Master's thesis

World Migration Network : modelling and analysis about human migration

Quentin L. Cappart[1]
Adrien P. Thonet[2]

Supervisor :
Jean-Charles Delvenne

Co-supervisor :
Pierre Dupont

Reader :
Jean-François Carpantier

Louvain-la-Neuve
2013-2014

[1]Master's degree in Computer Science and Engineering - Université catholique de Louvain - quentin.c@swing.be
[2]Master's degree in Mathematical Engineering - Université catholique de Louvain - adrien.thonet@hotmail.com

Acknowledgements

Foremost, I would like to express my gratitude to our supervisor Jean-Charles Delvenne for the continuous support, his useful comments and remarks throughout this entire year.

I would like to thank my relatives who took time to read this report to make it better : my father Philippe, my sister Amélie and especially my brother Sébastien for his precious advice as an economist.

Always working is not something pleasant. This is why I would like to thank my roommates from the Kot Méca and my 'almost roommates' from the Dépakot. You made this year a great year for me.

My sincere thanks go to all those who contributed in this report directly or not through discussions and encouragements.

I am deeply thankful to all my family for their support. You made me who I am.

Finally, I would like to thank my coworker Quentin for working with me for this report and bearing me during a whole year. You have been truly complementary to me.

Adrien

First of all, I would like to give my deepest thanks to our supervisor Jean-Charles Delvenne for his availability, his devotion and his valuable advice throughout this entire year. This work would not have been the same without his help.

My sincere thanks go to all those who contributed to improve this document through readings, reviewing, discussions and comments.

I would also like to thank all those who have helped me indirectly. I especially think about my family and my roommates for their encouragements and thanks to whom I could work in excellent conditions.

And the last but not least, I would like to give my best thanks to Adrien for this year we spent together. Working in team is not always easy but it has been a pleasure to work with you. Thank for your good ideas, your motivation and your cheerfulness.

Quentin

Contents

Chapter 1

Introduction

"All of us are migrants to this world for a few days!"

– Kandathil Sebastian, *Dolmens in the Blue Mountain*

Studying human migrations is not something new. It has been deeply studied through a large number of researches such as anthropology, history or economics. It is legitimate to wonder why such studies are useful and interesting. We can find several reasons of this interest. Firstly, diverse studies and observations have shown that international migrations have an impact on various fields like economics[151], education[60], health[98], culture[150] or labour market[137]. Therefore, if someone wants to study such fields, migrations are an important factor to take into account. Furthermore, for the last decades the migrations have become bigger and bigger. For instance, we had in 1990 a migration flow of 154.2 millions people and a flow of 231.5 millions in 2013 [123]. And because of other socio-economics factors, this trend is likely to have more importance [63]. There is therefore a real need to have efficient methods to analyse migrations.

One way to do it is to use a network perspective [63]. Concretely, countries are represented by nodes and the edges correspond to the migration flow between two countries. This gives thereby a weighted directed graph. Using a graph theory approach to model large and complex networks is not something new [145]. There are numerous examples of concrete applications : the Internet [64], social networks [28], food webs [59] and many more. It is also used to study human-mobility problems [49]. Given that large graphs can be used to model such problems, it would be interesting to apply it to migrations [63]. However, among all the methods and algorithms related to graph theory, few of them seem to have already been considered and used for migrations.

The main goal of this master's thesis is to present a global analysis of migrations by adopting a network perspective. We will do it through three different aspects :

1. The ranking of countries.
2. The detection of cohesive groups among countries.
3. The prediction of migrations.

Each of them will be explained with more details in the corresponding chapters. This master thesis is organised as follows :

- The Chapter 2 explains how we obtained the information used for our analysis.

- The Chapter 3 describes and compares several ranking methods, including PageRank [126], a tool initially conceived for ranking webpages.
- The Chapter 4 describes and compares various group detection methods by using a graph theory perspective.
- The Chapter 5 finally presents innovative gravity models by using the results of the previous chapter instead of the factors commonly used in the literature like GDP per capita [13].

Moreover, in the three last chapters, we describe on the one hand what are our contributions on the topic and on the other hand why the methods we use are relevant and interesting in the field of migrations.

Chapter 2

Database elaboration

"If you have built castles in the air, your work need not be lost; that is where they should be. Now put the foundations under them."

– Henry David Thoreau, *Walden*

2.1 Motivation

As previously mentioned, many aspects are involved in the study of migration networks, one of which is the identification of the reasons that can push people to move. The economic situation of the countries, the standard of living, the languages used, the distance, the past, etc. All these factors have a certain impact on the size and direction of migrations. That is why we need to consider them in the task of elaborating a gravity model. But in practice, it is not a trivial task. Indeed, before all analysis we need to find a data set related to these factors. Until now, there is no open database where we can have a full listing of this information.

That is why we have to do a work of data retrieval : exploring diverse sources and gathering the data found in order to design a database with relevant information concerning the countries and the migrations. The interest is twofold. On the one hand, all the useful information are centralised and can be directly used for the analysis that will greatly simplify our work. And on the other hand, the database could be used afterwards by other people interested in the topic.

The purpose of this section is to explain the process of data retrieval, with the issues encountered and how we manage to overcome them.

2.2 Relevant data

The most used data in all our work is the recap of migration flows. Doing such a recap is not trivial and includes a lot of issues. For instance the Development Research Group of the Trade in Integration Team elaborated a global origin-destination migration matrix for each decade from 1960 to 2000 [156] which is available via the World bank[162]. They explain how they built these matrices and what were the difficulties encountered :

- Data can be missing for some countries.
- The definition of migrant can vary depending on the country.

- The data recording can differ greatly depending on the destination country.
- The data concerning a migration flow from a country to another is typically collected by the destination country.

This is due to the difficulty of the source country to collect data about people who are not living in the country anymore. The consequence is that the quality of migration statistics depends chiefly on the rigour with which destination country records the immigration flow. For some countries it can be a problem.

In this report, we took two data sets, one from the World Bank [162] for the year 1960, 1970, 1980, 1990 and 2000 and the other one from F.Docquier [55] for 1990 and 2000.

Regarding the factors introduced previously, the first step is to identify which one will be used for elaborating the gravity models. Although this task may seem trivial, it presents some difficulties. At this step of the work, we don't know if a factor is relevant or not. To overcome this issue, we take factors commonly used on the state of the art models of migrations as well [13]. At this list, we add two others factor, the PageRank and the community, which have until now never in the gravity models for migrations. The following list summarizes all the factors considered with the source of the corresponding data :

- **Population** : Number of inhabitants in the considered country [163].
- **Coordinates** : Longitude and latitude of the central point of the country [124].
- **Distance** : The distance between two countries which we compute from the coordinates of the countries. More specifically, we computed the spherical distance[89] :

$$\texttt{dist(A,B)} = \left| r \times \arccos\Big(\sin(\texttt{lat(A)})\sin(\texttt{lat(B)}) + \cos(\texttt{lat(A)})\cos(\texttt{lat(B)})\cos(\texttt{long(B)} - \texttt{long(A)}) \Big) \right|$$

  with `dist(A,B)` the distance between the country A and B, r the Earth radius[1] and the longitude/latitude in radian.
- **Languages** : The set of the languages commonly spoken in the country [1].
- **Root languages** : The root of the languages spoken in the country. For example, *Indo-European* is the root of *French* or *Portuguese*. We considered 23 different roots. If we consider the tree of languages according to their evolution, we take languages located on the top of the tree. The root chosen are presented in the Appendix D.1. We make this choice for a specific reason. What we want with the root language is to have a global linguistic factor. The current spoken language in a country may already act as local linguistic factor and thus, the more specific are the roots the more we tend to have a local factor which can be redundant with the current language. That is why we consider the most global roots.
- **English speaker** : Boolean set at true if English is spoken in the country. Obtained by own calculation from the languages spoken. The intuition behind this factor is that English is an international language and therefore favorable for the migrations.
- **GDP** : The Gross Domestic Product per inhabitants in the country [99] [161].
- **Borders** : The set of countries sharing a common land border with the country. Being neighbor can have an effect on the migrations [1].

[1] We took 6371 km.

- **Area** : The area in km^2 of the country [1].
- **Density** : The density in $\frac{nb\_hab}{km^2}$ of the country computed from population and area.
- **HDI** : The Human development index of the country [71]. This index is obtained by three factors [154]:
  1. The life expectancy of the country.
  2. The knowledge in the country. It is defined by the mean years of education[2].
  3. The standard of living of the country. It is defined by the Gross National Income (GNI) which is the GDP plus net receipts of primary income coming from abroad[74].

  The higher are these values, the bigger the HDI is. The intuition behind this parameter is that countries with a high HDI are more popular for immigration.
- **Death rate** : The death rate in the country [159].
- **Birth rate** : The birth rate in the country [158].
- **PageRank** : The PageRank of the countries for the year considered. The way to compute it will be explained in Section 3.6.
- **Community** : The community of the countries for the year considered. The way to compute it will be explained in Section 4.5.

2.3 Data retrieval

For each of the presented factors, we have to collect the corresponding data for each countries and for the years considered. Here begins the work of data retrieval. This task implies some issues :

1. Data can be missing.
2. Same data can have different names.
3. Collected data can be wrong or useless.

For all these reasons, before inserting the data in our database, we have to preprocess them. We do this work in three phases : the normalisation, the completion of missing values and the filtering.

2.3.1 Normalisation

As we said, the data collected come from diverse sources. Each of the sources have their own particular conventions to represent the data. At first sight the shape of the data is heterogeneous. The goal of the normalisation is to give a common shape to the data in order to efficiently classify them. Concretely, we make several things :

- Reducing the country names to an unique code. Depending on the source, the name of a country can have different spellings. For example, *Vietnam* can be written as *Viet-nam* too. Letting two different names for the same data is a concern for an automated processing. Indeed *Vietnam* and *Viet-nam* would be considered as two different countries. Sometimes

[2] And the expected mean.

the differences were bigger such as the Republic of the Union of Myanmar which was written Myanmar in some sources and Burma in other. To overcome this issue, we use two codes, the *ISO-3166 alpha2* and the *ISO 3166-1 alpha 3*, both with the purpose of determining a country in a unique way. Every time we encounter a country name, we take the equivalent ISO code.

- Reducing the different dialects to the main language. For example, *fr-BE*, the French language spoken in Belgium, and *fr-FR*, the French language spoken in France, are considered as two different languages. For our analysis, it is more relevant to regroup them to the root *fr*. For this work, we use regular expression to cut the dialect part.
- Harmonising the number to the English format. Some sources use different conventions.

After doing it, the normalised data can be easily used.

2.3.2 Missing values

Another issue is the lack of data. Sometimes values are not provided by the main sources. We use two ways to fill the values :

1. Exploring secondary sources to complete the holes.
2. Making an interpolation with the known data to obtain the missing ones. However, this method is dangerous. First, it is only based on the values, not on the situation of the country. For example, we observed that the population of several countries which belonged to USSR has decreased between the year 1990 and 2000 while the general trend about the population is to grow up. This decrease can be explained by the dissolution of the Soviet Union occurred in 1991. And secondly, if there are too many missing values, the interpolation can be very bad. In this case, we prefer to let a hole in our database instead of having bad values.

This task needs hand processing. Fortunately, missing data represents only a small percentage (less than 5%) of the whole set. Such data are mainly related to the GDP, the HDI and the population of small countries (Andorra, Lichtenstein, Monaco, etc.) and of islands countries (Falkland Islands, Niue, Tokelau, etc.). About the migrations, some data are missing too, but given that the migration data set is much larger than the others [3], a missing value has a negligible impact comparing to the other parameters.

2.3.3 Filtering values

Before using the data collected, it is necessary to check the consistency of the values. Given that our analysis will be based on these data, we want to minimise the risks of wrong values. That is why filtering data by removing incoherent values is necessary. Concretely we remove countries containing outliers values. Among these countries, there are many small islands, such as the *Channel islands*.

2.4 Database structure

With all the data collected and preprocessed, we are now able to create the database. We make two kinds of tables :

[3] for n countries, we have $n \times n$ different migrations per year.

- The tables related to the countries and containing only the information about them. We have information about 249 countries.
- The tables related to the migrations and the different relations between the countries. For 249 countries, this table has a maximum size of 249×249 entries.

These pieces of information are taken for the year 1990 and 2000.

Chapter 3

Ranking analysis

"A simple way to take measure of a country is to look at how many want in.. And how many want out."

– Tony Blair

3.1 Motivation

The ranking problem consists in finding an ordering function which assigns a score to each node of the graph [9]. In other words, the aim is to order the nodes according to one or several criteria. However, this task cannot be reduced to a unique algorithm or method to compute it. Indeed, it is highly dependent of the nature of the graph and of the ordering criteria. For this reason numerous ranking algorithms exist, each having their own specificity.

A first way is to use the topological ordering for acyclic directed graph [76]. This ordering gives each node of the graph a numeric label such that if we follow an edge of the graph, the label of the destination node will always be superior or equal to the label of the origin node. It is especially used in scheduling problems [67].

However the topological ordering cannot be used for directed graph without the acyclicity condition. A solution is to prune the fewest number of edges in order to make the graph acyclic. This process is called the minimum feedback arc set problem and is unfortunately NP-Hard [39]. Moreover, having an approximation better than an order 1.36 is also NP-hard [54].

Among a directed graph, we can also identify tournaments which are directed graph where each pair of nodes is connected to exactly one of the two possible directed edges [95]. There are also several ranking algorithms for such situations. Kenyon-Mathieu and Schudy [95] present a solution using a polynomial time approximation scheme for weighted tournaments. Coppersmith, Fleischer and Rurda [44] envisage another approach to solve this problem.

Another family of directed graphs is the one exhibiting a hierarchical structure. In other words the nodes of the graph will have a relative level against the other nodes. Luo and Magee [109] consider two interesting kinds of hierarchy in the context of network analysis :

- **The flow hierarchy**, which is determined by the directed flow between the nodes. In the best case, a node will be above another if there is an edge directed from the later to the

former which is a kind of topological ordering where all the nodes at the same level have the same label.

- **The containment hierarchy**, where the nodes are divided into groups that are further divided into subgroups and recursively over multiple levels.

