Modeling migration dynamics within the agent-based Harris-Todaro framework

Auteur : Pasture, Martin
Promoteur(s) : Tharakan, Joseph
Faculté : HEC-Ecole de gestion de l'Université de Liège
Diplôme : Master en sciences économiques, orientation générale
Année académique : 2023-2024
URI/URL : http://hdl.handle.net/2268.2/20342

MODELING MIGRATION DYNAMICS WITHIN THE AGENT-BASED HARRIS-TODARO FRAMEWORK

Jury:
Supervisor:
Joseph THARAKAN
Readers:
Jérôme SCHOENMAECKERS
Kelly CIOPPA

Master thesis by
Martin PASTURE
For a Master in Economics – General Orientation
Academic year 2023/2024

Acknowledgements

I would like to express my heartfelt gratitude to HEC Liège, my professors, and especially Joseph Tharakan for his invaluable guidance and support throughout my master's journey.

I also would like to express special thanks to Kelly Cioppa and Jérôme Schoenmaeckers for dedicating their time and expertise to read my master thesis.

I am also grateful to Martial Toniotti for insightful discussions.

Your encouragement and expertise have been instrumental in shaping this thesis.

This master's thesis is the result of my year at HEC as an economics student. As a 3rd master thesis after my studies in electrical engineering and in political philosophy. I have tried to use my skills and experience to produce a relevant piece of work.

I dedicate this master thesis to the study rooms made available by the Occupation of the ULiège for Palestine.

> "*But we know too well that our freedom is incomplete without the freedom of the Palestinians*"
>
> Nelson Mandela

Abstract

This master thesis explores after some theoretical development inspired from different authors, the dynamics of migration within urbanization processes using the agent-based Harris-Todaro model. As the world urbanizes at a rapid pace, it is important to develop tools that allow us to understand the changing world around us. Urbanization is characterized by an increasing proportion of populations residing in urban areas, driven by productivity differences between the urban and rural sectors. This thesis investigates how migration dynamics can be modeled within the Harris-Todaro framework, situating the research in the context of current global urban transitions. It also revisits classic texts by Marx and Smith to highlight theoretical tensions between the two authors. A literature review covers existing migration models, multi-agent systems, and dynamics within migration models. The thesis details the implementation of the model and the results obtained. By simulating various migration parameters, this study demonstrates the model's flexibility and ability to capture complex migration dynamics suitable for diverse socio-economic contexts. The two major contributions of this thesis within the agent-based Harris-Todaro model are the addition of the subjective judgment of agents and the historicity of the unemployed person's journey. The significant contribution of this work is that it also presents in a very detailed form the influence of the parameters and initial conditions of the model on its own dynamics. The findings lay the groundwork for future research, advocating for the integration of empirical data to further refine the model and extend its usefulness in understanding and planning urban development.

Total word count : 18234

Contents

List of Figures

List of Tables

List of Abbreviations, Acronyms and Notations

Summary of Notations

M_{ij} = migration from region i to region j,
P_i, P_j = population in region i and j, respectively,
D_{ij} = distance between region i and region j.
V_0 = present discounted value of expected gain from migrating over the migrant's time horizon,
p_t = probability of being employed in the urban sector at time t,
y_u, y_r = fixed real urban and rural incomes, respectively,
β^t = discount factor for time period t,
C_0 = cost of moving incurred at time $t = 0$.
$w_m =$ wage in manufacture;
$w_a =$ wage in agriculture;
$\lambda =$ urban unemployment rate.
$Y_i =$ output, where $i = a, m$;
$L_i =$ labour input in production of Y_i, where $i = a, m$;
$\bar{N} =$ fixed amount of land;
$\bar{K}_i =$ fixed capital input, where $i = a, m$.
$p =$ terms of trade;
$\Leftrightarrow$ price of agricultural good express in manufactured goods;
$P_i =$ price of goods, where $i = a, m$.
$w_a =$ wage in agriculture.
$w_m =$ manufacturing wage;
$\bar{w}_m =$ the minimum wage in urban sector.
$L_u =$ is the total urban labor force
$w_u^e =$ expected urban wage;
$L_m =$ labor input in manufacturing jobs;
$L_u =$ labor input in urban sector.
$\bar{L} =$ fixed total labor supply;
$L_a =$ labor input in agriculture;
$\bar{L}_r, \bar{L}_u =$ initial endowments of labour in rural and urban sector, respectively.
$\lambda =$ urban unemployment rate.

Summary of Notations

$Y_a =$ production level of the agricultural good;
$L_a =$ amount of workers used in the agricultural production;
$A_a > 0 =$ parametric constant;
$0 < \phi < 1 =$ parametric constants.
$Y_m =$ production level of the manufactured good;
$L_m =$ amount of workers used in the manufacture production;
$A_m > 0 =$ parametric constant;
$0 < \alpha < 1 =$ parametric constants.
$w_a =$ wage in rural sector;
$p =$ price of the agricultural good;
both expressed in units of manufactured good.
$w_m =$ wage in urban sector;
$L_u =$ amount of workers in the urban sector.
GDP = Gross Domestic Product
BRICS = is an intergovernmental organization comprising Brazil, Russia, India, China, South Africa
LDC = Least Developed Countries

Chapter 1

Introduction

Urbanization, the process by which populations shift from rural to urban areas, has experienced unprecedented growth over the past two centuries. These urban environments, far removed from the natural resources essential for human material subsistence, have fostered the expansion of industry and services. In 2007, for the first time in history, half of humanity resided in urban areas, a trend that England had already experienced since 1851 [5]. Projections indicate that by 2050, the global urban population will double, with nearly seven out of ten people worldwide residing in urban settings.

Urbanization represents a historical movement that reshapes societal forms, characterized by the increase in the urban population relative to the total population. It is crucial to differentiate between urbanization and urban growth; while urbanization pertains to the proportion of the population residing in urban areas, urban growth refers to the absolute increase in urban population. This master thesis focuses on urbanization, emphasizing the rise in the percentage of the population living in urban environments.

The distinction between urban and rural areas is a complex subject, with definitions of cities varying across disciplines and perspectives. Economic criteria, such as labor productivity and job types, serve as the basis for defining cities within the context of this master thesis.

A common notion posits that industrial growth drives urbanization. However, this idea requires nuanced consideration, as urbanization also influences industrialization in a reciprocal manner. Metropolises generate positive externalities that, in turn, contribute to industrial growth across various dimensions.

This master's thesis examines theoretical views of early industrialisation and urbanisation by bringing Karl Marx and Adam Smith into dialogue, and examines the complex dynamics of rural-urban migration, exploring urbanisation within the framework of economic analysis and an agent-based model using the *Harris-Todaro model.* This research aims to provide tools for studying the dynamics of urbanization in agent-based models.

The research question is: **How can migration dynamics be modeled within the agent-based Harris-Todaro framework?**

1.1 Urbanization in history

Urbanization is a very recent process in human history. The genus *Homo* has inhabited the Earth for three million years. The oldest tools discovered to date, shaped pebbles, date back to between 2.44 and 1.92 million years ago [6]. These tools made of hard stone provide the first archaeological evidence of the cognitive and motor abilities necessary for their manufacture by hominids, over 3 million years ago.

The Israeli site of Gesher Benot Ya'aqov, dated to 800,000 years ago, seems to be the oldest evidence of fire domestication, but Homo species have been using fire since 1.5 million years ago. Evidence of fire domestication intensified between 400,000 and 200,000 years ago [7].

Homo sapiens have been living on this planet for nearly 300,000 years [8]. They were nomadic and practiced hunting and gathering. It is through labor that human beings produce their means of existence. The specificity of human beings lies in producing and reproducing their existence. Humans are not animals that find their means of existence directly in their environment, but through their work, they transform their environment to adapt to it and themselves.

10,000 years ago, during what is called the Neolithic Revolution, humans became sedentary. For the first time in their history, humans no longer needed to migrate to find the resources necessary for their subsistence. They were able to produce them by raising livestock, fishing and cultivating plants. This marked the beginning of the creation of property [9]. Society produced more than it needed and could begin to store in granaries or herds. It also marked the beginning of class society, where a particular group of society appropriates this surplus. It is this surplus that allows for the creation of lifestyles that are not directly focused on the production of physiological needs. However, at this time, humans mainly worked to produce their physiological needs, thus interacting with nature. The overwhelming majority of the population were peasants.

Urbanization has a rich history that dates back to ancient civilizations such as the Indus Valley, Mesopotamia, and Egypt, where small centers of population engaged in trade and manufacturing coexisted with rural areas focused on subsistence agriculture. Until the 18th century, there was a balance between rural and urban populations, with limited urban growth due to stagnant agriculture [10].

However, the onset of the British agricultural and industrial revolution in the late 18th century marked a significant shift. The rapid increase in agricultural productivity freed up laborers, leading to unprecedented urban population growth. Cities like Manchester and Birmingham experienced booming commerce and industry, attracting migrants from rural areas. This trend continued into the 19th century, with cities expanding spatially and industrially. Trade networks expanded globally, allowing for the importation of food and goods. By the end of the century, a large proportion of populations in Western countries lived in cities. Table 1.1 from Law's paper [5] shows us the fast urbanization of England and Wales in the second part of the 19th century. We observe that even with a population growth, the rural population is shrinking in absolute terms. Figure 1.1 represents the evolution of the urban population shares in different regions of the world. We can see that the United Kingdom was the first country to reach 50% of urban population in the middle of the 18th century. Urbanization rates have been on a steep upward trajectory across all regions, with less than 16% of the global population residing in urban areas in 1900.

Year	Total Population	Urban Population	% of Total	Rural Population	% of Total
1851	17,927,609	8,990,809	50.2	8,936,800	49.8
1861	20,066,224	10,960,998	54.6	9,105,226	45.4
1871	22,712,266	14,041,404	61.8	8,670,862	38.2
1881	25,794,439	17,636,646	67.9	8,337,793	32.1
1891	29,002,525	20,895,504	72.0	8,107,021	28.0
1901	32,527,843	25,058,355	77.0	7,469,488	23.0
1911	36,070,492	28,162,936	78.1	7,907,556	21.9

Table 1.1: Census Estimates of Urban and Rural Population in England and Wales, 1851-1911 [5].

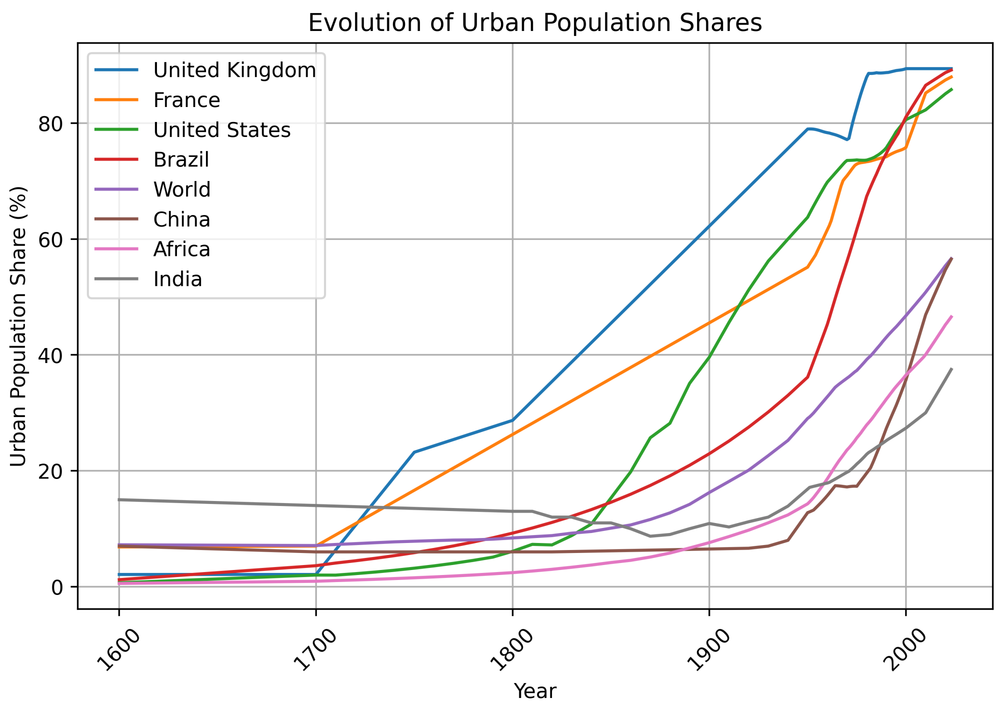


Figure 1.1: Evolution of urban population shares [1].

Since the 1950s, significant urbanization occurring in the developing world as well. At the turn of the 20th century, only 15% of the world population lived in cities [11].

In 2007, for the first time in history, more than half of the global population lived in cities, marking a pivotal moment in urbanization. This is represented in Figure 1.2.

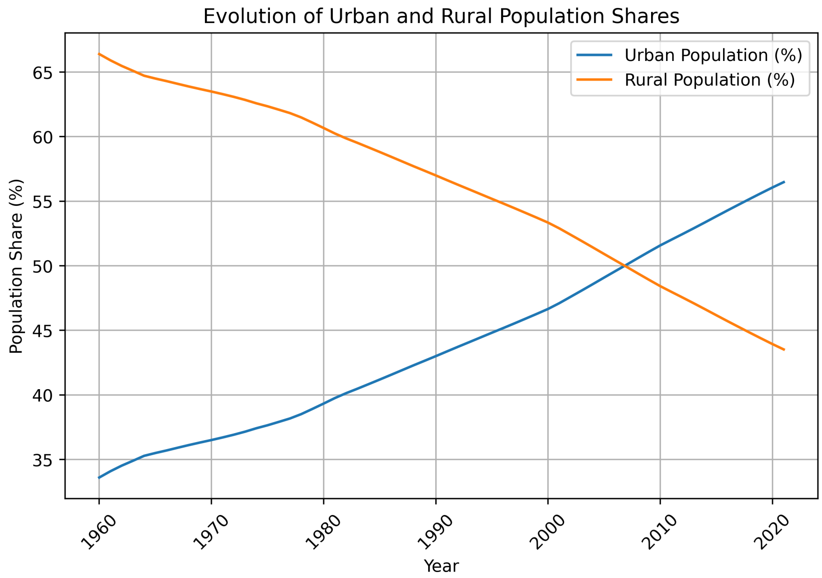


Figure 1.2: Evolution of urban an rural population shares [1].

Figure 1.3 shows the evolution of both urban and rural population. The world's population has risen from 3 billion in 1960 to 7.88 billion in 2021 (representing +163% of population growth), the urban population has risen from 1 billion to 4.45 billion (representing +345% of urban population growth) and the rural population has risen from 2 billion to 3.43 billion (representing +72% of rural population growth).

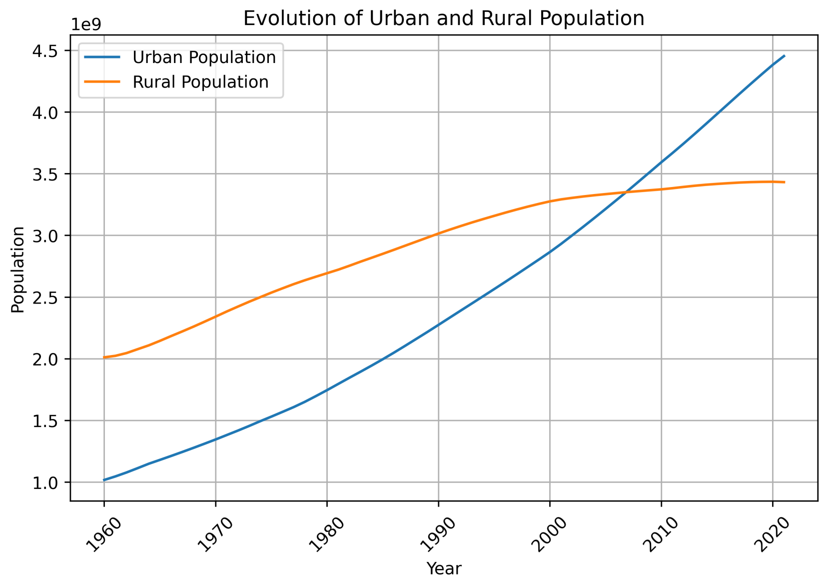


Figure 1.3: Evolution of ubran and rural population [1].

Urbanization is projected to continue its upward trend, driven by increasing incomes and shifts away from agriculture-based employment. Discrepancies in urban population figures arise primarily due to variations in the definition or delineation of what constitutes an 'urban' population.

1.2 Urbanization around the world

Urbanization around the world continues to accelerate, with an increasing proportion of the global population residing in urban areas. This growth was not uniform across all regions; some areas urbanized at a faster pace than others. As shown in Figure 1.1, the Western countries are significantly more urbanized than those in the Global South. As we have 55% of urbanization at world level, lots of countries have today more people living in urban areas than in rural areas. Figure 1.4 shows if countries have a majority of people in the cities or in the countryside. We see that Sub-Saharan Africa, Central Asia, South Asia and South-East Asia and a few other countries have not yet reached 50% urbanisation rate.

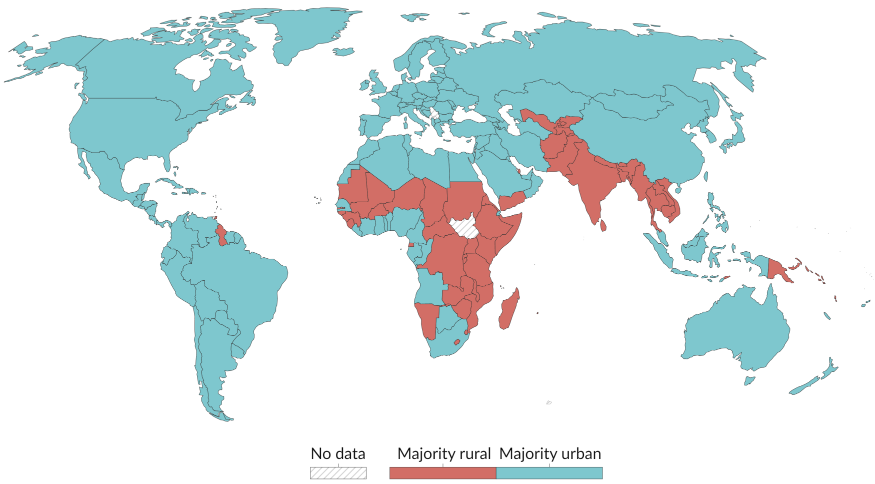


Figure 1.4: Do more people live in urban or rural areas ? 2023. Taken from[2].

Figure 1.5 illustrates the GDP per capita in the world. The poorest countries are the same as the least urbanized countries in Figure 1.4. This correlation will be developed in the next section.

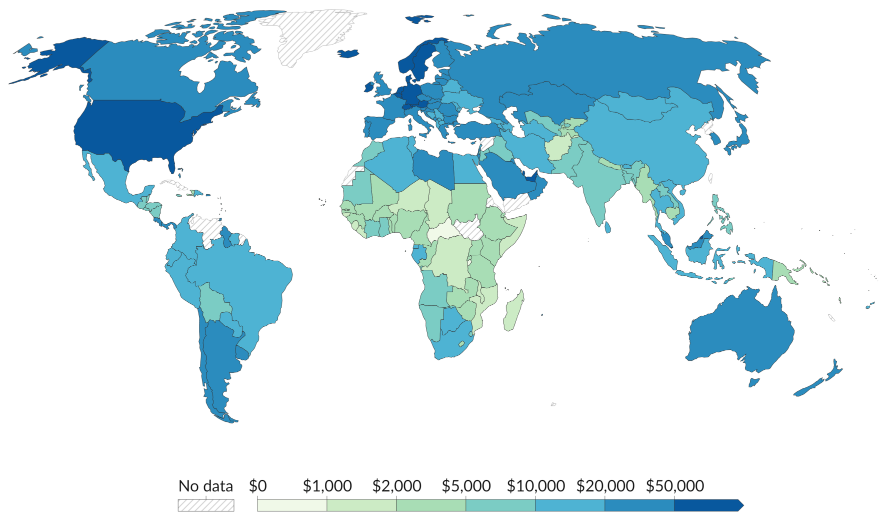
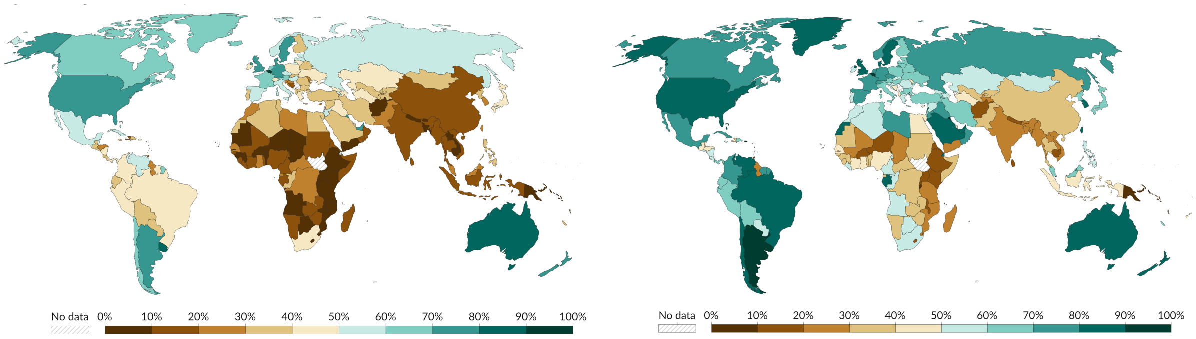


Figure 1.5: GDP per capita, 2021. Taken from[3].

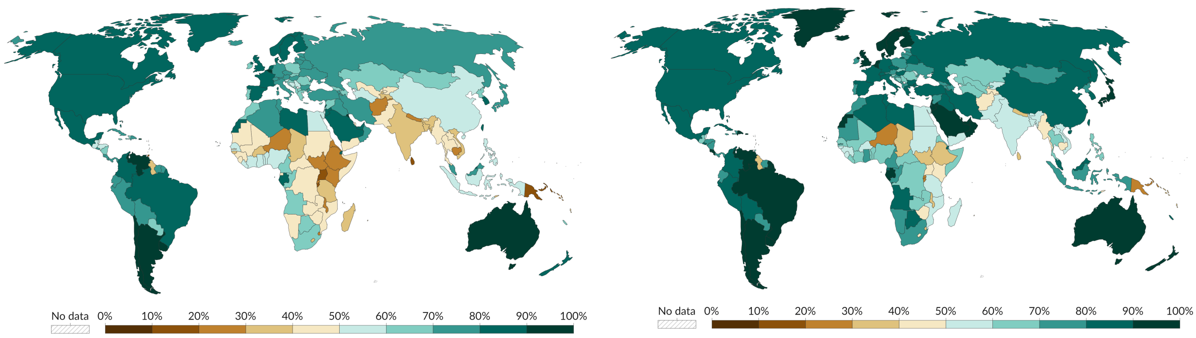
Figures 1.6 and 1.7 represent the urbanization rates in the world for the year 1961, 2000, 2023 and a prediction for 2050. The data presented reveals a changing global urban landscape. In 1961, urbanization was low at under 35%, showing unequal urbanization and development worldwide. By 2000, this inequality persisted, with North America and Europe highly urbanized while Africa and Asia remained behind. In 2023, urbanization became more even, but disparities remained. Projections for 2050 suggest a more balanced urbanization globally where the countries of the global south are catching up with the advanced countries in terms of urbanization.



(a) Share of population living in urban area, 1961. Taken from[2].

(b) Share of population living in urban area, 2000. Taken from[2].

Figure 1.6: Regression between Log GDP and Urbanization Rate (1960-2021).



(a) Share of population living in urban area, 2023. Taken from[2].

(b) Share of population living in urban area, 2050. Taken from[2].

Figure 1.7: Regression between Log GDP and Urbanization Rate (1960-2021).

Figure 1.8 shows the annual growth rate of urban agglomarations in the world. The poorest and less urbanized countries have the bigger urban agglomerations growth rate. Western European countries reached high levels of urbanization rate and do not know a annual growth rate of urban agglomerations while Eastern European countries know a decrease of annual growth rate of urban agglomerations with 300 inhabitants or more. Syria has been ravaged by war since 2011, causing massive destruction in urban areas and displacing millions of people. Many Syrians have fled to rural areas or taken refuge in neighboring countries, which could reduce the relative urban population.

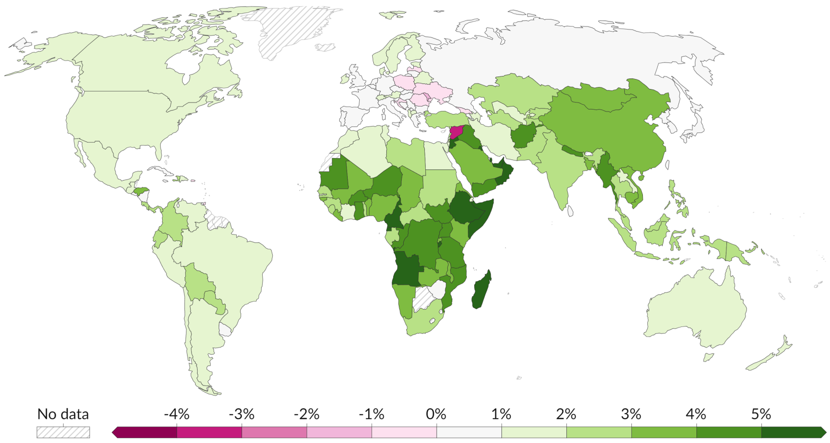


Figure 1.8: Annual growth rate of urban agglomerations with 300 inhabitants or more, 2015. Taken from [2].

1.3 Urbanization and economics

Economics significantly influences urbanization patterns, shaping the trajectory of urban growth and development. Figure 1.9 represents the population share in urban areas and the logarithm of GDP per capita. The analysis of the relationship between the logarithm of GDP and the urbanization rate revealed a correlation coefficient of 0.658 for the year 2021. This correlation suggests a moderate positive relationship between the two factors, implying that as urbanization increases, so does GDP per capita, and vice versa. Each time the wages are multiplied by 10, the country gains approximately 15% in urbanization rate, and vice versa. Figure 1.10 shows that the correlation and the coefficient decrease compared to the 1960s-1980s.

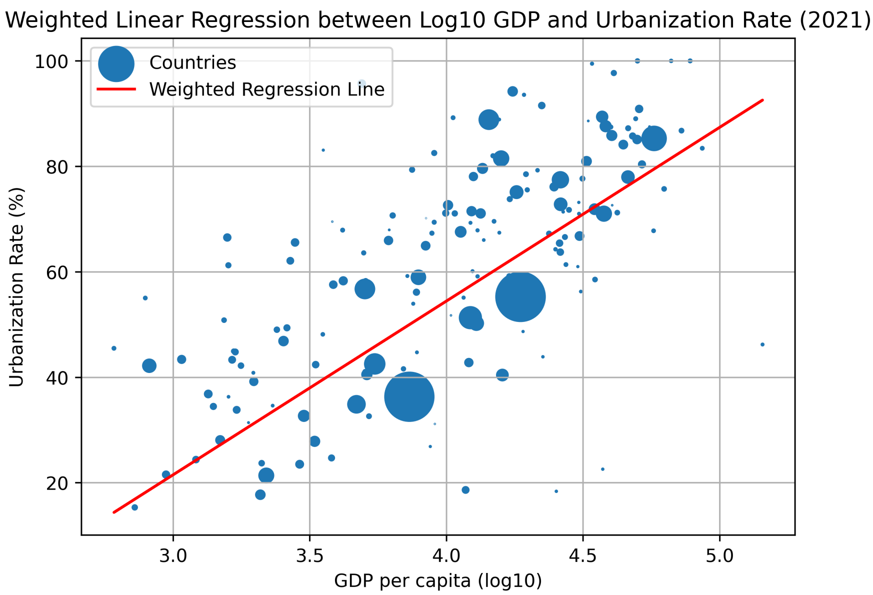
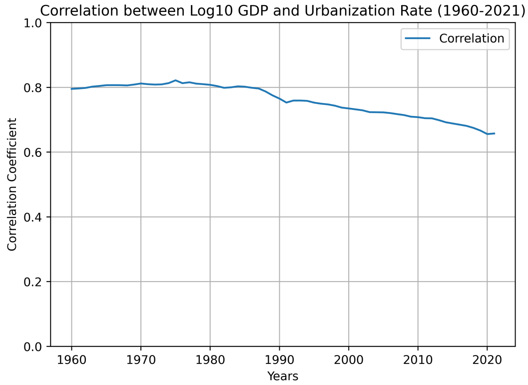
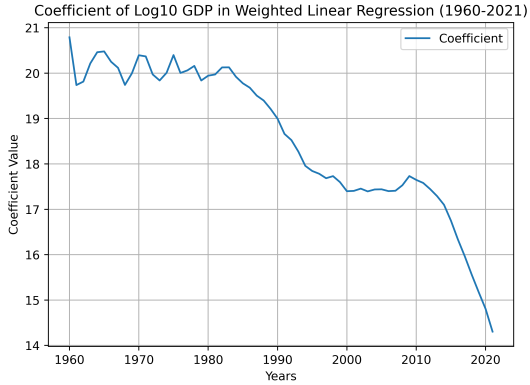


Figure 1.9: Share of the population living in urbanized areas vs. GDP per capita, 2021 [1].



(a) Correlation between Log GDP and Urbanization Rate (1960-2021) [1].



(b) Coefficient of Log GDP in Weighted Linear Regression (1960-2021) [1].

Figure 1.10: Regression between Log GDP and Urbanization Rate (1960-2021).

Figure 1.11 illustrates the relationship between employment in agriculture and urban population. In 2021, there was a significant negative correlation of -0.693 between employment in agriculture and urban population, indicating an inverse relationship between these two variables. For each percent increase in urban population, the country loses approximately 1% of the share of employment in agriculture. The findings of Chen et al.

