double time = this.getTickCountDouble();
    double alogTime = Math.log(time);

    logTimeVsLogInacAg.setConnected(true);
    logTimeVsLogInacAg.setXRange(0, 100);
    logTimeVsLogInacAg.setYRange(0, 100);
    logTimeVsLogInacAg.setAxisTitles("log time","log inactive agents");

    logTimeVsLogInacAg.plotPoint(alogTime, logActiveAgents, 0);

    logTimeVsLogInacAg.updateGraph();
    logTimeVsLogInacAg.fillPlot();

  }
```

//method to calculate density from the centre

```
  public void calculateDensity() {
    int a, b, i;
    double R, dx, dy;
    int[] massArray;
    massArray = new int[50];

    densityPlot.setConnected(true);
    densityPlot.setXRange(0, 100);
```

```
densityPlot.setYRange(0, 100);



for (i = 0; i < 50; i++){
  massArray [i] = 0;
}

//density of total inactive agents (patches)- dataset 0
 for (i = 0; i < 50; i++){
    for (a = 0; a <  spaceSize; a++){
      for (b = 0; b < spaceSize; b++){
        if (aCondition.isThereAnInactiveAgentAtList(space, a, b)){
          dx = Math.pow((a - 50), 2);
          dy = Math.pow((b - 50), 2);
          R = Math.sqrt(dx + dy);
            if (R <= i){
              massArray[i]++;
            }

        }//if aCondition
      }//for y
    }//for x
    //plot points densityTotal
    densityPlot.plotPoint(i, massArray[i], 0);
  }//for i

//density of red inactive agents (patches) - dataset 1
  for (i = 0; i < 50; i++){
     for (a = 0; a <  101; a++){
       for (b = 0; b < 101; b++){
         if (aCondition.isThereRedInactive(space, a, b)){
           dx = Math.pow((a - 50), 2);
           dy = Math.pow((b - 50), 2);
           R = Math.sqrt(dx + dy);
             if (R <= i){
               massArray[i]++;
             }
```

```
          }//if aCondition
        }//for y
      }//for x

      //plot points densityTotal
      densityPlot.plotPoint(i, massArray[i], 1);
    }//for i



//density of yellow inactive agents (patches) - dataset 2
for (i = 0; i < 50; i++){
  for (a = 0; a <  101; a++){
    for (b = 0; b < 101; b++){
        if (aCondition.isThereYellowInactive(space, a, b)){
        dx = Math.pow((a - 50), 2);
        dy = Math.pow((b - 50), 2);
        R = Math.sqrt(dx + dy);
        if (R <= i){
          massArray[i]++;
        }

      }//if aCondition
    }//for y
  }//for x

  //plot points densityYellow
  densityPlot.plotPoint(i, massArray[i], 2);
}//for i

//density of blue inactive agents (patches) - dataset 3
for (i = 0; i < 50; i++){
  for (a = 0; a <  101; a++){
    for (b = 0; b < 101; b++){
        if (aCondition.isThereBlueInactive(space, a, b)){
        dx = Math.pow((a - 50), 2);
        dy = Math.pow((b - 50), 2);
```

```
          R = Math.sqrt(dx + dy);
          if (R <= i){
            massArray[i]++;

          }
        }//if aCondition
      }//for y
    }//for x

    //plot points densityYellow
    densityPlot.plotPoint(i, massArray[i], 3);

  }//for i

     // plot display:
     densityPlot.setAxisTitles("radius", "mass inactive agents");
     densityPlot.addLegend(0, "Total inactive", Color.black, Plot.FILLED_CIRCLE);
     densityPlot.addLegend(1, "Red inactive", Color.red, Plot.FILLED_CIRCLE);
     densityPlot.addLegend(2, "Yellow inactive", Color.yellow, Plot.FILLED_CIRCLE);
     densityPlot.addLegend(3, "Blue inactive", Color.blue, Plot.FILLED_CIRCLE);
     densityPlot.display();
     densityPlot.updateGraph();
     densityPlot.fillPlot();