Focusing on the concept of flow hierarchy Luo and Magee [109] designed several metrics to measure the hierarchical character of a directed graph. Such piece of information can be used to select a particular ranking algorithm. For example the QuickRank algorithm [77] is particularly efficient for networks having a hierarchical structure.

Up to now, all the ranking algorithms presented above were designed in order to match a particular nature of the graph considered[1]. However, there exist also algorithms which were developed for networks having a specific function. It is the case of the ranking algorithm [57] which is intended for social communication systems like social networks.

Another deeply studied network is the Web. Indeed, this network can be represented by a directed graph where the nodes are webpages and where the edges correspond to the hyper links between the pages [32]. The task of ranking the webpages has been initiated by Larry Page and Sergey Brin, the two co-founders of Google [126]. Their goal was to develop an algorithm to sort all the webpages related to the query of a user by giving a score to each page. They called this score the PageRank [40].

Concerning migrations themselves, the commonest way to rank countries is to consider only the number of migrants [153] or the number of migrants normalised by the population [38]. To the best of our knowledge, there is not any source or reference which considers a network approach for ranking countries[2]. It is why a main aspect of our work is to select some ranking algorithms, adapt them when necessary in order to apply it to the migration network.

The purpose of this chapter is to present how we adapt and apply the ranking methods in the field of the migrations. From the resulting analysis we will detect what are the pros, the cons and limits of these methods.

We base our work on the migration data of the World Bank [169] which contains about 17 million migrations that has been made between 1990 and 2000. This represents about 740 000 migrations by country, including islands and dependent states.

3.2 In/Out Degrees

The simplest way to rank the countries consists of only considering the number of immigrants for each country. According to this criterion, the most attractive countries will be those having the highest number of immigrants. In practice, this can be done by calculating and comparing the weighted in-degree of each countries. For the majority of the methods detailed on this chapter, we represent the results in two ways :

1. A global view showing the score of every country on a world map.
2. A detailed view taking up the countries having best score and sometimes the ones having the worst score.

[1] Simple, directed, weighted, etc.
[2] With exception of an article in preparation by de Montjoye and Rocher [50].

We obtain thus the Map 3.1 and the Ranking 3.1. All the maps and the results we obtained are based on the the World Bank's data of migrations [162] occurred from 1990 to 2000 unless otherwise mentioned.

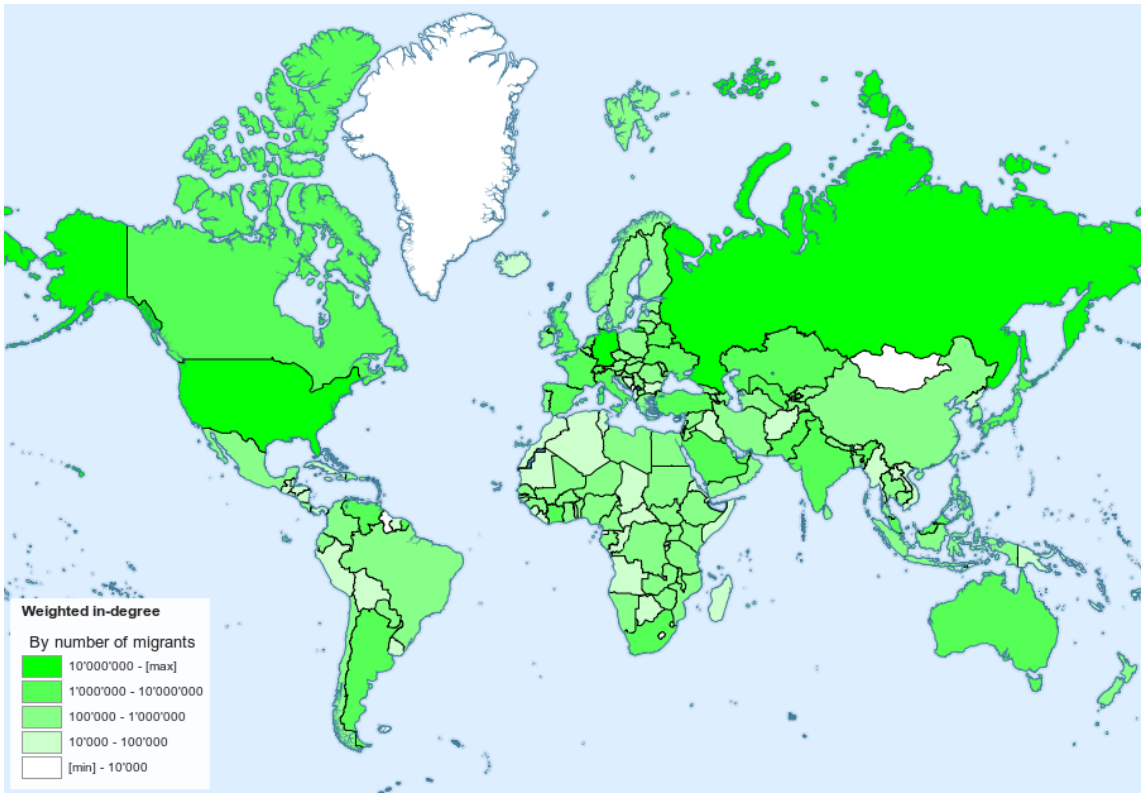


Figure 3.1: Map of weighted in-degree.

Ranking	Country
1	United States
2	Russian Federation
3	Germany
4	France
5	India
6	Canada
7	Ukraine
8	Saudi Arabia
9	United Kingdom
10	Australia

Table 3.1: Top ten of most attractive countries according to their weighted in-degree.

A direct comment about these results is that all the countries appearing on the top have a high population. From the Ranking 3.1 Australia is the country with the smallest population among all the countries above with 19 million peoples (See Appendix D.2.6). Similarly, we can do another ranking by considering the number of emigrants which will give an intuition of the most repellent countries. Here, we will use the weighted out-degree of each countries, which yields the Map 3.2 and the Ranking 3.2.

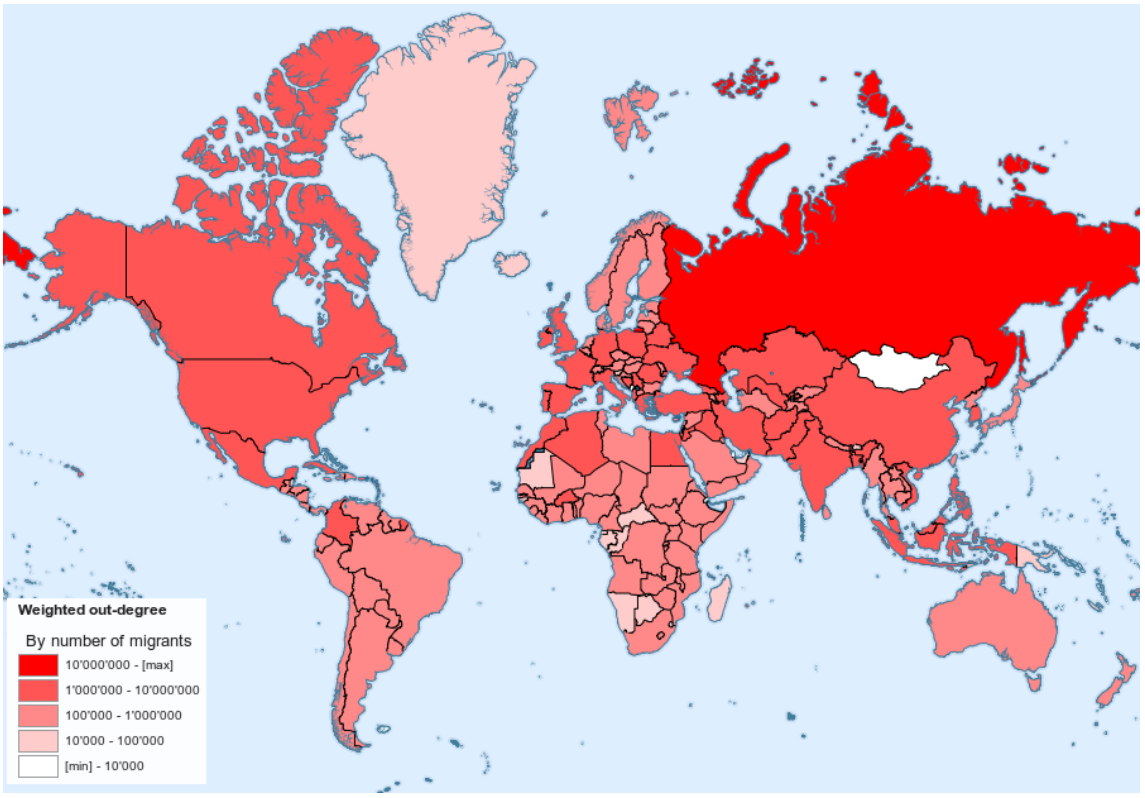


Figure 3.2: Map of weighted out-degree.

Ranking	Country
1	Russian Federation
2	Mexico
3	India
4	Ukraine
5	China
6	Poland
7	Bangladesh
8	United Kingdom
9	Pakistan
10	Germany

Table 3.2: Top ten of most repellent countries according to their weighted out-degree.

Once again, we observe that only populated countries appear in the ranking (Poland is the least populated in the top 10 with 38'654'000 people D.2.70). This brings us to the first issue of the simple weighted degree method : the most populated countries are too advantaged compared to the least populated. Indeed, a low populated country, like the Qatar with 640'000 people (D.2.72) could never send as much people as countries like India. It is why such countries will never appear in such rankings.

Furthermore, we can notice that some countries appear in both rankings. We can wonder if they are attractive, repulsive or both. We do not have enough information to tell it only with the weighted degree method and we do not want to base on conclusions on this single ranking. Because of these two issues, we need to elaborate more sophisticated methods to obtain a more

refined ranking.

3.3 In/Out degree divided by the population

This second method consists in dividing the number of immigrants, or similarly the emigrants, by the population of their home country. As you can guess, the purpose of this method is to overcome the issue raised by the first method about the population. Indeed, by normalising like that the migration flows, we can really see what is the proportion of migrants for each country. Afterward, we obtain the Map 3.3 and the Ranking 3.3.

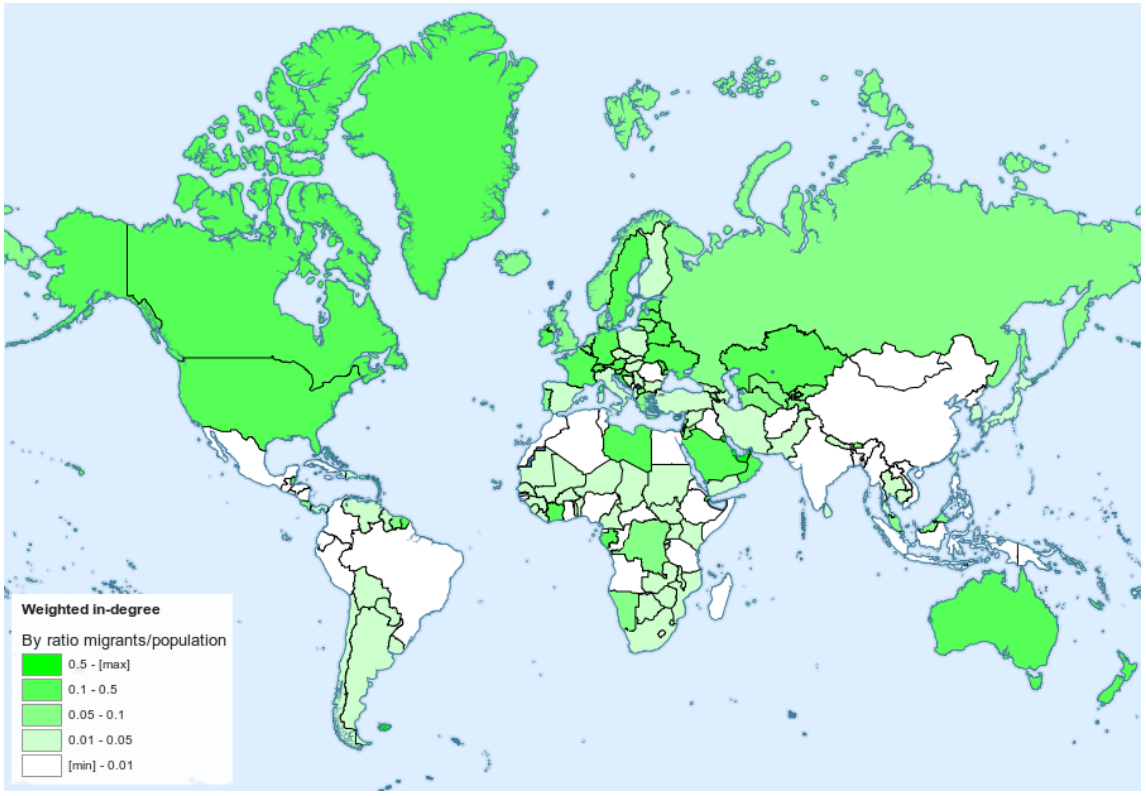


Figure 3.3: Map of weighted in-degree divided by the population

Ranking	Country
1	Kuwait
2	Qatar
3	United Arab Emirates
4	Monaco
5	Andorra
6	Nothern Mariana Islands
7	Cayman Islands
8	Macao SAR, China
9	Falkland Islands
10	Virgin Islands (US)

Table 3.3: Top ten of most attractive countries according to their weighted in-degree divided by the population.

The results are different from what we obtained previously. Firstly, no country of the Ranking 3.1 is included in the new one. We manage to find explanations of why these countries are present on this new ranking. For all of them, their presence in the ranking is not fortuitous. We split the countries into the different reasons :

- Some of these countries are oil producing countries. It is the case of Kuwait, Qatar and United Arab Emirates (See Appendix D.2 for specific values of GDP percentages).
- Some of these countries are offshore financial centers D.2. It is the case of United Arab Emirates, Monaco, Andorra, Cayman Islands, Macao SAR.
- Some of these countries are tax haven countries D.2. It is the case of Monaco, Andorra, Bahrain, Cayman Islands, Macao SAR and Virgin Islands (U.S).
- Some of these countries have a very high (top 20 in the World) GDP per capita (Purchasing Power Parity) [83] : Kuwait, Qatar, United Arab Emirates, Monaco, Cayman Islands, Falkland Islands.