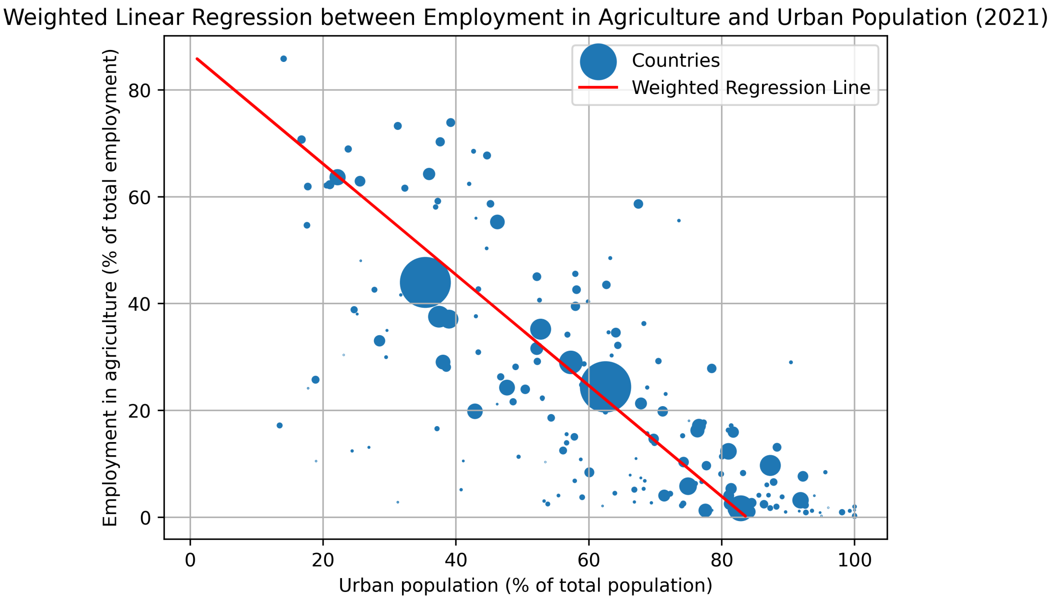
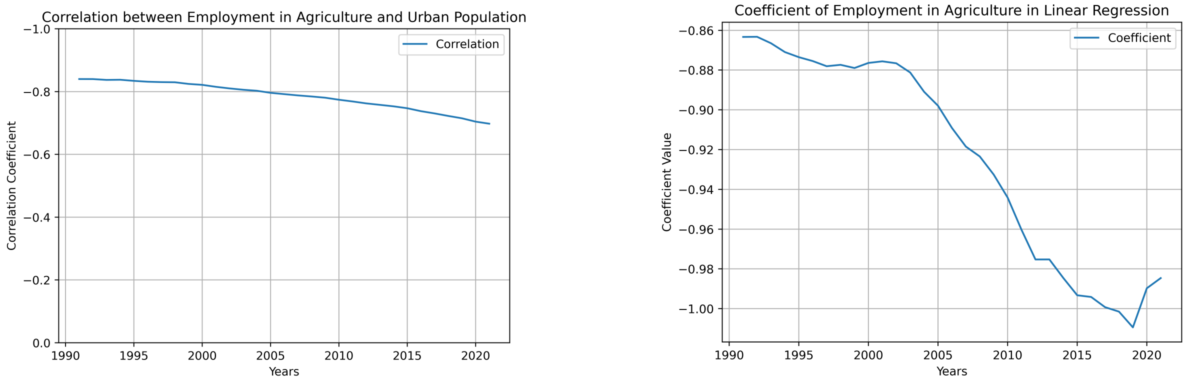


Figure 1.11: Share of the population employed in agriculture vs. urban population, 2021 [1].

in 2023 [12] indicate a strong correlation between the pace of urbanization and levels of vulnerable employment . The analysis reveals a statistically significant negative regression coefficient (-0.168) associated with urbanization, suggesting that urbanization exerts a negative impact on vulnerable employment. Specifically, for every 1% increase in the urbanization rate, there is a corresponding decrease of 0.168% in the rate of vulnerable employment.



(a) Correlation between Employment in Agriculture and Urban Population (1991-2021) [1].

(b) Coefficient of Employment in Agriculture in Weighted Linear Regression (1991-2021) [1].

Figure 1.12: Regression between Log GDP and Urbanization Rate (1960-2021).

The rise of the BRICS (Brazil, Russia, India, China and South Africa) beyond the traditional G7 economies (United States, Japan, Germany, United Kingdom, France, Italy and Canada) in terms of GDP per purchasing power parity marks a major turning point in the world economy. This transition represented in Figure 1.13 reflects a significant shift in the distribution of economic power and a redistribution of the cards on the world stage.

The BRICS have emerged as dynamic engines of economic growth, drawing on their vast populations, abundant natural resources, and efforts to modernize infrastructure and manage rapid urbanization. Their economic rise has been fueled by rapid urbanization, investment-friendly policies, structural reforms, increased integration into the global economy.

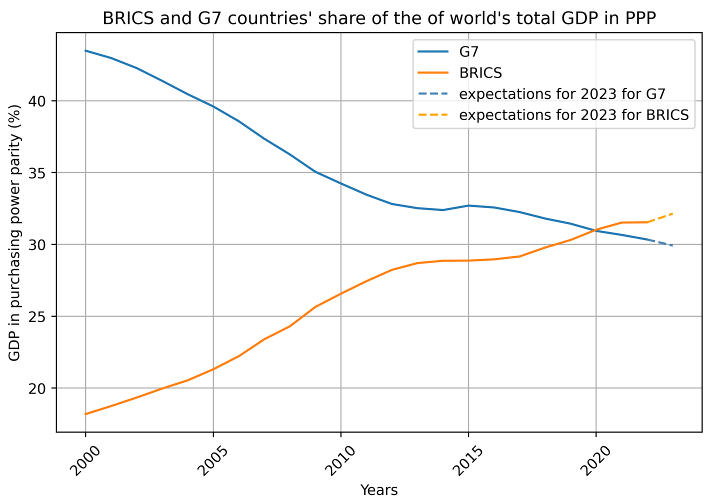


Figure 1.13: BRICS and G7 countries' share of the world's total gross domestic product (GDP) in purchasing power parity (PPP) from 2000 to 2023 [4].

Chapter 2

Theoretical development

2.1 The causes of urbanization

What are the foundations of urbanization? How did this historical movement begin? In the first part, we saw that urbanization occurs as capitalism develops in Western Europe, particularly in England, where an industrial revolution will see the establishment of cities and industry. We reviewed the global situation of urbanization and the links with the economy. In this part, we will tackle the hypotheses explaining urbanization. This section aims to be more theoretical and to explore the origins of modern cities.

Several hypotheses propose to explain why we observe a population movement from the countryside to the city. We will discuss the differences between the two environments, the specific social relations, and the differences in production. We will also take a look at economic theory, especially Adam Smith and Karl Marx. We will try to see how they provide explanations for this industrialization and how they explain the conditions of capitalism. As we will see, this condition revolves around initial accumulation.

2.2 Agriculture and industry

Several factors are behind this process of urbanization, which involves a rapid increase in population accompanied by greater productivity in agriculture, allowing fewer people to produce more food and ultimately industrialization.

Firstly, it can be observed that social relations, production relationships, are not the same and undergo profound transformations in the West before the industrial revolution. In a schematic way, to simplify, the economy is often reduced to two sectors. The first is the agricultural sector, the second is the industrial sector. There are many methods to define these two sectors; they can be seen simply as two different activity sectors, as two separate businesses with different productivities and production functions. They can also be considered as different sectors in terms of production relations; at that time, agriculture was in a regime of serfdom or sharecropping, and industry was in a wage labor regime.

2.3 Two sector economy

A two-sector economy is an important concept in economic science and particularly in development economics. It gives a simple framework to analyze economies. The two sector model divides the economy into two distinct sectors: the primary sector and the secondary sector. This is based on the initial paper of Sir Arthur Lewis from 1954 entitled "Economic Development with Unlimited Supplies of Labor" [13] that present a dual sector model that explains the growth of a developing economy in terms of a labour transition between the two sectors.

The primary sector, includes activities related to the agriculture and production of natural resources. It is the interface of society with nature. In agrarian economies, the primary sector dominates, with a high proportion of the population working in agricultural sector.

On the other hand, the secondary sector, transforms raw materials into finished goods. It is also known as manufacturing or industrial sector. It relies on labor and capital. Industrialization leads to the growth of the secondary sector as economies transition from rural to industrialized economy.

2.4 Adam Smith and Karl Marx

The aim of this theoretical section is to explore the thinking of two classic authors of political economy on the birth of capitalism and the industrial revolution. Their vision of the foundations of the current movement of society and history will be briefly outlined on the basis of quotes. Two fundamental concepts will be explored, namely initial accumulation and the opposition between town and country. We will also show how they relate to each other and to other fundamental concepts such as the division of labour. This idea comes to me from reading Chapter 24 of Book 1 of Karl Marx's Capital [14] and a lecture by Jean Batou at the Sorbonne on 31 March 2012 [15]. The quotes come from my personal reading. My reading of Karl Marx was done in French, so I allow myself to quote him directly in French.

2.4.1 Initial accumulation

Let's start with Adam Smith's thinking. In *An Inquiry into the Nature and Causes of the Wealth of Nations*, Adam Smith explains the origin of the division of labour through the initial accumulation of reserves, such as means of subsistence, means of production, money...

In the introduction to the second book of this work, Adam Smith begins with what might be called a *robinsonnade*. In other words, a founding myth that explains the prehistory of society as the life of the isolated individual. It is the story of Robinson Crusoe on his island. Adam Smith begins:

> "In that rude state of society, in which there is no division of labour, in which exchanges are seldom made, and in which every man provides everything for

himself, it is not necessary that any stock should be accumulated, or stored up before-hand, in order to carry on the business of the society. Every man endeavours to supply, by his own industry, his own occasional wants, as they occur. When he is hungry, he goes to the forest to hunt; when his coat is worn out, he clothes himself with the skin of the first large animal he kills: and when his hut begins to go to ruin, he repairs it, as well as he can, with the trees and the turf that are nearest it." (Adam Smith, An Inquiry into the Nature and Causes of the Wealth of Nations – Book 2 – Introduction [16]).

At this stage, each isolated individual has to provide entirely for his or her own needs. Obviously, this is just a myth and has no real historical basis, because humans are social animals who have lived in groups since the dawn of time. This link between the great fascination with *robinsonnades* from the eighteenth century onwards and this explanation of humanity's alleged past was criticised by Marx in his work :

> « Des individus produisant en société – donc une production socialement déterminée des individus, c'est là naturellement le point de départ. Le chasseur et le pêcheur singuliers et singularisés, avec lesquels débutent Smith et Ricardo, appartiennent aux fictions sans imagination de la robinsonnade du XVIIIe siècle [. . .]. C'est dans la société de la libre concurrence que l'individu singulier apparaît détaché des liens naturels, etc., qui font de lui, à des époques historiques antérieures, l'élément accessoire d'un conglomérat humain limité et déterminé. Dans la tête des prophètes du XVIIIe siècle, sur les épaules desquels se tiennent encore complètement Smith et Ricardo, cet individu du XVIIIe siècle [. . .] apparaît comme un idéal dont l'existence serait une existence passée. Non pas comme un individu issu de l'histoire mais comme un individu posé par la nature. » (Karl Marx, Contribution à la critique de l'économie politique, Éditions sociales, p31-32 [17])

What is being said here when it is mentioned that individuals are a socially determined production, is that the notion of the individual is not an immanent notion (valid at all times). That the individual is the result of the society of free competition, which allows human beings to appear detached from all natural ties, whereas objectively these individuals each need society to survive. So the idea of the individual has not always been self-evident for human beings, but is a historical product. The individual is therefore not posited by a hypothetical human nature, but is the result of social organisation. The above extract can be compared with this passage:

> « La bourgeoisie a joué dans l'histoire un rôle éminemment révolutionnaire. Partout où elle a conquis le pouvoir, elle a foulé aux pieds les relations féodales, patriarcales et idylliques. Tous les liens complexes et variés qui unissent l'homme féodal à ses "supérieurs naturels", elle les a brisés sans pitié pour ne laisser subsister d'autre lien, entre l'homme et l'homme, que le froid intérêt, les dures exigences du "paiement au comptant". » (Karl Marx et Friedrich Engels, Manifeste du parti communiste [18])

However, the reality is that the division of labour exploded with the industrial revolution. For example, at the hunter-gatherer stage described by Adam Smith, hunter-gatherer communities had to meet their needs collectively, with little exchange with other human groups. What's more, these hunter-gatherer societies had no surplus production. They took what nature had to offer and were nomadic. They take what the environment provides and migrate regularly once these resources are no longer abundant. It was from the Neolithic revolution onwards that societies were able to accumulate a surplus of production in relation to their consumption, at which point we see the birth of class society (in which parts of the population monopolise this surplus of production) and we also see the beginning of the division of labour. This point is developed in section 1.1. Let's now talk about the difference between the absence and presence of the division of labour in society. Adam Smith continues:

> "But when the division of labour has once been thoroughly introduced, the produce of a man's own labour can supply but a very small part of his occasional wants. The far greater part of them are supplied by the produce of other men's labour, which he purchases with the produce, or, what is the same thing, with the price of the produce, of his own. But this purchase cannot be made till such time as the produce of his own labour has not only been completed, but sold. A stock of goods of different kinds, therefore, must be stored up somewhere, sufficient to maintain him, and to supply him with the materials and tools of his work, till such time at least as both these events can be brought about. A weaver cannot apply himself entirely to his peculiar business, unless there is before-hand stored up somewhere, either in his own possession, or in that of some other person, a stock sufficient to maintain him, and to supply him with the materials and tools of his work, till he has not only completed, but sold his web. This accumulation must evidently be previous to his applying his industry for so long a time to such a peculiar business." (Adam Smith, An Inquiry into the Nature and Causes of the Wealth of Nations – Book 2 – Introduction [16])

In this paragraph, he gives us the conditions for the division of labour. In the next sentence, he writes

> "the accumulation of stock must, in the nature of things, be previous to the division of labour." (Adam Smith, An Inquiry into the Nature and Causes of the Wealth of Nations – Book 2 – Introduction [16])

This division of labour does not meet the needs of the individual worker. At best, his personal labour only meets a small fraction of his needs. To do this, the worker needs to be able to sell the product of his labour, a commodity, on a market in order to obtain in exchange the commodities produced by other workers and which he needs. But to complete this process, the worker must first be able to consume and satisfy his needs, he must have access to his means of existence, and he must also have access to the means of production, tools, raw materials, etc., necessary for his work, which will enable him to produce a commodity.

It is this necessity that makes initial accumulation necessary, and it is not at the stage where labour is divided that it can be found. This accumulation is not the result of the capitalist mode of production, but its point of departure. The question now is how this initial accumulation came about. Further on in book 2 of An Inquiry into the Nature and Causes of the Wealth of Nations, in chapter 1 entitled Of the division of stocks, Adam Smith explains the two types of man in society, the poor worker and the capitalist:

> “When the stock which a man possesses is no more than sufficient to maintain him for a few days or a few weeks, he seldom thinks of deriving any revenue from it. He consumes it as sparingly as he can, and endeavours, by his labour, to acquire something which may supply its place before it be consumed altogether. His revenue is, in this case, derived from his labour only. This is the state of the greater part of the labouring poor in all countries. But when he possesses stock sufficient to maintain him for months or years, he naturally endeavours to derive a revenue from the greater part of it, reserving only so much for his immediate consumption as may maintain him till this revenue begins to come in. His whole stock, therefore, is distinguished into two parts. That part which he expects is to afford him this revenue is called his capital. The other is that which supplies his immediate consumption, and which consists either, first, in that portion of his whole stock which was originally reserved for this purpose; or, secondly, in his revenue, from whatever source derived, as it gradually comes in; or, thirdly, in such things as had been purchased by either of these in former years, and which are not yet entirely consumed, such as a stock of clothes, household furniture, and the like. In one or other, or all of these three articles, consists the stock which men commonly reserve for their own immediate consumption.” (Adam Smith, An Inquiry into the Nature and Causes of the Wealth of Nations – Book 2 – Chapter 1 : Of the division of stocks [16])

Adam Smith says nothing about the origin of this initial accumulation. In the common imagination, we can undoubtedly cite the fable of the cicada and the ant by Jean de La Fontaine, where the figure of the ant shows us the figure of the hoarding saver and the cicada the unconscious squanderer. The ant plays the role of the entrepreneur who accumulates capital prudently through his own work, while the cicada plays the role of the worker who is too bohemian to secure his future through his own work.

La Cigale, ayant chanté
Tout l’été,
Se trouva fort dépourvue
Quand la bise fut venue :
Pas un seul petit morceau
De mouche ou de vermisseau.
Elle alla crier famine
Chez la Fourmi sa voisine,
La priant de lui prêter
Quelque grain pour subsister
Jusqu’à la saison nouvelle.
« Je vous paierai, lui dit-elle,
Avant l’Oût, foi d’animal,
Intérêt et principal. »

La Fourmi n'est pas prêteuse :
C'est là son moindre défaut.
« Que faisiez-vous au temps chaud ?
Dit-elle à cette emprunteuse.
— Nuit et jour à tout venant
Je chantais, ne vous déplaise.
— Vous chantiez ? J'en suis fort aise.
Eh bien ! Dansez maintenant. »
(Jean de La Fontaine, 1668, Fables livre I, [19])

And Karl Marx recounted the vision of political economy dominant in his time, in which accumulation was the result of personal labour and a sense of savings.

> « Il était une fois, il y a bien longtemps de cela, une élite laborieuse d'un côté, intelligente et avant tout économe, et, de l'autre, une bande de canailles fainéantes, qui gaspillait sans compter les biens de cette élite. [...] Dans la suave économie politique, c'est l'idylle qui a toujours régné. Droit et « travail » furent de tout temps les uniques moyens d'enrichissement, exception faite chaque fois, naturellement, de « cette année ». En réalité, les méthodes de l'accumulation initiale sont tout ce qu'on voudra sauf idylliques. » (Karl Marx, 2016, Le Capital livre 1, Éditions sociales, p691-692 [14])

Here we question the fact that 'law and labour' were the only means of enrichment. Marx criticises the idyllic view of capital accumulation as a series of equitable opportunities. He highlights the violent and coercive aspects of the process, which have their roots in acts of expropriation. Karl Marx sets out his analysis of the initial accumulation that made capitalist industrial production possible in chapter 24 of *Capital*.

> « Nous avons vu au chapitre IV que pour transformer la monnaie en capital, il ne suffisait pas qu'il y ait production de marchandises et circulation de marchandises. Il fallait d'abord quelqu'un qui possède de la valeur ou de la monnaie, et quelqu'un qui possède de la substance créatrice de valeur [de la force de travail] ; un possesseur de moyens d'existence et de moyens de production et un possesseur de force de travail (et de rien d'autre) ; et il fallait qu'ils se trouvent l'un en face de l'autre, l'un en tant qu'acheteur, l'autre en tant que vendeur. C'est la séparation entre le produit du travail et le travail lui-même, entre les conditions objectives du travail et la force de travail subjective, qui était donc la base effective, le point de départ du processus de production capitaliste. » (Karl Marx, 2016, Le Capital livre 1, Éditions sociales, p.553 [14])

To achieve this, the market must be polarised to allow the labour market. The fundamental condition for capitalist production is that on the one hand there are workers without the means of subsistence and the means of production, and on the other hand there are owners of these two means. There has to be a separation between the workers and the ownership of the conditions under which this work is carried out.

We therefore have the situation of a possessor of the means of subsistence (to support the worker) and the means of production (to make the worker work) and, on the other hand, a free worker, i.e. freed from traditional and symbolic social ties (workers belong to themselves) but also freed from all private property, which forces him to sell himself in order to subsist. This is how Karl Marx described the conditions of capitalism and wage labour.

> « Le travailleur ressort en permanence du processus comme il y était entré : source personnelle de richesse mais dépouillé de tous les moyens de réaliser cette richesse pour lui-même. Étant donné qu'avant même qu'il entre dans le processus, son propre travail lui est rendu étranger, que le capitaliste se l'approprie et qu'il est incorporé au capital, ce travail s'objective constamment pendant le processus en un produit d'autrui, étranger [qui n'est pas à lui]. » (Karl Marx, 2016, Le Capital livre 1, Éditions sociales, p.553 [14])

> « C'est avec cette polarisation du marché que sont données les conditions fondamentales de la production capitaliste. Le rapport capitaliste présuppose le divorce entre les travailleurs et la propriété des conditions de réalisation du travail. [. . .] Le mouvement historique qui transforme les producteurs en travailleurs salariés apparaît ainsi, d'un côté, comme leur affranchissement de la servitude et de la loi des corporations, et c'est ce côté seul que retiennent nos historiographes bourgeois. Mais, de l'autre côté, ces affranchis de fraîche date ne deviennent vendeurs d'eux-mêmes qu'après avoir été dépouillés de tous leurs moyens de production et de toutes les garanties qu'offraient pour leur existence les anciennes institutions féodales. Et l'histoire de cette expropriation est inscrite dans les annales de l'humanité en caractère de sang et de feu. » (Karl Marx, 2016, Le Capital livre 1, Éditions sociales, p.692-693 [14])

We can see that the transition to capitalist society is progressive in the sense that it dissolves the old symbolic ties, as we saw in the extract from the Communist Manifesto. Karl Marx analyses this initial accumulation, the necessary basis of capitalism, further on in *Capital*.

> « La soi-disant accumulation initiale n'est donc pas autre chose que le processus historique de désassemblage du producteur d'avec les moyens de production. » (Karl Marx, 2016, Le Capital livre 1, Éditions sociales, p.692 [14])

With this formulation, Marx plays on the etymology of the word 'accumulation', cumulate meaning to reunite, to assemble, whereas capital disassembles what was reunited.

> « L'accumulation primitive du capital. Inclut la centralisation des conditions du travail. C'est l'autonomisation des conditions de travail vis-à-vis du travailleur et du travail lui-même. Son acte de séparation historique qui transforme les conditions de travail en capital et le travail en travail salarié. Par là est donné la base de la production capitaliste. » (Karl Marx, 2024, Théories sur la plus-value – tome 3, Éditions sociales, p366 [20])

And what are the historical acts of separation on which this accumulation is based? Marx takes the example of Great Britain:

« Pillage des biens d'Église, aliénation frauduleuse des domaines de l'État, vol de la propriété communale, transformation usurpatoire de la propriété féodale et de la propriété du clan en propriété privée moderne, menée à son terme avec un terrorisme impitoyable : autant de méthodes idylliques de l'accumulation initiale. C'est par elles que furent conquis les champs pour l'agriculture capitaliste, que la terre fut incorporée au capital, et que fut créé pour l'industrie des villes l'apport nécessaire en prolétariat exploitable à merci. » (Karl Marx, 2016, Le Capital livre 1, Éditions sociales, p.709-710 [14])

It is here that we come to a junction with the second concept that will interest us in the rest of this theoretical section, namely the opposition between town and country. It is therefore expropriation which is the basis of this accumulation and which is also the basis for the influx of labour into the urban sector. The initial accumulation of capital was a process that prepared the ground for industrialisation. This process involved massive expropriations, forcing many peasants and rural dwellers to lose their land and traditional livelihoods. This loss created a dispossessed class of people who had no choice but to migrate to the cities in search of work, supplying the labour force for the emerging factories of the industrial age. In the rest of the chapter, he explains the various laws against vagrancy, which were intended to discipline all these former peasants, who had no means of production, to find work with a 'labour giver'.

« Au XVIIIe siècle on interdit en même temps l'émigration aux Gaëls chassés de leurs terres, pour les forcer à aller à Glasgow et dans d'autres villes industrielles. » (Karl Marx, 2016, Le Capital livre 1, Éditions sociales, p.706 [14])

« Ce prolétariat sans feu ni lieu, privé de toute protection juridique, chassé de son terroir par la dissolution des suites féodales et par des expropriations violentes et successives, ne pouvait en aucune manière être absorbé par les manufactures naissantes aussi rapidement qu'il avait été engendré. En outre, ces hommes brusquement arrachés à la trajectoire habituelle de leur existence ne pouvaient se faire aussi brusquement à la discipline de leur nouvel état. Ils se transformèrent massivement en mendiants, voleurs, vagabonds, partie par vocation, mais le plus souvent sous la pression des circonstances. D'où, à la fin du XVe et pendant tout le XVIe siècle, dans l'Europe occidentale, une cruelle législation draconienne contre le vagabondage. » (Karl Marx, 2016, Le Capital livre 1, Éditions sociales, p.710 [14])

Initial accumulation is therefore, in the final analysis, the abolition of ownership of one's own personal labour and of the product of that labour. By selling his labour-power, the worker alienates it, rents out his labour to the buyer of that labour-power.

« L'accumulation initiale du capital ne signifie rien d'autre que l'expropriation des producteurs immédiats, la dissolution de la propriété privée fondée sur le travail personnel. » (Marx, Le Capital, Éditions sociales, p.733 [14])

« La base de tout ce processus, c'est l'expropriation hors de sa terre du producteur rural, du paysan. » (Marx, Le Capital, Éditions sociales, p.694 [14]).

2.4.2 The opposition between the city and the countryside

The opposition between town and country is a theme that runs throughout Karl Marx's work. He will only mention it marginally, but we will try to see how the theoretical concepts he develops allow him to explain this opposition.

> « Une fois que la production capitaliste s'est emparée de l'agriculture, ou dans la mesure où elle s'en est plus ou moins emparée, la demande de population ouvrière agricole diminue de façon absolue à mesure que progresse l'accumulation du capital qui fonctionne dans ce domaine, sans que la répulsion qu'elle subit se complète, comme c'est le cas dans l'industrie non agricole, d'une attraction plus grande. Une partie de la population rurale se trouve donc continuellement sur le point de faire le saut dans le prolétariat urbain ou manufacturier, et en train de guetter les circonstances favorables à cette conversion (Manufacture désignant ici toute industrie non agricole). Cette source de surpopulation relative coule donc en permanence. Mais son afflux permanent vers les villes présuppose une surpopulation latente continue à la campagne, dont le volume ne devient visible qu'à partir du moment où les canaux d'évacuation s'ouvrent largement de façon exceptionnelle. L'ouvrier agricole se trouve donc réduit au minimum de salaire et a toujours un pied dans le marécage fangeux du paupérisme. » (Marx, Le Capital, Éditions sociales, p.624 [14]).

Urbanisation was largely fuelled by this migration from rural to urban areas, stimulated by the economic opportunities offered by industrialisation. As Adam Smith noted, the division of labour became more complex and specialised with the development of industry. In industrial cities, the division of labour intensified, with individuals occupying very specific roles in the production process, in contrast to rural life where tasks were more generalist.

In his book, Smith highlights the way in which the initial accumulation of capital and the division of labour paved the way for a world in which the specialisation of functions became the key to economic growth. However, it is in the cities that this specialisation finds its most acute expression, attracting incessant flows of migrants from the countryside to rapidly expanding urban centres. These migratory movements, initially provoked by dispossession and the search for means of subsistence following agricultural losses or expropriation, as demonstrated in Subsection 2.4.1, gradually become a permanent feature of capitalist society, engendering massive urbanisation. Thus, in the *Manifesto*, it is mentioned:

> « La bourgeoisie a soumis la campagne à la ville. Elle a créé d'énormes cités; elle a prodigieusement augmenté la population des villes par rapport à celles des campagnes, et par là, elle a arraché une grande partie de la population à l'abrutissement de la vie des champs. [...] La bourgeoisie supprime de plus en plus l'émiettement des moyens de production, de la propriété et de la population. Elle a aggloméré la population, centralisé les moyens de production et concentré la propriété dans un petit nombre de mains. La conséquence totale de ces changements a été la centralisation politique. » (Karl Marx et Friedrich Engels, Manifeste du parti communiste [18])

Marx, critical of Smith's vision, deepened this analysis by revealing the brutal mechanisms underlying this accumulation and migration. In his view, the real dynamic of industrialisation lay not in saving and self-sufficiency, but rather in a series of expropriations and coercions that not only fed the factories of the industrial age with cheap labour, but also erected the social and economic barriers that define capitalism. Migration to the cities was therefore not just a quest for opportunity, but often a forced flight from the total loss of autonomous means of production, making workers dependent on wages for their survival.

Urbanisation, in this context, can be seen as the theatre of social transformation, where old feudal and communal ties are dismantled in favour of relations determined by capital. Cities become centres of economic power where capital and labour clash, redrawing class relations and intensifying social struggles. These urban centres are not only places of industrial production but also centres of resistance and protest, where the contradictions of capitalism are concentrated.

Industrialisation transformed not only economies but also social structures and modes of production. Factories exacerbated the division of labour on a scale never seen before and required a concentration of labour and resources, leading to major changes in the way towns were organised and in the social relations within these communities. This process often led to precarious living conditions for workers, giving rise to collective resistance. Workers who became proletarians were forced to sell their labour in order to survive, exacerbating social inequalities. This process created an urban proletariat whose survival depended entirely on the fluctuations of the labour market, marking a definitive break with agricultural and community lifestyles.

> « Plus le capital est accumulé rapidement dans une ville industrielle ou commerciale, et plus l'afflux du matériau humain exploitable est rapide et les logements improvisés des travailleurs misérables. Cela explique pourquoi Newcastle-upon-Tyne, en tant que centre d'un district où les carrières et les mines de charbon continuent à être de plus en plus productives, occupe après Londres la deuxième place dans l'Inferno du logement. 34 000 personnes, pas moins, habitent ici dans des logements d'une seule pièce. Il y a peu de temps, la police sanitaire de Newcastle Gateshead a détruit pour cause de danger public absolu un nombre considérable de maisons. La construction de nouvelles maisons progresse très lentement, les affaires vont très vite. Si bien qu'en 1865 la ville était bien plus encombrée qu'auparavant. » (Marx, Le Capital, Éditions sociales, p.642 [14]).