} //method

    public static void openFile(){
      String FileName="./density.txt";
     try {
         outputDensity = new DataOutputStream(
            new FileOutputStream(FileName));
      }
       catch ( IOException e) {
          System.err.println("Error during output\n");
          System.exit(1);
       }
    }
```

```
//method to calculate density from the centre
  public void computeDensityTotal() {
    int a, b, i;
    double R, dx, dy;
    int halfSpaceSize = spaceSize/2;
    int radiusSize = (spaceSize/2) + 1;
    int[] massArrayTotal;
    massArrayTotal = new int[radiusSize];
    int[] massArrayRed;
    massArrayRed = new int[radiusSize];
    int[] massArrayYellow;
    massArrayYellow = new int[radiusSize];
    int[] massArrayBlue;
    massArrayBlue = new int[radiusSize];
    int[] massArrayCyan;
    massArrayCyan = new int[radiusSize];
    int[] massArrayGreen;
    massArrayGreen = new int[radiusSize];

    openFile();

    for (i = 0; i < radiusSize; i++) {
      massArrayTotal[i] = 0;
      massArrayRed[i] = 0;
      massArrayYellow[i] = 0;
      massArrayBlue[i] = 0;
    }

    //density of total inactive agents (patches)
    for (i = 0; i < radiusSize; i++) {
      for (a = 0; a < spaceSize; a++) {
        for (b = 0; b < spaceSize; b++) {
          if (aCondition.isThereAnInactiveAgentAtList(space, a, b)) {
            dx = Math.pow( (a - (halfSpaceSize)), 2);
            dy = Math.pow( (b - (halfSpaceSize)), 2);
            R = Math.sqrt(dx + dy);
```

```
            if (R <= i) massArrayTotal[i]++;
           } //if aCondition
          } //for y
         } //for x
        }// end for 50 (i)

//density of red inactive agents (patches)
   for (i = 0; i < radiusSize; i++){
       for (a = 0; a <  spaceSize; a++){
        for (b = 0; b < spaceSize; b++){
          if (aCondition.isThereRedInactive(space, a, b)){
           dx = Math.pow((a - halfSpaceSize), 2);
           dy = Math.pow((b - halfSpaceSize), 2);
           R = Math.sqrt(dx + dy);
            if (R <= i){
             massArrayRed[i]++;
            }
          }//if aCondition
        }//for y
       }//for x
     }//for i


  //density of yellow inactive agents (patches)
  for (i = 0; i < radiusSize; i++){
   for (a = 0; a <  spaceSize; a++){
    for (b = 0; b < spaceSize; b++){
        if (aCondition.isThereYellowInactive(space, a, b)){
        dx = Math.pow((a - halfSpaceSize), 2);
        dy = Math.pow((b - halfSpaceSize), 2);
        R = Math.sqrt(dx + dy);
        if (R <= i){
          massArrayYellow[i]++;
        }
       }//if aCondition
     }//for y
   }//for x
```

```
  }//for i

  //density of blue inactive agents (patches)
  for (i = 0; i < radiusSize; i++){
    for (a = 0; a <  spaceSize; a++){
      for (b = 0; b < spaceSize; b++){
          if (aCondition.isThereBlueInactive(space, a, b)){
          dx = Math.pow((a - halfSpaceSize), 2);
          dy = Math.pow((b - halfSpaceSize), 2);
          R = Math.sqrt(dx + dy);
          if (R <= i){
            massArrayBlue[i]++;
          }
        }//if aCondition
      }//for y
    }//for x
  }//for i

//density of cyan inactive agents (patches)
for (i = 0; i < radiusSize; i++){
  for (a = 0; a <  spaceSize; a++){
    for (b = 0; b < spaceSize; b++){
        if (aCondition.isThereCyanInactive(space, a, b)){
        dx = Math.pow((a - halfSpaceSize), 2);
        dy = Math.pow((b - halfSpaceSize), 2);
        R = Math.sqrt(dx + dy);
        if (R <= i){
          massArrayCyan[i]++;
        }
      }//if aCondition
    }//for y
  }//for x
}//for i

//density of cyan inactive agents (patches)
for (i = 0; i < radiusSize; i++){
  for (a = 0; a <  spaceSize; a++){
```