Concerning Northern Mariana Islands, this country is said to be a tax haven and a heaven for tourists [75] but its presence on this top ranking could also come from data errors as we don't have much information about this country. For all these reasons, and probably others, people tend to come to these countries.

Besides, all the countries obtained are also low populated. The highest population is in the United Arab Emirates which is less than 3.2 million (D.2.87). This idea is confirmed too by looking the map which is globally lighter than the previous one which means that the most populated countries (generally the bigger on the map) lose their importance. Similarly, we can find the most repellent countries, yielding the Map 3.4 and the Ranking 3.4.

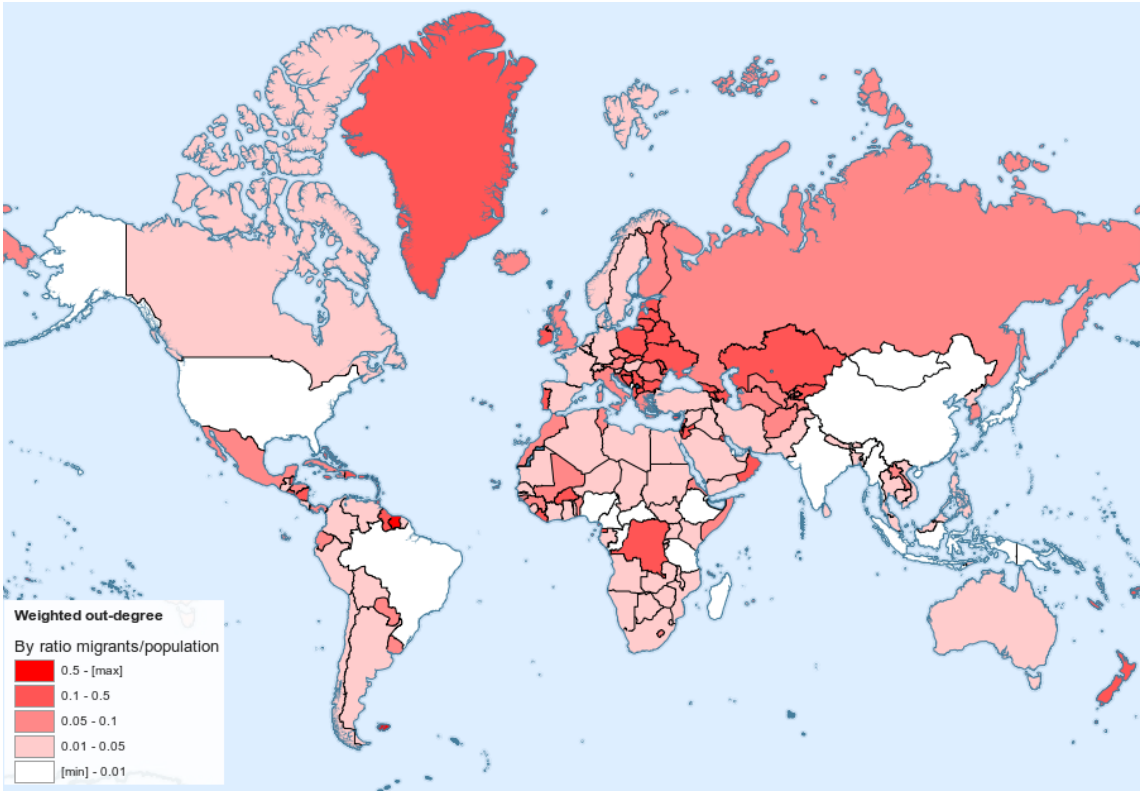


Figure 3.4: Map of weighted out-degree divided by the population.

Ranking	Country
1	Niue
2	Tokelau
3	Montserrat
4	Cook Islands
5	Palau
6	Virgin Islands (U.S.)
7	St. Kitts and Nevis
8	Grenada
9	Guadeloupe
10	Antigua and Barbuda

Table 3.4: Top ten of most repellent countries according to weighted out-degree divided by the population.

Here we obtain in the ranking only small islands and dependent states. Among them, Guadeloupe is the most populated with only less than 500'000 people D.2.34. We are not interested in these countries where the low population has a negligible effect on the normalisation comparing with the populated countries. Again, when we compare the Maps 3.2 and 3.4 we can see that in the second one the populated countries lose their importance while less populated have a higher rank. It confirms our idea about the previous ranking which gives too much importance to the populated countries.

However, it raises another issue about our new method. The lowly populated countries are now advantaged. One more time we have to find another method. The main idea that we want to show is that if a country attracts a lot of people and does not repel many, this country is attractive.

The notion of attractiveness is ambiguous. Indeed, this concept can refer to a particular portion of the population. For example, there exist countries that are only attractive for low skilled people and repulsive for high skilled people (see the Jamaica case D.2.42). However it is not what we want to study here, we focus on the migrations with a global point of view.

3.4 Ratio of migrations

Another way of not taking the population into account is to elaborate a ranking according to the ratio of immigrations and emigrations. Mathematically, it is expressed like this :

$$\text{ratio} = \frac{\text{in-degree}}{\text{out-degree}}.$$

With this ranking, we obtain the Map 3.5 and the Ranking 3.5 :

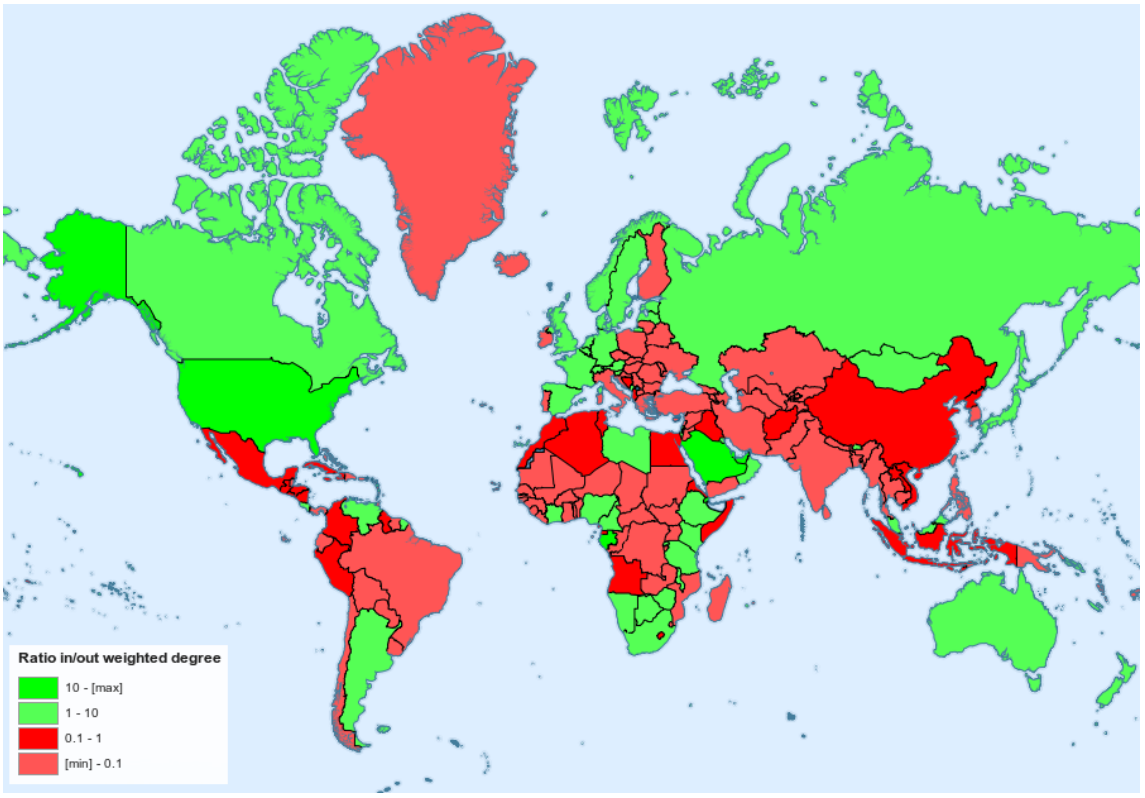


Figure 3.5: Map of migration ratio

The ranking would be :

Ranking	Country
1	Qatar
2	Mayotte
3	United Arab Emirates
4	Saudi Arabia
5	Djibouti
6	United States
7	Cayman Islands
8	Gabon
9	French Guiana
10	Andorra

Table 3.5: Top ten of most attractive countries according to their ratio $\frac{\text{in-degree}}{\text{out-degree}}$.

Several countries on top on this ranking were also on top of the previous ranking (Figure 3.3) such as Qatar, United Arab Emirates, Andorra, Cayman Islands. Besides, there are other countries in it. As previously, let us see if their presence is plausible :

- United StatesD.2.89 : Having the 2^{nd} HDI in the world in 1990, the 3^{rd} in 2000 and being the 1^{st} in the ranking of GDP in 1990 and 2000 it is natural to find United States in a ranking of attractiveness.
- DjiboutiD.2.21 : Djibouti is an excellent local attractor because of its strategic position near the Red Sea and the Indian Ocean. This results in a large immigration from close countries

(about 80'000 immigrations between 1990 and 2000 against less than 2'000 emigrations to those same countries).

- GabonD.2.28 : Having the 2^{nd} HDI in Africa in 1990 and the 3^{rd} HDI in 2000, the Gabon is a good local attractor. Between 1990 and 2000, we count more than 150'000 migrations from close countries to Gabon against no more than 10'000 emigrations to those countries.
- Saudi ArabiaD.2.76 : Saudi Arabia is one of the largest oil producer in the world [84].
- MayotteD.2.55: Mayotte is very close to the Comores (70 km), which is one of the poorest island in the world. As Mayotte is an overseas department of France, it is very attractive for people from the Comores [155].
- French GuianaD.2.27 : This overseas department of France attracts many French people since the mid-1980's [127]. Furthermore the country is a door to the EU for the neighbouring countries, which makes it a local attractor.

As we saw, we have on this top several countries like Mayotte, French Guiana, Gabon or Djibouti which are good attractors only locally. However, our goal is to have a ranking of more global attractors. To do so, a solution is to take the importance of the countries of the incoming people into account.

Before that, we can also look at the ten least attractive countries according to the same ratio :

Ranking	Country
1	Cuba
2	Iraq
3	Guyana
4	Vietnam
5	Jamaica
6	Eritrea
7	Haiti
8	El Salvador
9	Morocco
10	Bosnia and Herzegovina

Table 3.6: Top ten of least attractive countries according to their ratio $\frac{\text{in-degree}}{\text{out-degree}}$.

Again, we analyse each country separately to find one major reason that can help to explain the leak of people in those countries[3] :

- Cuba has known an economic crisis D.2.20.
- Civil War : El SalvadorD.2.22, Eritrea D.2.23, Haiti D.2.36.
- War : Iraq D.2.38, Bosnia and Herzegovina D.2.13.
- Post-War : Vietnam D.2.91.
- Diasporas : Important emigrations of foreign people from Guyana D.2.35 and Morocco D.2.60 due to the attractiveness of another countries for these people.
- Brain drain : Important emigrations of educated people from Jamaica D.2.42.

Through this ranking, we can see that migrations are affected by such events.

[3]More explanations about the events are presented in the appendices related to countries.

3.5 Eigenvector centrality

We stated previously the interest of taking into account the importance of the countries from where the people come from. One way to do it is to rank the countries according to the eigenvector centrality. We can define the importance of a country in a recursive way. If a country receives people from an important country it will gain more importance than if the country is less important.

3.5.1 Introduction

The eigenvector centrality is based on two concepts, the power and the centrality. Both notions are commonly studied in the field of social networks [79]. The following example illustrates these concepts in order to exhibit the difference between them.

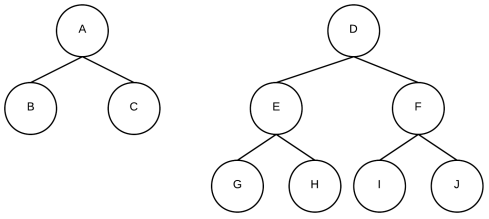


Figure 3.6: Example to distinguish power and centrality.

In the Figure 3.6 that we can interpret as a graph of connections between people, A and D have both two friends. In term of influence, D should be more important than A because the friends of D have friends as well. That is the concept of centrality. However, A should have more ease to convince his friends to do something, as they do not have any other friends. That is the concept of power.

However until 1983 no difference was made between the power and the centrality [26]. To overcome this lack, Philipp Bonacich has developed a new measure, called the eigenvector value which depends on a parameter to compute either the centrality or the power of a node in a graph [26] which can also be directed [27].

3.5.2 Mathematical concept

The main idea about the eigenvector centrality[120] is thereby to take importance of the nodes into account. Indeed, not all connections are equal. In general, connections to countries which are themselves influential will give to a country more influence than connections coming from less influential countries. Mathematically we have :

$$x_i = \frac{1}{\lambda} \sum_{j=1}^{n} A_{ij} x_j$$

where x_i is the centrality of vertex i and λ is a constant.

Using the Perron-Frobenius theorem[30], it can be shown that if we want the eigenvector to be non-negative, λ must be the largest eigenvalue of the adjacency matrix and x the corresponding eigenvector.