In this great painting of social and economic transformation, the theories of Smith and Marx are not simply historical narratives; they are tools for understanding how the forces of capital accumulation, migration, urbanisation and industrialisation have combined to shape modern societies. They allow us to grasp how modern economic rationality is based on the violent revolutionary change of the bourgeoisie, which abolishes the old feudal society through the dynamics of power and expropriation. This understanding is crucial not only for economists, but for anyone interested in the real dynamics that govern our societies and the possibilities for transforming them.

2.5 Urbanization in least developed countries

Urbanization transforming cities both demographically and socio-economically, has been a significant trend in least developed countries (LDCs) since the 1970s. This phenomenon has profound implications for economic development, social structures, and environmental sustainability in these regions [21].

Since the 1970s, urbanization in LDCs has accelerated due to several factors, including industrialization, globalization, and rural-urban migration driven by the search for better economic opportunities. The rate of urban growth in these countries has often outpaced the ability of urban infrastructure to adapt, leading to a unique set of challenges and opportunities. The Harris-Todaro model that we will use for this master thesis was built in the 1970s. Numerous theoretical and critical debates continue to be conducted.

David Harvey in 'The new imperialism: accumulation by dispossession' [22] [23], quotes Rosa Luxemburg [24].

> The other aspect of the accumulation of capital concerns the relations between capitalism and the non-capitalist modes of production which start making their appearance on the international stage. Its predominant methods are colonial policy, an international loan system – a policy of spheres of interest – and war. Force, fraud, oppression, looting are openly displayed without any attempt at concealment, and it requires an effort to discover within this tangle of political violence and contests of power the stern laws of the economic process

Beyond these considerations, urbanization has been a catalyst for economic development. In LDCs, cities have become centers of economic activity, fostering innovation, creating jobs, and facilitating economies of scale. Urban areas offer better access to education, healthcare, etc., which can enhance productivity. Globalization and the shift from agrarian economies to more diversified urban economies has destroyed local lifestyles and sectors like manufacturing and services have seen significant growth in urban settings, providing employment to millions. Furthermore, a notable feature of urbanization in LDCs is the expansion of the informal economy. This sector often constitutes a substantial part of urban (un)employment and economic activity. Informal economy represents from 75% to 89% in urban sector in least urbanized countries [25].

While urbanization has the potential to drive economic growth, it also presents several challenges as urban poverty and slums, infrastructure deficits or environmental degradation. Rapid urban growth has often led to the proliferation of informal settlements or slums. These areas are characterized by inadequate housing, poor sanitation, and limited access to essential services [26]. In the same perspective, many LDCs face significant infrastructure deficits, including inadequate transportation networks, electricity, and water supply. These shortcomings hinder the potential economic benefits of urbanization[27]. Finally, urbanization in LDCs leads to environmental challenges such as air and water pollution, waste management issues, and the loss of green spaces. The environmental impact of rapid urban growth is a critical concern for sustainable development [28].

In the various sources above, the solutions put forward by the United Nations defends the need to reinvent governance and planning for urban transitions to combat the problems exposed.

2.6 Theorical development and the Harris-Todaro model

This part of the master thesis does not fundamentally challenge the Harris-Todaro model. It underscores the importance of having a solid theoretical foundation and comparing it against empirical data to deepen our understanding of how our world operates. In this section, we have explored how misconceptions about the emergence of capitalism, the division of labor, and migration since the 15th century were originally presented by Adam Smith and later corrected by Karl Marx in *Capital*. As explained before in this work :

> Marx, critical of Smith's vision, deepened this analysis by revealing the brutal mechanisms underlying this accumulation and migration. In his view, the real dynamic of industrialisation lay not in saving and self-sufficiency, but rather in a series of expropriations and coercions that not only fed the factories of the industrial age with cheap labour, but also erected the social and economic barriers that define capitalism. Migration to the cities was therefore not just a quest for opportunity, but often a forced flight from the total loss of autonomous means of production, making workers dependent on wages for their survival.

Continuing in this critical vein, it seems crucial to ensure that the theoretical bases underpinning our model's hypotheses are supported by the empirical data available to us.

For instance, the notion that individual labor allowed workers to become capitalists could underlie hypotheses aiming to construct an agent-based model designed to illustrate how capitalism evolved from an agrarian society. However, this representation would likely deviate significantly from the objective reality of capitalism's development in Europe over the past 500 years. It is with this perspective that I returned to these classical texts of political economy and Marx. Unfortunately, the scope of a master's thesis did not permit further exploration of this subject. However, further work could be undertaken to verify the validity of these hypotheses.

Chapter 3

Literature Review

In this section, we will look at the papers that will be used to establish the preliminary knowledge for the developments in this master thesis.

The paper form Gallup [29] surveys theoretical models of migration decision-making in 1997. We will develop different models in this literature review and explain the general framework for understanding migration.

In this master thesis, we will takle to the Harris-Todaro model more in detail. We will therefore use different state of the art about it as the one written by Ingene [30], Bahns [31], Laing et al. [32], and also about dynamics of migration and of the model as Inoue's paper [33], de Haas [34].

Finally, we will focus on agent-based models like Silveira et al. [35], Espindola et al. [36], Cai et al. [37], Klabunde and Willekens [38], Boulahbel et al. [39] and Fu and Hao [40].

3.1 Migration models

3.1.1 Gravity model

As for the gravity model of trade introduced by Walter Isard in 1954 [41], the gravity model is used by geographers to takle spatial interactions such as journeys to work, migrations, tourism, the usage of public facilities, the transmission of information or capital, the market areas of retailing activities, international trade and freight distribution [42]. Economic activities involve generating goods and attracting demand. When things move between places, it means the benefits outweigh the costs of that interaction.

From Gallup's paper, we find a quote from the Principles of social science written in 1859 by Carey [43].

> "Man, the molecule of society, is the subject of Social Science.... The great law of Molecular Gravitation [is] the indispensable condition of the existence of the being known as man.... The greater the number collected in a given space, the greater is the attractive force that is there exerted.... Gravitation

is here, as everywhere, in the direct ratio of the mass, and the inverse one of distance." (Carey, Principles of social science, 1859, p41-43. Taken from [29])

This gives us the well known equation:

$$M_{ij} \propto \frac{P_i \cdot P_j}{D_{ij}} \tag{3.1}$$

where

$$\begin{aligned} M_{ij} &= \text{migration from region } i \text{ to region } j, \\ P_i, P_j &= \text{population in region } i \text{ and } j, \text{ respectively}, \\ D_{ij} &= \text{distance between region } i \text{ and region } j. \end{aligned}$$

Empirically, this model can provide highly accurate correlations, likely attributed to transportation costs, information flows and the economies of agglomeration. However, the drawback of the gravity model lies in its inability to capture individual behavior. It does not effectively depict the decision-making process behind migration, at least not in a convincing manner. So this model will not be of interest to us for this master thesis.

3.1.2 Human Capital and Expected Income

The first model of individual migration decision-making within the economic framework is Sjaastad's model in 1962 [44]. Sjaastad's model views migration as an investment decision in human capital, where individuals weigh the costs and benefits of moving. Migrants move when the expected gains, mainly driven by income differences, outweigh the costs, including monetary expenses, loss of professional experience, social ties, and uncertainty.

Todaro built in 1969 upon Sjaastad's ideas in [45], focusing on job uncertainty and the migrant's impact on urban unemployment. He proposed that migrants move if the expected income gain in the city outweighs moving costs. The decision hinges on comparing the present value of expected gains with the costs of moving. Todaro's model introduces variables like fixed rural and urban incomes, job probabilities, and migration costs. However, it has shortcomings: assumes known and fixed wages, lacks dynamic decision-making, implies a one-time migration, and presents an unusual job-finding process in cities. Despite these, it sheds light on urban job creation's effects on unemployment, observed in Todaro's experience in Nairobi.

$$V_0 = \sum_{t=0}^{T} (p_t \cdot y_u - y_r)\beta^t - C_0 \tag{3.2}$$

where

$$\begin{aligned} V_0 &= \text{present discounted value of expected gain from migrating over the} \\ &\quad\ \text{migrant's time horizon}, \\ p_t &= \text{probability of being employed in the urban sector at time } t, \\ y_u, y_r &= \text{fixed real urban and rural incomes, respectively}, \\ \beta^t &= \text{discount factor for time period } t, \\ C_0 &= \text{cost of moving incurred at time } t = 0. \end{aligned}$$

This equation represents the calculation of the present value of the expected gain from migrating over the migrant's time horizon, taking into account the probabilities of employment in the urban sector, fixed urban and rural incomes, the discount factor for each time period, and the initial cost of moving. If V_0 is positive, the decision to migrate has a positive utility for the rural worker, on the other hand, it has a negative utility, the worker will not migrate.

3.1.3 Harris-Todaro Model

In their 1970 original paper[46], Harris and Todaro intend to demonstrate that rural-to-urban migration, even in the presence of urban unemployment, is a rational choice for rural workers. At its core, their model is a two-sector economy between an urban sector based on manufacture or industry and a rural sector based on agriculture. The wages are fixed in both sectors but the wages are higher than market clearing level in manufacture. The migration cost is zero. Therefore, the decision to migrate depends on whether the expected wage in the city is higher than the guaranteed salary in the country. Rural to urban migration occurs if:

$$w_m(1-\lambda) > w_a \tag{3.3}$$

where

$$\begin{aligned} w_m &= \text{wage in manufacture;} \\ w_a &= \text{wage in agriculture;} \\ \lambda &= \text{urban unemployment rate.} \end{aligned}$$

Equilibrium is reached when agricultural and urban expected wages are equalized, halting rural-to-urban migration: $w_m(1-\lambda) = w_a$. Jobs in the city are considered to be in manufacturing where there are two possibilities for workers, the formal and informal sectors. In the formal sector, workers' unions have bargaining power over wages and work is governed by laws. In the informal sector, there are often no contracts and no special protection or benefits for workers compared with the formal sector. Here at the beginning we can assume that informal sector has a wage of zero.

After this paper, the "Harris-Todaro model" is kept as name for the two-sector economy model to model migration from rural to urban area. This model will receive lots of improvements and additions. Among the many different economist that worked on this model, we can cite three papers in order to get literature reviews of the Harris-Todaro model: Ingene [30], Bahns [31], Laing et al. [32]. The thesis written by Karin Margarethe Bahns in 2005 entitled "Rural-to-urban migration in developing countries - The Applicability of the Harris Todaro Model with a Special Focus on the Chinese Economy" gives us a basic and an extended version of the model.

3.1.4 Basic Harris-Todaro model

This basic model is taken from [31, 46]. We will give here a summary of the equations used to formalize the model and the explanations.

Note that a, m, F, and I stand for agriculture, manufacturing, formal and informal respectively. In what follows, all variables displayed as x^f to foreign values. Those taking the form of x^* show equilibrium values and those taking the form of $\bar{x}$ show fixed values.

The agricultural and manufacturing production functions take the following form:

$$Y_a = f_a(L_a, \bar{N}, \bar{K}_a), \quad f'_a > 0, f''_a < 0 \tag{3.4}$$

$$Y_m = f_m(L_m, \bar{K}_m), \quad f'_m > 0, f''_m < 0 \tag{3.5}$$

where

$$\begin{aligned} Y_i &= \text{ output, where } i = a, m; \\ L_i &= \text{ labour input in production of } Y_i, \text{ where } i = a, m; \\ \bar{N} &= \text{ fixed amount of land;} \\ \bar{K}_i &= \text{ fixed capital input, where } i = a, m. \end{aligned}$$

These sectors are labor-intensive for the rural sector and capital-intensive for urban sector. The rural sector utilizes labor, capital, and land for production, while the manufacturing sector employs only labor and capital. Although capital is specific to each sector, it's fully deployed in production. The total capital remains constant, the land is fully used and labor has the flexibility to transition between sectors but isn't fully engaged.

The terms of trade are expressed as the price of the agricultural good in terms of the manufactured good. Due to the assumption of homothetic preferences, the terms of trade can be expressed as a function of the relative outputs of the agricultural and the manufactured good.

$$p = \frac{P_a}{P_m} = \rho\left(\frac{Y_m}{Y_a}\right), \quad \rho' > 0 \tag{3.6}$$

where

$$\begin{aligned} p &= \text{ terms of trade;} \\ &\Leftrightarrow \text{ price of agricultural good express in manufactured goods;} \\ P_i &= \text{ price of goods, where } i = a, m. \end{aligned}$$

Both goods and labor markets are perfectly competitive. In the rural sector, the wage, perfectly flexible, is equal to the marginal productivity of labor in this sector.

$$w_a = pf'_a \tag{3.7}$$

where

$$w_a = \text{ wage in agriculture.}$$

In a perfectly competitive economy, urban producers are looking to maximise their profits. As a consequence, the manufacturing wage is equal to the value of the marginal product of labour in the formal urban sector. However, this wage is bounded by a minimum threshold (minimum wage), established institutionally, which dictates that it must be at least as high as a predetermined level denoted as $\bar{w}_m$. This institutionally fixed minimum wage is defined in relation to the manufacturing output.

$$w_m = pf'_m \geq \bar{w}_m \tag{3.8}$$

where

$$\begin{aligned} w_m &= \text{ manufacturing wage;} \\ \bar{w}_m &= \text{ the minimum wage in urban sector.} \end{aligned}$$

Equilibrium Conditions

Rural migrants base their migration decision not solely on the prevailing urban wage rate (or the fixed minimum wage $\bar{w}_m$), but rather on the concept of an expected wage. They factor in the possibility of not securing urban employment by considering the urban wage weighted by the urban unemployment rate. Thus, the decision to migrate depends on the minimum wage multiplied by the likelihood of finding employment in the urban area. This probability, represented by $\frac{L_m}{L_u}$, where L_u is the total urban labor force (comprising employed individuals and the unemployed L_{un}), signifies the ratio of employed workers to the total urban labor force. Naturally, this ratio is less than one in the presence of unemployment and equal to zero if there is no unemployment. Hence, the expected urban wage utilized by migrants in their decision-making process is determined by this consideration.

$$w_u^e = \frac{\bar{w}_m L_m}{L_u}, \quad \frac{L_m}{L_u} \leq 1. \tag{3.9}$$

where

$$\begin{aligned} w_u^e &= \text{ expected urban wage;} \\ \bar{w}_m &= \text{ the minimum wage in urban sector;} \\ L_m &= \text{ labor input in manufacturing jobs;} \\ L_u &= \text{ labor input in urban sector.} \end{aligned}$$

In this model, we have no population growth therefore we know that the total population is constant and that the population in both sector at the equilibrium is equal to the sum of workers in agriculture and manufacturing before migration.

$$\bar{L} = L_a + L_u = \bar{L}_r + \bar{L}_u \tag{3.10}$$

where

$$\begin{aligned} \bar{L} \quad &= \text{ fixed total labor supply;} \\ L_a \quad &= \text{ labor input in agriculture;} \\ \bar{L}_r, \bar{L}_u &= \text{ initial endowments of labour in rural and urban sector, respectively.} \end{aligned}$$

We also have that unemployment in the city is represented by $L_{un} = L_u - L_m$. Therefore we have the unemployment rate $\lambda = L_{un}/L_u$.

In equilibrium the agricultural wage and the urban expected wage are equalised, and rural-to-urban migration ceases.

$$w_a = w_u^e = \bar{w}_m(1 - \lambda) \tag{3.11}$$

where

$$\lambda = \text{ urban unemployment rate.}$$

3.2 Agent-based models for rural-urban migration

Finally, we will focus on agent-based models. These agent-based models can be defined as in MESA documentation [47]:

> Agent-based models are computer simulations involving multiple entities (the agents) acting and interacting with one another based on their programmed behavior. Agents can be used to represent living cells, animals, individual humans, even entire organizations or abstract entities. Sometimes, we may have an understanding of how the individual components of a system behave, and want to see what system-level behaviors and effects emerge from their interaction. Other times, we may have a good idea of how the system overall behaves, and want to figure out what individual behaviors explain it. Or we may want to see how to get agents to cooperate or compete most effectively. Or we may just want to build a cool toy with colorful little dots moving around.

The paper from Klabunde and Willekens (2016) [38] entitled : "Decision-Making in Agent-Based Models of Migration: State of the Art and Challenges" gives us a good understanding on the different agent-based models in the literature.

The papers from Silveira et al. [35] and Espindola et al. [36] will be used in order to create our agent-based model.

3.2.1 Model from Espíndola et al. (2006)

In order to build the model, we need to specify the production functions. In our two sector economy, the production function for agricultural and manufactured goods are described by Cobb-Douglas production functions.

Production functions

The rural sector production function is:

$$Y_a = A_a L_a^{\phi} \tag{3.12}$$

where

$$\begin{aligned}
Y_a \quad &= \text{ production level of the agricultural good;} \\
L_a \quad &= \text{ amount of workers used in the agricultural production;} \\
A_a > 0 \quad &= \text{ parametric constant;} \\
0 < \phi < 1 &= \text{ parametric constants.}
\end{aligned}$$

The urban sector production function is:

$$Y_m = A_m L_m^{\alpha} \tag{3.13}$$

where

$$\begin{aligned}
Y_m &= \text{production level of the manufactured good;} \\
L_m &= \text{amount of workers used in the manufacture production;} \\
A_m > 0 &= \text{parametric constant;} \\
0 < \alpha < 1 &= \text{parametric constants.}
\end{aligned}$$

Wages

In the rural sector, the wage (expressed with p by equation 3.7), is given by the derivative of the production function:

$$w_a = \phi A_a L_a^{\phi-1} p \tag{3.14}$$

where

$$\begin{aligned}
w_a &= \text{wage in rural sector;} \\
p &= \text{price of the agricultural good;} \\
&\quad \text{both expressed in units of manufactured good.}
\end{aligned}$$

The wage in the urban sector is also expressed by the derivative of the production function exept that we assume a minimum wage:

$$w_m = \alpha A_m L_m^{\alpha-1}, \quad \text{such that } L_m \leq L_u; \tag{3.15}$$

where

$$\begin{aligned}
w_m &= \text{wage in urban sector;} \\
L_u &= \text{amount of workers in the urban sector.}
\end{aligned}$$

The relative price of the agricultural good in terms of the manufactured good, p, varies according to the relative scarcity between agricultural and manufacturated goods.

$$p = \rho \left(\frac{Y_m}{Y_a} \right)^{\gamma} \tag{3.16}$$

where

$$\gamma = \text{parameter.}$$

Temporary equilibrium

We can calculate the temporary equilibrium of the model by using the equations developed above. We have the parametric constant vector $(A_a, A_m, \phi, \alpha, \rho, \gamma)$, an initial urban population L_u and a minimum wage w_m.

By rearranging the wage expression in Equation 3.15, we can find the employment level at the manufacturing sector:

$$w_m = \alpha A_m L_m^{\alpha-1}$$
$$\Leftrightarrow L_m^{1-\alpha} = \frac{\alpha A_m}{w_m}$$
$$L_m = \left(\frac{\alpha A_m}{w_m}\right)^{\frac{1}{1-\alpha}} \tag{3.17}$$

Replacing Equation 3.17 in the production function described in Equation 3.13, we get the production level of the manufacturing sector:

$$Y_m = A_m L_m^{\alpha}$$
$$\Leftrightarrow Y_m = A_m \left(\left(\frac{\alpha A_m}{w_m}\right)^{\frac{1}{1-\alpha}}\right)^{\alpha}$$
$$\Leftrightarrow Y_m = A_m \left(\frac{\alpha A_m}{w_m}\right)^{\frac{\alpha}{1-\alpha}}$$
$$\Leftrightarrow Y_m = A_m \cdot A_m^{\frac{\alpha}{1-\alpha}} \left(\frac{\alpha}{w_m}\right)^{\frac{\alpha}{1-\alpha}}$$
$$\Leftrightarrow Y_m = A_m^{\frac{1-\alpha}{1-\alpha}} \cdot A_m^{\frac{\alpha}{1-\alpha}} \left(\frac{\alpha}{w_m}\right)^{\frac{\alpha}{1-\alpha}}$$
$$\Leftrightarrow Y_m = A_m^{\frac{1-\alpha+\alpha}{1-\alpha}} \left(\frac{\alpha}{w_m}\right)^{\frac{\alpha}{1-\alpha}}$$
$$Y_m = A_m^{\frac{1}{1-\alpha}} \left(\frac{\alpha}{w_m}\right)^{\frac{\alpha}{1-\alpha}} \tag{3.18}$$

Knowing that the labor input (number of workers) in agriculture is the total population minus the urban population:

$$L_a = L - L_u \tag{3.19}$$

We can replace the agricultural labor input described in Equation 3.19 in Equation 3.12 and find the following production function.

$$Y_a = A_a (L - L_u)^{\phi} \tag{3.20}$$

We can compute p in function of Equation 3.18 and Equation 3.20

$$p = \rho \left[\frac{A_m^{\frac{1}{1-\alpha}} \left(\frac{\alpha}{w_m}\right)^{\frac{\alpha}{1-\alpha}}}{A_a (L - L_u)^{\phi}}\right]^{\gamma} \tag{3.21}$$

Therefore, by replacing Equation 3.21 in 3.14, we find a expression for agricultural wage in function of manufacturing wage and urban population.

$$w_a = \phi \rho A_a^{1-\gamma} A_m^{\frac{\gamma}{1-\alpha}} \left(\frac{\alpha}{w_m}\right)^{\frac{\alpha\gamma}{1-\alpha}} \frac{1}{(L - L_u)^{1-\phi+\gamma\phi}} \tag{3.22}$$

This is a temporary equilibrium since the difference between the agricultural wage and manufacturing wage continues to induce migration and to change the sectorial distribution of overall. Therefore, a change in the number of worker in a sector will change the wage in this sector (or the expected wage in urban sector if we are already at minimum wage). until the long term equilibrium is reached.

Long run equilibrium

The expected wage in urban sector is equal to the manufacturing wage times the employment rate, assuming that every worker has the same chance to be engaged.

$$w_u^e = \frac{L_m}{L_u} w_m \tag{3.23}$$

In the long run, the migration stops when the equilibrium is reached. This equilibrium is reached when the agricultural wage and the expected urban wage will adjust and become equal. At this point, there is no more incentive to migrate and the migration process stops.

$$w_u^e = w_a$$

The difference between the agricultural wage and the expected urban wage is zero. This equality is known in the economic literature as the Harris-Todaro condition.

$$w_u^e - w_a = 0 \tag{3.24}$$

Harris and Todaro generalized this condition to a threshold. The generalized Harris-Todaro condition explains that the net migration between rural and urban areas ceases when the differential of expected wages reaches a certain constant value, denoted as δ (where $\delta \neq 0$). In other words, once the wage differential attains this threshold value, there is no further incentive for individuals to migrate from one area to another.

$$w_u^e - w_a = \delta \tag{3.25}$$

This generalized Harris-Todaro condition in Equation 3.25 can be expressed with the agricultural wage found in Equation 3.22 and the expected urban wage found in Equation 3.23.

$$\frac{L_m}{L_u} w_m - \phi \rho A_a^{1-\gamma} A_m^{\frac{\gamma}{1-\alpha}} \left(\frac{\alpha}{w_m}\right)^{\frac{\alpha\gamma}{1-\alpha}} \frac{1}{(L - L_u)^{1-\phi+\gamma\phi}} = \delta \tag{3.26}$$

Stability

The long-run equilibrium is globally asymptotically stable [36]. This means that the economy tends towards this equilibrium regardless of the initial conditions. In this context, the long-run equilibrium is characterized by unemployment in the urban sector due to a relatively high minimum wage. The urban population derivative is expressed below. This derivative has a negative derivative that means that the speed of change in urban population is diminishing with growing urban population. If the difference between expected urban wage and agricultural wage is zero, the population is stable and the migration stops.

$$\dot{L}_u = \Psi(w_u^e - w_a), \text{ with } \Psi' > 0 \text{ and } \Psi(0) = 0 \tag{3.27}$$

By using the expressions that we formulate above, we obtain:

$$\dot{L}_u = \Psi\left(\frac{L_m}{L_u} w_m - \phi \rho A_a^{1-\gamma} A_m^{\frac{\gamma}{1-\alpha}} \left(\frac{\alpha}{w_m}\right)^{\frac{\alpha\gamma}{1-\alpha}} \frac{1}{(L - L_u)^{1-\phi+\gamma\phi}}\right), \text{ with } \Psi' > 0 \text{ and } \Psi(0) = 0 \tag{3.28}$$

3.2.2 Other models

In literature, there are lots of agent-based models for migration. We can find other equations for production function and wages in papers like the ones of Silveira et al. [35] and Cai et al. [37]. Here are the version they developped. It includes more parameters and the number of firms or farms in each sectors.

Production functions

The rural sector production function is:

$$Y_a = \xi_3 L_a^{\phi} \tag{3.29}$$

with

$$\xi_3 = A_a Z_a^{1-\phi} \tag{3.30}$$

where

$$Z_a = \text{ is the amount of farms which constitute the agricultural sector.}$$

The urban sector production function is:

$$Y_m = \xi_1 L_m^{\alpha} \tag{3.31}$$

with

$$\xi_1 = A_m Z_m^{1-\alpha} \left[\left(\frac{\eta}{1-\eta} \right)^{\eta} \left(1 - \frac{\eta}{b} \right) \right]^{\alpha} \tag{3.32}$$

where

$$\begin{aligned} Z_m &= \text{ is the amount of firms which constitute the manufacturing sector;} \\ 0 < \eta < 1 &= \text{ parametric constant;} \\ b > 0 &= \text{ parametric constant.} \end{aligned}$$

Wages

In the rural sector, the wage is:

$$w_a = \xi_4 L_a^{\phi-1} p \tag{3.33}$$

with

$$\xi_4 = \frac{A_a \phi}{Z_a^{\phi-1}} \tag{3.34}$$

The wage in the urban sector is:

$$w_m = \xi_2 L_m^{\alpha-1}, \quad \text{such that } L_m \leq L_u; \tag{3.35}$$

with

$$\xi_2 = \alpha A_m \left(\frac{\eta}{1-\eta} \right)^{\alpha\eta} \left[\left(1 - \frac{\eta}{b} \right) \frac{1}{Z_m} \right]^{\alpha-1} \tag{3.36}$$

3.3 Decision-Making in agent-based model of migration

In their paper, Klabunde and Willekens [38] have drawn up a state of the art and the challenges encounter in decision-making agent-based model of migration. Agent-based models of migration have primarily succeeded in explicitly modeling network interactions, which are often crucial determinants of migration decisions. They identify the critical choices that must be made in developing an agent-based model, namely the modelling of decision processes and social networks. They also discuss two challenges that hamper the widespread use of agent-based model in the study of migration and, more broadly, demography and the social sciences: (a) the choice and the operationalisation of a behavioural theory (decisionmaking and social interaction) and (b) the selection of empirical evidence to validate the model.

3.3.1 Decision theories

Different disciplines and researchers utilize various decision theories to model migration decisions. Economists often employ utility maximization models, while social scientists may use theories from cognitive psychology. Physicists, on the other hand, tend to prefer minimalistic models. Some authors may combine aspects from different theories, leading to somewhat arbitrary behavioral rules.

Klabunde and Willekens [38] suggest that researchers should aim to use a single decision theory, such as utility maximization or the theory of planned behavior, to model migration decisions. This approach can enhance the coherence and clarity the model.

Models based on theory, rather than solely on data, offer several advantages. They can provide insights beyond mere extrapolation and allow researchers to predict how agents will react to significant changes in their conditions. Additionally, theoretical models can be tested and compared to real-world data, even in situations where data availability is limited or inadequate.

The use of agent-based models based on behavioral theory enables researchers to test the implications of different theories and compare their predictions to real-world observable data. This approach facilitates a deeper understanding of migration dynamics and the factors influencing individual decision-making processes.

3.3.2 Role of empirical data

Agent-based models vary in the amount of empirical data they use, ranging from none to a large amount. Empirical data are used for estimation and validation.

Estimation involves determining plausible parameter values and selecting the most acceptable ones. This often involves minimizing the difference between observations and simulations.

Validation involves comparing model outcomes with data. Validation can be made in a three-step procedure. The procedure includes finding reasonable parameter ranges from data, qualitatively checking whether the model works as expected, and estimating parameters for a good fit.

An alternative approach is to build a statistical meta-model of the agent-based model, which can explore the parameter space more efficiently. Gaussian emulators are examples of such meta-models.

3.3.3 Individual migration choice in models

Understanding individual choices is crucial to building demographic and economic models. To understand individual migration choices, there are different methods for modelling individual behaviour, and the literature identifies, among other things, five criteria that influence a rural worker's decision to migrate.

In the context of the Harris-Todaro model, we will not include inequalities in wealth and education in the rural sector, the cost of migration or other policy interventions in our model.

Migration decisions are strongly influenced by the potential economic gains linked to skills that can be developed in urban areas. Education provides access to urban jobs, particularly well-paid jobs. This criterion highlights the fact that the distribution of wealth and skills plays an essential role in determining who migrates and who stays in rural areas.

Migration decisions are also influenced by the age of the potential migrant, as explained by Anna Klabunde [48]. Younger people are more likely to migrate due to their higher risk tolerance and longer time horizon to reap the benefits of their move, while older people may be more risk averse and less inclined to relocate.