```
    for (b = 0; b < spaceSize; b++){
        if (aCondition.isThereGreenInactive(space, a, b)){
        dx = Math.pow((a - halfSpaceSize), 2);
        dy = Math.pow((b - halfSpaceSize), 2);
        R = Math.sqrt(dx + dy);
        if (R <= i){
          massArrayGreen[i]++;
        }
      }//if aCondition
    }//for y
  }//for x
}//for i

 //print results in a file
   for (i = 0; i < radiusSize; i++) {
     try{
       outputDensity.writeBytes(i + ","+ massArrayTotal[i] + "," + massArrayRed[i] + "," +
massArrayYellow[i] + "," + massArrayBlue[i] + "," + massArrayCyan[i] + "," +
massArrayGreen[i] + " \n"); //aqui ele ta gravando no arquivo
       }
     catch(IOException ioException){
       System.out.println("Cannot save result");
     }
  }
          try{
            outputDensity.close();  //este comando fecha o arquivo
          }
          catch(IOException ioException){
            System.out.println("Cannot close file");
          }
       }// end method

   public double calculateMassPerTime(){
     int i;
     double derivative;
     int[] massPerTime = new int[200000];
     i = (int) this.getTickCountDouble();
```

```
    massPerTime[i] = this.countNumInactiveAgents();
    derivative = massPerTime[i] - massPerTime[i - 1];
    return derivative;
  }

//number of red active agents
public double countNumActiveRedAgents () {
        double count = 0;
        Agent ag1;
        for (int i = 0; i < agentList.size(); i++) {
          ag1 = (Agent) agentList.get(i);
          if (ag1.getActive() && (ag1.getCodeColor()== 'r'))
            count++;
        }
        return count;
}

//number of yellow active agents
public double countNumActiveYellowAgents () {
    double count = 0;
    Agent ag1;
    for (int i = 0; i < agentList.size(); i++) {
      ag1 = (Agent) agentList.get(i);
      if (ag1.getActive() && (ag1.getCodeColor() == 'y'))
        count++;
    }
    return count;
}

//number of blue active agents
public double countNumActiveBlueAgents (){
    double count = 0;
    Agent ag1;
    for (int i = 0; i < agentList.size(); i++) {
      ag1 = (Agent) agentList.get(i);
      if (ag1.getActive() && (ag1.getCodeColor() == 'b'))
        count++;
```

```
    }
    return count;
}
  ///////////////SPATIAL CONSTRAINTS

  public void readMatrix(String filename) {
    int matrixSize = spaceSize;
    DataInputStream input;
    String inputLine, element;
    StringTokenizer tokenizedLine;
    int line = 0;
    int column = 0;
    int[][] matrix = new int[matrixSize][matrixSize];

    try {
      input = new DataInputStream(new FileInputStream(filename));

      while ( (inputLine = input.readLine()) != null) {

        tokenizedLine = new StringTokenizer(inputLine);
        while (column < matrixSize){
        element = tokenizedLine.nextToken();
        matrix [column][line] = Integer.parseInt(element);
        column++;
        }
          line++;
          column = 0;
        }
        //create grey inactive agents at each point of matrix = 1
        for (int i = 0; i < matrixSize ; i++){
          for (int j = 0; j< matrixSize; j++){
            //System.out.print(matrix[i][j]+ " ");
            if (matrix[i][j] == 1){
              createGreyAgentAt(i, j);
            }
            if (matrix[i][j] == 2){
             createRedAgentAt(i, j);
```

```
            }
          }
        //System.out.println();
        }
      }

    catch (IOException e) {
       System.err.println("Cannot read matrix\n" + e.toString());
        }
      }

      public void createGreyAgentAt (int a, int b){

         int index = agentList.size()+ 1;
         Agent greyAgent = new Agent (space, index, this);
         //System.out.println("grey agent created at "+ a + "," + b );
         greyAgent.setXY(a, b);
         greyAgent.setActive(false);
         greyAgent.setColor(Color.darkGray);
         greyAgent.setCodeColor('d');
         agentList.add(greyAgent);
         //System.out.println(greyAgent.getWho() + " cinza inativo" );
      }

      public void createRedAgentAt (int a, int b){
        int index = agentList.size()+ 1;
        Agent redAgent = new Agent (space, index, this);
        //System.out.println("grey agent created at "+ a + "," + b );
        redAgent.setXY(a, b);
        redAgent.setActive(false);
        redAgent.setColor(Color.red);
        redAgent.setCodeColor('r');
        agentList.add(redAgent);
      //System.out.println(greyAgent.getWho() + " cinza inativo" );
    }