3.5.3 Application

Computing the eigenvector centrality to our problem gives us the following results :

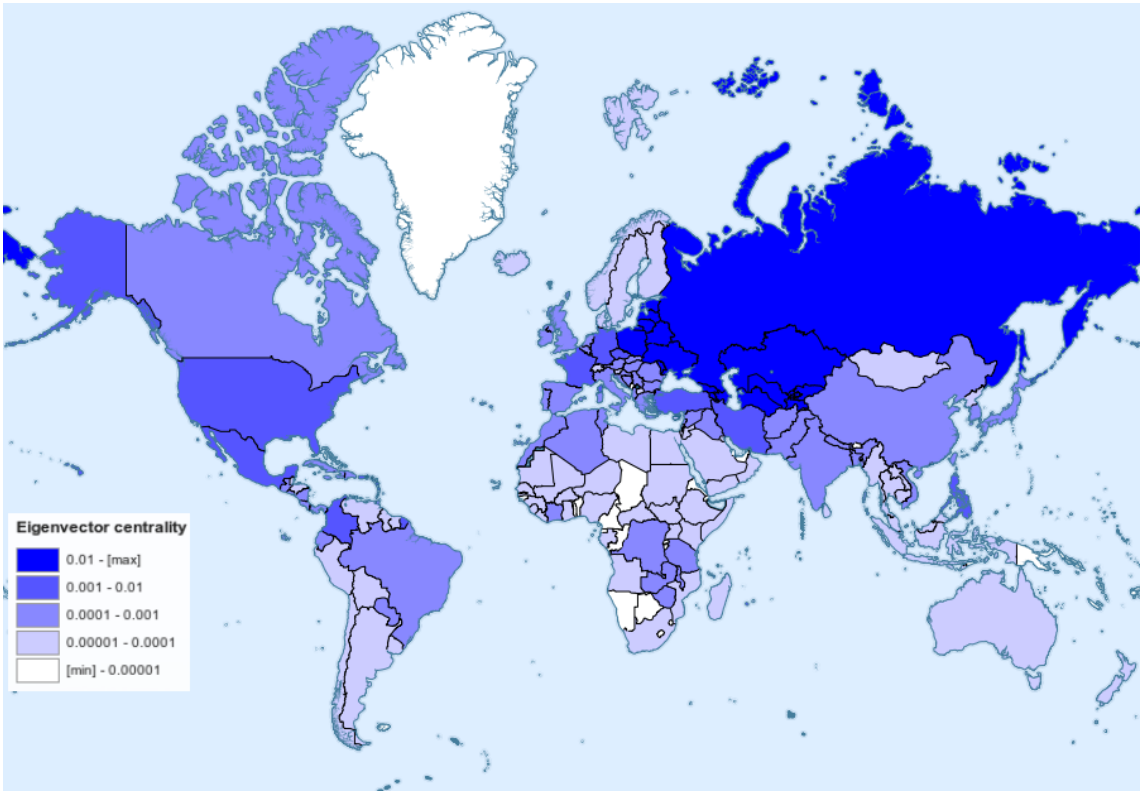


Figure 3.7: Map of eigenvector centrality measure.

Ranking	Country
1	Russian Federation
2	Ukraine
3	Kazakhstan
4	Belarus
5	Uzbekistan
6	Azerbaijan
7	Poland
8	Georgia
9	Armenia
10	Kyrgyz Republic

Table 3.7: Top ten of most attractive countries according to their eigenvector centrality.

We saw in Section 3.2 that Russia and Ukraine had together an important number of immigrations and emigrations. They both exchanged about 3.6 million people to each other which results in giving each other a big importance. In a smaller scale, the same fact happens between all the countries in the former USSR (see also Sections 4.3 and 4.5). As the eigenvector centrality takes the importance into account, all of them appear to be in the top of this ranking.

However this metric has some issues in our context. Indeed, according to the concept of eigen-

vector centrality, the more connections there are in the local network, the more central a country is [114]. The countries from the former USSR are really tight [4] and for this reason, the local effect makes them important to the detriment of the countries globally attractive. We want to avoid such situation in our ranking. That is why we analyse another method called the PageRank which is known to be more global.

3.6 PageRank

3.6.1 Introduction

The PageRank was invented by Larry Page and Sergey Brin in 1996, the two co-founders of Google. Their goal was to develop an algorithm to sort all the webpages related to the query of a user [126].

3.6.2 Mathematical concept

The idea of the PageRank[40] is related to the random walk. Let us consider a random person who moves out from country to country following the flow of people and who sometimes decides to go to any country randomly. Given that we consider weighted flow, if the number of migrants going from a country to another is high, the probability that this random person will follow this flow will be high too. By assuming that this person moves out an infinity of times, we can define the PageRank of a country as the proportion of times this random person has been in this country. This is also referred as the stationary probability.

As the eigenvector centrality, the PageRank has the idea that all the connections are not equal. The more important is a country a migrant comes from, the more importance it gives to the country he goes to. The big difference between these two concepts is that the PageRank does not take the population into account but only the proportion of migrants of the different countries.

Mathematically, to compute the PageRank we need to follow these first steps :

- Transform the adjacency matrix into a stochastic matrix.
- Add the probability to go to any countries.
- Find the eigenvector related to the highest eigenvalue λ.

Using the Perron-Frobenius theorem [30], it can be shown that λ is equal to 1 and is unique. The left eigenvector related to this eigenvalue is the PageRank. According to Perron-Frobenius, the PageRank always exists and as mentioned above, it is equal to the stationary probability.

3.6.3 Obtaining the stochastic matrix

As said above, the first step is to transform the adjacency matrix into a stochastic matrix :

$$H_{ij} = \frac{A_{ij}}{\sum_{j=1}^{N} A_{ij}}$$

[4] It will be highlighted later on the community Section 4.5.

where N is the number of countries and A is the adjacency matrix. H is thereby a matrix where every row sums up to 1.

However, for the sake of robustness, we need to make some changes on H. Indeed, if one country is not connected to any others[5], the country becomes a dangling node in the graph of migrations and the row of H corresponding to this country will be entirely full of zeros which can generate a problem in our calculation.

To avoid this problem for such situations, the method commonly used [40] is to add a uniform connection to all nodes :

$$\mathbf{\hat{H}} = \mathbf{H} + \frac{1}{N}(\mathbf{w_{N\times 1}} \mathbf{1_{1\times N}})$$

with $w_i = 1$ if row i is full of zeros and 0 otherwise. The intuition behind this modification is that, if a country is isolated, it will have the same probability to go to any country.

Moreover, we also want the random person to sometimes move out to any country randomly. For that purpose, we need to make another modification in the matrix :

$$\mathbf{G} = \theta\ \mathbf{\hat{H}} + (1-\theta)\frac{1}{N}\mathbf{1_{N\times N}}$$

where $\theta \in [0, 1]$ is the proportion of times the random walker follows the flow and $1 - \theta$ is the proportion where the random walker moves out randomly to any country. The purpose of the term $\frac{1}{N}\mathbf{1_{N\times 1}}\mathbf{1_{1\times N}}$ is to have a matrix linking every country to all other countries with the same probability.

Finally, we can use the matrix G to compute the PageRank. As previously explained, the PageRank correspond to the left eigenvector and the eigenvalue equals to 1. We have thereby the following equation :

$$\mathbf{G}^T\ \pi = \pi \tag{3.1}$$

where π is the PageRank. If we want to normalise the solution, we can add the constraint $\sum_i \pi_i = 1$.

Now, the issue is to solve this equation.

3.6.4 Methods to compute the PageRank

Different methods could be applied to solve the PageRank equation. First of all, we present three of them in order to select afterward the most efficient [93].

Gauss-Seidel

The Gauss-Seidel method is an iterative numerical method used to solve system of linear equations as $\mathbf{Ax} = \mathbf{b}$. The idea of this algorithm is to decompose the matrix $\mathbf{A}$ into a lower triangular component $\mathbf{L}$, the diagonal component $\mathbf{D}$ and a upper triangular component $\mathbf{U}$. We have thereby the following decomposition :

[5] It can be due to data errors for instance.

$$\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U}.$$

The iteration followed by the method is

$$(\mathbf{L} + \mathbf{D})\mathbf{x}_{k+1} = \mathbf{b} - \mathbf{U}\mathbf{x}_k$$

which is equivalent to compute with the fixed point method

$$x_{k+1} = (\mathbf{L} + \mathbf{D})^{-1}(b - \mathbf{U}\mathbf{x_k}).$$

However, our problem is more specific than the general case $\mathbf{A}\mathbf{x} = \mathbf{b}$. Indeed, we want to solve $\mathbf{G}^T\ \pi = \pi$ which is equivalent to solve

$$(\mathbf{G}^T - \mathbf{I})\ \pi = 0$$
$$\sum_i \pi_i = 1$$

To ensure the convergence of the method, one way is to have the spectral radius $\rho((\mathbf{L} + \mathbf{D})^{-1}(-\mathbf{U})) < 1$ [15]. This is true if $\mathbf{A}$ is strictly diagonally dominant or if $\mathbf{A}$ is diagonally dominant and irreducible [15]. A necessary and sufficient condition of irreducibility for a graph is that all the nodes are strongly connected, which is the case in our problem. It is obvious that $\mathbf{A}$ is also diagonally dominant due to its construction.

Jacobi

The Jacobi method is another iterative numerical method to solve problems of type $\mathbf{A}\mathbf{x} = \mathbf{b}$. It is a generalisation of the Gauss-Seidel method. Instead of decomposing $\mathbf{A} = \mathbf{L} + \mathbf{D} + \mathbf{U}$, we take the following decomposition

$$\mathbf{A} = \mathbf{M} - \mathbf{N}$$

where M is an invertible matrix.

It is a generalisation because if we take $\mathbf{M} = \mathbf{L} + \mathbf{D}$ and $\mathbf{N} = -\mathbf{U}$, $\mathbf{M}$ is invertible as $\mathbf{L} + \mathbf{D}$ is a triangular matrix with a non-zero diagonal.
The iteration followed is

$$\mathbf{M}\mathbf{x}_{k+1} = \mathbf{N}\mathbf{x}_k + b.$$

The convergence is ensured with the same conditions as Gauss-Seidel's algorithm [15].

Power Method

The power method is an iterative numerical method to find the eigenvector related to the highest eigenvalue of a matrix with real coefficients. The idea is to take a random vector v_0 with non-zero elements and to do iteratively the following operation :

$$v_{n+1} = \frac{\mathbf{A}v_n}{\|\mathbf{A}v_n\|_2}.$$

If **A** has an eigenvalue that is strictly greater in magnitude than its other eigenvalues and if the starting vector v_0 has a non-zero component in the direction of an eigenvector associated with the dominant eigenvalue, v_n is ensured to converge to an eigenvector associated with the dominant eigenvalue.

In our situation, we know that the highest eigenvalue is 1 and is unique. So if we take a random vector v_0 with non-zero elements, the convergence is ensured. For this reason and because of the speed of the algorithm in practice, this is the method that Google uses [62] and that we use as well.

3.6.5 Application

Using the PageRank to characterise the countries, we obtain the Ranking 3.8 and the Map 3.8.

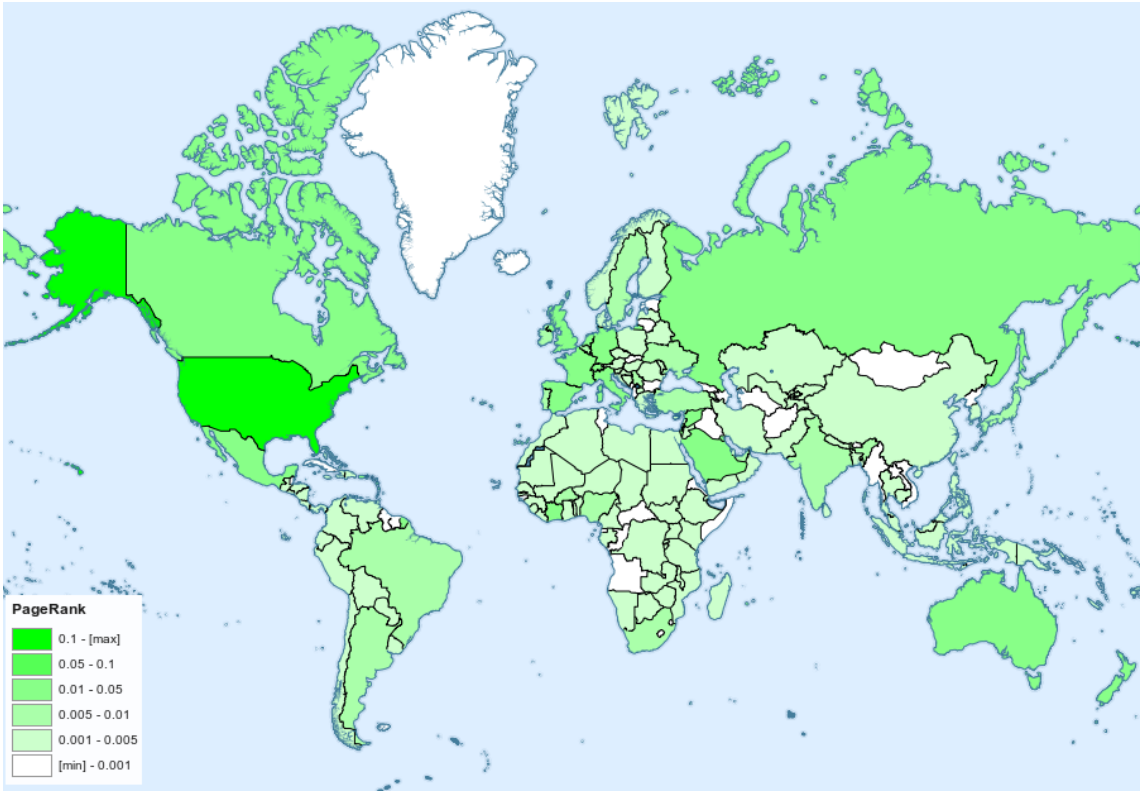


Figure 3.8: Map of PageRank in 2000.

Ranking	Country
1	United States
2	Canada
3	United Kingdom
4	Germany
5	Australia
6	France
7	West Bank and Gaza
8	Mexico
9	Puerto Rico
10	Saudi Arabia

Table 3.8: Top ten of most attractive countries according to their PageRank.

As for the previous ranking, we try to find a plausible reason concerning the presence of these countries on the top ranking. We identify severals reasons :

- High HDI : Australia (0.906, 2^{nd}), United States (0.897, 3^{rd}) , Canada (0.879, 6^{th}), Germany (0.864, 12^{th}), France (0.846, 18^{th}), United Kingdom (0.833, 22^{nd})
- High GDP : United States (1^{st}), Germany (3^{rd}), United Kingdom (4^{th}), France (5^{th}), Canada (8^{th}), Mexico (9^{th}), Australia (14^{th}), Saudi Arabia (23^{rd})
- Tax haven or Offshore Financial Center : Puerto Rico D.2.71.
- Oil producing country : Saudi Arabia D.2.76.
- Local attractor of oil producing countries[6] : West Bank and Gaza D.2.93.
- Major emigrations of the United States : Mexico (16%), Canada (12.4%), Puerto Rico (11.2%), United Kingdom (7.6%), Germany (5.3%), France (3.8%)

As the United States is by far the most attractive country according to the PageRank evaluation, the major emigrations coming from this country give to the destination country a good place in the ranking. Nevertheless, other attractive countries rise in top of the ranking for other reasons.