Paul R. Masson [49] addresses the concept of urban poverty traps, where high unemployment rates in urban areas can reduce household wealth below the cost of acquiring skills, trapping them in poverty. The model shows how migration, motivated by the search for better opportunities, can paradoxically lead to an increase in urban poverty due to unemployment and inadequate skills acquisition.

The above three criteria will not be used because they do not fit with our initial hypotheses of worker homogeneity.

The importance of understanding the cognitive processes underlying migration decisions is emphasised by [50]. These include how individuals perceive risks and benefits, gather and process information, and how their preferences and social networks influence their choices. Agent-based demographic models (ABMs) need to take account of the complexity of human behaviour, which involves multiple levels of decision-making and interaction.

Finally, understanding individual decision-making in migration can contribute to the design of policies that are better adapted to migrants' motivations and constraints. For example, policies aimed at providing better information on job opportunities and living conditions in potential destinations can help individuals to make more informed migration decisions.

These last three criteria will be used, namely: the influences of social networks, decision-making that takes account of the agent's past choices and the influence of policies linked to these dynamics.

3.4 Dynamics of agent-based models

This master thesis focus on the dynamics of migration and on the Harris-Todaro model. We will here focus on the literature review that tackle the dynamics of agent-based models for rural-urban migration.

3.4.1 Boulahbel-Bachari et al.

The model from 2017 by Boulahbel-Bachari, Saadi and Bah [39] given in the reference book *Applied Computational Intelligence and Mathematical Methods: Computational Methods in Systems and Software* exposes an agent-based modelling approach of migration dynamics. This subsection will present this dynamics.

This model consider various individual attributes such as age, gender, employment status, level of education, marital status, and the presence or absence of migratory networks. These factors significantly impact an individual's likelihood to migrate.

The identification of potential migrants relies on computing a discriminant function concerning these variables for individuals who are either unemployed or informally employed, aged between 16 and 59, and residing in rural areas. This function is expressed as a linear combination of these variables ($F_j(t)$). A person is classified as a potential migrant if they meet the following criterion:

$$F_j(t) \times u \times \theta > T \tag{3.37}$$

where

$$\begin{aligned} F_j(t) &= \text{ a linear combination of employment status, age and residential sector;} \\ u, \theta &= \text{ are parameters;} \\ T &= \text{ is the parametric threshold.} \end{aligned}$$

Once potential migrants are found, these potential migrants will compare their wage with the expected wage of the other sector and if the expected wage is higher, the potential migrant will migrate.

All unemployed people or informal workers aged between 16 and 59 are actively looking for work. Skilled workers can apply for both skilled and unskilled jobs, while unskilled workers can only apply for unskilled jobs. Every year, job opportunities arise in each region due to retirements, the creation of new jobs or the closure of existing businesses.

Those with the best chances of finding a formal job are hired first. However, persistent unemployment can lead to discouragement if the individual fails to find a job after numerous attempts. It is assumed that the probability of being asked for a job increases with the individual's qualifications and the length of their job search. This probability is represented mathematically as follows:

$$P_i^j(t) = P_i^j(t)(t-1) + (1 - P_i^j(t)(t-1)) \cdot \pi_i^j(t) \tag{3.38}$$

where

$P_i^j(t)(t)$	is the probability of being required for agent i in region j at time t,
$\pi_i^j(t)$	is the probability of finding a job for agent i in region j at time t.

3.4.2 Cai et al.

Cai et al. [37] developed the dynamics of the agent-based model. They tried to introduce dynamics into previous agent-based models and especially the model from Silveira et al. described in Subsection 3.2.1. Equation 3.39 explains the propensity of worker i to work in the rural or urban sector. Workers are influenced by their previous decisions, their social environment and the difference between the agricultural wage and the expected urban wage.

$$\dot{x}_i(t) = \underbrace{ax_i(t)}_{\text{inertia}} + \underbrace{f\sum_{j=1}^{N} w_{ij}(x_j - x_i)}_{\text{social influence}} + \underbrace{bv(t)}_{\text{rational judgment}} \tag{3.39}$$

where

$x_i(t)$	=	measures the propensity of worker i to work in the rural or urban sector. $x_i > 0$ implies that the worker intends to work in urban sector, whereas $x_i < 0$ implies a propensity to work in rural sector;
a	=	parametric constant;
$ax_i(t)$	=	reflects the inertia of worker i;
f	=	parametric constant;
w_{ij}	=	weight of the social link between worker$i$ and worker j;
$f\sum_{j=1}^{N} w_{ij}(x_j - x_i)$	=	represents the social influence from one's (i) companions.
b	=	parametric constant;
bv	=	stands for one's (i) rational judgment. The expected wage differential between urban and rural sectors can be regarded as the input variable v of system.

with

$$v(t) = (1 - \lambda)w_m - w_a \tag{3.40}$$

where

v is positive if rural wage is lower than expected wage in urban sector.

v is negative if rural wage is bigger than expected wage in urban sector.

The weight of social links between two workers w_{ij} can be expressed as a matrix or a graph. If we have a population of 100 workers distributed on a grid and that each social link is represented by a edge between two vertex, we get the graph represented in Figure 3.1.

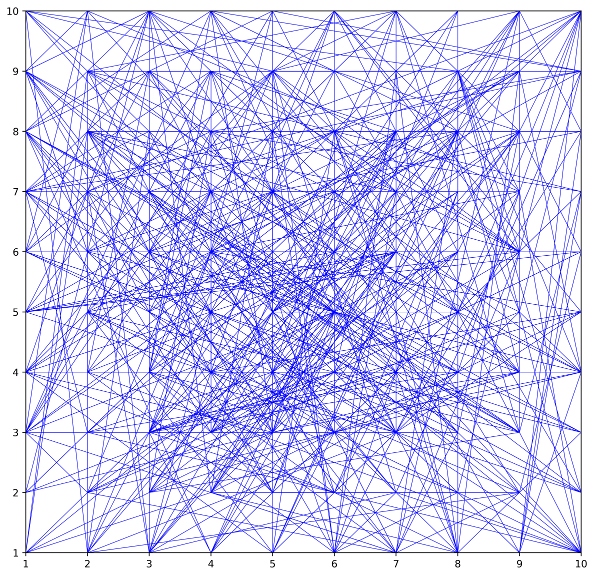


Figure 3.1: Social links.

Migration decision

Each period, every worker reviews their situation and makes a choice to either migrate or remain in their current sector. A worker may consider migrating if they have a strong propensity to work in the alternative sector. For a given worker, the likelihood of successful migration is denoted as P_i and is calculated as the absolute value of x_i divided by the sum of the absolute value x_i and β. $\beta \in \mathbb{R}^+$ is a parameter to modulate the sensitivity. Migration is more sensitive with a smaller β.

$$P_i = \frac{|x_i|}{|x_i| + \beta} \tag{3.41}$$

where

$P_i(t) =$ the probability of migration for worker$i$.

$x_i(t) =$ measures the propensity of worker i to work in the rural or urban sector.

β is a parameter.

Consensus analysis

A central theme in agent-based model is the "consensus problem" in system dynamics. Consensus refers to a state where agents converge on a common decision or behavior, in this case, the propensity to migrate. Cai [37] explores conditions under which consensus is reached or avoided within the population, using mathematical formulations tied to the stability and dynamics of the system.

In the case of the agent-based dynamic model, we don't want to reach a consensus where the whole population makes the same choices. This would result in a high degree of instability that would prevent equilibrium from being reached. Without going into the details of the demonstration, we can summarise the condition for the absence of consensus in the proposition below.

Proposition to avoid consensus [37]. The propensity dynamics can avoid consensus if and only if

$$a > f \operatorname{Re}(\lambda_2)$$

where λ_2 is the nonzero eigenvalue of Laplacian matrix of the weighted directed graph of social relations that has the minimal real part. The Laplacian matrix of the weighted directed graph can be computed as $L = D - W$ where D is the in-degree matrix and W is the adjacency matrix.

In order to avoid consensus, a should be sufficiently large, otherwise f or $\operatorname{Re}(\lambda_2)$ should be small enough. If $a \leq 0$, then consensus is inevitable. In other words, the private inclination should overwhelm the social influence.

Chapter 4

Developments and implementation of agent-based models

We use MESA library to simulate our agent-based model. MESA is an Apache2 licensed agent-based modeling (or ABM) framework in Python [51]. Documentation is available online [52, 47].

Mesa is a modular framework for building, analyzing and visualizing agent-based models. MESA allows to create specific objects as models and agents. ABMs represent complex systems by simulating the behavior of individual agents and their interactions within an environment. Mesa facilitates the implementation of such models by offering a structured framework where agents can be defined with specific behaviors, rules, and interactions. This library enables researchers and practitioners across disciplines to explore emergent phenomena, study system dynamics, and test hypotheses within simulated environments.

4.1 MESA implementation and algorithm

In this part, we will describe the Python implementation and the algorithm of the Harris-Todaro agent-based model [36] in order to run the model as we described in Subsection 3.2.1. The python code of this implementation is available in Annex A. Figure 4.1 explains the algorithm of the agent-based model with pseudo-code. A Python implementation of the model is available on github [53] and was used to write the code in Annex A. Modifications have been done in order to get the right implementation of Espindola's model [36].

The code operates in several steps. First, it initializes parameters, followed by the creation of the population in the `origin_list` and their migration choices in the `migra_list`. Subsequently, the Mesa model and its constituent agents are initialized. Once everything is set up, the model iterates autonomously the number of iterations requested, allowing it to reach equilibrium through its own processes.

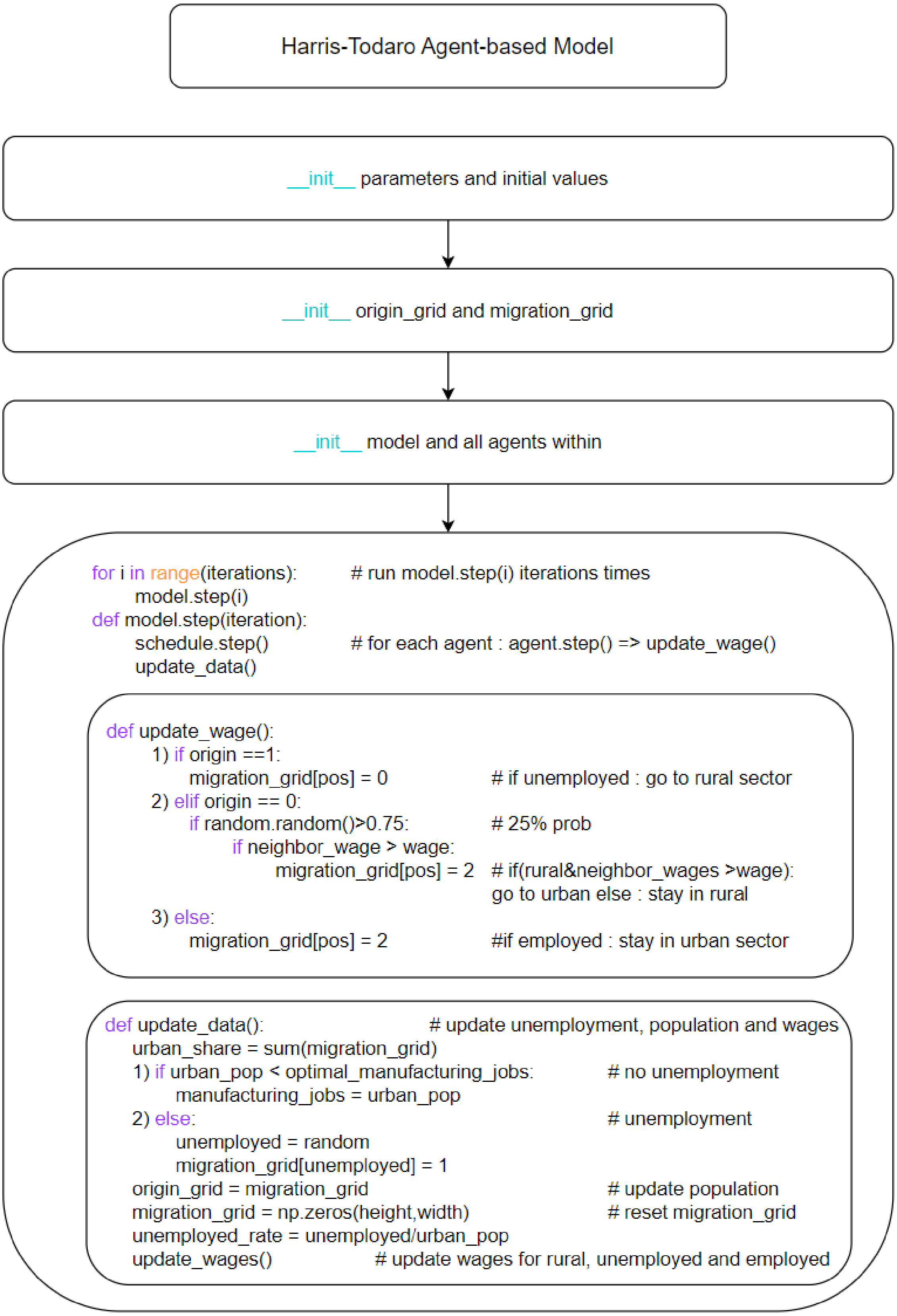


Figure 4.1: Harris-Todaro Agent-based Model Algorithm.

4.1.1 Initialization

Parameters and initial values

Figure 4.2 is the begining of the code. At the begining of the execution, Python initializes all parameters and initial values for our equations as explained in Section 3. The parameters values are represented in Table 4.1.

```
#**************************************************************************
#**************************************************************************
# MAIN

if __name__ == '__main__':


    #======================================================================
    # Initialize all parameters and initial conditions
    #
    #======================================================================
    A_a = 1      # Parametric constant in rural sector production function
    A_m = 1      # Parametric constant in urban sector production function
    phi = 0.3    # Parametric constant in rural sector production function
    alpha = 0.7  # Parametric constant in urban sector production function
    sigma = 1    # Parametric constant in trade
    rho = 1      # Parametric constant in trade
    W_m = 0.8    # Minimum wage in urban sector
    L_mu = 0.2       # labor input in urban sector (employed and unemployed)
    L_a = 1 - L_mu   # labor input in rural sector
    L_m = 0.2        # labor input in manufacturing jobs
    L_mu_share = [L_mu]            # urbanization (urban/total population share)
    unemploy_rate = [1-L_m/L_mu]   # urban unemployment rate
    Lm_opt = compute_Lm_opt()      # optimal manufacturing jobs

    width = 300          # size of the population = width x height
    height = 300
    iterations = 20      # number of iterations of the model
```

Figure 4.2: Initialization of parameters and initial values.

Parameter	Value
A_a	1
A_m	1
ϕ	0.3
α	0.7
σ	1
ρ	1
W_m	0.8
L_u	0.2
L_a	$1 - L_u = 0.8$
L_m	0.2
λ	$1 - \frac{L_m}{L_u} = 0$

Table 4.1: Parameters and initial values

Origin and migration matrices

Then, two matrices of 300 times 300 people (9000 people in total) are initialized, one with the situation of all the people at the start of the step and the other with the decision to migrate or to stay in urban or rural sector during the step. Figure 4.3 and 4.5 explain how the list are created. Figure 4.4 shows the plot of `origin_list` where every white pixel represents an agent working in rural sector and every blue pixel represents an agent in urban sector, dark blue representing manufacturing worker and light blue representing unemployed worker.

```
#==========================================================================
# Initialize origin and migration matrices
#
#==========================================================================
origin_list, migra_list = init_list(height,width) # Initialization of pop
plot_list(origin_list, 'origin')                  # plot initial population
```

Figure 4.3: Initialization of population.

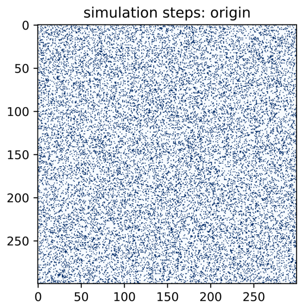


Figure 4.4: Plot of `origin_list` at initialization.

```
#===============================================================================
# Initialization of the population matrices
# origin list = random proportion of 2 for urban people
# migration list = only zeros
#
# 0: stay in the countryside
# 1: migrate to the city without wage
# 2: migrate to the city with wage
#===============================================================================
def init_list(width, height):
    matrix = np.zeros((height, width))                  # create the origin list
    migra_list = np.zeros((height, width))              # create the migra list
    total  = width*height                               # total population
    to_urban = int(total*L_mu)                          # total urban population
    index = random.sample(range(0, total), to_urban)    # create a list of random
    for i in range(to_urban):                           # implementing urban peopl
        row = int(index[i]/width)
        col = index[i]%width
        matrix[row][col] = 2
    return matrix, migra_list                     # return matrices
```

Figure 4.5: Function for initializing lists.

Model and Agents

Figure 4.6 creates the model while Figure 4.7 and 4.8 show the constructors for the class `MigrationModel` and the class `Harris_Todaro_Agent`. A class is a code template for creating objects. Objects have member variables and have behaviour associated with them.

```
#==========================================================================
# Initialize model and agents
#
#==========================================================================
model = MigrationModel(height*width, width, height) # Init of MESA MODEL
```

Figure 4.6: Initialization function for Mesa agent.

```
#==============================================================================
# MESA MODEL
#
#==============================================================================
"""A model with some number of agents."""
class MigrationModel(Model):

   #===========================================================================
   # MESA MODEL initialization
   #
   #===========================================================================
    def __init__(self, L, width, height):
        self.num_agents = L                              # Size of population
#        self.grid = MultiGrid(width, height, False)
        self.grid = SingleGrid(width, height, False)
        self.schedule = RandomActivation(self)
        # Scheduler will call AGENT in a random order
        self.running = True
        self._steps = 0
        self._time = 0

        for i in range(self.num_agents):      # Fullfill the grid by MESA AGENT
            a = Harris_Todaro_Agent(i, self) # Initialize MESA AGENT
            self.schedule.add(a)             # Add AGENT to schedule
            x = int(i/self.grid.width)
            y = i%self.grid.width
            self.grid.place_agent(a, (x, y)) # Add AGENT on the grid
```

Figure 4.7: Initialization function for Mesa model.

```
#==============================================================================
# MESA AGENT
#
# Agent = agent id (number from 0 to Pop-1)
#==============================================================================
class Harris_Todaro_Agent(Agent):

    #==========================================================================
    # MESA AGENT initialization
    #
    #==========================================================================
    def __init__(self, unique_id, model):
        super().__init__(unique_id, model)      # init agent
        x = int(unique_id/model.grid.width)
        y = unique_id%model.grid.width
        global origin_list                      # import origin list
        if origin_list[x, y] == 0:              # wage = rural agent
            self.wage = compute_Wa()
        else:                                   # wage = employed urban agent
            self.wage = compute_Wm()
```

Figure 4.8: Initialization function for Mesa agents.

4.1.2 Iterations

When the model is initialized, it just evolves based on the parameters and equations implemented. Figure 4.9 executes the function `step()` of the model `iterations` times. The function `step()` of the model iterates the model following the rules for each agent. Figure 4.10 shows the `step()` function for the class `MigrationModel`. In this function, there are two important functions called: `schedule.step()` that run the function `step()` for each agent in a random order and `update_data()` that update market prices.

```
#==============================================================================
# Iterate the model
#
#==============================================================================
for i in tqdm(range(iterations)):  # Running the MESA MODEL n-times
    model.step(i)   # running the MESA MODEL for 1 iteration
```

Figure 4.9: Execution of the model.

```
#==============================================================================
# Run a step/an iteration for the model
#
# i = iteration number
#==============================================================================
def step(self, i):
    # self.datacollector.collect(self)
    self.schedule.step()                    # step all agent
    self.update_data()                      # update data model
    global origin_list
    global migra_list
    # print(migra_list)
    plot_list(origin_list, str(i))
```

Figure 4.10: `Step()` function of the class `MigrationModel`.

Schedule step

Figure 4.11 shows the function `step()` that launch function `update_wage()`. This instruction make all agent do their step function. The behavior of a `Harris-Todaro_Agent` is the following :

- **Unemployed:** migrate to rural sector.
- **Agricultural worker:** migrates to urban sector if their neighborhood's wage is higher than them, stay at rural sector otherwise. There are two implementations, first it compares the number of neighbors having higher wages to the number of neighbors having lower wages, second it compares the mean wage of neighbors.
- **Manufacturing worker:** Stay at urban sector.

```python
#==============================================================================
# update agent wage
#
#==============================================================================
def update_wage(self):
    global origin_list
    global migra_list

    is_possible_mig = random.random()                        # give a random number between 0 and 1
    if origin_list[self.pos[0], self.pos[1]] == 1:           # if no wage = go to rural sector
        migra_list[self.pos[0], self.pos[1]] = 0

    elif origin_list[self.pos[0], self.pos[1]] == 0:         # if rural sector && rand>0.75 => 25% prob
        if is_possible_mig > 0.75:
            neighbors = self.model.grid.get_neighborhood(
            self.pos,
            moore=False,                                     # Moore == TRUE => 8 surrounding squares, if Moor == FALSE => Neumann - 4 surrounding squares
            include_center=False)                            # Excluding yourself

            total_wage = 0
            satisfaction = 0

            espindola=False                                  # Here we use either the paper's model or the github implementation
            if (espindola==True):
                for neighbor in neighbors:                       # Add all neighbors' wages
                    cellmates = self.model.grid.get_cell_list_contents(neighbor)
                    if (self.wage > cellmates[0].wage):          # If neighbor wage is lower than rural wage = satisfaction increase
                        satisfaction += 1
                    if (self.wage < cellmates[0].wage):          # If neighbor wage is higher than rural wage = satisfaction decrease
                        satisfaction -= 1
                if satisfaction<0:                               # If the satifaction is negative : migrate to the city
                    migra_list[self.pos[0], self.pos[1]] = 2
            else:
                for neighbor in neighbors:                       # Add all neighbors' wages
                    cellmates = self.model.grid.get_cell_list_contents(neighbor)
                    total_wage += cellmates[0].wage
                if total_wage/len(neighbors) > self.wage:    # If the mean neighbor wage is higher than rural wage = migrate to the city
                    migra_list[self.pos[0], self.pos[1]] = 2
    else:
        migra_list[self.pos[0], self.pos[1]] = 2             # If manufacturing wage = stay in urban sector

#==============================================================================
# Run a step/an iteration for the agent
#
#==============================================================================
def step(self):
    # self.move()
    self.update_wage()                                       # update agent wage
```

Figure 4.11: `Step()` function of the class `Harris_Todaro_Agent`.

The `Step()` function operates in the following way:

1. Firstly, it checks if the agent is unemployed. If the agent is unemployed, then they return to the countryside.
2. Secondly, if the agent is in the countryside, there is a 25% chance they will consider whether to migrate or not. If we are within the 25% chance, then the agent compares their wage in the countryside with the wages of their neighbors. Here, there are two implementations for the non-dynamic model, either by the average or by the number. According to the article by Espíndola et al. [36], the agent

calculates their satisfaction based on the number of their neighbors who have a higher wage than theirs. If they have more neighbors with a lower wage, they remain in the countryside; conversely, if their satisfaction is negative, that is, they have more neighbors who have a higher wage than them, they decide to migrate to the city. In the second case, the implementation found on GitHub [53], the agent averages the wages of their neighbors and decides to migrate if this average is higher than their wage, otherwise, they stay in the rural sector.

3. Thirdly, if the agent is employed in the urban sector, they remain there.

Update data

Figure 4.12 shows the function `update_data` that update data of the model.

> **If urban population is lower than manufacturing jobs - Full employment**
>
> **If urban population is higher than manufacturing jobs - Unemployment**
> The unemployment is randomly distributed among urban workers.

Update the wages of all agents to reflect their new status.

Let us explain the operation of the function `Update_data()`. This function intervenes after all agents have individually decided whether to migrate or not. We start by importing the global variables we need for execution, then we calculate the share of the urban population. Next, if the urban population is less than the *optimal* population (noting that the optimal population is reached when the wage is equal to the minimum wage but there is no unemployment, i.e., the limit before the appearance of unemployment), then there is no unemployment and everyone can be given a job in the manufacturing sector. If the urban population exceeds the *optimal* population, then we randomly select among them the workers/agents who will be unemployed for the next iteration of the program. Once that is done, we reset the migration list, add unemployment and wage data to our lists, and update the wages of the agents according to their new occupation.

```python
#==============================================================================
# MESA MODEL uptade
#
#==============================================================================
def update_data(self):
    global migra_list
    global origin_list
    global L_a
    global L_m
    global L_mu
    global Lm_opt
    global L_mu_share
    global unemploy_rate
    #==============================================================================
    # Compute urban share
    # Append the list : L_mu
    L_mu = np.sum(migra_list)/(2*self.grid.width*self.grid.height)          # L_mu = urban population / total population
    L_mu_share.append(L_mu)                                                 # Append the list : L_mu_share

    #==============================================================================
    # If there is less urban population than manufacturing jobs (NO UNEMPLOYMENT)
    if L_mu <= Lm_opt:                                                      # If urban population is lower than optimal urban population
        origin_list = migra_list                                            # Set population list to migration list (step migration)
        L_a = 1-L_mu                                                        # Update rural population
        L_m = L_mu                                                          # All urban population is employed in manufactures
        migra_list = np.zeros((self.grid.height, self.grid.width))          # Reset migration list
    #==============================================================================
    # If there is more urban population than manufacturing jobs (UNEMPLOYMENT)
    else:
        L_a = 1-L_mu                                                        # Update rural population
        L_m = Lm_opt                                                        # Manufacturing workers < urban population

        total_migr = np.sum(migra_list)/2                                           # Count all people migrating or staying to urban sector
        unemployed = int(total_migr - self.grid.width*self.grid.height*Lm_opt)      # Count all people unemployed (total urban pop - total jobs)
        index = random.sample(range(0, int(total_migr)), unemployed)                # Finding random urban population to be unemployed
        count = 0
        for i in range(self.grid.height):                                   # Setting unemployed urban population
            for j in range(self.grid.width):
                if migra_list[i][j] != 0:
                    if count in index:
                        migra_list[i][j] = 1
                    count += 1
        origin_list = migra_list                                            # Set population list to migration list (step migration)
        migra_list = np.zeros((self.grid.height, self.grid.width))          # Reset migration list


    #==============================================================================
    # Append the list : unemploy_rate
    unemploy_rate.append(1-L_m/L_mu)
    W_a_ev.append(compute_Wa())
    W_m_ev.append(compute_Wm())
    #==============================================================================
    # Update wages
    for i in range(self.grid.height):                                       # Change all agents
        for j in range(self.grid.width):
            pos = (i, j)
            cellmates = self.grid.get_cell_list_contents(pos)
            if origin_list[i][j] == 2:
                cellmates[0].wage = compute_Wm()                            # Manufacturing wages
            elif origin_list[i][j] == 1:
                cellmates[0].wage = 0                                       # No wages for unemployed
            else:
                cellmates[0].wage = compute_Wa()                            # Agricultural wages
```

Figure 4.12: `Update_data()` function of the class `MigrationModel`.

4.2 Modeling migration dynamics

The problem with the model presented above is that there is no dynamic behavior of agent. The agent chose to migrate or not in function of their expected wage in urban sector. The result of this model need several iterations in order to reach the equilibrium but that is not due to implemented dynamic behavior but it follows the mesa framework that use the neighbors of each agent and the probability of 25% implemented arbitrarily. This section takles to the challenge to implement dynamic behavior as presented in Section 3.4 in order to study it.

To implement dynamics, the only need is to change the behavior of the agents. Therefore the code will mostly remain the same than in Subsection 4.1. The changes in the code will be explained below.

Subsection 3.4.2 propose to implement dynamics in the agent-based model. Equation 4.1 define the propensity to migrate for an agent. The solution proposed in this master thesis adds a subjective judgement in order to model the propensity to return to the rural sector if the agent is unemployed. This subjective judgment has no influence if the worker is an agricultural worker, low influence if the worker is manufacturing worker and higher influence if the worker is unemployed.

$$\dot{x}_i(t) = \underbrace{ax_i(t)}_{\text{inertia}} + \underbrace{f\sum_{j=1}^{N} w_{ij}(x_j - x_i)}_{\text{social influence}} + \underbrace{bv(t)}_{\text{rational judgment}} + \underbrace{cs_i(t)}_{\text{subjective judgment}} \tag{4.1}$$

where

$x_i(t)$	= measures the propensity of worker i to work in the rural or urban sector. $x_i > 0$ implies that the worker intends to work in urban sector, whereas $x_i < 0$ implies a propensity to work in rural sector;
a	= parametric constant;
$ax_i(t)$	= reflects the inertia of worker i;
f	= parametric constant;
w_{ij}	= weight of the social link between worker$i$ and worker j;
$f\sum_{j=1}^{N} w_{ij}(x_j - x_i)$	= represents the social influence from one's (i) companions.
b	= parametric constant;
bv	= stands for one's (i) rational judgment. The expected wage differential between urban and rural sectors can be regarded as the input variable v of system;
c	= parametric constant;
cs_i	= stands for one's (i) subjective judgment. The differential between the agent wage and rural wage.

with

$$v(t) = (1 - \lambda)w_m - w_a \tag{4.2}$$

where

v is positive if rural wage is lower than expected wage in urban sector.
v is negative if rural wage is bigger than expected wage in urban sector.

and with

$$s_i(t) = w_i(t) - w_a \tag{4.3}$$

where

$s_i(t)$ is positive if rural wage is lower than agent wage.
$s_i(t)$ is negative if rural wage is bigger than agent wage.

Migration choice

As in Cai et al. [37], the migration choice is stochastic. The probability to migrate is defined in Equation 4.4.

$$P_i = \frac{|x_i|}{|x_i| + \beta} \tag{4.4}$$

where

$P_i(t) =$ the probability of migration for worker$i$.
$x_i(t) =$ measures the propensity of worker i to work in the rural or urban sector.
β is a parameter.