//////END SPATIAL CONSTRAINTS
```

```
////////////////////////////////////////////////////////////////////////////////
////EXPORT MATRIX ASCIIGRID
       public void exportASCIIfile(){
         int[][] fragstatsMatrix = new int[spaceSize][spaceSize];

         for (int x = 0; x < spaceSize; x++) {
           for (int y = 0; y < spaceSize; y++) {
             //create fragstatsMatrix:
             if (aCondition.isThereAnInactiveAgentAtList(space, x, y) == true){
               Agent inactiveHere = aCondition.getAnInactiveAgentAtList(space, x, y);
               char colorHere = inactiveHere.getCodeColor();
               if (colorHere == 'r') fragstatsMatrix[x][y] = 1;
               if (colorHere == 'y') fragstatsMatrix[x][y] = 2;
               if (colorHere == 'b') fragstatsMatrix[x][y] = 3;
               if (colorHere == 'c') fragstatsMatrix[x][y] = 4;
               if (colorHere == 'g') fragstatsMatrix[x][y] = 5;
               //if cell is grey (spatial constraint), treat it as background:
               if (colorHere == 'd') fragstatsMatrix[x][y] = 9;
             }
             else {
               fragstatsMatrix[x][y] = 9;
             }

           } //for x
         }//for y

         //open file
         try {
           float time = this.getTickCount();
           outputASCIIGRID = new DataOutputStream(new
FileOutputStream("./fragstats_"+time+".txt"));
           }
         catch ( IOException e) {
           System.err.println("Error during output\n");
           System.exit(1);
           }
```

```
        //print matrix to
        for (int i = 0; i < spaceSize; i++){
          for (int j = 0; j < spaceSize; j++) {
            try {
              outputASCIIGRID.writeBytes(fragstatsMatrix[i][j] + " ");
            }
            catch (IOException ioException) {
              System.out.println("Cannot save result");
            }
           } //for j
            //new line after each
            try {
              outputASCIIGRID.writeBytes("\n");
            }
            catch (IOException ioException) {
              System.out.println("Cannot save result");
            }
        }//for y

      }// export ASCIIfile method

//////////////////////////////////////////////
    // a required method
    public Schedule getSchedule() {
      return schedule;
    }

    // a required method - displayed on the Controller toolbar
    public String getName() {
      return "Model";
    }

    public static void main(String[] args) {
      SimInit init = new SimInit();
      Model model = new Model();
      init.loadModel(model, "", false);
    }
```

```
  }
```

Agent Class

```
package periproject;


import uchicago.src.sim.space.Multi2DTorus;
import uchicago.src.sim.gui.Drawable;
import uchicago.src.sim.gui.SimGraphics;
import uchicago.src.sim.gui.*;
import uchicago.src.sim.util.Random;

import java.awt.*;
import java.util.ArrayList;
import java.awt.Color.*;

public class Agent implements Drawable {

//agent parameter
  private int who;
  private Model model;
  private Color color;
  private float agentDirection;
  private boolean active;
  private char codeColor;
  private Conditions aCondition;
  private int x, y;
  private int age = 0;
  private boolean toOutskirts = false;
  private int cons;

  public int steps;


//space
  public Multi2DTorus space;


  public Agent(Multi2DTorus space, int who, Model model) {
    this.who = who;
    this.space = space;
    this.model = model;
    aCondition = new Conditions();
  }

//////////////   W A L K    ///////////////
public void walk() {

  int rd1, rd2, rd3;
  float walkDirection;
  //distance allows agent to walk in a different manner.
  int distance;
  distance = 1;
```