Furthermore, through the Map 3.8, we can also identify some local attractors[7] such as the United States, the Russian Federation, the Ivory Coast and the Saudi Arabia. We will come back on local attractors later (see Section 4.5).

3.6.6 Robustness of PageRank

Another aspect to consider is the sensibility of data errors. In other words, we wonder if some perturbations or modifications in our data will affect PageRank. Let us study this problematic. As we stated, the general problem is to solve

$$\mathbf{G}^T \pi = \pi$$

with

[6]This attractiveness is related to several events in Palestine such as the Madrid Conference of 1991 which lead to the Oslo I Accord in 1993. More details are available in Appendix D.2.93.

[7]Countries which are more attractive than the countries around.

$$\mathbf{G} = \theta\ \hat{\mathbf{H}} + (1-\theta)\frac{1}{N}\mathbf{1}\mathbf{1}^{\mathbf{T}}.$$

We would like to study the perturbed problem

$$\tilde{\mathbf{G}}^T\ \tilde{\pi} = \tilde{\pi}$$

with

$$\tilde{\mathbf{G}} = \theta\ (\hat{\mathbf{H}} + \mathbf{E}) + (1-\theta)\frac{1}{N}\mathbf{1}\mathbf{1}^{\mathbf{T}}$$

where E is the perturbation. This leads [93] to

$$\|\tilde{\pi} - \pi\|_1 \leq \frac{\theta}{1-\theta}\|E\|_\infty$$

or in other words

$$\sum_{i=1}^{N}|\tilde{\pi_i} - \pi_i| \leq \frac{\theta}{1-\theta}\max_{1\leq i\leq N}\sum_{j=1}^{N}|E_{ij}| \leq \frac{2\ \theta}{1-\theta}.$$

Indeed $\max\limits_{1\leq i\leq N}\sum\limits_{j=1}^{N}|E_{ij}| \leq 2$ because $\left(\hat{\mathbf{H}} + \mathbf{E}\right)$ has to be a stochastic matrix and $\hat{\mathbf{H}}$ is already a stochastic matrix. The value 2 is reached if every element which had a non-zero value becomes zero, which means that not any data is similar. We can also prove [93] that changing θ a bit will not alter the PageRank a lot either.

This gives us a theoretical bound for the sensitivity of the PageRank according to data errors but we would like to see what happens in practice. To do so, we collect the data from another source[8]. These data has been reviewed by several economists and are applied to only 190 countries instead of 227 for the data from the World Bank. The amount of migrations between these two databases is completely different. For instance, we have about 167 million migrations according to the World Bank data and only about 112 million migrations according to the other source. Before applying the PageRank on the other database, it is interesting to know how different the data would be. The Table 3.9 shows the main results.

Criterion	Data(World Bank) - data(Docquier)
Sum of the migrations	49'000'000
Max difference	3'050'000
Min difference	-952'000
Mean difference	1357
Variance	786'000'000
Standard deviation	28'000

Table 3.9: Difference between the data from the World Bank and F. Docquier.

It is interesting to see that the mean difference is low while the variance is huge, we can thereby expect big differences between the two datasets. Indeed, the standard deviation is more than twenty times bigger than the mean difference. We can also highlight that the average migration between two countries is around 3000 people for F.Docquier data and 4500 for World Bank data.

[8] François Docquier [13]

Now we present the results of the PageRank algorithm applied to the data from F. Docquier and compare it with the previous results :

Ranking	Country	Previous Ranking
1	United States	1
2	Canada	2
3	United Kingdom	3
4	Australia	5
5	Germany	4
6	Occupied Palestinian Territory	7
7	France	6
8	Russia	12
9	Spain	11
10	Saudi Arabia	10

Table 3.10: Comparison of the top ten of most attractive countries according to their PageRank for 2 different data sources.

Only two countries are in the top ten of the first ranking and not in the second one :

- Mexico which goes to 8^{th} position to the 18^{th}.
- Puerto Rico which is not in the 190 countries taken into account by the second data set.

This proves us that even with a big data variation, the PageRank is a reliable index, relatively insensitive to perturbations or errors. We can also verify if the theoretical bound is met or not. We had previously found that

$$\sum_{i=1}^{N} |\tilde{\pi}_i - \pi_i| \leq \frac{2\ \theta}{1-\theta}.$$

In our case, we have

$$\sum_{i=1}^{N} |\tilde{\pi}_i - \pi_i| = 0.22 \lll 2\frac{0.85}{0.15} \approx 11.33.$$

As we can see the sum of the variations of the PageRank is much lower than the theoretical bound. This is a really good news to justify the robustness of the PageRank. Furthermore, the average value of $|\tilde{\pi}_i - \pi_i|$ is thereby equal to 1.17×10^{-3}. This is an excellent result as the single change of the number of countries induces an average error by country of 3.4×10^{-4} for $|\tilde{\pi}_i - \pi_i|$.

3.6.7 Evolution of PageRank

We can also wonder if the PageRank varies a lot between two periods of time. Given that the PageRank is not sensitive to data variation from different database, we can hope that it will remain stable in this case too. Ideally, the PageRank will change only if there are major differences between two periods. To analyse it, we apply the algorithm to the data from the World Bank in 1990. As previously, we start by computing the main differences in the data :

Criterion	Data(World Bank 2000) - data(World Bank 1990)
Sum of the migrations	30'000'000
Max difference	4'700'000
Min difference	-1'600'000
Mean difference	580
Variance	724'000'000
Standard deviation	27'000

Table 3.11: Difference between data from the World Bank in 2000 and in 1990.

The content of this table is similar to the one of the Table 3.9 previously obtained. The Table 3.12 presents the differences of ranking between the year 1990 and 2000.

Ranking 1990	Country	Ranking 2000
1	United States	1
2	Canada	2
3	United Kingdom	3
4	Australia	5
5	Germany	4
6	France	6
7	West Bank and Gaza	7
8	Saudi Arabia	10
9	Puerto Rico	9
10	Mexico	8

Table 3.12: Comparison of countries according to their PageRank for the year 1900 and 2000.

All countries in the top ten in 1990 are also in the top ten in 2000, with only small variations. This shows us that the PageRank is stable and does not change a lot between two period of times which proves its stability.

3.7 Inverted PageRank

As the PageRank gives a relevant and stable ranking of attractiveness, it may be used to build a ranking of repulsiveness. The idea is that the more people leave a country to other repulsive countries, the more repulsive is that country. This concept is really close to the PageRank. It corresponds to the PageRank applied to the transposed matrix of migrations. Indeed, in this matrix all the flows are inverted. We obtain the following results :

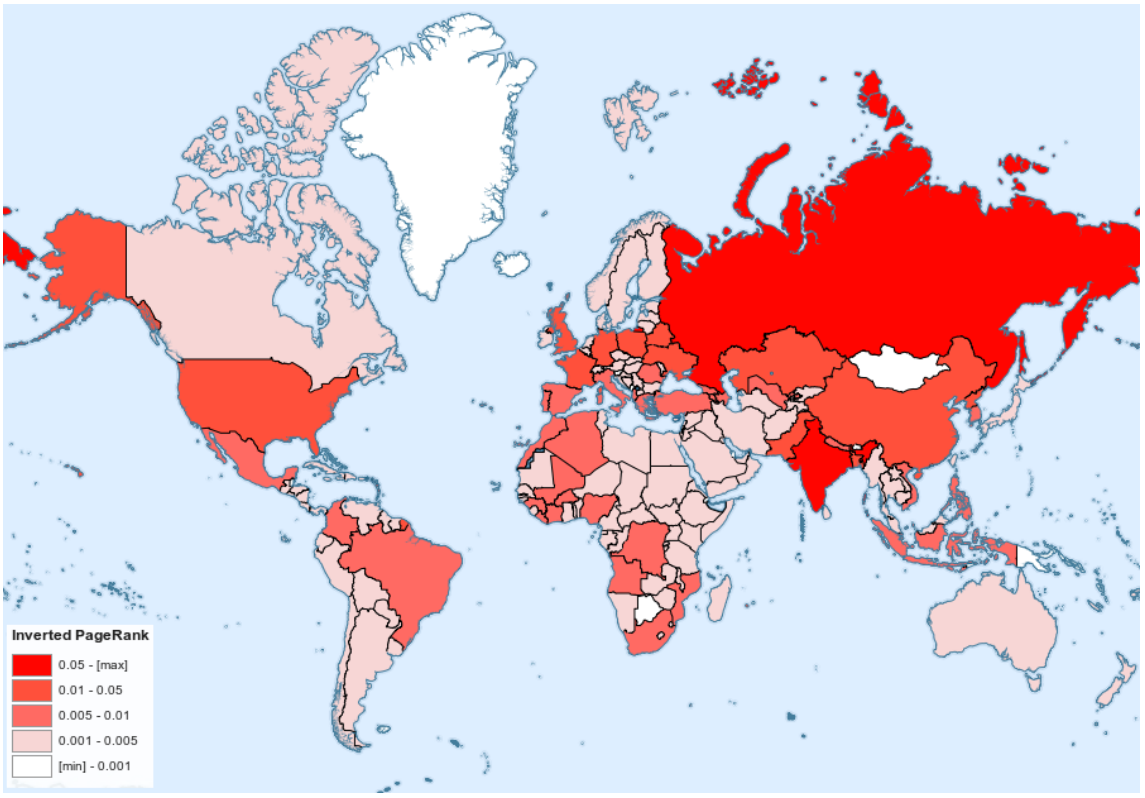


Figure 3.9: Map of inverted PageRank.

Ranking	Country
1	India
2	Russian Federation
3	Bangladesh
4	Ukraine
5	China
6	France
7	United States
8	Pakistan
9	United Kingdom
10	Kazakhstan

Table 3.13: Top ten of most repulsive countries according to their inverted PageRank.

These ranking and map gives an idea of which countries are the most repulsive. The majority of the countries present on this ranking were also in the Table 3.2 of repulsiveness according the out degree.

Furthermore as previously said, the least attractive countries do not interest us because they are mainly small islands, dependent states or at best countries where people do not tend to go. However, the least repulsive countries are more interesting to analyse because those are countries where people do not tend to leave once they are in it. We have thereby the following ranking of least repulsive countries :

Ranking	Country
1	Saint Pierre and Miquelon
2	Norfolk Islands
3	Maldives
4	Liechtenstein
5	San Marino
6	Cayman Islands
7	Gibraltar
8	Monaco
9	Djibouti
10	Bermuda

Table 3.14: Top ten of least repulsive countries according to their inverted PageRank.

Again, we can find several reasons of why these countries are on this ranking :

- Saint Pierre and Miquelon has known only 169 emigrations between 1990 and 2000. That is the reason why they appear in this ranking. We have a similar situation for Norfolk and the Maldives.
- Some others countries already appeared in previous top ranking (3.5 and 3.3. It is the case for Cayman Islands, Monaco and Djibouti.
- The last countries are also either Offshore financial centers or tax havens like Liechtenstein D.2.48, San Marino D.2.75, GibraltarD.2.32 and Bermuda D.2.12.

3.8 Ratio of PageRank and inverted PageRank

We have now a ranking of attractiveness and a ranking of repulsiveness thanks to two measures. However, we would like to build a single ranking including all the information. One way to do it is to consider the ratio between the pageRank and the inverted PageRank :

$$\text{Ratio of PageRank} = \frac{\text{PageRank}}{\text{inverted PageRank}}$$

Using this measure we obtain the Table 3.15 of the top ranking , the Table 3.16 of the bottom ranking[9] and the Map 3.10.

Ranking	Country	Ranking PageRank	Ranking Inverted PageRank
1	Canada	2	69
2	Saudi Arabia	10	148
3	Puerto Rico	9	132
4	United States	1	7
5	United Arab Emirates	25	176
6	Australia	5	54
7	West Bank and Gaza	7	44
8	Syrian Arab Republic	15	105
9	Switzerland	20	112
10	Israel	16	99

Table 3.15: Top ten of most attractive countries according to their ratio of PageRanks.

[9]Let us recall that being highly ranked in Inverted PageRank means to be repulsive.

Several countries appearing on the Table 3.15 were also present on the Table 3.8 of the most attractive country according to the PageRank. Indeed, only the United Arab Emirates which had the 25^{th} position, Syrian Arab Republic (15^{th}), Switzerland (20^{th}) and Israel (16^{th}) were not in the top 10 of attractiveness. We can also analyse the bottom of the ranking :

Ranking	Country	Ranking PageRank	Ranking Inverted PageRank
1	Bangladesh	107	3
2	Korea, Dem. Rep.	192	13
3	China	83	5
4	Azerbaijan	161	22
5	India	23	1
6	Georgia	171	34
7	Morocco	121	17
8	Albania	166	38
9	Pakistan	58	8
10	Armenia	148	33

Table 3.16: Top ten of least attractive countries according to their ratio of PageRanks.

Unlike the Table 3.15, the results here are more spread than the ones of the Table 3.13.