4.2.1 Dynamic implementation

Table 4.2 gives the parameters for two different implementation of the dynamic model. Subtable 4.2a is a no consensus case where the model will be stable. Subtable 4.2b is a consensus case where the agent will reach consensus and therefore stability is not reached.

Parameter	Value
a	0.0008
f	0.001
b	0.02
c	0.01
β	3
`social_link`	0.1
`sparse_factor`	0.9

(a) Parameters that induce no consensus

Parameter	Value
a	0.002
f	0.004
b	0.02
c	0.01
β	2
`social_link`	0.1
`sparse_factor`	0.8

(b) Parameters that induce consensus

Table 4.2: Parameters values for both no-consensus and consensus cases.

Figure 4.13 shows the implementation of the `step()` function of the class `Harris_Todaro_Agent`. This function implement Equation 4.1 and Equation 4.4.

It calculates labour market wages in the rural and urban sectors. It calculates inertia, then social influence, then rational judgement and finally subjective judgement.

Secondly, the code calculates each agent's probability of migrating. If the propensity is negative, the agent migrates to the rural sector, otherwise to the urban sector. Otherwise, the agent does not migrate.

```
#===============================================================================
# update agent wage
#
#===============================================================================
def update_wage(self):
    global origin_list
    global migra_list
    global x_matrix
    global dotx_matrix
    global weights_matrix_matrix
    global width
    global height
    global x_ev

    Wm = compute_Wm()
    Wa = compute_Wa()


    x_i = x_matrix[self.pos[0]][self.pos[1]]
    social_influence = 0
    for i in range(height):
        for j in range(width):
            social_influence += weights_matrix_matrix[self.pos[0]][self.pos[1]][i][j] ...
                * (x_matrix[i,j]-x_matrix[self.pos[0]][self.pos[1]])
    v = ((L_m/L_mu) * Wm - Wa)
    s_i = (self.wage - Wa)

    dotx_matrix[self.pos[0]][self.pos[1]] = a * x_i + f * social_influence + b * v + c * s_i

    x_i_t1 = x_matrix[self.pos[0]][self.pos[1]]+dotx_matrix[self.pos[0]][self.pos[1]]
    prob = np.abs(x_i_t1)/(np.abs(x_i_t1)+beta)

    if random.random() < prob:
        if x_matrix[self.pos[0]][self.pos[1]]+dotx_matrix[self.pos[0]][self.pos[1]] > 0:
            migra_list[self.pos[0]][self.pos[1]]=2
        else:
            migra_list[self.pos[0]][self.pos[1]]=0
    else:
        if origin_list[self.pos[0]][self.pos[1]]==2:
            migra_list[self.pos[0]][self.pos[1]]=2
        if origin_list[self.pos[0]][self.pos[1]]==1:
            migra_list[self.pos[0]][self.pos[1]]=2

#===============================================================================
# Run a step/an iteration for the agent
#
#===============================================================================
def step(self):
    # self.move()
    self.update_wage()                                  # update agent wage
```

Figure 4.13: `Step()` function of the class `Harris_Todaro_Agent`.

`Step()` implementation

Let's discuss the implementation of the `Step()` function in the context of the dynamic modeling of this master's thesis. As seen in Equation 4.1, we have a differential equation composed of different parts that symbolize various causes influencing an agent's migration.

Firstly, we retrieve the value of x_i from the `x_matrix`, which is the propensity of agent i at time t to migrate. The `dotx_matrix` represents all the values for the different $\dot{x}_i$. The value of $\dot{x}_i$ is composed of:

- `a` $\cdot$ `x_i` is the inertia of the agent.

- `f` $\cdot$ `social_influence` is the social influence of other agents.

- `b` · `v` is the rational judgment based on objective conditions.

- `c` · `s_i` in the subjective judgment based on individual situation.

Next, we calculate the probability of migrating according to Equation 4.4. If this probability is met (we compare it with a random number following a uniform distribution between 0 and 1):

- If the propensity to migrate is positive and the agent is in the rural sector, then they migrate to the urban sector.

- If the propensity to migrate is negative and the agent is in the urban sector, then they migrate to the rural sector.

If the probability is not met, we leave the agent in their initial sector (either urban or rural). Note that `migra_list` represents the same matrix as `origin_list` but at time $t + 1$.

4.3 Modeling dynamic of unemployment

In most of the models described in the Literature Review, unemployment is distributed randomly. Under our assumptions, all workers are similar, so we cannot find reasons to employ some rather than others. On the other hand, redistributing jobs at each iteration among the different workers (or agents) in urban area (employed and unemployed) in a random way seems a rather far-fetched hypothesis, where employee turnover would be at a maximum. To overcome this problem, the solution offered by this master's thesis is to propose a probability of finding a job to each unemployed worker. In this way, unemployed workers have a chance of remaining unemployed and a chance of leaving unemployment, and we avoid redistributing unemployment randomly at each iteration.

Figure 4.14 explains the implementation of the unemployment. First, it compute the number of unemployed workers. After that, it lists all the unemployed workers remaining in the urban sector and gives them a probability of finding work described in Table 4.3. Then we look to see if there is more or less unemployment. If there is more unemployment than at previous step, we choose random agents in the urban population; conversely, if there is less unemployment than at previous stap, we choose random agents in the unemployed agents of previous step and give them a job.

Parameter	Value
`find_job`	0.2

Table 4.3: Probability to find a job as unemployed worker in urban sector.

4.3.1 Initial situation

Modeling the Economy

The economy is modeled using specific production functions that relate directly to the wages. The development is made in Subsection 3.2.1. Figure 4.15 shows the Python

```python
#==============================================================================
        # If there is more urban population than manufacturing jobs (UNEMPLOYMENT)
        else:
            L_a = 1-L_mu                                                            # Update rural population
            L_m = Lm_opt                                                            # Manufacturing workers < urban population

            # Count all people migrating or staying to urban sector
            total_migr = np.sum(migra_list)/2
            # Count all people unemployed (total urban pop - total jobs)
            unemployed = round(total_migr - self.grid.width*self.grid.height*Lm_opt)

            count = 0
            rural_now = []                  # all worker in rural sector
            unemployed_still = []           # all unemployed workers that remains in the urban sector
            for i in range(height):
                for j in range(width):
                    if migra_list[i][j] == 0:
                        rural_now.append(count)
                    if origin_list[i][j] == 1 and migra_list[i][j] != 0:
                        if random.random()> find_job:                       # prob to get a job for unemployed worker still in urban sector
                            unemployed_still.append(count)                  # remains unemployed
                    count += 1
            total_new_unemployed = unemployed - len(unemployed_still)       # number of new worker unemployed
            potential_new_unemployed = set(range(width*height))             # finding potential unemployed
            potential_new_unemployed = potential_new_unemployed - set(rural_now) - set(unemployed_still)

            if total_new_unemployed>=0:     # if their is more unemployed than previous step
            # Finding random urban population to be unemployed
                index = sorted(random.sample(list(potential_new_unemployed), total_new_unemployed))
                count = 0
                for i in range(height):
                    for j in range(width):
                        if migra_list[i][j] != 0:
                            if count in index or count in unemployed_still:
                                migra_list[i][j] = 1
                        count += 1
                                                # Set population list to migration list (step migration)
            else :                          # if their is less unemployed than previous step
            # Finding random unemployed to remain unemployed
                index = sorted(random.sample(list(unemployed_still), unemployed))
                count = 0
                for i in range(height):
                    for j in range(width):
                        if migra_list[i][j] != 0:
                            if count in index:
                                migra_list[i][j] = 1
                        count += 1

            origin_list = migra_list                                                # Set population list to migration list (step migration)
        x_matrix = dotx_matrix + x_matrix
```

Figure 4.14: `Update_data()` function of the class `MigrationModel`.

function that computes the wages for both sectors. The wage function for each sector can be derived as follows:

$$\text{Manufacturing:} \quad \alpha \cdot A_m \cdot L_m^{\alpha-1} \tag{4.5}$$

$$\text{Agriculture:} \quad \phi \cdot A_a \cdot P \cdot L_a^{\phi-1} \tag{4.6}$$

4.3.2 Social network

Each agent in the model has a unidirectional social network. This design implies that while one agent can be influenced by others, the reverse does not necessarily occur. Initially, connections between agents are established randomly, with each agent influencing the others with a weight distributed according to a uniform probability. After this initial configuration, only the top 10% of these connections, according to their strength, are retained, so that each agent has links with around 10% of the total population. This model makes it possible to study the impact of significant but limited social influence on economic behaviour and migration patterns. This initial randomization reflects a scenario in which social influences are not always reciprocal or equally distributed, but it can be modified if necessary. The python code is described in Figure 4.16.

```
#==============================================================================
# Compute wage in manufacturing jobs
#
#==============================================================================
def compute_Wm():
    return alpha * A_m * math.pow(L_m, alpha-1)


#==============================================================================
# Compute wage in agricultural jobs
#
#==============================================================================
def compute_Wa():
    Y_a = A_a * math.pow(L_a, phi)
    Y_m = A_m * math.pow(L_m, alpha)
    P = rho * math.pow(Y_m/Y_a, sigma)
    return phi * A_a * P * math.pow(L_a, phi-1)
```

Figure 4.15: Wage calculation functions.

```
weights_matrix_matrix = [[np.random.rand(height, width) * social_link for _ in range(width)] for _ in range(height)]
for rows in weights_matrix_matrix:
    for inner_matrix in rows:
        for i in range(height):
            for j in range(width):
                if inner_matrix[i, j] < sparse_factor:
                    inner_matrix[i, j] = 0
```

Figure 4.16: Initialising the social network.

4.3.3 Propensity to migrate

The initial propensity to migrate is a crucial parameter set at the start of the simulation. Each agent is assigned a propensity to migrate drawn at random from a uniform distribution between -1 and 1. This range represents a spectrum from a propensity to migrate to the rural sector (-1) to a propensity to migrate to the urban sector (1). The uniform initial distribution ensures that agents are as likely to start with a propensity to migrate to the rural sector as with a propensity to migrate to the urban sector. This initial distribution has an important impact on the dynamics during the first stages of the simulation. The initial dynamics tend to have as many rural workers as urban workers in the population. However, as the simulation progresses, this initial propensity is influenced by economic conditions, such as wage disparities and job opportunities. Over time, these economic factors become the dominant force dictating migration patterns, gradually eclipsing the initial random propensity parameters.

Chapter 5

Results and discussion

In this chapter, we present the results of the various simulations. After reproducing the non-dynamic agent-based model of Espíndola et al. [36], we will present the results of the model of Cai et al. [37]. Finally we will look at the results obtained by our dynamic model.

5.1 Espíndola et al. model results

The first model is the one presented in Espíndola et al. [36] and it does not implement dynamics but it can already bring us some responses about the equilibrium of the Harris-Todaro agent-based model.

Figure 5.1 shows the eveolution of `origin_list` through iterations. Where `origin_list` represent all agent of our model in a square grid. They are represented in white if they are in the rural sector and in blue if they are in the urban sector. There are two shades of blue, dark blue and light blue. Dark blue squares represent a employed worker in manufactural firm and light blue squares represent an unemployed worker in urban sector.

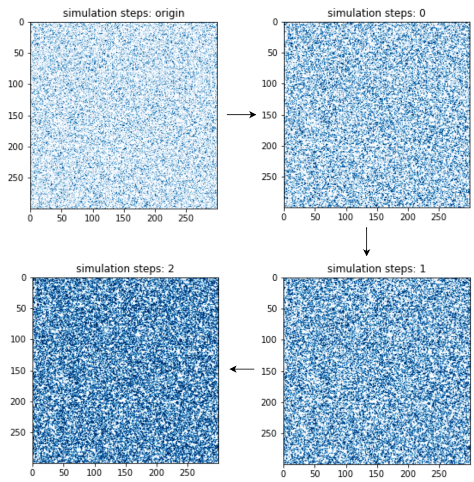
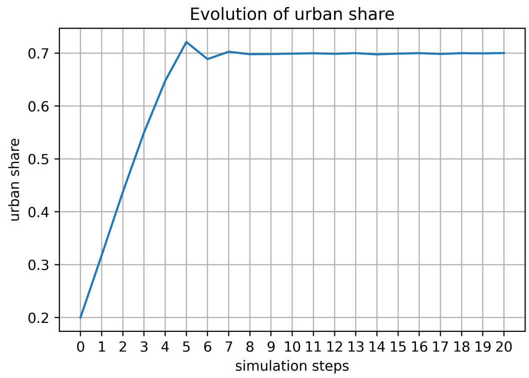


Figure 5.1: Evolution of `origin_list` through iterations. Model by Espíndola et al.

Figure 5.2 represents the result of the agent-based model. The itinal situation of the simulation is 20% of urban share and full employment. In Figure 5.2a, the urban population share increases from 20% to 70% and the unemployment rate increases from 0 to around 8%. Precise values are available in Table 5.1.



(a) Evolution of the urban population share. Model by Espíndola et al.

(b) Evolution of the unemployment rate.

Figure 5.2: Result of the Harris-Todaro agent-based model by Espíndola et al.

Iteration	urban share (%)
1	20.0
2	31.8
3	43.7
4	55.0
5	64.8
6	72.1
7	68.9
8	70.3
9	69.8
10	69.8
11	69.9
12	70.0
13	69.9
14	70.0
15	69.8
16	69.9
17	70.0
18	69.9
19	70.0
20	70.0

Iteration	unemployment rate (%)
1	0.0
2	0.0
3	0.0
4	0.0
5	1.05
6	11.1
7	6.97
8	8.81
9	8.21
10	8.25
11	8.35
12	8.44
13	8.31
14	8.48
15	8.16
16	8.36
17	8.47
18	8.27
19	8.46
20	8.41

Table 5.1: L_mu_share Values and unemployment rate. Model by Espíndola et al.

Figure 5.3 represents the ages in agricultural wage, manufacturing wage and expected urban wage. At the begining, the expected urban wage is equal to the manufacturing wage since their is no unemployment in the urban sector. Since the unemployment appears, we see that the expected urban wage drop below the minimum manufacturing wage in the urban sector. The equilibrium reach the equality between the expected urban wage and agricultural wage. At this situation, the migration stops because agents in rural sector will not find higher expected wage in urban sector.

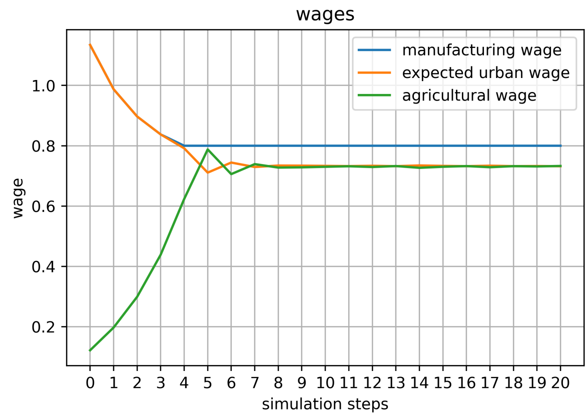


Figure 5.3: Evolution of wages in the Harris-Todaro agent-based model by Espíndola et al.

Iteration	Manufactural wage	Expected wage	Agricultural wage
1	1.134	1.134	0.122
2	0.987	0.987	0.197
3	0.897	0.897	0.299
4	0.838	0.838	0.439
5	0.800	0.792	0.623
6	0.800	0.711	0.788
7	0.800	0.744	0.706
8	0.800	0.729	0.739
9	0.800	0.734	0.728
10	0.800	0.734	0.728
11	0.800	0.733	0.730
12	0.800	0.732	0.732
13	0.800	0.734	0.729
14	0.800	0.732	0.733
15	0.800	0.735	0.727
16	0.800	0.733	0.730
17	0.800	0.732	0.733
18	0.800	0.734	0.729
19	0.800	0.733	0.732
20	0.800	0.732	0.732

Table 5.2: Evolution of wages in function of simulation steps. Model by Espíndola et al.

Here, the dynamics of the equilibrium is essentially due to the 25% probability assigned in the function `update_wage`. The study of the dynamics can therefore be found in Section 5.3.

5.2 Cai et al. model results

The second model is the one presented in Cai et al. [37]. This implementation is reproduced and can have some differences with the results presented in the original paper. This model implements dynamics for the Harris-Todaro agent-based model.

This model takes into account the inertia of agents, the social influence and the rational judgment as exposed in Equation 3.39.

As explained in Subsection 3.4.2, the model should not reach consensus in order to have stability. Therefore, we know that the proposition : $a > f\mathrm{Re}(\lambda_2)$ allows us to avoid consensus. Tables 4.2a and 4.2b gives two sets of parameters, the first without consensus, the second with consensus.

5.2.1 Without consensus

Figure 5.4 represents the total population at the last simulation step. Each square represents an agent and the colour tells us whether the agent is in the rural sector (purple), is unemployed in the urban sector (green) or employed in manufacturing job in the urban sector (yellow).

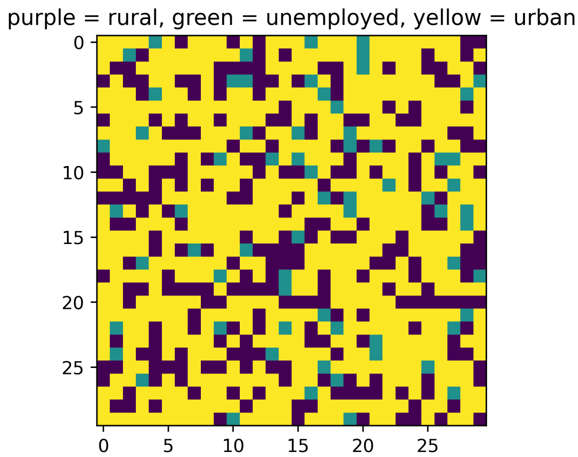
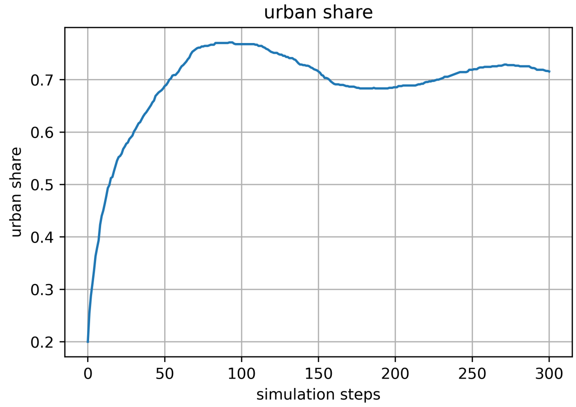
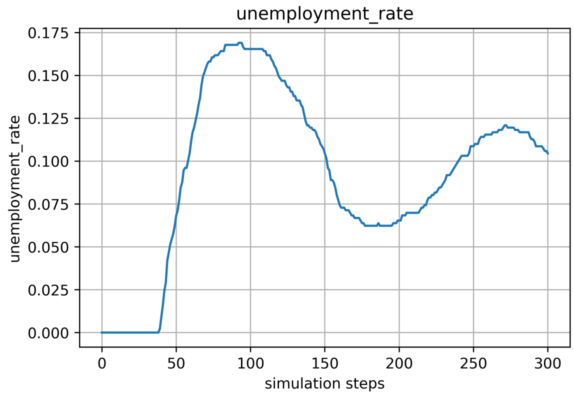


Figure 5.4: Representation of whole population at final simulation step for the case where there is no consensus. Model by Cai et al.

Figure 5.5a and Figure 5.5b represent the urban share and the unemployment rate in the population during the execution. The urban share stabilizes around 70% of the total population while the unemployment hovers around 10%.
Figure 5.6 represents the different wages in function of the simulation steps. Since dynamics have been implemented, we see that the agricultural wage is higher than the manufacturing wage. This is due to the inertia and social influence in the model. We see that the equilibrium tends to be reached and that expected urban wage and agricultural wage converge.



(a) Urban share in function of the simulation steps without consensus. Model by Cai et al.



(b) Unemployment rate in function of the simulation steps without consensus. Model by Cai et al.

Figure 5.5: Urban share and unemployment evolution without consensus. Model by Cai et al.

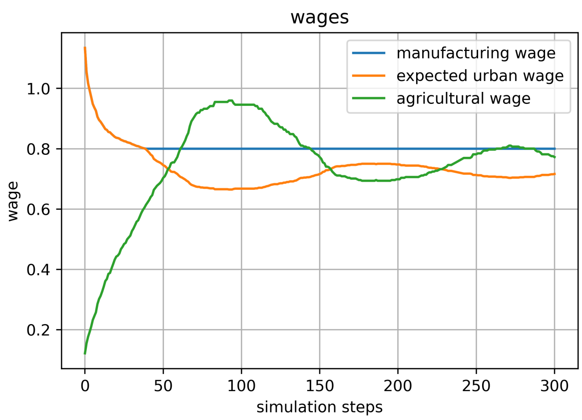


Figure 5.6: Wages in function of the simulation steps without consensus. Model by Cai et al.

5.2.2 With consensus

Figure 5.7 represents the total population at the last simulation step. Compared to Figure 5.4, we see that Figure 5.7 has more green squares and less purple ones. This implies that the urban share is higher and the unemployment rate too.

Figure 5.8a and Figure 5.8b represent the urban share and the unemployment rate in the population during the execution. Since, the proposition to avoid consensus is not fulfilled, we see that the system stability is much lower. We can see that there is stability as the system tends to converge towards a single urban share value. The stability is very low because the oscillation is high and is not rapidly attenuated.
Figure 5.9 represents the different wages in function of the simulation steps. In the long term, an equilibrium should be reached where the expected urban wage and agricultural wage are equal.

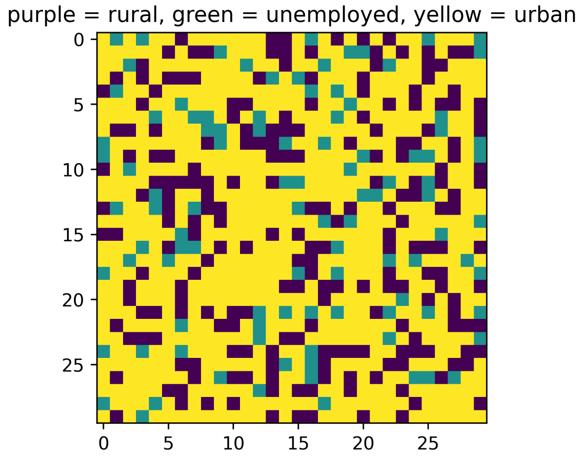
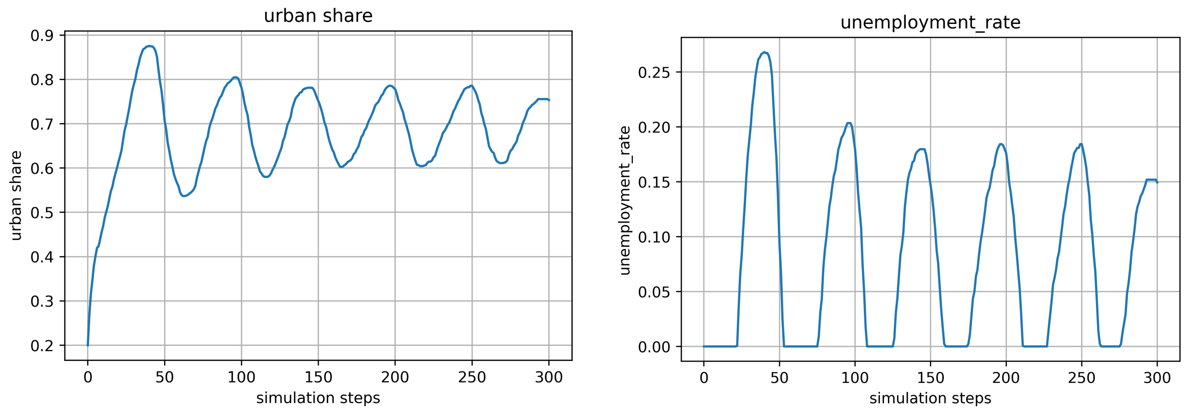


Figure 5.7: Representation of whole population at final simulation step for the case where there is consensus. Model by Cai et al.



(a) Urban share in function of the simulation steps with consensus.

(b) Unemployment rate in function of the simulation steps with consensus.

Figure 5.8: Urban share and unemployment evolution with consensus. Model by Cai et al.

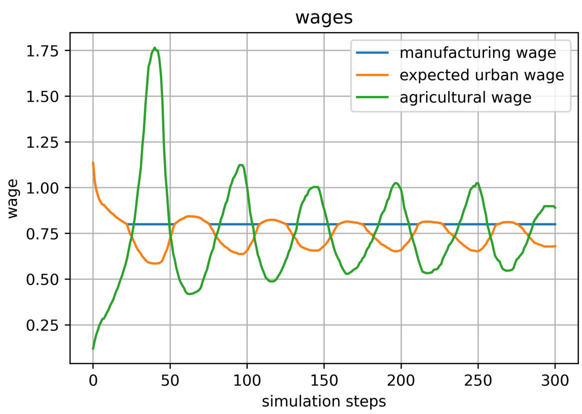


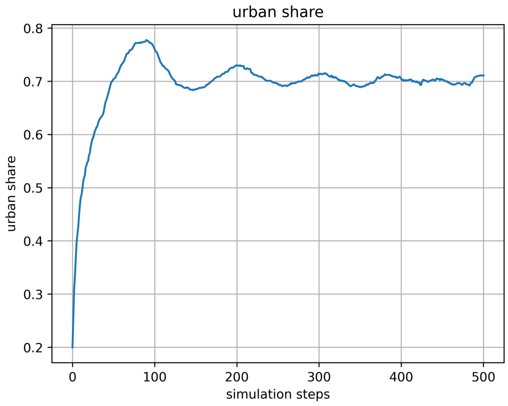
Figure 5.9: Wages in function of the simulation steps with consensus. Model by Cai et al.

5.3 Dynamic model results

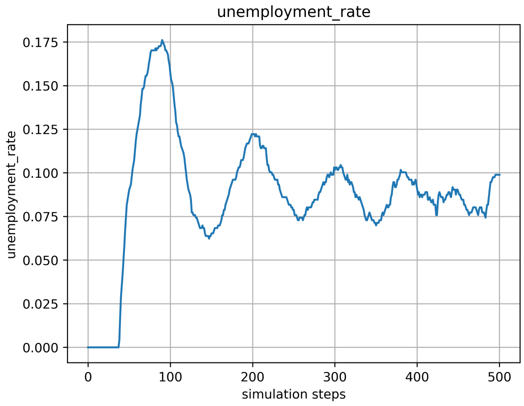
This is the most interesting part of the master thesis. The results presented in this section will serve as a basis for the discussion section.

5.3.1 Without consensus

Figure 5.10a and Figure 5.10b represent the urban share and the unemployment rate in the population during the execution of the dynamic model developed for this master thesis. To see the dynamic of the model during the successive simulation steps, there is a video available in Annex C (Adobe Acrobat is needed to see video in pdf) or available here: https://youtu.be/5s_QM4XytaI.



(a) Evolution of urban share.



(b) Evolution of unemployment rate.

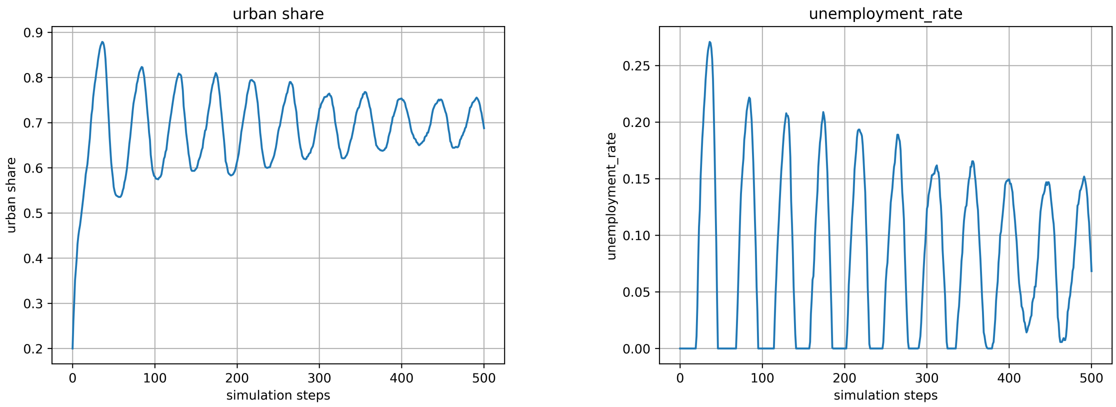
Figure 5.10: Evolution of urban share and unemployment rate. No consensus case.

Figure 5.11 represents the different wages in function of the simulation steps. We see that the model is stable and that it converges around 70% of urban share in Figure 5.10a, 8% of unemployment rate in Figure 5.10b and about 0.73 of expected urban wage or agricultural wage in Figure 5.11.

Figure 5.11: Evolution of wages. No consensus case.

5.3.2 With consensus

Figure 5.12a and Figure 5.12b represent the urban share and the unemployment rate in the population during the execution. To see the dynamic of the model during the successive simulation steps, there is a video available in Annex D or available here: https://youtu.be/OLU6xn6U2xg.



(a) Evolution of urban share. (b) Evolution of unemployment rate.

Figure 5.12: Evolution of urban share and unemployment rate. Consensus case.

Figure 5.13 represents the different wages in function of the simulation steps.