```
    //Change the direction of agent
    //option A: 20 degrees to the right, 20 degrees to the left
    rd1= Random.uniform.nextIntFromTo(0, 20);
    rd2 = -(Random.uniform.nextIntFromTo(0, 20));
    this.setAgentDirection(rd1 + rd2 + this.agentDirection);

    //option B: set direction random 360
    rd3 = Random.uniform.nextIntFromTo(0, 360);//test
    //this.setAgentDirection(rd3+ this.agentDirection);//teste 230 em vez de 20, 20
    //this.setAgentDirection(rd3);

    this.forward(distance);

    //option C: not setting agentDirection!
    //walkDirection = this.agentDirection + rd1 + rd2;
    //this.forward(distance, walkDirection);
  }

//////////////forward
  public void forward(float distance) {

    boolean freeOfContraints = true;
    int count = 0;
    //this method is a loop to avoid spatial constraints
    while (freeOfContraints){
      count ++;
      int a, b;
      int newXtoInt;
      int newYtoInt;
      a = this.getX();
      b = this.getY();

      float agentDirectionInRadians = toRadians(this.agentDirection);
      float newX = (float) (a + distance * Math.cos(agentDirectionInRadians));
      float newY = (float) (b + distance * Math.sin(agentDirectionInRadians));

      newXtoInt = Math.round(newX);
      newYtoInt = Math.round(newY);

      int normalizedX = space.xnorm(newXtoInt);
      int normalizedY = space.xnorm(newYtoInt);

      //if there is a spatial contraint at position change direction
      if (aCondition.isThereGrey(space, normalizedX, normalizedY)== true){
        //change direction to random ? ----check what is best here!
        float oldDirection = this.getAgentDirection();
        //float rDirRight = Random.uniform.nextIntFromTo(0, 25);
        //float rDirLeft = Random.uniform.nextIntFromTo(0, 25);
        //this.setAgentDirection(oldDirection + 180 +  rDirRight - rDirLeft);
        float rDir01 = Random.uniform.nextIntFromTo(0, 90);
        float rDir02 = Random.uniform.nextIntFromTo(0, 90);
        this.setAgentDirection(oldDirection + 180);
        this.setAgentDirection((this.getAgentDirection() - rDir01 + rDir02)% 360);
```

```
    }
    else {
      //set XY at position
      this.setXY(normalizedX, normalizedY);

      //set boolean to false and leave the loop!
      freeOfContraints = false;
    }

  }//while loop

    //to check if wrapping feature of torus is working:
    if (this.x > this.model.getSpaceSize()) System.out.println(
        "x maior que espaco");
    if (this.y > this.model.getSpaceSize()) System.out.println(
        "y maior que espaco");

  }

///Conversion from degree to radians which is required by the Math.cos methods
///The Math.toRadians method requires a double, that's why it is not been used here
  float toRadians (float agentDirection) {
    return (float) (agentDirection * Math.PI / 180.0);
  }

   ////////////FINDSPACEWALK
     public void findSpaceWalk(int numSteps) {
       int rd1, rd2, rd4;
       //Random.createUniform();
       rd1 = Random.uniform.nextIntFromTo(0, 25);//40
       rd2 = Random.uniform.nextIntFromTo(0, 25);
       rd4 = Random.uniform.nextIntFromTo(0, 50);
       //this.setAgentDirection(this.agentDirection + rd1 - rd2);
       this.setAgentDirection(this.agentDirection + rd4);
       this.forward(numSteps);
     }


   ////////////////////////////////////////////////////////////////////////////////////////

     //assenta o agente na posicao x,y
     public void setXY(int a, int b) {
       if (space.getObjectsAt(this.x, this.y).contains(this))
        space.removeObjectAt(this.x, this.y, this);
       this.setX(a);
       this.setY(b);
       space.putObjectAt(this.x, this.y, this);
     }


     //assenta o agente na posicao x,y
     public void moveToXY (int a, int b) {
       if (space.getObjectsAt(this.x, this.y).contains(this))
```

```
    space.removeObjectAt(this.x, this.y, this);
   this.setX(a);
   this.setY(b);
   space.putObjectAt(this.x, this.y, this);
 }