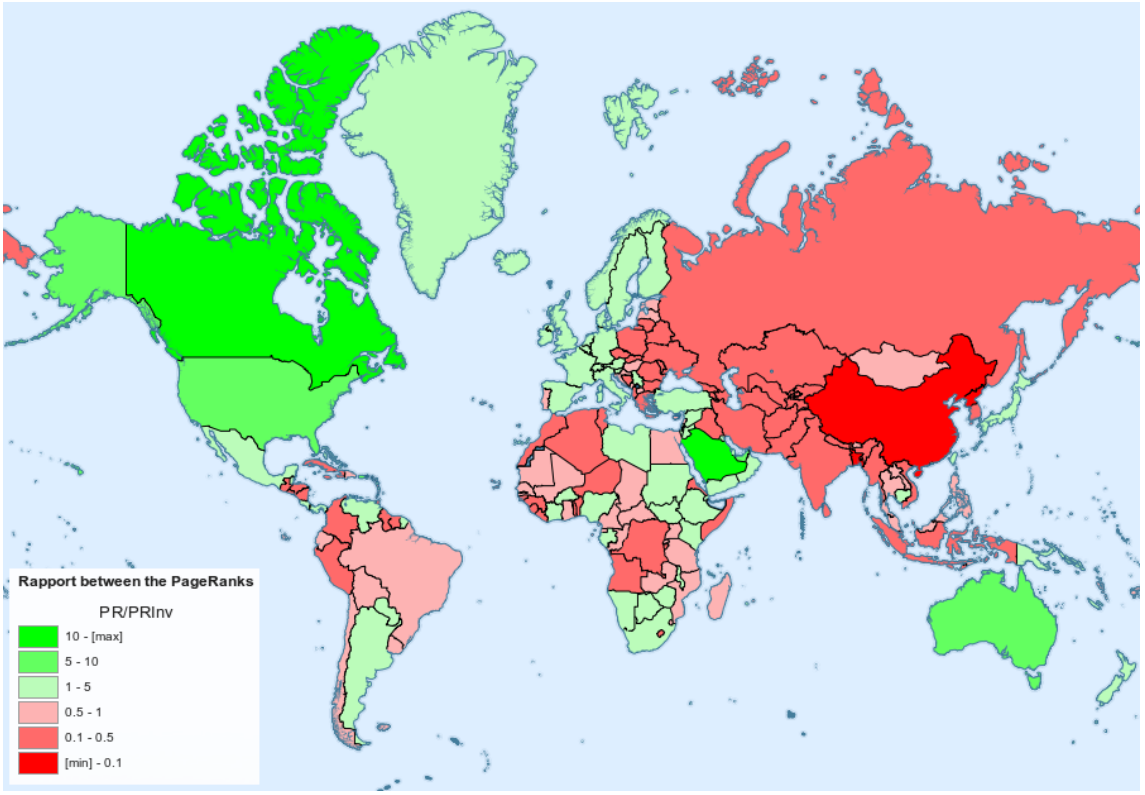


Figure 3.10: Map of ratio of PageRanks.

With the Map 3.10 we have thereby a global view of which countries are attractive and which ones are repulsive.

3.9 Difference of PageRank and inverted PageRank

Another way to merge the ranking of attractiveness and repulsiveness into a single ranking is to calculate for each country the difference between its PageRank and its inverted PageRank value. The Tables 3.17 and 3.18 recap respectively the top ranking and the bottom ranking of attractiveness.

Ranking	Country	Ranking PageRank	Ranking Inverted PageRank
1	United States	1	7
2	Canada	2	69
3	Australia	5	54
4	United Kingdom	3	9
5	Germany	4	12
6	West Bank and Gaza	7	44
7	Mexico	8	31
8	Puerto Rico	9	132
9	Saudi Arabia	10	148
10	France	6	6

Table 3.17: Top ten of most attractive countries according to their difference of PageRanks.

Ranking	Country	Ranking PageRank	Ranking Inverted PageRank
1	India	23	1
2	Russian Federation	12	2
3	Bangladesh	107	3
4	Ukraine	32	4
5	China	83	5
6	Pakistan	58	8
7	Kazakhstan	54	10
8	Korea, Dem. Rep.	192	13
9	Poland	45	11
10	Belarus	97	14

Table 3.18: Top ten of least attractive countries according to their difference of PageRanks.

It is interesting to see that this new ranking highlights at the same time the countries that were attractive with the PageRank and the countries that were repulsive with the Inverted PageRank. Indeed, the ranking obtained matches fairly well the ranking of the Tables 3.8 and 3.13. It was not the case of the previous measure which took the ratio of the PageRank although the Maps 3.11 and 3.10 have a similar trend. Another advantage of this measure is that it limits the value between -1 and +1. This measure can be visualised on the Map 3.11.

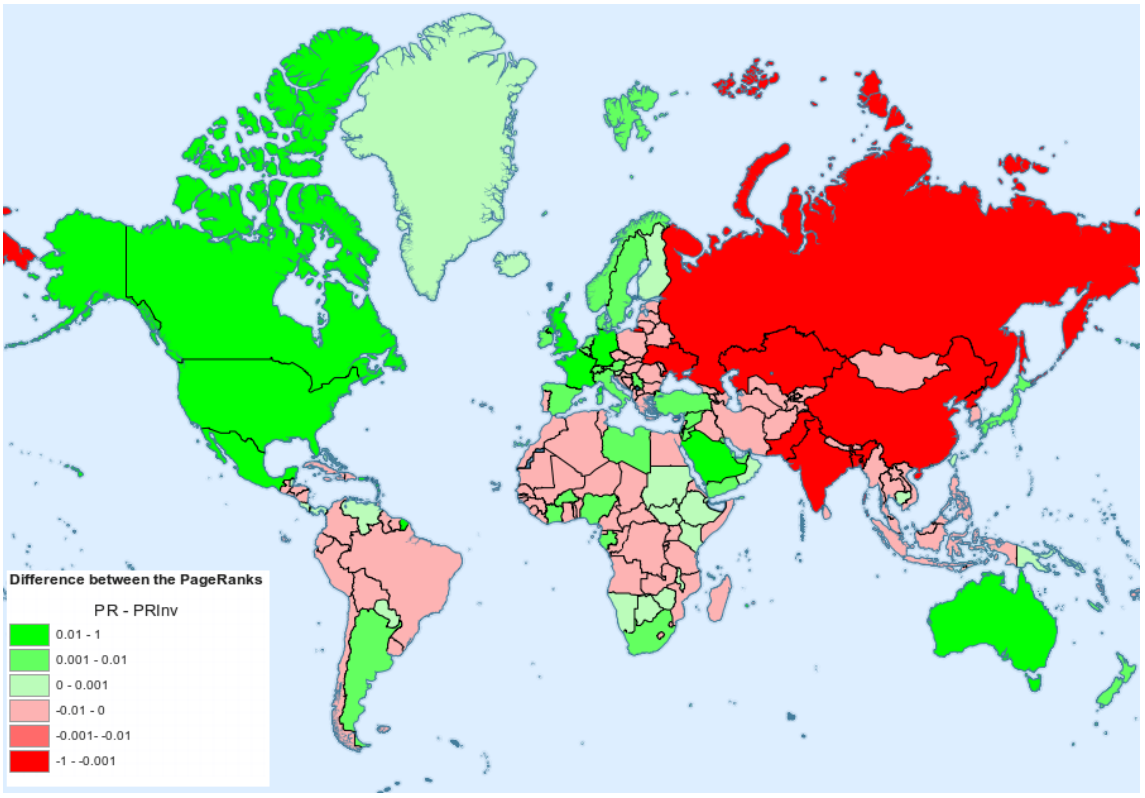


Figure 3.11: Map of difference of PageRanks.

3.10 Topological order

3.10.1 Principles

As we saw, there are many ways to rank the nodes of a graph. On this section, we consider a completely different method, the topological order [76].

Definition 3.10.1 (Topological order)**.** *A topological order is an ordering* $g : x \to \mathbb{N}$ *of the nodes* x *of a graph where if there is an edge from the node* u *to* v *we have* $g(u) < g(v)$*.*

It is easier to visualise it by applying this concept on a concrete case. Let us consider a food chain graph where there is a directed edge from u to v if the animal v is a predator of u. According to this definition, the animals on the bottom of the chain have the lowest score because they have no incoming edges. Each of their predators will have an higher score because they have an incoming edge from their preys. The predators of these lasts have an higher score too and we continue this process recursively to the top of the chain where the animals with no predator have the highest score.

To compute the topological order, we use the following algorithm :

```
1 //Input : A directed acyclic graph G
2 //Output : A topological order of G
3
4 TopologicalOrder(G):
5       S := Set of nodes with an indegree of 0
6       level := 0
7       numberOfNodes := G.size()
```

```
8       while(S.size() < numberOfNodes):
9            level++
10           for all nodes n in G:
11                if(G.indegree(n) == 0):
12                     topOrder[n] := level
13                     G.remove(n) // All the edges touching n are removed too
14                     S.add(n)
```

There exist several algorithms to compute the topological order of a DAG[10]. This one has the advantage of giving the same level at every node having at the same iteration a null in-degree which is more relevant for the task of ranking countries. Moreover, it ensures the uniqueness of the topological order because at each iteration level, all the nodes with a null in-degree obtain the same level.

Concerning the time complexity, it is $O(n+m)$, the sum of the number of nodes (n) and edges (m) in the graph, because they are both only considered once :

- n with the iteration from the nodes of the graph.
- m with the computation of the in-degree of each node.

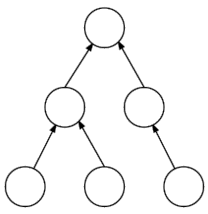
However, as we said, this algorithm works only on directed acyclic digraph. We wonder if it is applicable in our graph. Obviously, it is not entirely acyclic but we can adapt the topological order algorithm by aggregating some nodes according to their strongly connected component.

Definition 3.10.2 (Strongly connected component)**.** *A strongly connected component of a directed graph G is a subgraph G' of G where from every node of G', we can reach every node of G'.*

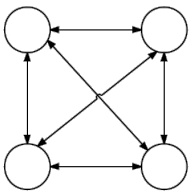
By using this concept, we can thus give the same topological level at every node that belongs to the same strongly connected component. However we need to be careful about that. Indeed, if the graph has too many cycles, there is a risk to have only few components containing each many nodes. This configuration yields a very poor topological order because many nodes have the same level. The topological order can only be accurate if we have many components containing each not too many nodes. It is why we need to see first how cyclic is our graph. It can be done by calculating the hierarchy of the graph.

3.10.2 Hierarchy measure

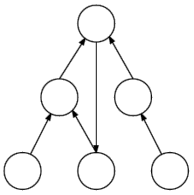
The purpose of measuring the hierarchy is to analyse the hierarchical character of the graph. Concretely, this measure is done by giving a score called the hierarchy degree, to the graph. This score is obtained by a function $h : G \to [0, 1]$ taking the graph as input and returning the score. On the extreme cases, we have $h = 1$ if the graph is fully hierarchic and $h = 0$ if it is fully non hierarchic. The Figure 3.12 shows three different graphs, each with a particular hierarchy degree.



(a) Fully hierarchic graph.



(b) Fully non-hierarchic graph.



(c) Semi hierarchic graph.

Figure 3.12: Example of graphs.

[10]Directed acyclic graph

There are different ways to calculate the hierarchy score. Indeed, it entirely depends on how the metric function h is defined. The purpose of this section is to describe two kinds of metrics and explain which one we take to measure the hierarchy and why.

Metric based on cycles

The main idea of this metric [109] is that cycles in a graph detriment the hierarchy. Indeed, when several nodes are implied in the same cycle, we can not tell which one is hierarchically superior to the other ones [11]. We can see it on the Figure 3.12 where the Graph b has a score of zero because all of its edges are involved in the same global cycle while the Graph a is fully hierarchic because there is no cycle. Following this idea, this first metric consists in calculating the score by taking the ratio of the number of edges that doesn't belong to any cycle to the total number of edges. Mathematically, it is formulated like this :

$$h = \frac{\sum_{i=1}^{|E|} e_i}{|E|}.$$

with $e_i = 1$ if the edge i belongs to no cycle and $|E|$ the number of edges in the graph.

However, although this metric is suitable for unweighted graphs, it presents some issues in the opposite case. Indeed, with a weighted graph, some edges are more important than others. If we do not take the weights into account, it means that we consider all the edges as similar but having a cycle with a weight of 10 is not the same of one with a weight of 100. It is why this metric has to be refined.

Metric based on weighted cycles

This second metric [109] overcomes the issues raised by the first one by including the weights in the computation of the score. The expression is thereby the following :

$$h = \frac{\sum_{i=1}^{|E|} w_i e_i}{\sum_{i=1}^{|E|} w_i}$$

with w_i the weight of the edge i.

Now, if the edges that belongs to no cycle have a more important weight, the score will thus increase. We use this metric to characterise the migration graph.

To implement it, we used the Property 3.10.2.

Property 3.10.1. *In a directed graph, every edge forming a cycle belongs to the same strongly connected component.*

Proof. By contradiction. Let us assume that we have a cycle of two nodes, n_1 and n_2, but which do not belong to the same strongly connected component. If there is a cycle, we have a a path from n_1 to n_2 and from n_2 to n_1 (definition of cycle). If there are these two paths, from both nodes we can reach the other ones. According to the Definition 3.10.2, they belong to the same strongly connected component. We have a contradiction, so the property is true. □

Property 3.10.2. *In a directed graph, an edge belongs to a cycle if and only if it belongs to a strongly connected component.*

[11] At this step, we do not consider yet the weights of the edges.

Proof. We need to prove both implications :

- `Belong to a cycle` $\implies$ `Belong to a strongly connected component` : By contradiction. Let us suppose that an edge belongs to a cycle but to not any strongly connected component. According to the definition of cycle, from any edge that belongs to the cycle, we can reach any other in the cycle. Because there are such paths, theses edges belong to the same strongly connected component. We have a contradiction, the statement is thereby true.
- `Belong to a strongly connected component` $\implies$ `Belong to a cycle` : By contradiction. Let us suppose that we have a strongly connected component but no cycle. Applying the definition of cycle and strongly connected component, if an edge belongs to a strongly connected component, it belongs to a cycle with at least another edge. We have a contradiction, the statement is thereby true.

□

Algorithm

Using the Property 3.10.2, we can obtain the set of acyclic edges only by doing the difference between the set of all the edges of the graph and the set of the edges which belong to a strongly connected component. The following pseudo code implements the metric :

```
1  Input : A directed graph G
2  Ouput : the metric h (value [0,1])
3
4  computeMetric(G):
5       H = G.getStronglyConnectedComponents() // Give the set of edges
6       A = {} // Set of acyclic edge.
7       for all edges e in G:
8            if(e not in H):
9                 A.add(e)
10
11      acyclicWeight = 0
12      for all edges e in A:
13           acyclicWeight = acyclicWeight + e.weight()
14
15      totalWeight = 0
16      for all edges e in G:
17           totalWeight = totalWeight + e.weight()
18
19      return acyclicWeight/totalWeight
```

Complexity

Let us analyse the complexity of this algorithm :

- By using the Tarjan algorithm [146], the strongly connected component can be obtained on $O(n+m)$, with n the number of nodes and m the number of G.
- The first *for loop* iterates on m edges.
- By using array list for storing the set of edges, we can decide if an edge is contained in the array list in $O(m)$ by testing each elements. We test m different edges, we have thus $O(m^2)$ for the first loop.