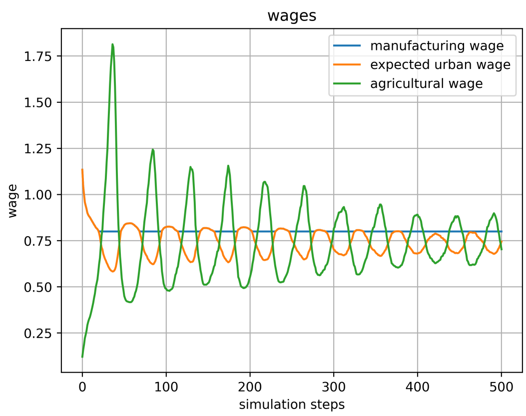


Figure 5.13: Evolution of wages. Consensus case.

5.3.3 Influence of parameters

In the following pages, we will discuss the influence of the different parameters on the dynamics of the model. It is important to realise that the model is stochastic and that, as a result, a given execution of the code will not give the same result as another, even with the same parameter. Consequently, the influence of the change in parameter is the only factor that changes between different executions of the code. This is why we sometimes find inconsistencies when we compare different simulations with different parameters. But we will try to identify the general trends

5.3.4 Influence of inertia

In this part, we can see that as inertia increases, oscillations slow down. We can also see a tendancy to have higher peaks in the dynamics. Figures 5.14, 5.15a and 5.15b represent the evolution of the wages for different values of parameter a that represents the influence of past propensity on current propensity and reinforces the agent's past propensity. Figure 5.16a and Figure 5.16b represent the evolution of urban share and unemployment rate.

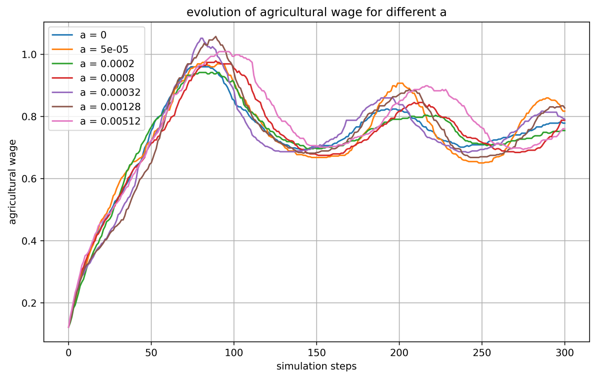
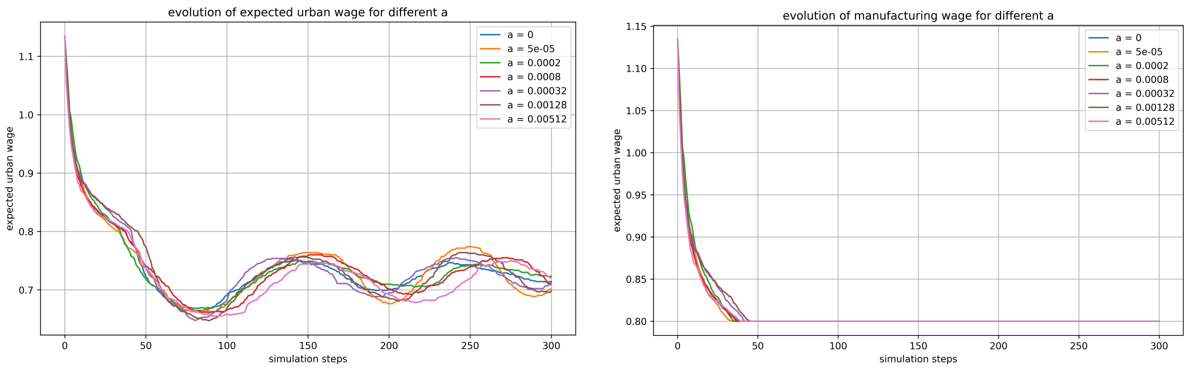
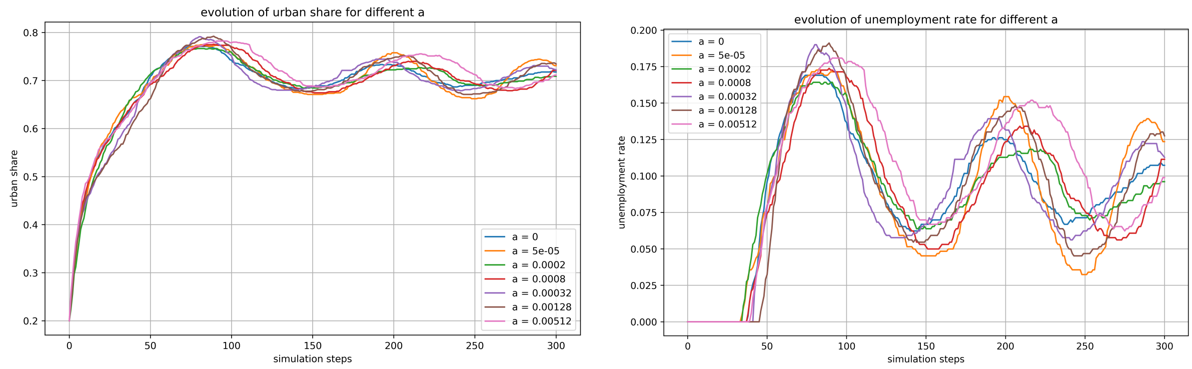


Figure 5.14: Evolution of agricultural wage for different value of parameter a.



(a) Evolution of expected urban wage. (b) Evolution of manufacturing wage.

Figure 5.15: Expected urban wage and manufacturing wage for different value of parameter a.



(a) Evolution of urban share.

(b) Evolution of unemployment.

Figure 5.16: Evolution of urban share and unemployment rate for different value of parameter a.

5.3.5 Influence of social network

By studying the influence of the social network on migration dynamics, we can see the transition from the state of non-consensus to the state of consensus, which revolves around the value $f = 0.00075$. In Figure 5.17 for values above $f = 0.00075$, we see a large oscillation which shows that the consensus is reached and that a large part of the population moves successively from migration to the city and then back to the countryside, and so on, according to Subsection 3.4.2. The higher the parameter of influence of the social network, the more the system's response oscillates. For the social

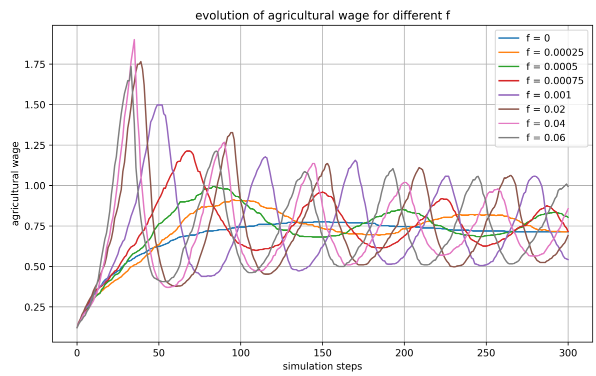
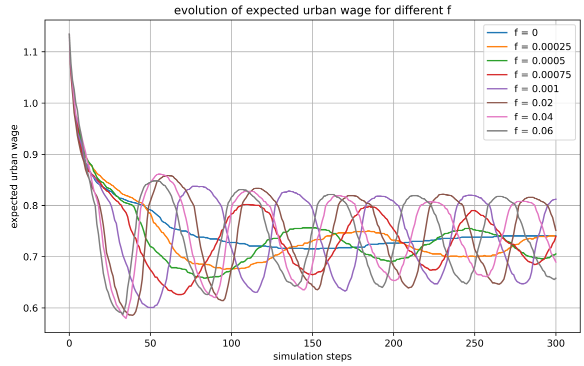


Figure 5.17: Evolution of agricultural wage for different value of parameter f.

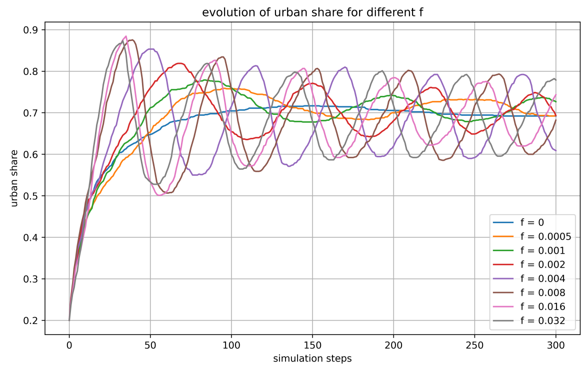
network, it is not the only way to model the dynamics. Each agent's social network can also be modified in order to



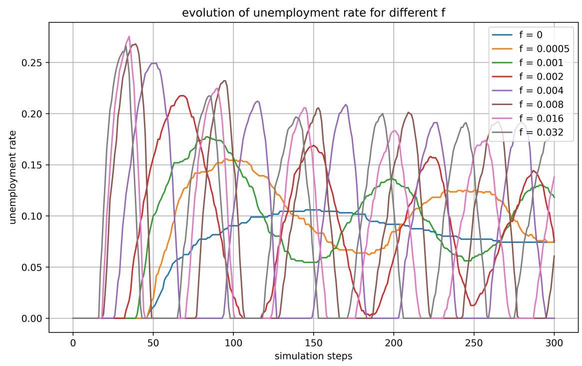
(a) Evolution of expected urban wage.

(b) Evolution of manufacturing wage.

Figure 5.18: Expected urban wage and manufacturing wage for different value of parameter f.



(a) Evolution of urban share.



(b) Evolution of unemployment rate.

Figure 5.19: Evolution of urban share and unemployment rate for different value of parameter f.

5.3.6 Influence of rational judgment

The influence of rational judgement increases the speed of response and reduces the risk of overshooting the equilibrium value. The higher the value of rational judgement, the faster and more fluidly equilibrium will be reached. When the parameter b of the rational judgment is equal to 0, the subjective judgment is the only aspect of the model that relies agent to the economic situation. When b is high, the agent will react fast to the objective condition of the labour market. If expected wage in the city is higher than in rural sector, there will be migration from rural to urban sector.

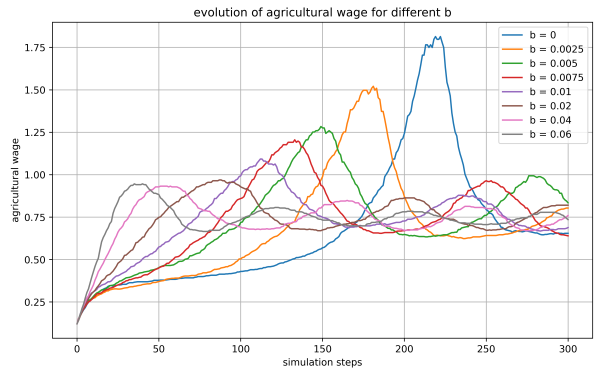
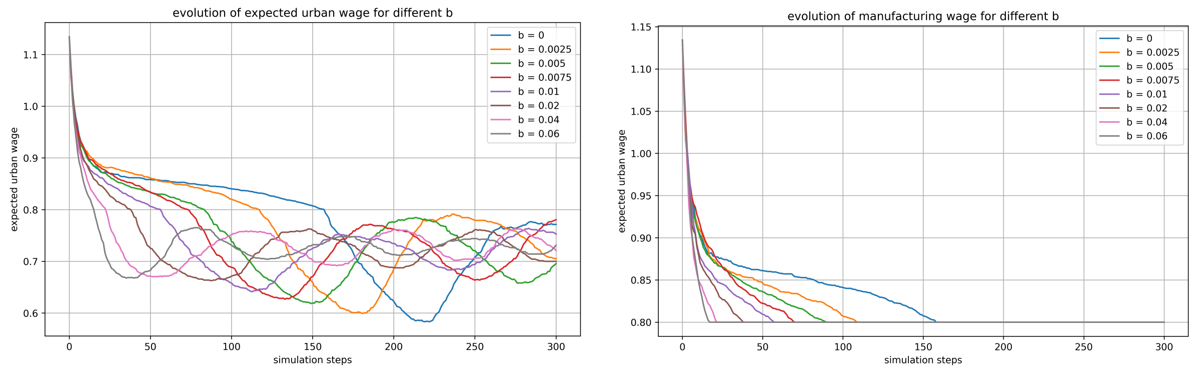


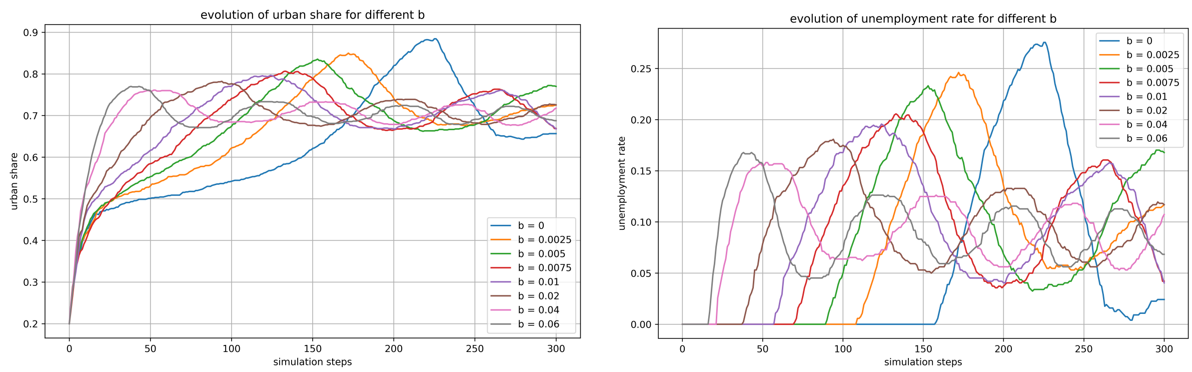
Figure 5.20: Evolution of agricultural wage for different value of parameter b.



(a) Evolution of expected urban wage.

(b) Evolution of manufacturing wage.

Figure 5.21: Expected urban wage and manufacturing wage for different value of parameter b.



(a) Evolution of urban share.

(b) Evolution of unemployement rate.

Figure 5.22: Evolution of urban share and unemployment rate for different value of parameter b.

5.3.7 Influence of subjective judment

The influence of subjective judgement is a major factor in achieving equilibrium quickly. The greater the subjective influence, the quicker unemployed agents will return to the rural sector if they don't find a job in the meantime. In figure 5.25, we see that the higher the subjective judgment, the faster the transition.

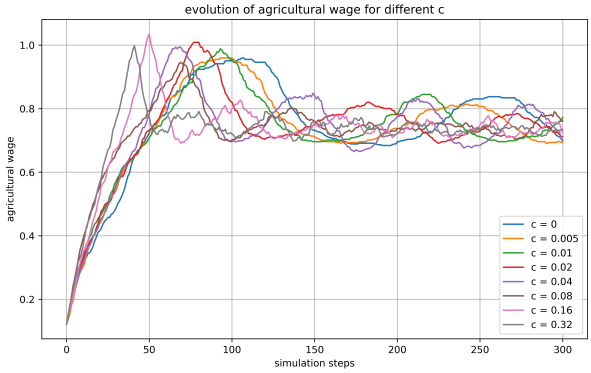
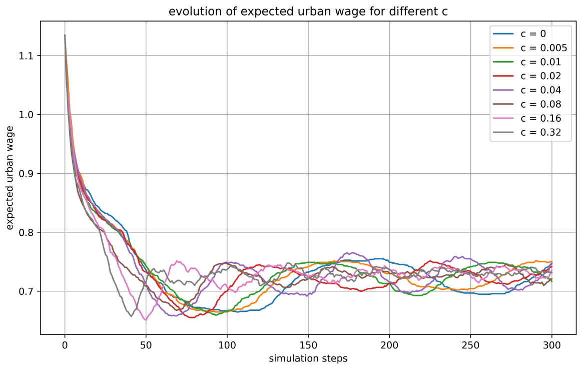
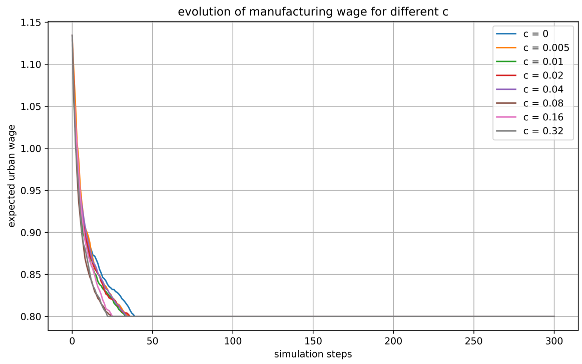


Figure 5.23: Evolution of agricultural wage for different value of parameter c.

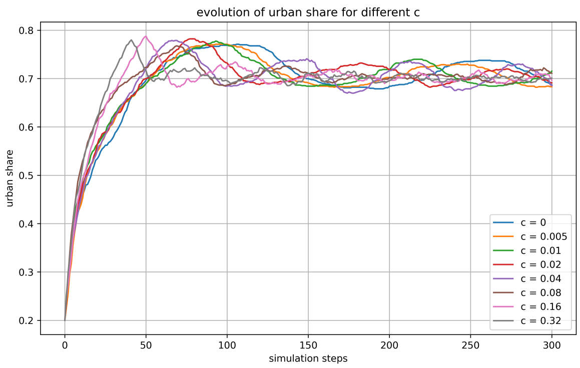


(a) Evolution of expected urban wage.

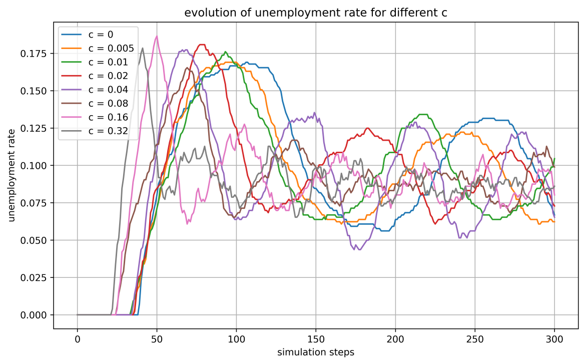


(b) Evolution of manufacturing wage.

Figure 5.24: Expected urban wage and manufacturing wage for different value of parameter c.



(a) Evolution of urban share.



(b) Evolution of unemployment rate.

Figure 5.25: Evolution of urban share and unemployment rate for different value of parameter c.

5.3.8 Probability and stochastic sensitivity

As expected, β has an influence on the sensitivity of migration. For $\beta = 0$, agents migrate for sure if their propensity to migrate indicates the other sector. Figure 5.28 shows that for $\beta = 0$, there is faster migration and also less oscillation because it stabilizes faster. For large values of β, we can see that a positive propensity to migrate to the city does not specifically materialise in a migration path. There is a probability of migrating, which reduces the number of agents actually migrating to the city.

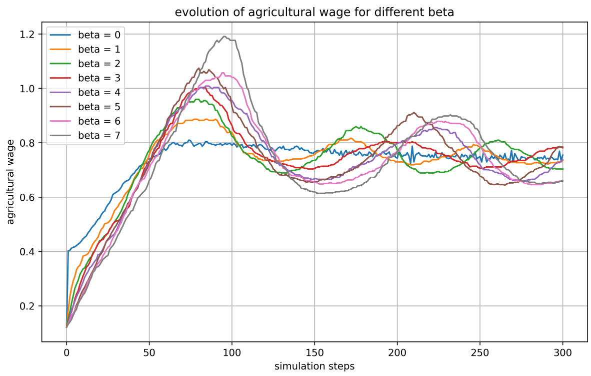
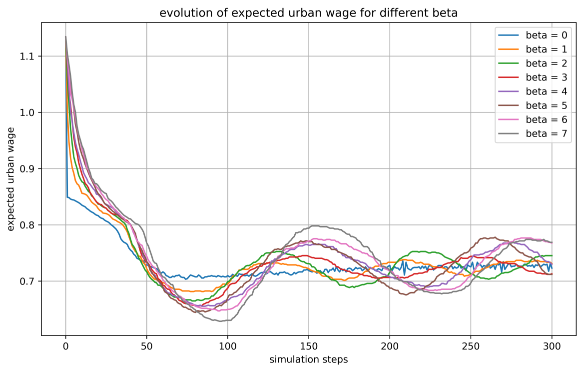
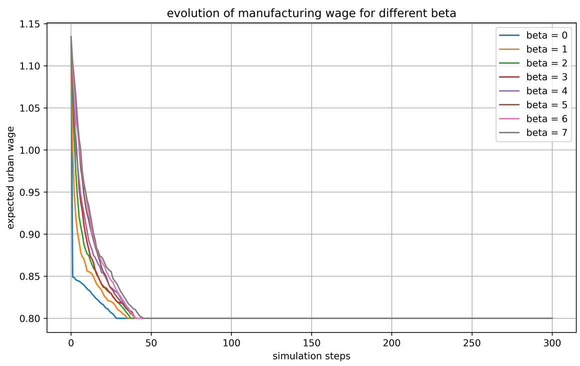


Figure 5.26: Evolution of agricultural wage for different value of parameter β.

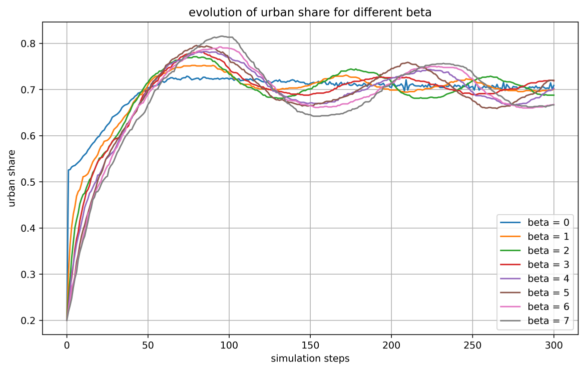


(a) Evolution of expected urban wage.

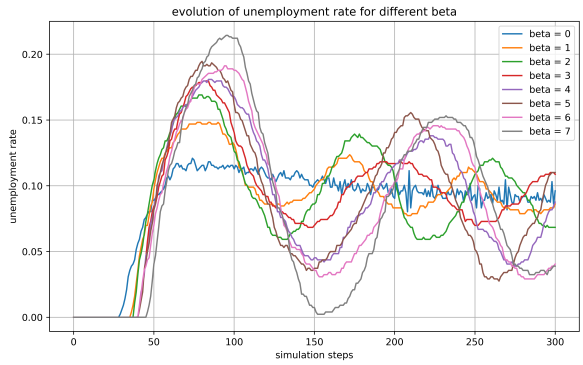


(b) Evolution of manufacturing wage.

Figure 5.27: Expected urban wage and manufacturing wage for different value of parameter β.



(a) Evolution of urban share.



(b) Evolution of unemployment rate.

Figure 5.28: Evolution of urban share and unemployment rate for different value of parameter β.

5.3.9 Influence of probability in Employment Decisions and Subjective Judgment

Subjective judgment and the probability of unemployed agents being hired in the next iteration can impact the overall dynamics of the agent-based Harris-Todaro model. However, it primarily adds an individual dimension that enriches each agent's personal journey with a real narrative.

Indeed, the `find_job` function directly models the chances for a migrant who has arrived in the city to stay there in the long term or not. If `find_job` is very low, an unemployed person has no income, and increased subjective judgment will likely increase their propensity to return to the countryside. After a few iterations spent unemployed, the agent will return to the countryside. Conversely, if the probability of finding a job (`find_job`) is high, then the unemployed are more likely to stay in the city as they will find work relatively quickly.

By closely watching the video at https://youtu.be/5s_QM4XytaI, if you observe the unemployed agents, you can see their individual stories unfold. Some will find employment fairly quickly, while others, after several iterations spent unemployed, will give up and return to the countryside to work and earn a salary. This aspect of the dynamics is not solely a global or "macro" dynamic but also allows for modeling individual pathways to better reflect the external world.

5.3.10 Influence of minimum fixed wage

The minimum wage significantly influences the unemployment rate in urban areas. If the wage gap between the manufacturing sector and the agricultural sector is substantial, it diminishes as migration progresses. However, a high minimum wage can induce unemployment in the urban population. In Figure 5.31b and Figure 5.30b we see that the level of this minimum wage directly influences the unemployment rate as well as wage levels in the manufacturing sector. Interestingly, it does not seem to affect the proportion of the urban population within society as shown in Figure 5.31a.

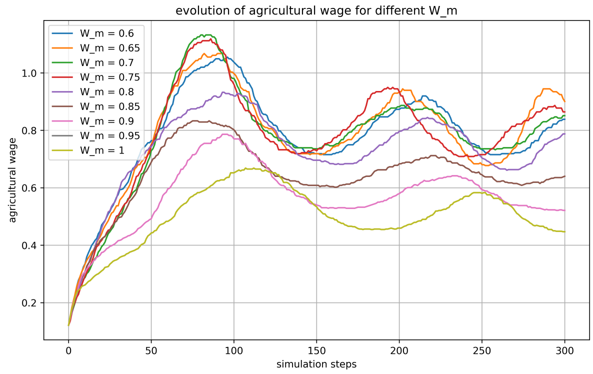
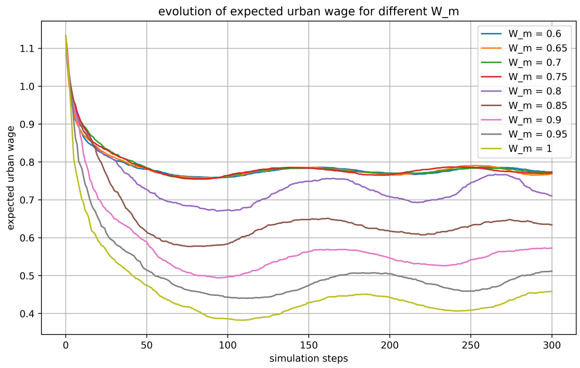
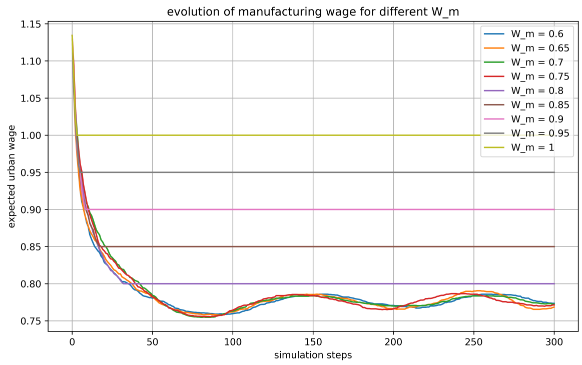


Figure 5.29: Evolution of agricultural wage for different value of minimum fixed wage $\bar{w}_m$.

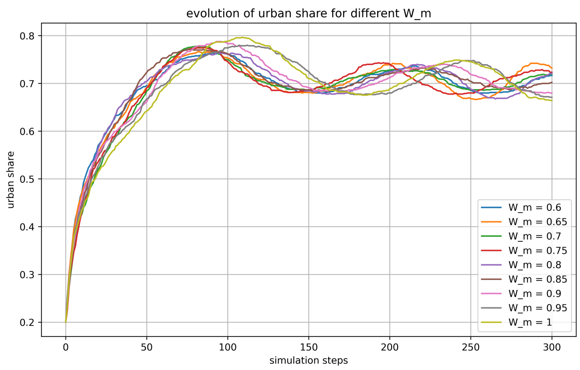


(a) Evolution of expected urban wage.

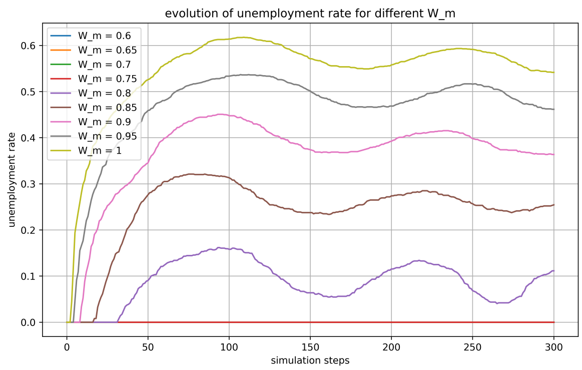


(b) Evolution of manufacturing wage.

Figure 5.30: Expected urban wage and manufacturing wage for different value of minimum fixed wage $\bar{w}_m$.



(a) Evolution of urban share.



(b) Evolution of unemployment rate.

Figure 5.31: Evolution of urban share and unemployment rate for different value of minimum fixed wage $\bar{w}_m$.

5.3.11 Influence of initial urban population share and initial propensity matrix

The initial share of the urban population, denoted by `L_mu`, plays a minimal role in the equilibrium and migration dynamics of the agent-based Harris-Todaro model. Figure 5.32 clearly shows how the trajectories are quite similar. The only significative differences are observable on first iterations. However, it is important not to overlook that the initial conditions regarding people's propensity to migrate have a significant influence on the dynamics of the early iterations. Dans le code, la `x_matrix` est initialisée comme étant une matrice représentant la propension de tous les agents individuels. Les conditions initiales utilisées dans ce mémoire sont une distribution uniforme entre $[-1; 1]$. Cette distribution initiale tend à expliquer une répartition de la population à 50% dans le secteur rural et 50% dans le secteur urbain ce qui a une grande influence sur la dynamique au début, Par exemple dans la Figure 5.28a, on voit que quand $\beta = 0$, la part de la population urbaine monte très rapidement vers 50% (et un peu plus avec l'effet social entrainé). Dans la Figure 5.33, on voit que la distribution initiale de `x_matrix` influence la dynamique de la transition urbaine. Table 5.3 avoir la distribution initiale des x_i.

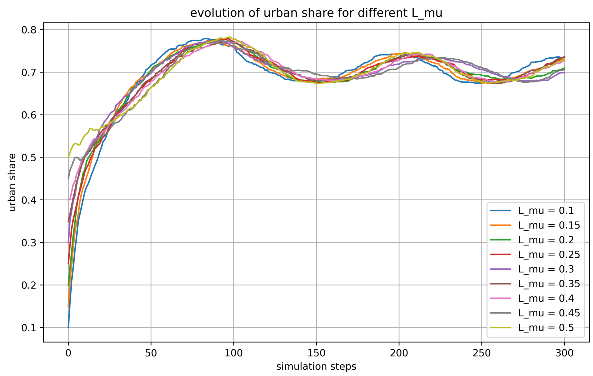


Figure 5.32: Evolution of urban share for different value of initial urban share.