//////////findspace////////////////////////////////////////////////////////////
 public void findSpace(int findspaceSteps) {

   char activeAgColor;
   char inactiveAgColor;
   Agent inactiveAg;
   //int steps = 0;
   int steps = findspaceSteps;


   while (this.active == true)  {

     //defines numSteps either as equal to the parameter findspaceSteps
     // or as a random value between 1 and the parameter value (to avoid regularities)
     //Random Option: //numSteps = Random.uniform.nextIntFromTo(1, findspaceSteps);

     if (this.model.activateOutskirtsRule) {
       //if agent has abandoned a cell before, then it will have findSpaceSteps changed
       if (this.toOutskirts == true) {
         if (this.codeColor == 'r') {
           steps = this.model.findspaceSteps2;
         }
         if (this.codeColor == 'y') {
           steps = this.model.findspaceSteps3;
         }
       } // end if outskirts
     }


       //call findspaceWalk
       this.findSpaceWalk(steps);

       //module 2 -- choose size of steps according to density values
       //if density is bigger, then agent will do findspace again

       if (this.model.activateOutskirtsRule) {
         if (this.codeColor != 'b') {
           boolean n = true;
           while (n == true) {
             int densityHere = this.checkDensity();
             if (densityHere >= this.model.d) {
               if (this.codeColor == 'r') {
                 steps = this.model.findspaceSteps2;
               }
               if (this.codeColor == 'y') {
                 steps = this.model.findspaceSteps3;
               }
               this.findSpaceWalk(steps);
             } //if density
```

```
        else {
          n = false;
        } // else density
      } //if while

    } //if not blue
  }


  activeAgColor = this.getCodeColor();

  int inactivesHere = aCondition.getNumberInactiveAgentAtList(space, this.x, this.y);
  if (inactivesHere == 0) {
    this.settleSimple();
  }
  else{
  inactiveAg = aCondition.getAnInactiveAgentAtList(space, this.getX(), this.getY());
  inactiveAgColor = inactiveAg.getCodeColor();
  if ((activeAgColor == 'r') && (inactiveAgColor != 'r')) this.build(activeAgColor,
inactiveAgColor, inactiveAg);
  if ((activeAgColor == 'y') && (inactiveAgColor == 'b')) this.build(activeAgColor,
inactiveAgColor, inactiveAg);
  if ((activeAgColor == 'b') && (inactiveAgColor == 'g')) this.build(activeAgColor,
inactiveAgColor, inactiveAg);
  }//else
 }//while
}//findspace


public void build (char activeAgColor, char inactiveAgColor, Agent inactiveAg) {
  if ((activeAgColor == 'r') && (inactiveAgColor == 'b')) {
    this.expellInactiveAgent(inactiveAg);
    this.settleSimple();
  }
  if ((activeAgColor == 'r') && (inactiveAgColor == 'y')) {
    this.expellInactiveAgent(inactiveAg);
    this.settleSimple();
  }
  if ((activeAgColor == 'y') && (inactiveAgColor == 'b')) {
    this.expellInactiveAgent(inactiveAg);
    this.settleSimple();
  }
  if ((activeAgColor == 'b') && (inactiveAgColor == 'g')) {
    //System.out.println("blue on green");
    this.destroyGreenAgent(inactiveAg);
    this.settleSimple();
  }
}

//////////////////////////////////////////////////////////////////////////////

///////////settle simple
public void settleSimple() {
  if (this.codeColor == 'b'){
    if (this.model.activateConsolidationOld) {
```

```
          this.consolidate();
        }
      }
      int inactivesHere;
      inactivesHere = aCondition.getNumberInactiveAgentAtList(space, this.x, this.y);
      if (inactivesHere == 0) {
      this.setActive(false);
      this.setAge(0);
      }
    }

//////////expellInactiveAgent
    public void expellInactiveAgent(Agent inactiveAg) {
        inactiveAg.setActive(true);
        inactiveAg.setAge(0); //added on 10 august
        space.removeObjectAt(inactiveAg.x, inactiveAg.y, inactiveAg);
inactiveAg.findSpace(model.findspaceSteps);
      }

      public void destroyGreenAgent(Agent greenAg){
        space.removeObjectAt(greenAg.x, greenAg.y, greenAg);
        this.model.getAgentList().remove(greenAg);
      }