- For the two other loops, we iterate also on the edges for which the number is bounded at m. We have therefore $O(m)$ for these two loops.

So this algorithm runs on $O(n+m) + O(m^2) = O(n+m^2)$.

Results

Using the algorithm, we can finally have the hierarchy score of the graph. The score obtained is null which means that every edge of the graphs is implied in at least one cycle. Actually, it could be seen as well by noticing that every edge belongs to the same strongly connected component. By the the Property 3.10.1 the graph is strongly connected. In this actual state, it is impossible to give a hierarchy to the graph. To do it, we have to prune several edges in order to exhibit a backbone permitting the computation of a topological order. Given that aggregating some nodes into strongly connected component is a waste, we need to find another solution like obtaining a spanning tree of the graph.

3.10.3 Spanning tree

The issue is now to prune some edges in order to exhibit a fully hierarchic subgraph containing no cycle. We did it by using the concept of spanning tree.

Definition 3.10.3 (Spanning tree)**.** *A spanning tree T of a directed graph G is a tree (connected graph with no directed cycle) including all the nodes of G and some of its edges.*

The concept of spanning tree raises another issue. Generally, a graph has more than one spanning tree. Furthermore, given that our migration graph is quite dense, there are many different spanning trees. The goal is to have the most relevant backbone of the graph so that we can apply the topological order. A way to do it is to keep all the possible highest edges. If we have the choice to prune one edge between one of a weight of 100'000 and one of 100, we prefer to prune the lowest one and to keep the highest in the backbone. Following this argument, this task turns to having a structure tending to a maximum spanning tree.

Definition 3.10.4 (Maximum spanning tree)**.** *A maximum spanning tree is a spanning tree of a graph G where its weight, the sum of the weighted edges, is higher or equal than the weight of all other spanning tree of G.*

There exists two well known algorithms to compute the maximum spanning tree of undirected graph, Kruskal's algorithm and Prim's algorithm [76]. However these algorithms only work on undirected graphs and our graph is directed. To overcome this issue, we designed two heuristics based on Kruskal with in order to obtain a spanning tree approximating the maximum [12].

Directed spanning tree

The idea of the first heuristic is quite simple, we sort the edges on decreasing order by their weight and we try to add each edge to the spanning tree. If adding the edge does not create a cycle, it is added on the spanning tree. We only stop when each edge has been tested.

```
1 Input : A directed graph G
2 Ouput : A spanning tree of G
3
4 directedSpanningTree(G):
5       T := {} // Spanning tree containing no edge and all the nodes of G
```

[12] We did not find any source explaining if these heuristics have already been used.

```
6       for all nodes n in G:
7            define a cluster C(n) for each node n
8       Q := priorityQueue(G) // Q contains the edges of G sorted decreasingly
9       while(Q is not empty):
10           e := Q.pop() // take and remove the first edge of Q
11           Csrc = C(e.src()) // Cluster containing the source of the edge.
12           Cdest = C(e.dest()) // Cluster containing the destination of the edge.
13           if(Csrc != Cdest) // The nodes are in two different clusters
14                T.add(e)
15                Union(Csrc,Cdest)
16      return T
```

The purpose of the cluster is to efficiently decide if adding an edge will create a cycle. A cluster is just a set of nodes that grows up and that can be merged with another. If both nodes of an edge belong to the same cluster, it means that there is a cycle. Concerning the priority queue, it is an abstract data structure where each element has a priority. Every time that we pop an element of the queue, it is always the one with the highest priority that is returned. The priority queue can be implemented in a heap.

Heap

Definition 3.10.5 (Heap)**.** *A heap is a data structure consisting of a complete binary tree where the object contained in a node has a priority lower or equal than the priority of its father node. This property must be satisfied for every node. The root node will thereby contain the object with the highest priority.*

Each node A of the tree can be represented by the tuple $(\texttt{index}, \texttt{object})$ where `index` is the node identifier. The identifiers are unique and attributed in an increasing order from the root and following a left-to-right attribution.

The main advantage of this representation is that the heap can be easily stored in an array. For example, the heap in the Figure 3.13 yields the array in the Table 3.19.

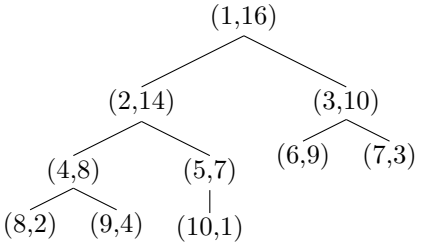


Figure 3.13: Example of a heap.

Index	1	2	3	4	5	6	7	8	9	10
Value	16	14	10	8	7	9	3	2	4	1

Table 3.19: Array representation.

For a node of index i, we have the following relation :

$$\texttt{Father of } i = \lfloor \frac{i}{2} \rfloor$$

$$\texttt{Left child of } i = 2i$$

$$\texttt{Right child of } i = 2i + 1$$

We can recombine the heap where $A[i]$ is the node of index i. Before using a heap, we need to consider three things :

1. How to build the heap.
2. How to access the element with the highest priority while maintaining the heap structure.
3. How to push an element in the heap while maintaining the heap structure.

Building the heap

Building a heap requires two specific functions :

- `pile(i)` : let us suppose that the left and the right subtree of node $A[i]$ are both heaps but $A[i]$ is lower than one of its children. We swap the node $A[i]$ with its children of highest priority. After that we recursively continue this process in the modified subtree to preserve the heap structure.
- `buildHeap(A)` : build the heap level by level. We first consider the leaves (all of them are heaps of one node) and after we go to the next level by performing a `pile` if necessary. We recursively do it until reaching the root.

This yields the following algorithm to build a heap :

```
1  Input : An array of object
2  Output : An heap
3
4  buildHeap(A):
5       for i from A.size() to 1 // from the leaves to the root
6            A.Pile(i)
7       return A
```

Let us see the complexity of these procedures :

- `pile(i)` : on the worst case, we have to go from the root to a leaf. Given that we have a binary tree, the complexity is $O(\log n)$ with n the number of nodes.
- `buildHeap(A)` : on the worst case, the tree is fully complete[13]. When we are at the leaves level, there are at most $n/2$ heaps, but no operation to do, because they are all already heaps. At the next level, we have $n/4$ structures, with at most one `pile` to perform for every structure because they have a depth of 1. Recursively, we obtain this expression :

$$\texttt{number of operations : } \quad 0 \times \frac{n}{2} + 1 \times \frac{n}{4} + \cdots + i \times \frac{n}{2^{i+1}} = O(n).$$

[13] The last level is full of leaves.

Popping element

Popping the element with highest priority is easy. According to the heap structure, this element will be located at the root node. We can directly access it. The issue is to maintain the heap without this element. To do so, we move the rightmost leaf to the root and we perform a `pile` on it. Popping an element has thus a time complexity of $O(n)$.

Pushing element

The procedure for pushing an element is quite similar for popping. We add the element at the first available position of the tree while preserving the completeness of the tree, and we perform a reversed `pile` on this node, to rise it up until its value is lower than its father node. Pushing an element has thus a time complexity of $O(n)$.

Complexity

Now every structure used in the directed spanning tree algorithm are known, let us analyse the complexity of it :

- As said before, by using a heap for the priority queue, it takes $O(\log m)$ to build it and $O(\log k)$ to pop an element with m the number of edges and k the number of element in the priority queue. In our case, $k = m$ at the first iteration an is decremented of 1 at each iteration.
- There are m elements in the queue, so we iterate m times.
- Merging the cluster is done in $O(1)$.

This yields a final time complexity of $O(m \log m)$. Now that we have a directed acyclic graph, we can finally compute the topological order.

Results

Firstly, it is important to mention that the spanning tree obtained is not maximum. Indeed, although Kruskal gives the maximum spanning tree for undirected graph, it is not the case on directed graphs. What we have is just a heuristic. With the structure obtained, we can now apply our topological order algorithm to have a ranking of the countries. Applying it, from the 20088 edges of the initial graph, we keep approximatively 55% of them, 11114 edges, to have the acyclic graph. The topological order applied to it gives 171 different levels. The Table 3.20 recaps the top ten countries having the highest levels.

Ranking	Country
1	Bahrain, Monaco
3	Andorra, Qatar
5	Luxembourg
6	United Arab Emirates, San Marino
8	United States
9	Marshall Islands
10	Australia

Table 3.20: Top ten countries according to the topological order of the first spanning tree.

Firstly, we can notice that the number of levels is high regarding the number of countries considered (227). Indeed, there is nearby one particular level per country with at most three countries per level. This is because even the half of the edges have been pruned there are still many edges yielding a higher probability to have many levels.

Furthermore, according to the concept of the topological order, the countries that belongs to the highest level do not have any outgoing edges at all. The ranking obtained confirms this idea. Indeed, eight of these top countries are tax havens countries (Bahrain, Monaco, Andorra, Qatar, Luxembourg, United Arab Emirates, San Marino, Marshall Islands) where we tend to go but not to leave it. It joins the concept of attractive and non repulsive countries. This ranking gives thereby an idea of which countries are at the same time attractive and non repulsive.

A last comment is about the design of our topological order algorithm applied on the directed acyclic graph obtained. As we said, we give the same level at each node having at the same iteration a null in-degree. However, it is possible to construct the topological order differently. For example, the ordering illustrated on the Figure 3.14 is correct.

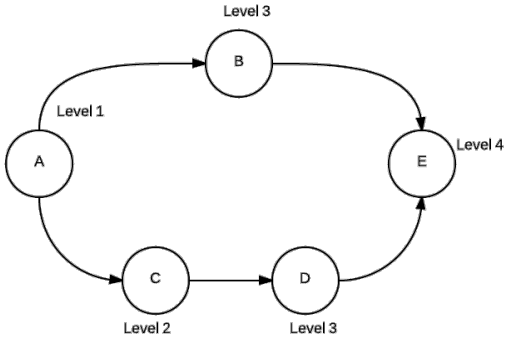


Figure 3.14: Another topological order.

Here, the assignment of level is not directly made when the in-degree of a node is null. It is why the node B has a level of 3 and not 2 as with our algorithm. However, for two main reasons we decided to did not pay attention to this particularity. First, the spanning tree used is the result of an heuristic and interpreting the structure of the tree is thereby not an easy task. We do not know which order is the most relevant in this case. Furthermore, the difference between the orders only concerns some intermediate nodes and we are mainly interested in the top rank where it is less likely to have a difference.

Undirected spanning tree

Another idea to have a relevant spanning tree of a directed graph is to transform the directed edge into undirected and apply after the classic Kruskal's algorithm. Finally, we retrieve a directed graph by keeping only the edges remaining on the spanning tree.

```
1 Input : A directed graph G
2 Ouput : A spanning tree of G
3
4 undirectedSpanningTree(G):
5       Changing the directed edge of G into non directed
6       T := G.kruskal() // Apply the kruskal algorithm on H
7       Recover the directed edge of G from the non directed of T
8       return G
```

Complexity

The complexity of each function is the following :

- To change each directed edges into undirected we need to consider once all edges of the graphs, yielding $O(m)$. Following the same argument, the recovering of the directed edges is done on $O(m)$ too.
- With the best implementation of Kruskal we can have the maximum spanning tree on $O(m \log m)$.

This yields a time complexity of $O(m \log m)$.

Results

As previously, let us compute the topological order on the retrieved directed spanning tree. The Table 3.21 recaps the ranking.

Topological order	Ranking	Country
5	1	United States
4	2	Nigeria, Syria, Turkey
3	5	23 countries
2	28	54 countries
1	82	more than 100 countries

Table 3.21: Ranking of countries according to the topological order of the second spanning tree.

Unlike the first spanning tree, we can notice that we have only 5 levels. Indeed, given that the spanning tree is formed from an undirected graph, adding an edge has a higher probability to create a cycle. It is why we have here less edges than previously, only 193 edges which is less than 1% of the initial set, yielding a lower number of levels. The Figure 3.15 represents this spanning tree on a world map [14].

Figure 3.15: Undirected spanning tree.

We can notice that some countries, like the United States or Russia, are important hubs in the spanning tree. Moreover, we can differentiate the hubs. Indeed, the United States have edges

[14] Every edge crossing the map from east to west are linked to the United States.

from/to countries located all around the world while the countries interacting with Russia are from a more specific area. It is why we can tell the the United state is a global hub and Russia a local hub. at the first sight we can observe a pyramidal distribution where we have a big number of countries on the first level and only one at the last. However apart from that, it is more difficult to interpret. Indeed, except the United States having the first position which can make sense, there is no particular reason to have Nigeria, Syria and Turkey, at the 4th level. The problem comes from the number of edges which is too low. With less than 1% of the initial set of edges, it is difficult, even impossible, to maintain a backbone keeping the representativeness of the initial graph. It is why this ranking seems to be less relevant than the one obtained with the directed spanning tree.

3.11 Conclusion

In this chapter we saw many different ranking methods, each with their own particularities, pros, cons and limits. We summarize here our main findings in this chapter :

- We started by considering simple ranking methods, each having some drawbacks to have a relevant ranking. By finding a way to improve the previous methods, we converged to the idea of the PageRank, a tool initially conceived for ranking webpages, which is also effective to rank countries, as shown on the Table 3.8.
- The PageRank is a robust and stable method as shown on the Tables 3.10 and 3.12. Indeed, if there are not too many differences between two sets of data, the PageRank gives a similar ranking for both sets.
- The PageRank can be reverted to obtain a ranking of repulsiveness. This ranking allows us to obtain the least repulsive countries (Table 3.14).
- By using the PageRank and the inverted PageRank together, we obtain a general ranking showing at the same time which countries are attractive and which ones are repulsive. The final result of the PageRank section is represented on the Map 3.11 which highlights the differences between the countries.
- Another way to rank countries is the topological order. However, before using this method, we must have an acyclic graph. It can be done by using heuristic to obtain a good spanning tree of a directed graph.
- The first heuristic using a directed spanning tree presented good results (Table 3.20) by highlighting at the same time attractive and non repulsive countries such as tax haven countries.
- The simple spanning tree heuristic did not give exploitable results given there are too many edges pruned with this method. However through the Map 3.15, we can see which countries have many exchanges. This heuristic allows to obtain a spanning tree highlighting global and local hubs.