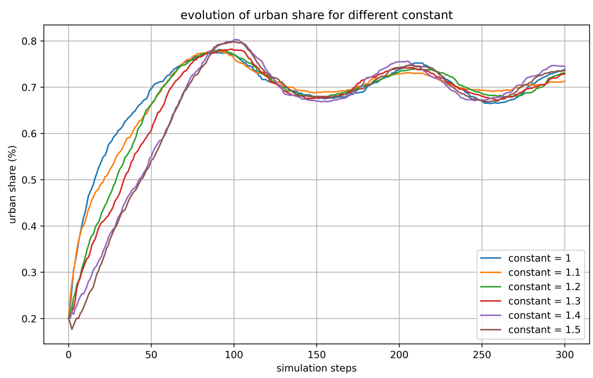


Figure 5.33: Evolution of urban share for different value of initial propensity matrix.

Constant	Distribution
1	$[-1; 1]$
1.1	$[-1.1; 0.9]$
1.2	$[-1.2; 0.8]$
1.3	$[-1.3; 0.7]$
1.4	$[-1.4; 0.6]$
1.5	$[-1.5; 0.5]$
1.6	$[-1.6; 0.4]$
1.7	$[-1.7; 0.3]$
1.8	$[-1.8; 0.2]$

Table 5.3: Initial distribution in `x_matrix`.

5.3.12 Influence of agricultural wage

Altering the parameter ϕ in agricultural wages changes the initial conditions of the model and sets a different equilibrium at the end of the simulation. Figure 5.34 clearly demonstrates that the higher the ϕ parameter, the higher the wage in the rural sector. Increasing the wage (or labor productivity) in the rural sector helps to reduce the urban population share in the final equilibrium. With the parameter $\phi = 0.3$, which we use throughout this work, we find an urban population share around 0.7. The higher the wage in the agricultural sector, the less people are attracted to the wages in the urban sector because the difference between the two is smaller, resulting in a higher equilibrium wage. Figure 5.36a represent the different evolution of urban shares for different value of parameter ϕ.

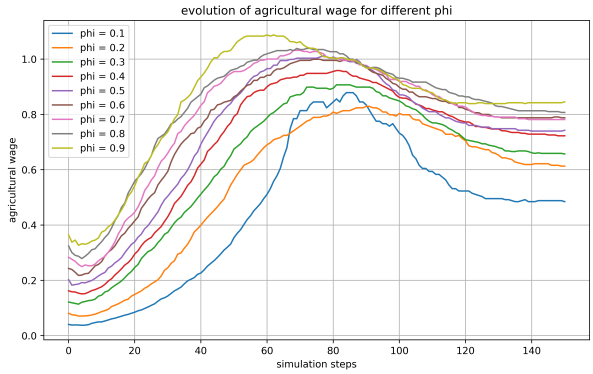
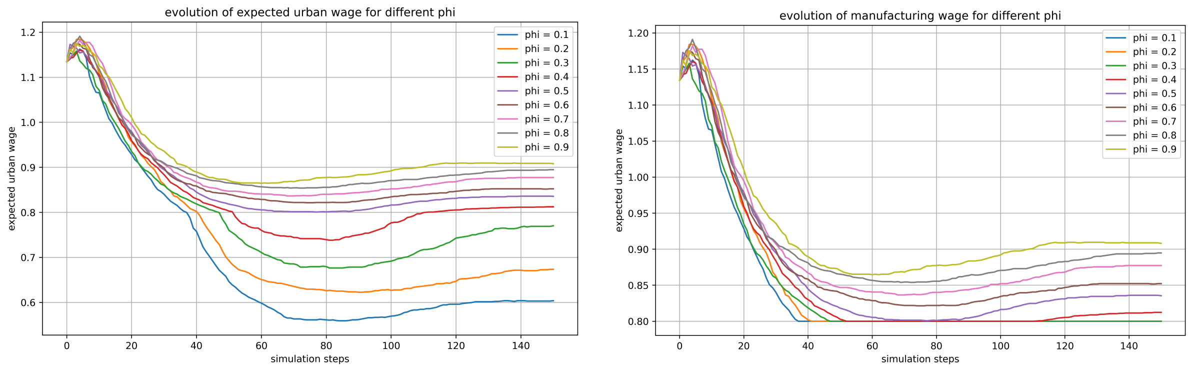


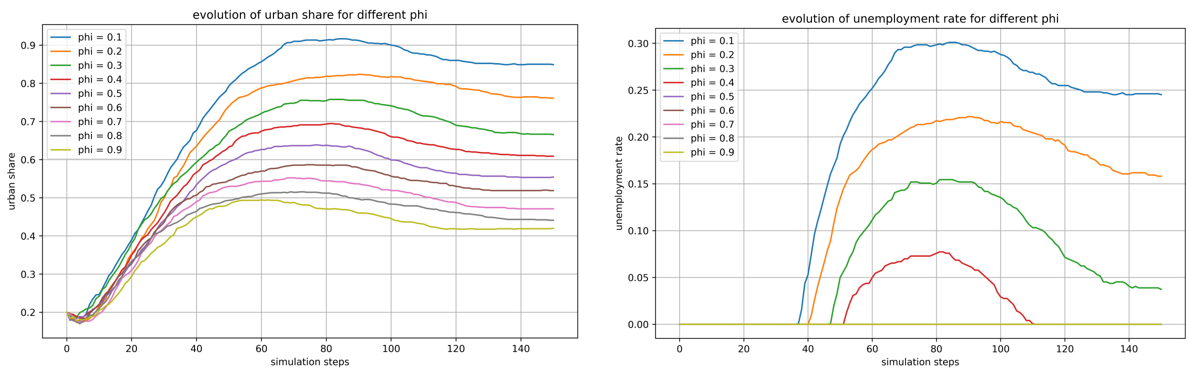
Figure 5.34: Evolution of agricultural wage for different value of parameter ϕ.



(a) Evolution of expected urban wage.

(b) Evolution of manufacturing wage.

Figure 5.35: Expected urban wage and manufacturing wage for different value of parameter ϕ.



(a) Evolution of urban share.

(b) Evolution of unemployment rate.

Figure 5.36: Evolution of urban share and unemployment rate for different value of parameter ϕ.

5.3.13 Influence of manufacturing wage

Modifying the parameter α in the manufacturing wage alters the initial conditions of the model and establishes a different equilibrium at the end of the simulation. As α increases, the manufacturing wage also increases (reflecting the productivity of labor in the urban sector, not the minimum wage), which tends to increase the urban population and decrease unemployment. Consequently, as the urban population grows and the rural population decreases, rural wages also rise. For values of $\alpha = 0.4$ and lower, there is no unemployment at equilibrium because the minimum wage is 0.8, and the equilibrium occurs at a manufacturing wage higher than the minimum wage.

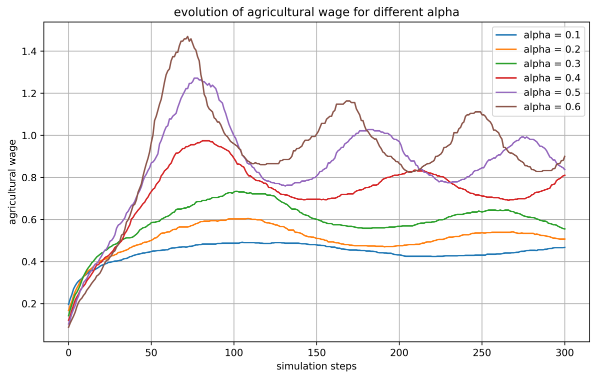
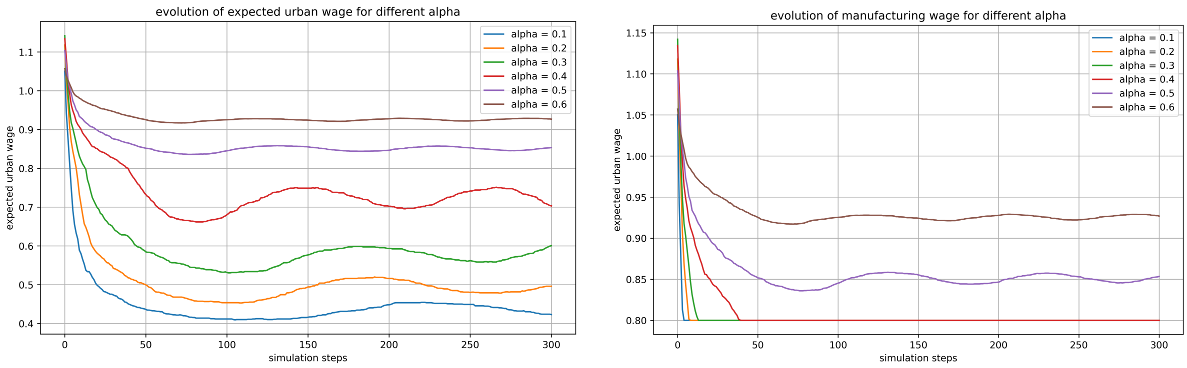
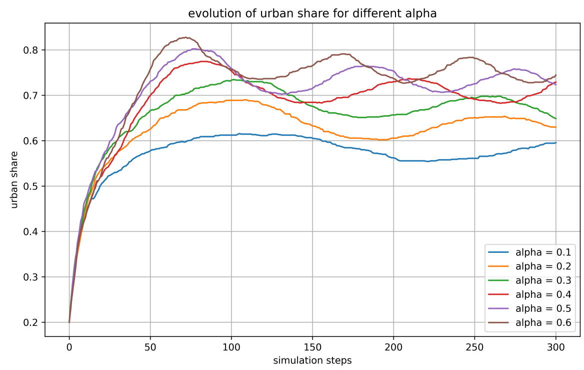


Figure 5.37: Evolution of agricultural wage for different value of parameter α.



(a) Evolution of expected urban wage. (b) Evolution of manufacturing wage.

Figure 5.38: Expected urban wage and manufacturing wage for different value of parameter α.



(a) Evolution of urban share.

(b) Evolution of unemployment rate.

Figure 5.39: Evolution of urban share and unemployment rate for different value of parameter α.

Chapter 6

Conclusions

In this master thesis, we have explored the complexities of migration dynamics within the framework of the agent-based Harris-Todaro model. Our investigation has revealed that this model provides a robust platform for simulating and understanding the dynamics of rural-to-urban migration, driven by economic incentives and modulated by a series of adjustable parameters.

In the introduction, we revisited the context of the urban transition that the world is currently experiencing. We examined various aspects that showcase changes from different perspectives, enabling us to understand the history and the historical phase we are living through. In the second chapter, we explored classical texts to critically analyze the theoretical foundations and assumptions of classical political economy.

In the literature review, we discussed migration models in economics and presented the foundational works that this thesis builds upon. In the chapter titled "Developments and Implementation of Agent-Based Models," we explained how we have recreated existing models from the literature and introduced the original model presented in this work.

Finally, in the results and discussion section, we presented and then discussed the outcomes of the Python model executions. These images are the result of hours of computation, noting that for each parameter explored, the program ran for approximately 25 to 40 minutes.

6.1 Modeling migration dynamics within the agent-based Harris-Todaro framework

The core question of this research was posed as follows:

> How can migration dynamics be modeled within the agent-based Harris-Todaro framework?

This master thesis has effectively addressed this question by demonstrating the flexibility and capability of the Harris-Todaro model within an agent-based framework to simulate various migration dynamics. The initial setup of the model and its customizable parameters allow for a detailed approximation of the myriad dynamics observable in real-world scenarios.

The parameters used in the model range from tangible constants, which can be directly estimated from empirical data, to more abstract variables that can be tailored to fit the needs of the simulation. This versatility ensures that the model can be adapted to reflect diverse migration patterns and trends seen across different geographical and socio-economic contexts.

This master thesis has successfully filled the research gap by using a technical approach to model migration. While my expertise in technical modeling has guided the focus of this thesis, there is an opportunity for further research. Integrating sociological research to study the actual behaviors of migrants moving from rural to urban sectors could refine how individual choices are modeled. Such interdisciplinary approaches could enhance the realism and applicability of the model, providing a deeper understanding of urban transition dynamics.

Although this thesis has primarily concentrated on the technical aspects of modeling within my personal competencies, it opens up avenues for future work to continue enhancing this model. The goal would be to capture more accurately the realities of urban transition, making significant strides towards bridging theoretical modeling and real-world applicability.

6.2 Comprehending Past Dynamics and Shaping Present Strategies

The Harris-Todaro model has repeatedly demonstrated its relevance in the social realities of economic development and migration dynamics. It provides an analytical framework that allows us to understand the urbanization that we have witnessed in the 20th and especially in the 21st century. Moreover, the emergence of new technical tools such as the ability to model complex economic behaviors in an agent-based framework represents a significant advancement in understanding the incentive and attraction factors that lead to migration.

This model has shown that not only we can understand the economic causes of current urbanization trends, but we can also offer policymakers a means to foresee future developments and thus give opportunities to plan urban development. The reduction of productivity disparities between rural and urban areas, as demonstrated by the model, stabilizes migration flows. This suggests that targeted economic policies can effectively manage urban growth and prevent the expansion of slums.

Through the application of the Harris-Todaro model, we can see how agent-based modeling provides robust insights into the mechanisms driving urbanization. By simulating various scenarios, policymakers can use these insights to develop strategies that not only address current urban challenges but also anticipate future issues. This proactive approach to urban planning is essential for creating sustainable and equitable urban environments that can adapt to the evolving needs of their populations.

6.3 Policy Implications and Recommendations

The inevitability of urbanization, as demonstrated by the model, emphasizes the critical need for policymakers to approach urban development with foresight and precision. It is not enough to passively observe the migration trends from rural to urban areas, proactive planning and strategic intervention are essential to ensure that these migrations foster equitable and sustainable urban growth. Effective urban planning must, therefore, be anticipatory, leveraging predictive models like the Harris-Todaro to preempt and mitigate adverse effects such as overcrowding, insufficient infrastructure, and the proliferation of slums.

Additionally, the agent-based Harris-Todaro model underscores the significant role of economic disparities in driving migration. Implementing policies that equalize opportunities across regions—by enhancing rural infrastructure, expanding educational access, and bolstering rural industries—can reduce the economic impetus for migration. Such measures would promote a more equitable demographic distribution and facilitate healthier urban development.

6.4 Future Research Directions

This thesis has laid the groundwork for further research into the dynamics of migration and urbanization. Future studies could expand upon this foundation by incorporating additional layers of complexity into the model, such as environmental factors, international migration, and the impacts of global economic changes. A sociological examination to understand the journeys and choices of migrants, and what influences them—whether consciously or unconsciously—seems essential to refine how agent-based models operate.

Furthermore, integrating empirical data into the simulation process could validate the model under real and concrete conditions, thereby testing its applicability to specific regions or scenarios. Unfortunately, this could not be accomplished within the scope of this master's thesis. This thesis represents a further step in the collective and scientific work surrounding the Harris-Todaro model, with each advance providing new insights into the complex interplay between human decision-making and the economic forces that shape our society.

In conclusion, this thesis has demonstrated the utility of the Harris-Todaro model within an agent-based framework for analyzing migration patterns. It provides a powerful tool for researchers and policymakers alike, capable of informing decisions that shape the urban landscapes of tomorrow. By understanding the past and anticipating the future, we can strive to create urban environments that foster inclusivity, sustainability, and prosperity for all.

Annex A - Code of the agent-based model: Espindola.py

```
from mesa import Agent, Model
import mesa
from mesa.datacollection import DataCollector
from mesa.time import RandomActivation
from mesa.space import MultiGrid, SingleGrid
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
import math
import random
from tqdm import tqdm
import cv2
import stats





#**********************************************************************
#**********************************************************************
# COMPUTING ECONOMICS VARIABLES

#======================================================================
# Compute wage in manufacturing jobs
#
#======================================================================
def compute_Wm():
    return alpha * A_m * math.pow(L_m, alpha-1)

#======================================================================
# Compute wage in agricultural jobs
#
#======================================================================
def compute_Wa():
    Y_a = A_a * math.pow(L_a, phi)
    # print(Y_a)
    Y_m = A_m * math.pow(L_m, alpha)
    P = rho * math.pow(Y_m/Y_a, sigma)
    return phi * A_a * P * math.pow(L_a, phi-1)

```

```
#==========================================================================
# Compute optimal urban population from manufactural wage
#
#==========================================================================
def compute_Lm_opt():
    return math.pow(alpha*A_m/W_m, 1/(1-alpha))






#**************************************************************************
#**************************************************************************
# INITIALIZATION

#==========================================================================
# Initialization of the population matrices
# origin list = random proportion of 2 for urban people
# migration list = only zeros
#
# 0: stay in the countryside 1: migrate to the city without wage 2: ...
    migrate to the city with wage
#==========================================================================
def init_list(width, height):
    matrix = np.zeros((height, width))                     # create ...
        the origin list
    migra_list = np.zeros((height, width))                 # create ...
        the migration list
    total = width*height                                   # total ...
        population
    to_urban = int(total*L_mu)                             # total ...
        urban population
    index = random.sample(range(0, total), to_urban)       # create a ...
        list of random numbers representing the urban people
    for i in range(to_urban):                              # ...
        implementing urban people in the origin matrix
        row = int(index[i]/width)
        col = index[i]%width
        matrix[row][col] = 2
    return matrix, migra_list                              # return ...
        matrices






#**************************************************************************
#**************************************************************************
# UTILITIES

#==========================================================================
# Save an image showing the entire population for step == title
# white pixel = 0 = rural
# light blue pixel = 1 = urban unemployed
```

```
# dark blue pixel = 2 = urban employed
#==========================================================================
def plot_list(data, title):
    # tips = sns.load_dataset("tips")
    # sns.jointplot(x = "total_bill", y = "tip", data = tips, kind ...
        ="hex", color="lightcoral")
    # with sns.axes_style("dark"):
    #     sns.jointplot(x, y, kind="hex")
    plt.title("simulation steps: "+title)
    plt.imshow(data, cmap='Blues')
    # plt.colorbar()
#    plt.savefig("./image/"+title+".png")
    plt.savefig("C:/Users/marti/Documents/ECO_THESIS/image/" + ...
        title + ".png")

#==========================================================================
# Save an image showing the evolution of "title"
#
#==========================================================================
def plot_line(data, title):
    y = data
    x = a = np.arange(0,len(data),1)
    plt.figure(figsize=(8,4))
    plt.plot(x,y,"b--",linewidth=1)
    plt.xlabel("simulation steps")
    plt.ylabel(title)
    plt.title(title)
    plt.savefig(title+'.png')

#==========================================================================
# Save a video showing the evolution of the entire population by step
#
#==========================================================================
def gen_video(length):
    video=cv2.VideoWriter( 'result.avi', ...
        cv2.VideoWriter_fourcc(*'MJPG'), 2, (1280,720))
    video.write(cv2.resize( cv2.imread( ...
        'C:/Users/marti/Documents/ECO_THESIS/image/origin.png') , ...
        (1280,720)))
    for i in range(0,length):
#        img=cv2.imread('./image/'+name[i]+'.png')   #Read image
        img=cv2.imread('C:/Users/marti/Documents/ECO_THESIS/image/' ...
            + str(i) + '.png')
        # Redimensionner l'image
        img=cv2.resize(img,(1280,720))
        video.write(img)

#==========================================================================
# Save an image showing the initial population (origin matrix)
#
#==========================================================================
def plot_grid(data):
    g = sns.heatmap(data, cmap="viridis", annot=True, cbar=False, ...
        square=True)
    g.figure.set_size_inches(4, 4)
    g.set(title="Number of agents on each cell of the grid");
```

```python







#**********************************************************************
#**********************************************************************
# MESA

#======================================================================
# MESA MODEL
#
#======================================================================
"""A model with some number of agents."""
class MigrationModel(Model):

   #===================================================================
   # MESA MODEL initialization
   #
   #===================================================================
    def __init__(self, L, width, height):
        self.num_agents = L                                 # Size of ... population
#        self.grid = MultiGrid(width, height, False)     # Create a ... grid (MultiGrid = multiple agents/cell)
        self.grid = SingleGrid(width, height, False)    # Create a ... grid (SingleGrid = single agent/cell)
        self.schedule = RandomActivation(self)          # Scheduler ... will call AGENT in a random order
        self.running = True
        self._steps = 0
        self._time = 0

        for i in range(self.num_agents):                # Fullfill ... the grid by MESA AGENT
            a = Harris_Todaro_Agent(i, self)            # ... Initialize MESA AGENT
            self.schedule.add(a)                        # Add AGENT ... to schedule
            x = int(i/self.grid.width)
            y = i%self.grid.width
            self.grid.place_agent(a, (x, y))            # Add AGENT ... on the grid

    #==================================================================
    # MESA MODEL uptade
    #
    #==================================================================
    def update_data(self):
        global migra_list
        global origin_list
        global L_a
        global L_m
        global L_mu
```

```
        global Lm_opt
        global L_mu_share
        global unemploy_rate
        #==================================================================
        # Compute urban share
        # Append the list : L_mu
        L_mu = ...
            np.sum(migra_list)/(2*self.grid.width*self.grid.height) ...
                     # L_mu = urban population / total population
        L_mu_share.append(L_mu) ...
                                                  # Append ...
            the list : L_mu_share

        #==================================================================
        # If there is less urban population than manufacturing jobs ...
            (NO UNEMPLOYMENT)
        if L_mu ≤ Lm_opt:              # If urban population is lower ...
            than optimal urban population
            origin_list = migra_list  # Set population list to ...
                migration list (step migration)
            L_a = 1-L_mu               # Update rural population
            L_m = L_mu                 # All urban population is ...
                employed in manufactures
            migra_list = np.zeros((self.grid.height, ...
                self.grid.width))          # Reset migration list
        #==================================================================
        # If there is more urban population than manufacturing jobs ...
            (UNEMPLOYMENT)
        else:
            L_a = 1-L_mu    # Update rural population
            L_m = Lm_opt    # Manufacturing workers < urban population

            total_migr = np.sum(migra_list)/2 ...
                                                  # Count ...
                all people migrating or staying to urban sector
            unemployed = int(total_migr - ...
                self.grid.width*self.grid.height*Lm_opt)       # ...
                Count all people unemployed (total urban pop - total ...
                jobs)
            index = random.sample(range(0, int(total_migr)), ...
                unemployed)                # Finding random urban ...
                population to be unemployed
            count = 0
            for i in range(self.grid.height): ...
                                              # Setting ...
                unemployed urban population
                for j in range(self.grid.width):
                    if migra_list[i][j] != 0:
                        if count in index:
                            migra_list[i][j] = 1
                        count += 1
            origin_list = migra_list ...
                                                  # Set ...
                population list to migration list (step migration)
            migra_list = np.zeros((self.grid.height, ...
                self.grid.width))          # Reset migration list
```

```


        #===================================================================
        # Append the list : unemploy_rate
        unemploy_rate.append(1-L_m/L_mu)

        #===================================================================
        # Update wages
        for i in range(self.grid.height): ...
                                                 # Change all agents
            for j in range(self.grid.width):
                pos = (i, j)
                cellmates = self.grid.get_cell_list_contents(pos)
                if origin_list[i][j] == 2:
                    cellmates[0].wage = compute_Wm() ...
                                                  # Manufacturing wages
                elif origin_list[i][j] == 1:
                    cellmates[0].wage = 0 ...
                                                             # No ...
                       wages for unemployed
                else:
                    cellmates[0].wage = compute_Wa() ...
                                                  # Agricultural wages

    #=======================================================================
    # Run a step/an iteration for the model
    #
    # i = iteration number
    #=======================================================================
    def step(self, i):
        # self.datacollector.collect(self)
        self.schedule.step()                    # step all agent
        self.update_data()                      # update data model
        global origin_list
        global migra_list
        # print(migra_list)
        plot_list(origin_list, str(i))

#===========================================================================
# MESA AGENT
#
# Agent = agent id (number from 0 to Pop-1)
#===========================================================================
class Harris_Todaro_Agent(Agent):

    #=======================================================================
    # MESA AGENT initialization
    #
    #=======================================================================
    def __init__(self, unique_id, model):
        super().__init__(unique_id, model)          # init agent
        x = int(unique_id/model.grid.width)
        y = unique_id%model.grid.width
        global origin_list                          # import origin list
        if origin_list[x, y] == 0:                  # wage = rural agent
            self.wage = compute_Wa()
```

```
        else:                                       # wage = employed ...
            urban agent
            self.wage = compute_Wm()

    #=====================================================================
    # update agent wage
    #
    #=====================================================================
    def update_wage(self):
        global origin_list
        global migra_list

        is_possible_mig = random.random()                       # give ...
            a random number between 0 and 1
        if origin_list[self.pos[0], self.pos[1]] == 1:          # if no ...
            wage = go to rural sector
            migra_list[self.pos[0], self.pos[1]] = 0

        elif origin_list[self.pos[0], self.pos[1]] == 0:        # if ...
            rural sector && rand>0.75 => 25% prob
            if is_possible_mig > 0.75:
                neighbors = self.model.grid.get_neighborhood(
                self.pos,
                moore=False,                                    # Moore ...
                    == TRUE => 8 surrounding squares, if Moor == ...
                    FALSE => Neumann - 4 surrounding squares
                include_center=False)                           # ...
                    Excluding yourself

                total_wage = 0
                for neighbor in neighbors:                      # Add ...
                    all neighbors' wages
                    cellmates = ...
                        self.model.grid.get_cell_list_contents(neighbor)
                    total_wage += cellmates[0].wage
                if total_wage/len(neighbors) > self.wage:       # If ...
                    the mean neighbor wage is higher than rural wage ...
                    = migrate to the city
                    migra_list[self.pos[0], self.pos[1]] = 2
        else:
            migra_list[self.pos[0], self.pos[1]] = 2            # If ...
                manufacturing wage = stay in urban sector

    #=====================================================================
    # Run a step/an iteration for the agent
    #
    #=====================================================================
    def step(self):
        # self.move()
        self.update_wage()                                      # ...
            update agent wage
```

```
#***********************************************************************
#***********************************************************************
# MAIN

if __name__ == '__main__':


    #=================================================================
    # Initialize all parameters and initial conditions
    #
    #=================================================================
    A_a = 1      # Parametric constant in rural sector production ...
        function        initial = 1
    A_m = 1      # Parametric constant in urban sector production ...
        function        initial = 1
    phi = 0.3    # Parametric constant in rural sector production ...
        function        initial = 0.3
    alpha = 0.7 # Parametric constant in urban sector production ...
        function        initial = 0.7
    sigma = 1    # Parametric constant in trade ...
                                          initial = 1
    rho = 1      # Parametric constant in trade ...
                                          initial = 1
    W_m = 0.8    # Minimum wage in urban sector ...
                                          initial = 0.8
    L_mu = 0.2       # labor input in urban sector (employed and ...
        unemployed)      initial = 0.2
    L_a = 1 - L_mu   # labor input in rural sector ...
                                       initial = 1 - L_mu = 0.8
    L_m = 0.2        # labor input in manufacturing jobs ...
                                 initial = 0.2
    L_mu_share = [L_mu]           # urbanization (urban/total ...
        population share)                                  LIST = 1 ...
        value per iteration
    unemploy_rate = [1-L_m/L_mu]  # urban unemployment rate ...
                            initial = [1-L_m/L_mu] = 0      LIST = 1 ...
        value per iteration
    Lm_opt = compute_Lm_opt()      # optimal manufacturing jobs

    width = 300          # size of the population = width x height = ...
        300 x 300 = 90000
    height = 300
    iterations = 20      # number of iterations of the model


    #=================================================================
    # Initialize origin and migration matrices
    #
    #=================================================================
    origin_list, migra_list = init_list(height,width)       # ...
        Initialization of population
```

```
    plot_list(origin_list, 'origin')                         # plot ... initial population (origin list)


    #======================================================================
    # Initialize model and agents
    #
    #======================================================================
    model = MigrationModel(height*width, width, height)      # ... Initialization of MESA MODEL



    #======================================================================
    # Iterate the model
    #
    #======================================================================
    for i in tqdm(range(iterations)):                        # ... Running the MESA MODEL n-times
        model.step(i)    # running the MESA MODEL for 1 iteration
# results of initial conditions
#     L_mu_share = [0.2, 0.3175222222222222, 0.4362666666666667, ... 0.5481222222222222, 0.6458, 0.7205, 0.6899222222222222, ... 0.7029555555555556, 0.6967888888888889, 0.6994222222222222, ... 0.6987444444444444, 0.6993777777777778, 0.6993888888888888, ... 0.7000222222222222, 0.6999888888888889, 0.6995222222222223, ... 0.7002111111111111, 0.6990777777777778, 0.6991222222222222, ... 0.6995888888888889, 0.6983777777777778]
#     unemploy_rate = [0.0, 0.0, 0.0, 0.0, 0.007808574861531525, ... 0.11067699881412496, 0.07126165571162502, 0.08848117453453563, ... 0.08041412173465468, 0.08387637396107783, 0.08298773972525209, ... 0.083818155632021, 0.08383271090796607, 0.08466159785565419, ... 0.08461800962081822, 0.08400733815305583, 0.0849085194640018, ... 0.08342498828777489, 0.08348325659321887, 0.08409462681419144, ... 0.08250628421583239]

    #======================================================================
    # Plots and prints
    #
    #======================================================================
    plot_grid(origin_list)
    plot_line(L_mu_share, "urban share")
    plot_line(unemploy_rate, "unemployment_rate")
    gen_video(iterations)
    print(L_mu_share)
    print(unemploy_rate)
```