//////////////////////////////////////////////////////////////////////////////
//methods module 2
/////////////////////////////////////////////////////////////////////////////

 public int checkDensity(){
   int densi = 0;
   int i;
   Agent ngbAg;
   ArrayList neighbours = space.getMooreNeighbors(this.x, this.y, false);
   for (i = 0; i < neighbours.size(); i++){
     ngbAg = (Agent) neighbours.get(i);
     if (ngbAg.active == false){ // only inactive neighbours!
       char ngbAgColour = ngbAg.getCodeColor();
       if (ngbAgColour == 'r'){
         densi++;
       }
     }
   }
   return densi;
 }

public void selectSteps(){
  int densityHere = this.checkDensity();
  if (densityHere >= this.model.d) {
    if (this.codeColor == 'r') steps = this.model.findspaceSteps2;
    if (this.codeColor == 'y') steps = this.model.findspaceSteps3;
  }
  else {
    steps = this.model.findspaceSteps;
  }
}
```

```
// consolidation rule
public void consolidate(){
  this.setCons(cons + 1);
  if (cons >= this.model.consLimit) {
    this.setCodeColor('c');
    this.setColor(Color.cyan);
    this.setActive(false);
  }
}
///////////////////////////////////////////////////////////////////////////////
//methods required by drawable interface
///////////////////////////////////////////////////////////////////////////////

     public int getX() {
       return this.x;
     }

     public int getY() {
       return this.y;
     }

     public void draw(SimGraphics g) {
         if (this.active == false) // only draw the inactive
             g.drawFastRoundRect(color);
     }
///////////////////////////////////////////////////////////////////////////////
///////other required methods

     public void setX(int newx) {
       this.x = newx;
     }

     public void setY(int newy) {
       this.y = newy;
     }

     public boolean getActive() {
       return this.active;
     }

     public void setActive(boolean bool) {
       this.active = bool;
     }

     public float getAgentDirection() {
       return this.agentDirection;
     }

     public void setAgentDirection (float value) {
       this.agentDirection = value;
     }

     public int getWho() {
       return this.who;
```

```
    }

    public Model getModel() {
      return model;
    }

    public void setCodeColor(char c) {
      this.codeColor = c;
    }

    public char getCodeColor() {
      return this.codeColor;
    }

    public void setColor(Color color) {
      this.color = color;
    }

    public Color getColor() {
      return color;
    }

    public void setAge (int a) {
      this.age = a;
    }

    public int getAge() {
      return this.age;
    }

    public void setToOutskirts (boolean b) {
      this.toOutskirts = b;
    }

    public boolean getToOutskirts() {
      return this.toOutskirts;
    }
    //consolidation Old version variable
    public void setCons (int a) {
      this.cons = a;
    }

    public int getCons() {
      return this.cons;
    }

  }
////////////////////////////////////////////////////////////////////////////////
/*   //////generates a similar agent and includes it on AgentList
       public Agent generateASimilarAgent(int index) {

         //create new agent and set variables equal to 'this'.
         Agent newAgent = new Agent(space, index, model);
         newAgent.setActive(true);
         newAgent.setColor(this.getColor());
```

```
          newAgent.setCodeColor(this.getCodeColor());

          //set agentDirection as random
          int r = Random.uniform.nextIntFromTo(0,360);
          newAgent.setAgentDirection(r);

          newAgent.setXY( (space.getSizeX() / 2), space.getSizeY() / 2); (0,0)
          this.model.addNumActiveAgents(newAgent);

          return newAgent;
        }
  */
```

Conditions Class

```
package periproject;


import uchicago.src.sim.space.*;

import java.util.List;
import java.awt.*;

public class Conditions {
  public Conditions() {
  }

   //public Multi2DTorus space;

   ////checks if the list of objects at a cell contains a GREY agent
  public boolean isThereGrey(Multi2DTorus space, int a, int b) {
    Agent ag;
    List listObjectAtPosition = space.getObjectsAt(a, b);
    int size = listObjectAtPosition.size();
    for (int i = 0; i < size; i++) {
      ag = (Agent) listObjectAtPosition.get(i);
      if ( (ag.getColor() == Color.darkGray) ){
        return true;
      }//if
    }//for
    return false;
  }


////checks if the list of objects at a cell contains a red inactive agent
  public boolean isThereRedInactive(Multi2DTorus space, int a, int b) {
    Agent ag;
    List listObjectAtPosition = space.getObjectsAt(a, b);
    int size = listObjectAtPosition.size();
    for (int i = 0; i < size; i++) {
      ag = (Agent) listObjectAtPosition.get(i);
      if ( (ag.getColor() == Color.red) && !ag.getActive())
      return true;
    }
    return false;
```

```
  }

  ////checks if the list of objects at a cell contains a yellow inactive agent
  public boolean isThereYellowInactive(Multi2DTorus space, int a, int b) {
    Agent ag;
    List listObjectAtPosition = space.getObjectsAt(a, b);
    int size = listObjectAtPosition.size();
    for (int i = 0; i < size; i++) {
      ag = (Agent) listObjectAtPosition.get(i);
      if ( (ag.getColor() == Color.yellow) && !ag.getActive())
      return true;
    }
    return false;
  }

  ////checks if the list of objects at a cell contains a blue inactive agent
  public boolean isThereBlueInactive(Multi2DTorus space, int a, int b) {
    Agent ag;
    List listObjectAtPosition = space.getObjectsAt(a, b);
    int size = listObjectAtPosition.size();
    for (int i = 0; i < size; i++) {
      ag = (Agent) listObjectAtPosition.get(i);
      if ( (ag.getColor() == Color.blue) && !ag.getActive())
      return true;
    }
    return false;
  }

////checks if the list of objects at a cell contains a cyan inactive agent
  public boolean isThereCyanInactive(Multi2DTorus space, int a, int b) {
    Agent ag;
    List listObjectAtPosition = space.getObjectsAt(a, b);
    int size = listObjectAtPosition.size();
    for (int i = 0; i < size; i++) {
      ag = (Agent) listObjectAtPosition.get(i);
      if ( (ag.getColor() == Color.cyan) && !ag.getActive())
        return true;
    }
    return false;
  }

////checks if the list of objects at a cell contains a green inactive agent
  public boolean isThereGreenInactive(Multi2DTorus space, int a, int b) {
    Agent ag;
    List listObjectAtPosition = space.getObjectsAt(a, b);
    int size = listObjectAtPosition.size();
    for (int i = 0; i < size; i++) {
      ag = (Agent) listObjectAtPosition.get(i);
      if ( (ag.getColor() == Color.green) && !ag.getActive()){
        return true;
      }
    }
    return false;
  }
```

```
////checks if the list of objects at a cell contains an inactive agent
  public boolean isThereAnInactiveAgentAtList(Multi2DTorus space, int a, int b) {
    Agent ag;
    List listObjectAtPosition = space.getObjectsAt(a, b);
    for (int i = 0; i < listObjectAtPosition.size(); i++) {
      ag = (Agent) listObjectAtPosition.get(i);
      if (ag.getActive() == false) {
        return true;
      }
    }
    return false;
  }


////checks if the list of objects at a cell contains an inactive agent and returns this agents
  public Agent getAnInactiveAgentAtList(Multi2DTorus space, int a, int b) {
    Agent ag;
    List listObjectAtPosition = space.getObjectsAt(a, b);
    for (int i = 0; i < listObjectAtPosition.size(); i++) {
      ag = (Agent) listObjectAtPosition.get(i);
      if (ag.getActive() == false) {
        return ag;
      }
    }
    return null;
  }

  ////checks if the list of objects at a cell contains an inactive agent and returns this agents
  public int getNumberInactiveAgentAtList(Multi2DTorus space, int a, int b) {
    Agent ag;
    int count;
    count = 0;
    List listObjectAtPosition = space.getObjectsAt(a, b);
    for (int i = 0; i < listObjectAtPosition.size(); i++) {
      ag = (Agent) listObjectAtPosition.get(i);
      if (ag.getActive() == false) {
        count++;
      }
    }
    return count;
  }

}
```