Annex B - Code of the dynamical agent-based model: Cai.py

```
from mesa import Agent, Model
import mesa
from mesa.datacollection import DataCollector
from mesa.time import RandomActivation
from mesa.space import MultiGrid, SingleGrid
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
import math
import random
from tqdm import tqdm
import cv2
import stats





#***********************************************************************
#***********************************************************************
# COMPUTING ECONOMICS VARIABLES

#=======================================================================
# Compute wage in manufacturing jobs
#
#=======================================================================
def compute_Wm():
    if (True):
        return alpha * A_m * math.pow(L_m, alpha-1)
    Xi2 = alpha * A_m * math.pow(eta/(1-eta), alpha * eta) * ...
        math.pow((1-eta/mu)/Zm, alpha - 1)
    return Xi2 * math.pow(L_m, alpha-1)

#=======================================================================
# Compute wage in agricultural jobs
#
#=======================================================================
def compute_Wa():
    if (True):
        Y_a = A_a * math.pow(L_a, phi)
```

```
        # print(Y_a)
        Y_m = A_m * math.pow(L_m, alpha)
        P = rho * math.pow(Y_m/Y_a, sigma)
        return phi * A_a * P * math.pow(L_a, phi-1)
    Xi3 = A_a *  math.pow(Za, 1 - phi)
    Y_a = Xi3 * math.pow(L_a, phi)
    # print(Y_a)
    Xi1 = A_m * math.pow(Zm, 1 - alpha) * ...
        math.pow(math.pow(eta/(1-eta), eta)*(1-eta/mu), alpha)
    Y_m = Xi1 * math.pow(L_m, alpha)
    P = rho * math.pow(Y_m/Y_a, sigma)
    Xi4 = phi * A_a / math.pow(Za, phi - 1)
    return Xi4 * P * math.pow(L_a, phi-1)

#========================================================================
# Compute optimal urban population from manufactural wage
#
#========================================================================
def compute_Lm_opt():
    return math.pow(alpha*A_m/W_m, 1/(1-alpha))






#************************************************************************
#************************************************************************
# INITIALIZATION

#========================================================================
# Initialization of the population matrices
# origin list = random proportion of 2 for urban people
# migration list = only zeros
#
# 0: stay in the countryside 1: migrate to the city without wage 2: ...
   migrate to the city with wage
#========================================================================
def init_list(width, height):
    origin_matrix = np.zeros((height, width))   # create the origin ...
        list
    migra_list = np.zeros((height, width))      # create the ...
        migration list


    x_matrix = np.zeros((height, width))
    dotx_matrix = np.zeros((height, width))
    x_matrix = 2*np.random.rand(height, width)-1

    weights_matrix_matrix = [[np.random.rand(height, width) * ...
        social_link for _ in range(width)] for _ in range(height)]
    for rows in weights_matrix_matrix:
        for inner_matrix in rows:
                for i in range(height):
                    for j in range(width):
                        if inner_matrix[i, j] < sparse_factor:
```

```
                            inner_matrix[i, j] = 0


    total = width*height                                # total ...
        population
    to_urban = int(total*L_mu)                          # total ...
        urban population
    index = random.sample(range(0, total), to_urban)    # create a ...
        list of random numbers representing the urban people
    for i in range(to_urban):                           # ...
        implementing urban people in the origin matrix
        row = int(index[i]/width)
        col = index[i]%width
        origin_matrix[row][col] = 2
    return origin_matrix, migra_list, x_matrix, dotx_matrix, ...
        weights_matrix_matrix    # return matrices






#************************************************************************
#************************************************************************
# UTILITIES

#========================================================================
# Save an image showing the entire population for step == title
# white pixel = 0 = rural
# light blue pixel = 1 = urban unemployed
# dark blue pixel = 2 = urban employed
#========================================================================
def plot_list(data, title):
    # tips = sns.load_dataset("tips")
    # sns.jointplot(x = "total_bill", y = "tip", data = tips, kind ...
        ="hex", color="lightcoral")
    # with sns.axes_style("dark"):
    #     sns.jointplot(x, y, kind="hex")
    plt.figure(dpi = 500)
    plt.title("simulation steps: "+title)
    # plt.imshow(data, cmap='Blues')
    plt.imshow(data)
    plt.colorbar()
#     plt.savefig("./image/"+title+".png")
    plt.savefig("C:/Users/marti/Documents/ECO_THESIS/image/" + ...
        title + ".png")
    plt.close()

#========================================================================
# Save an image showing the evolution of "title"
#
#========================================================================
def plot_line(data, title):
    y = data
    x = np.arange(0,len(data),1)
    plt.figure(dpi = 500)
```

```
    plt.plot(x,y)
    plt.xlabel("simulation steps")
    plt.ylabel(title)
    plt.title(title)
    plt.grid()
    plt.savefig(title+'.png')



#==========================================================================
# Save an image showing the evolution of "title"
#
#==========================================================================
def plot_line_dot(data, title, dot):
    y = data
    x = np.arange(0,len(data),1)
    plt.figure(figsize=(8,4), dpi = 500)
    plt.plot(x,y)
    plt.plot(dot, y[dot], marker='o', markersize=10)
    plt.xlabel("simulation steps")
    plt.ylabel(title)
    plt.title("simulation steps: "+str(dot))
    plt.grid()
    plt.savefig("C:/Users/marti/Documents/ECO_THESIS/image/" + ...
        title + str(dot) + ".png")
    plt.close()


#==========================================================================
# Save a video showing the evolution of the entire population by step
#
#==========================================================================
def gen_video(length,name):
    video=cv2.VideoWriter(name+'.avi', ...
        cv2.VideoWriter_fourcc(*'MJPG'), 10, (1280,720))
    if name == 'result':
        img_name = ""
        video.write(cv2.resize( v2.imread( ...
            'C:/Users/marti/Documents/ECO_THESIS/image/origin.png'), ...
            (1280,720)))
    if name == 'x':
        img_name = "x"
    if name == 'dotx':
        img_name = "dotx"
    if name == "urban_share":
        img_name = "urban_share"
    if name == "unemployment_rate":
        img_name = "unemployment_rate"
    for i in tqdm(range(0,length)):
#         img=cv2.imread('./image/'+name[i]+'.png')  #Read image
        img=cv2.imread('C:/Users/marti/Documents/ECO_THESIS/image/' ...
            + img_name + str(i) + '.png')
        # Redimensionner l'image
        img=cv2.resize(img,(1280,720))
        video.write(img)
    if name == "urban_share" or name == "unemployment_rate":
```

```
        img=cv2.imread('C:/Users/marti/Documents/ECO_THESIS/image/' ...
            + img_name + str(length) + '.png')
        # Redimensionner l'image
        img=cv2.resize(img,(1280,720))
        video.write(img)

#========================================================================
# Save an image showing the initial population (origin matrix)
#
#========================================================================
def plot_grid(data):
    g = sns.heatmap(data, cmap="viridis", annot=False, cbar=False, ...
        square=False)
    g.figure.set_size_inches(4, 4)
    g.set(title="purple = rural, green = unemployed, yellow = urban");







#************************************************************************
#************************************************************************
# MESA

#========================================================================
# MESA MODEL
#
#========================================================================
"""A model with some number of agents."""
class MigrationModel(Model):

   #=====================================================================
   # MESA MODEL initialization
   #
   #=====================================================================
    def __init__(self, L, width, height):
        super().__init__()
        self.num_agents = L                               # Size of ...
            population
#         self.grid = MultiGrid(width, height, False)     # Create a ...
   grid (MultiGrid = multiple agents/cell)
        self.grid = SingleGrid(width, height, False)     # Create a ...
            grid (SingleGrid = single agent/cell)
        self.schedule = RandomActivation(self)           # Scheduler ...
            will call AGENT in a random order
        self.running = True
        self._steps = 0
        self._time = 0

        for i in range(self.num_agents):                 # Fullfill ...
            the grid by MESA AGENT
            a = Harris_Todaro_Agent(i, self)             # ...
                Initialize MESA AGENT
```

```
            self.schedule.add(a)                          # Add AGENT ...
                to schedule
            x = int(i/self.grid.width)
            y = i%self.grid.width
            self.grid.place_agent(a, (x, y))              # Add AGENT ...
                on the grid

    #===================================================================
    # MESA MODEL uptade
    #
    #===================================================================
    def update_data(self):
        global migra_list
        global origin_list
        global x_matrix
        global dotx_matrix
        global L_a
        global L_m
        global L_mu
        global Lm_opt
        global L_mu_share
        global unemploy_rate
        global unemployed_tot
        global W_a_ev
        global W_m_ev
        global find_job
        global random_unemployment
        #===============================================================
        # Compute urban share
        # Append the list : L_mu
        L_mu = ...
           np.sum(migra_list)/(2*self.grid.width*self.grid.height) ...
                    # L_mu = urban population / total population
        L_mu_share.append(L_mu)           # Append the list : L_mu_share
        unemployed = 0
        #===============================================================
        # If there is less urban population than manufacturing jobs ...
            (NO UNEMPLOYMENT)
        if L_mu <= Lm_opt:                # If urban population is ...
           lower than optimal urban population
            origin_list = migra_list      # Set population list to ...
                migration list (step migration)
            L_a = 1-L_mu                  # Update rural population
            L_m = L_mu                    # All urban population is ...
                employed in manufactures

        #===============================================================
        # If there is more urban population than manufacturing jobs ...
            (UNEMPLOYMENT)
        else:
            L_a = 1-L_mu     # Update rural population
            L_m = Lm_opt     # Manufacturing workers < urban population


            total_migr = np.sum(migra_list)/2 ...
                                                        # Count ...
```

```
            all people migrating or staying to urban sector
        unemployed = round(total_migr - ...
            self.grid.width*self.grid.height*Lm_opt)       # ...
            Count all people unemployed (total urban pop - total ...
            jobs)

        if random_unemployment:

            index = random.sample(range(0, int(total_migr)), ...
                unemployed)                   # Finding random ...
                urban population to be unemployed
            count = 0
            for i in range(self.grid.height): ...
                                                  # Setting ...
                unemployed urban population
                for j in range(self.grid.width):
                    if migra_list[i][j] != 0:
                        if count in index:
                            migra_list[i][j] = 1
                        count += 1

        else:
            count = 0
            rural_now = []
            unemployed_still = []
            for i in range(height):
                for j in range(width):
                    if migra_list[i][j] == 0:
                        rural_now.append(count)
                    if origin_list[i][j] == 1 and ...
                        migra_list[i][j] != 0:
                        if random.random()> find_job:
                            unemployed_still.append(count)
                    count += 1
            total_new_unemployed = unemployed - ...
                len(unemployed_still)
            potential_new_unemployed = set(range(width*height))
            potential_new_unemployed = potential_new_unemployed ...
                - set(rural_now) - set(unemployed_still)

            if total_new_unemployed≥0:
                index = sorted(random.sample( ...
                    list(potential_new_unemployed), ...
                    total_new_unemployed))                # ...
                    Finding random urban population to be unemployed
                count = 0
                for i in range(height): ...
                                                  # Setting ...
                    unemployed urban population
                    for j in range(width):
                        if migra_list[i][j] != 0:
                            if count in index or count in ...
                                unemployed_still:
                                migra_list[i][j] = 1
                        count += 1
```

```
                                        # Set population list to ...
                                            migration list (step ...
                                            migration)
                    else :
                        index = ...
                            sorted(random.sample(list(unemployed_still), ...
                            unemployed))
                        count = 0
                        for i in range(height): ...
                                                    # Setting ...
                            unemployed urban population
                            for j in range(width):
                                if migra_list[i][j] != 0:
                                    if count in index:
                                        migra_list[i][j] = 1
                                count += 1

                origin_list = migra_list    # Set population list to ...
                    migration list (step migration)
            x_matrix = dotx_matrix + x_matrix

            #================================================================
            # Append the list : unemploy_rate
            unemploy_rate.append(1-L_m/L_mu)
            unemployed_tot.append(unemployed/(self.grid.width * ...
                self.grid.height))
            W_a_ev.append(compute_Wa())
            W_m_ev.append(compute_Wm())
            #================================================================
            # Update wages
            for i in range(self.grid.height): ...
                                                # Change all agents
                for j in range(self.grid.width):
                    pos = (i, j)
                    cellmates = self.grid.get_cell_list_contents(pos)
                    if origin_list[i][j] == 2:
                        cellmates[0].wage = compute_Wm() ...
                                                # Manufacturing wages
                    elif origin_list[i][j] == 1:
                        cellmates[0].wage = 0 ...
                                                        # No ...
                            wages for unemployed
                    else:
                        cellmates[0].wage = compute_Wa() ...
                                                # Agricultural wages

        #====================================================================
        # Run a step/an iteration for the model
        #
        # i = iteration number
        #====================================================================
        def step(self, i):
            # self.datacollector.collect(self)
            self.schedule.step()                # step all agent
            self.update_data()                  # update data model

```

```
        global origin_list
        global migra_list
        global x_matrix
        global dotx_matrix
        global debug
        plot_list(origin_list, str(i))
        if debug:
            plot_list(x_matrix, "x"+str(i))
            plot_list(dotx_matrix, "dotx"+str(i))
        migra_list = np.zeros((self.grid.height, self.grid.width)) ...
                    # Reset migration list
        dotx_matrix = np.zeros((self.grid.height, self.grid.width))



#========================================================================
# MESA AGENT
#
# Agent = agent id (number from 0 to Pop-1)
#========================================================================
class Harris_Todaro_Agent(Agent):

    #====================================================================
    # MESA AGENT initialization
    #
    #====================================================================
    def __init__(self, unique_id, model):
        super().__init__(unique_id, model)      # init agent
        x = int(unique_id/model.grid.width)
        y = unique_id%model.grid.width
        global origin_list                      # import origin list
        if origin_list[x, y] == 0:              # wage = rural agent
            self.wage = compute_Wa()
        else:                                   # wage = employed ...
            urban agent
            self.wage = compute_Wm()

    #====================================================================
    # update agent wage
    #
    #====================================================================
    def update_wage(self):
        global origin_list
        global migra_list
        global x_matrix
        global dotx_matrix
        global weights_matrix_matrix
        global width
        global height
        global x_ev

        Wm = compute_Wm()
        Wa = compute_Wa()

        social_influence = 0
        for i in range(height):
```

```
            for j in range(width):
                social_influence += ...
                    weights_matrix_matrix[self.pos[0]][self.pos[1]][i][j] ...
                    * (x_matrix[i,j]-x_matrix[self.pos[0]][self.pos[1]])

        dotx_matrix[self.pos[0]][self.pos[1]] = a * ...
            x_matrix[self.pos[0]][self.pos[1]] + f * ...
            social_influence + b * ((L_m/L_mu) * Wm - Wa) + c * ...
            (self.wage - Wa)

        prob = np.abs(x_matrix[self.pos[0]][self.pos[1]] + ...
            dotx_matrix[self.pos[0]][self.pos[1]] ...
            )/(np.abs(x_matrix[self.pos[0]][self.pos[1]] + ...
            dotx_matrix[self.pos[0]][self.pos[1]] )+beta)

        if random.random() < prob:
            if x_matrix[self.pos[0]][self.pos[1]] + ...
                dotx_matrix[self.pos[0]][self.pos[1]] > 0:
                migra_list[self.pos[0]][self.pos[1]]=2
            else:
                migra_list[self.pos[0]][self.pos[1]]=0
        else:
            if origin_list[self.pos[0]][self.pos[1]]==2:
                migra_list[self.pos[0]][self.pos[1]]=2
            if origin_list[self.pos[0]][self.pos[1]]==1:
                migra_list[self.pos[0]][self.pos[1]]=2

    #======================================================================
    # Run a step/an iteration for the agent
    #
    #======================================================================
    def step(self):
        # self.move()
        self.update_wage()                                  # ...
            update agent wage









#**************************************************************************
#**************************************************************************
# MAIN

if __name__ == '__main__':


    #======================================================================
    # Initialize all parameters and initial conditions
    #
    #======================================================================
```

```
    A_a = 1      # Parametric constant in rural sector production ...
        function        initial = 1
    A_m = 1      # Parametric constant in urban sector production ...
        function        initial = 1
    phi = 0.3    # Parametric constant in rural sector production ...
        function        initial = 0.3
    alpha = 0.7 # Parametric constant in urban sector production ...
        function        initial = 0.7
    sigma = 1    # Parametric constant in trade ...
                                  initial = 1
    rho = 1      # Parametric constant in trade ...
                                  initial = 1


    eta = 0.5
    Zm = 3
    Za = 3
    mu = 1

    # a = 0.8
    # f = 0.01
    # b = 1
    # beta = 4
    # social_link = 0
    # sparse_factor = 0.9 * social_link

    debug = True
    concensus = False
    find_job = 0.2
    random_unemployment = False
    if concensus:
        a = 0.002
        f = 0.004
        b = 0.02
        c = 0.01
        beta = 2
        social_link = 0.1
        sparse_factor = 0.08
    else:
        a = 0.0008
        f = 0.001
        b = 0.02
        c = 0.01
        beta = 3
        social_link = 0.1
        sparse_factor = 0.09

    W_m = 0.8    # Minimum wage in urban sector ...
                                  initial = 0.8
    L_mu = 0.2       # labor input in urban sector (employed and ...
        unemployed)      initial = 0.2
    L_a = 1 - L_mu   # labor input in rural sector ...
                                initial = 1 - L_mu = 0.8
    L_m = 0.2        # labor input in manufacturing jobs ...
                          initial = 0.2
```

```
    L_mu_share = [L_mu]          # urbanization (urban/total ...
       population share)                              LIST = 1 ...
       value per iteration
    unemploy_rate = [1-L_m/L_mu]  # urban unemployment rate ...
                           initial = [1-L_m/L_mu] = 0      LIST = 1 ...
       value per iteration
    unemployed_tot = [0]  # urban unemployment rate ...
                           initial = [1-L_m/L_mu] = 0      LIST = 1 ...
       value per iteration
    Lm_opt = compute_Lm_opt()     # optimal manufacturing jobs

    width = 30          # size of the population = width x height = ...
       300 x 300 = 90000
    height = 30
    iterations = 500     # number of iterations of the model


    #===================================================================
    # Initialize origin and migration matrices
    #
    #===================================================================
    origin_list, migra_list, x_matrix, dotx_matrix, ...
       weights_matrix_matrix = init_list(height,width)      # ...
       Initialization of population
    plot_list(origin_list, 'origin')                       # plot ...
       initial population (origin list)


    #===================================================================
    # Initialize model and agents
    #
    #===================================================================
    model = MigrationModel(height*width, width, height)     # ...
       Initialization of MESA MODEL
    W_m_ev = [compute_Wm()]
    W_a_ev = [compute_Wa()]


    #===================================================================
    # Iterate the model
    #
    #===================================================================
    print("Running model : ")
    for i in tqdm(range(iterations)):                      # ...
       Running the MESA MODEL n-times
        model.step(i)    # running the MESA MODEL for 1 iteration


    exp_urban_w = [W_m_ev[0]*(1-unemploy_rate[0])]
    for i in range(len(W_m_ev)-1):
        exp_urban_w.append(W_m_ev[i+1]*(1-unemploy_rate[i+1]))
    #===================================================================
    # Plots and prints
    #
    #===================================================================
    plot_grid(origin_list)
```

```
    plot_line(L_mu_share, "urban share")
    plot_line(unemploy_rate, "unemployment_rate")
    plt.figure(dpi = 500)
    plt.plot(W_m_ev, label="manufacturing wage")
    plt.plot(exp_urban_w, label="expected urban wage")
    plt.plot(W_a_ev, label="agricultural wage")
    plt.xlabel("simulation steps")
    plt.ylabel("wage")
    plt.title("wages")
    plt.legend()
    plt.grid()
    plt.savefig('wages.png')
    plt.show()
    print("Video population : ")
    gen_video(iterations,"result")
    if debug:
        print("Creating images : ")
        for i in tqdm(range(iterations+1)):
            plot_line_dot(L_mu_share, "urban_share",i)
            plot_line_dot(unemploy_rate, "unemployment_rate",i)
        print("Video urban share : ")
        gen_video(iterations,"urban_share")
        print("Video unemployment rate : ")
        gen_video(iterations,"unemployment_rate")
        print("Video propensity")
        gen_video(iterations, "x")
        print("Video propensity derivative")
        gen_video(iterations, "dotx")
    # print(L_mu_share)
    # print(unemploy_rate)
```

Annex C - Videos of the results of thee dynamical agent-based model without consensus

Open pdf on Adobe Acrobat to read video Figure 6.1.

Figure 6.1: Video - no consensus.

Annex D - Videos of the results of thee dynamical agent-based model with consensus

Open pdf on Adobe Acrobat to read video Figure 6.2.

Figure 6.2: Video - consensus.

Bibliography

[1] Hannah Ritchie and Max Roser. Urbanization. *Our world in data*, 2018.

[2] World bank based on data from the un population division – processed by our world in data. "urban population" [dataset]. world bank based on data from the un population division [original data]. https://ourworldindata.org/urbanization [Accessed: April 2024].

[3] World bank on may 2023 – with minor processing by our world in data. https://ourworldindata.org/grapher/gdp-per-capita-worldbank [Accessed: April 2024].

[4] Statista Research Department. Brics and g7 countries' share of the world's total gross domestic product (gdp) in purchasing power parity (ppp) from 2000 to 2023, 2024.

[5] Christopher M Law. The growth of urban population in england and wales, 1801-1911. *Transactions of the Institute of British Geographers*, pages 125–143, 1967.

[6] Mohamed Sahnouni, Josep María Parés, Mathieu Duval, Isabel Cáceres, Zoheir Harichane, Jan van der Made, Alfredo Pérez-González, Salah Abdessadok, Nadia Kandi, Abdelkader Derradji, et al. Evidence of stone tools and cutmarked bones dated to 2.44 and 1.92 millions years ago from aïn boucherit (sétif, algeria) and their implications on the first human occupation in north africa. Agence du Service Géologique de l'Algérie, 2021.

[7] Michael Chazan. Toward a long prehistory of fire. *Current Anthropology*, 58(S16):S351–S359, 2017.

[8] Jean-Jacques Hublin, Abdelouahed Ben-Ncer, Shara E Bailey, Sarah E Freidline, Simon Neubauer, Matthew M Skinner, Inga Bergmann, Adeline Le Cabec, Stefano Benazzi, Katerina Harvati, et al. New fossils from jebel irhoud, morocco and the pan-african origin of homo sapiens. *Nature*, 546(7657):289–292, 2017.

[9] Samuel Bowles and Jung-Kyoo Choi. The neolithic agricultural revolution and the origins of private property. *Journal of Political Economy*, 127(5):2186–2228, 2019.

[10] Andrey Korotayev and Leonid Grinin. The urbanization and political development of the world system: A comparative quantitative analysis. *HISTORY & MATHEMATICS: HISTORICAL DYNAMICS AND DEVELOPMENT OF COMPLEX SOCIETIES, P. Turchin, LE Grinin, VC de Munck, AV Korotayev, eds*, pages 115–153, 2006.

[11] Patricia Clarke Annez and Robert M Buckley. Urbanization and growth: Setting the context. *Urbanization and growth*, 1:1–45, 2009.

[12] Mingxing Chen, Xinrong Huang, Jiafan Cheng, Zhipeng Tang, and Gengzhi Huang. Urbanization and vulnerable employment: Empirical evidence from 163 countries in 1991–2019. *Cities*, 135:104208, 2023.

[13] William Arthur Lewis et al. Economic development with unlimited supplies of labour. 1954.

[14] Marx Karl. *Le Capital livre 1*. Editions sociales, 2016.

[15] Jean Batou. "l'accumulation primitive au xxie siècle", 2012. https://lesfilmsdelan2.org/ann%C3%A9e-2011-2012 [Accessed: May 2024].

[16] Adam Smith. *An inquiry into the nature and causes of the Wealth of Nations*. MetaLibri Digital Library, 2007.

[17] Karl Marx. *Contribution à la critique de l'économie politique*. Editions sociales, 2014.

[18] Karl Marx and Friedrich Engels. *Manifeste du parti communiste*. EPO, 2023.

[19] Jean de La Fontaine. *La cigale et la fourmi*. Bibliothèque et Archives nationales du Québec, 2006.

[20] Marx Karl. *Théories sur la plus-value – tome 3*. Editions sociales, 2024.

[21] Department of Economic United Nations and Population Division Social Affairs. World urbanization: Prospects: The 2018 revision, 2019. https://population.un.org/wup/Publications/Files/WUP2018-Report.pdf [Accessed: May 2024].

[22] David Harvey. The new imperialism: accumulation by dispossession. In *Karl Marx*, pages 213–237. Routledge, 2017.

[23] David Harvey. Le «nouvel impérialisme»: accumulation par expropriation. *Actuel Marx*, (1):71–90, 2004.

[24] Rosa Luxemburg. *The accumulation of capital*. Routledge, 2015.

[25] International Labour Office – Geneva: ILO. Women and men in the informal economy: A statistical picture (third edition), 2018. https://webapps.ilo.org/wcmsp5/groups/public/---dgreports/---dcomm/documents/publication/wcms_626831.pdf [Accessed: May 2024].

[26] Un-Habitat. World cities report 2016: Urbanization and development-emerging futures, 2016. https://unhabitat.org/world-cities-report-2016 [Accessed: May 2024].

[27] African Development Bank. African economic outlook 2019, 2019. https://www.afdb.org/fileadmin/uploads/afdb/Documents/Publications/2019AEO/AEO_2019-EN.pdf [Accessed: May 2024].

[28] Hajer M. Baynes T. Bergesen J. Labbé F. Musango J.K. Ramaswami A. Robinson B. Salat S. Suh S. Currie P. Fang A. Hanson A. Kruit K. Reiner M. Smit S. Tabory S. A Report by the International Resource Panel. United Nations Environment Programme Swilling, M. The weight of cities: Resource requirements of future urbanization, 2018. https://www.resourcepanel.org/reports/weight-cities [Accessed: May 2024].

[29] John Gallup. Theories of migration. *Harvard Institute for International Development (HIID) Papers*, 1997.

[30] Charles A Ingene. The state of the art in modeling migration in ldcs: a comment. *Journal of regional science*, 41(3):529–543, 2001.

[31] Karin Margarethe Bahns. Rural-to-urban migration in developing countries: The applicability of the harris todaro model with a special focus on the chinese economy. 2005.

[32] Derek Laing, Chuhwan Park, and Ping Wang. A modified harris-todaro model of rural-urban migration for china. *Critical issues in China's growth and development*, pages 245–264, 2005.

[33] Tadashi Inoue. On the dynamic characteristics of the open harris-todaro model. *Korea and the World Economy*, 10(2):259–284, 2009.

[34] Hein De Haas. The internal dynamics of migration processes: A theoretical inquiry. *Journal of ethnic and migration studies*, 36(10):1587–1617, 2010.

[35] Jaylson J Silveira, Aquino L Espíndola, and TJP Penna. Agent-based model to rural–urban migration analysis. *Physica A: Statistical Mechanics and its Applications*, 364:445–456, 2006.

[36] Aquino L Espíndola, Jaylson J Silveira, and TJP Penna. A harris-todaro agent-based model to rural-urban migration. *Brazilian journal of physics*, 36:603–609, 2006.

[37] Ning Cai, Hai-Ying Ma, and M Junaid Khan. Agent-based model for rural–urban migration: A dynamic consideration. *Physica A: Statistical Mechanics and its Applications*, 436:806–813, 2015.

[38] Anna Klabunde and Frans Willekens. Decision-making in agent-based models of migration: State of the art and challenges. *European Journal of Population*, 32:73–97, 2016.

[39] Samira Boulahbel-Bachari, Nadjia El Saadi, and Alassane Bah. Agent based modelling approach of migration dynamics. In *Applied Computational Intelligence and Mathematical Methods: Computational Methods in Systems and Software 2017, vol. 2*, pages 338–349. Springer, 2018.

[40] Zhaohao Fu and Lingxin Hao. Agent-based modeling of china's rural–urban migration and social network structure. *Physica A: Statistical Mechanics and its Applications*, 490:1061–1075, 2018.

[41] Walter Isard. Location theory and trade theory: short-run analysis. *The Quarterly Journal of Economics*, 68(2):305–320, 1954.

[42] Jean-Paul Rodrigue. *The geography of transport systems*. Routledge, 2020.

[43] Henry Charles Carey. *Principles of social science*, volume 3. Lippincott, 1859.

[44] Larry A Sjaastad. The costs and returns of human migration. *Journal of political Economy*, 70(5, Part 2):80–93, 1962.

[45] Michael P Todaro. A model of labor migration and urban unemployment in less developed countries. *The American economic review*, 59(1):138–148, 1969.

[46] John R Harris and Michael P Todaro. Migration, unemployment and development: a two-sector analysis. *The American economic review*, 60(1):126–142, 1970.

[47] Project Mesa Team. Mesa documentation release, 2024. https://mesa.readthedocs.io/_/downloads/en/stable/pdf/.

[48] Anna Klabunde. Computational economic modeling of migration. *Available at SSRN 2470525*, 2014.

[49] Paul R Masson. Migration, human capital, and poverty in a dual-economy model of a developing country. *IMF Working Papers*, 2001(128), 2001.

[50] Jonathan Gray, Jason Hilton, and Jakub Bijak. Choosing the choice: Reflections on modelling decisions and behaviour in demographic agent-based models. *Population Studies*, 71(sup1):85–97, 2017.

[51] Mesa project. https://github.com/projectmesa/mesa/.

[52] Mesa documentation. https://mesa.readthedocs.io/en/stable/.

[53] Jianwei Wang Guaiyoui https://github.com/guaiyoui. Python implementation of the paper : A harris-todaro agent-based model to rural-urban migration (espíndola 2006), 2021. https://github.com/guaiyoui/abm_migration.