Chapter 4

Group detection

"Life is a constant struggle between being an individual and being a member of the community."

– Sherman Alexie, *The Absolutely True Diary of a Part-Time Indian*

4.1 Motivation

In a general way, the task of detecting groups consists in analysing the links existing between different objects in order to gather the objects having strong relations together. This concept is highly related to graph theory [68] where the objects are the nodes and the relations are the edges. Like the ranking task, there is a multiplicity of specific strategies to group the nodes in a relevant way. It is why several methods exist according to the criteria by which we want to group the nodes.

Most of these methods tend to form cohesive groups where the connections are important inside the group. It is for instance the idea of the n-cliques, n-clans, n-clubs, k-core and others [18, 117]. Unfortunately, most of these methods are computationally difficult [130, 61]. It is not the case of the k-core which can be computed in a polynomial time complexity [18]. For example, this method has been used to decompose the Internet graph at the autonomous system level [16] in order to characterise this network according to the degree distribution and thereby eventually discover a specific architecture of the system [64].

Another method heavily used in networks is the community detection [68]. It enables to highlight nodes tightly connected together by forming groups or clusters in the graph. The communities formed can be used afterward to identify the structures of a network corresponding to important functions [69]. There are numerous applications of this concept on concrete networks as the social networks for humans [36] and animals [110], the Internet routing network [142], the food webs [97] and many more.

From a practical point of view, Newman and Girvan [120] studied this problem and proposed a way to solve it by designing a measure called the modularity which has the purpose of characterising the quality of the community partition. Several algorithms [25, 94, 102] use this concept to solve this problem.

The community method aims therefore to form groups of tightly connected nodes and where the connections extra-communities are weak. It is not the case of the core-periphery [29] model which consists in dividing the graph into two groups. One group, the core contains the densely

connected nodes, and the second group, the periphery, the sparsely connected nodes. Compared to the community model where the connections between two communities are penalised, there is no penalty to have connections between the core and the periphery. The only penalties are due to the periphery-periphery links. Several variations of this model exist.
For example, we can consider :

- A unique core [131].
- Several cores by discriminating local and global cores [46].
- A combination of the community and the core-periphery model [136].

Furthermore, the core-periphery has been studied and applied to undirected [131], directed [46, 29] and weighted [136] networks. Again, this method has numerous applications in concrete graphs like social networks, cellular networks, ecological networks, etc.[1]

All the methods concerning group detection described above use thereby a graph approach. However, to the best of our knowledge, none of these methods are already applied to the migration graph. We observe a similar situation with the ranking. As for the ranking, we will adapt these methods to analyse the migration network.

The purpose of this chapter is to present several methods that can be used to perform such group detections in the migration network. For each, we will analyse their specificities as well as the results obtained. Again, we base our work on the migration data of the World Bank [169].

4.2 Balanced cycles

The simple way to highlight groups in a graph is to consider the cycles. In its basic form, this task is equivalent to computing the strongly connected components. We already mentioned the meaning behind this idea in the topological order section where we tried to group countries by such strongly components. However we saw that the migration graph is strongly connected. The direct consequence is that every country is involved on the same global cycle yielding a single group which is not acceptable for our analysis. If we want to use cycles to detect groups, we must refine this concept. We do it by stating three more conditions for having a cycle of countries :

- The cycle must have a good cohesion. It means that the weight of the edges forming the cycle must be important. This condition is controlled by a cohesion threshold k. The arithmetic mean of the edges forming the cycle must be superior to k. For a cycle of n edges e of weight w, we have the following condition :

$$\frac{\sum_{i=1}^{n} w_i}{n} \geq k.$$

  If we increase k, we are more restrictive on the cycles.

- The cycle must be homogeneous. It means that the weight of the edges forming the cycle must be close together. This condition is controlled by a homogeneity factor $l \in [0, 1]$. The weight of each edge on the cycle must be superior to the arithmetic mean of the cycle weight multiplied by l. Mathematically, we have

$$\forall e_i \in \texttt{cycle} \ : \ w_i \geq \frac{\sum_{i=1}^{n} w_i}{n} \times l.$$

[1] All of them are presented in [46].

This condition ensures that there is no significant disparity between the weight of the edges. On the extreme case $l = 1$, which means that each edge on the cycle must have the same weight. Consequently, closer l is to 1, less tolerant we are on the disparity of weights.

- The cycle must have a determined length. It means that the number of countries involved on the cycle is fixed. We call this parameter j. With V the set of nodes on the cycle, we have

$$|V| = j$$

Using these conditions we are thereby more likely to obtain cycles. The number obtained depends directly on the parameters. The intuition behind this evolved concept of cycle is to group countries that have a balanced exchange together. None of the countries that belong to the same cycle is more attractive than the other ones if we consider only the current cycle. We present now some algorithms that we designed to detect such cycles.

4.2.1 2-cycle

In a first time, we only consider the cycle of length 2. The resulting cycles are obtained with the following algorithm.

```
1  // Input : G = Graph(V,E), k,l
2  // Output : the 2-cycles present on G
3
4  2cycleDetector(G,k,l):
5       cycleList = {}
6       for all nodes u in G:
7            for all nodes v in G:
8                 mean = (uv + vu)/2 // uv : edge between node u and v.
9                 threshold = mean*l
10                if(uv.weight >= threshold and
11                   vu.weight >= threshold and
12                   mean >= k):
13                     cycleList.add(uv + vu) // Add the cycle u-v-u
14      return cycleList
```

The idea of this algorithm is to iterate on every 2-cycle in the graph and verify if the cycle is balanced regarding to the the cohesion threshold k and the homogeneity factor l. About the complexity, we iterate twice on all the nodes, yielding $O(n^2)$. We execute this algorithm several times with diverse values of k and l. The Table 4.1[2] shows the evolution of the number of cycles according to the parameters.

[2] As in the Chapter 3, all the results obtained are based on the the World Bank's data of migrations [162] occurred from 1990 to 2000 unless otherwise mentioned.

k	l	number of 2-cycles
10'000	0.8	66
10'000	0.9	30
10'000	0.95	15
10'000	0.99	2
100'000	0.8	15
100'000	0.9	7
100'000	0.95	2
100'000	0.99	1
1'000'000	0.8	2
1'000'000	0.9	1
1'000'000	0.95	1
1'000'000	0.99	1

Table 4.1: Evolution of the number of 2-cycles according to k and l.

We observe that increasing the value of k or l gives less cycle. Comparing the number of simple cycles [3] connecting all the graph, we have here a restrained number of cycles. Indeed, the more we constraint the cycles, the less likely is the probability to obtain them. Besides, we manage to represent the most important cycles obtained. The Map 4.1 recaps and localises them.

Figure 4.1: Most important ($k = 100'000$, $l = 0.8$) 2-cycles.

From this map we can detect different cycles :

- A cycle with United kingdom and South Africa.
- A cycle with India and Nepal.
- Cycles involving countries of western Europe : France, Spain, United kingdoms or Netherlands.
- Cycles involving Russia and USSR countries.

The last two kinds of cycles are particularly interesting. As we saw previously with the eigenvector centrality ranking on Section 3.5, the former USSR countries kept an important relation together. What we learn more is that these relations are balanced. None of these countries drains the populations of others [4]. This is mainly true about Russia which is involved in many cycles.

[3] We call simple cycles the cycles on their basic definition : without considering the conditions added.
[4] For the countries involved on the cycles.

At a smaller scale, we have the same phenomenon with some western Europe countries where the migrations are balanced too.

Concerning the India-Nepal cycle, it is more surprising given that the population of these countries are different. Indeed, India is more than fourty times more populated than Nepal. Despite this, their relation stays balanced. All the previous cycles concerned countries relatively geographically close. It is not the case of the United Kingdoms and South Africa cycle which however remain stable. Both are members of the Commonwealth and that can explain this relationship.

4.2.2 3-cycle

Similarly, we can do the same process to find the 3-cycles. On this purpose we adapt the previous algorithm.

```
1  // Input : G = Graph(V,E), k,l
2  // Output : the 3-cycles present on G
3
4  3cycleDetector(G,k,l):
5       cycleList = {}
6       for all nodes u in G:
7            for all nodes v in G:
8                 for all nodes m in G:
9                      mean = (uv + vm + mu)/3
10                     threshold = mean*l
11                     if(uv.weight >= threshold and
12                        vm.weight >= threshold and
13                        mu.weight >= threshold and
14                        mean >= k):
15                          cycleList.add(uv + vm + mu) // Add the cycle u-v-m-u
16      return cycleList
```

The idea remains similar : We iterate on every 3-cycle in the graph and verify if the cycle is balanced. Here, we iterate three times on the number of nodes, giving $O(n^3)$. As previously, let us see how the cycle number varies according to the parameters. The Table 4.2 recaps the results.

k	l	number of 3-cycles
10'000	0.8	61
10'000	0.9	19
10'000	0.95	6
10'000	0.99	0
100'000	0.8	8
100'000	0.9	4
100'000	0.95	2
100'000	0.99	0
1'000'000	0.8	0
1'000'000	0.9	0
1'000'000	0.95	0
1'000'000	0.99	0

Table 4.2: Evolution of the number of 3-cycles according to k and l.

For each combination of parameters we have less 3-cycles than 2-cycles. Moreover, for the most restrictive values of k or l we obtain no cycle. It confirms the intuitive idea that longer balanced cycles are less common. Indeed, instead of having a balanced exchange with only two countries, we have three countries involved which generate one more exchange and thus a new restriction. We represent the cycles on the Map 4.2.

Figure 4.2: Most important ($k = 100'000$, $l = 0.8$) 3-cycle.

Furthermore, the Table 4.3 details the most important 3-cycles.

Country 1	Country 2	Country 3
United States (1)	Germany (4)	Spain (11)
United States (1)	Germany (4)	Netherlands (19)
United States (1)	United Kingdom (3)	Ireland (35)
United Kingdom (3)	Germany (4)	Australia (5)
Germany (4)	Spain (11)	France (6)
Germany (4)	Turkey (28)	Austria (37)
West Bank and Gaza (7)	Jordan (41)	Saudi Arabia (10)
Ukraine (32)	Poland (45)	Belarus (97)

Figure 4.3: 3-cycles obtained with $k = 100'000$ and $l = 0.8$ with the PageRank ranking of the countries.

Intuitively, we could think that the countries implied in several 2-cycles are irremediably implied in 3-cycles, but the Map 4.2 invalidates this idea. Indeed, if we consider Russia, we see that it belongs to no 3-cycles. However some countries having 2-cycles, such as Germany, have 3-cycles as well. Moreover, Germany is implied in five 3-cycles from the eight obtained. It gives up the intuition that Germany is a country relatively stable by having balanced exchange with others. We cannot thereby state a general rule about the relation of 2-cycles and 3-cycles.

The Table 4.3 details the most important 3-cycles. It is important to specify that the 3-cycles are directed. This means that the flow from "Country 1" to "Country 2" is balanced with the flow from "Country 2" to "Country 3" and the flow from "Country 3" to "Country 1" and not in the other direction. The different countries in the cycle have either a really similar ranking or are neighbour countries. This follows the intuitive idea that in highly developed countries, people tend to move either to neighbour countries or to other highly developed countries.

4.2.3 Generic algorithm

The previous algorithms have two shortcomings :

- They are implemented in a naive way. Indeed, they only consist by testing all the possibilities of cycles of a given length in the graph. Following this idea, for a graph of n nodes, finding

cycles of length j gives a time complexity of $O(n^j)$ which is inefficient for long cycles.

- They suffer from a lack of genericity. Indeed, they are only adapted for cycles of a particular length. Each time we want to find other cycles, we have to modify the algorithm.

The purpose of this section is to present a generic algorithm which can find the j-cycles with j a parameter indicating the size of the cycle. The idea of the algorithm is to find all the j-cycles in the graph and after verify if the conditions of balance are respected. Here is a pseudo code of this algorithm :

```
1  // Input : G = Graph(V,E), k,l,j
2  // Output : the j-cycles present on G
3
4  cycleDetector(G,k,l,j):
5       cycleList = {}
6       tmpCycleList = findAllCycle(j) // Give all the simple cycles of length j.
7
8       for c in tmpCycleList:
9            if mean(c) >= k and all e in c >= mean(c)*k:
10                cycleList.add(x)
11      return cycleList
```

According to A. Panhaleux [128], the function `findAllCycle()` can find the cycles of length j in $O(m^{2-\frac{2}{j}})$ if j is even and in $O(m^{2-\frac{2}{j+1}})$ if it is odd with m the number of edges in the graph. Regarding the iteration on the number of cycles, its complexity is negligible in comparison with the first term.

However we do not implement this algorithm. As we saw the number of n-cycles [5] decreases when n increases. Given that we only had eight cycles remaining for the 3-cycles, increasing more the length did not seem very relevant to us.

As we saw through this section, the main goal of the cycle method is to detect set of countries having balanced exchanges. The 2-cycles and the 3-cycles already give an idea of the countries having this property. However, in a global context this method suffers from an important superficiality. Indeed, by considering only the cycles, we prune all the other migrations without paying attention of their importance. It results on a loss of informations. What we want to do is to form group without pruning edges. It is why we need to elaborate other methods having different purposes than only considering the balanced exchanges.

4.3 K-core

There are many ways to determine groups in a graph, each ones with their own specificities. For example, it is possible to form groups only by considering the strength by which some nodes are connected to the others. It is the idea of the k-core [18]. For a directed graph we can have three different k-cores depending on the kind of edges considered : the incoming edges, the outcoming or both. In this section, we envisage these three kinds of k-core.

4.3.1 Directed k-core

The article [18] presents the k-core for a simple undirected and unweighted graph. The idea can be outlined by the following definition.

[5] We are always speaking about the cycles respecting the conditions of balance stated before.

Definition 4.3.1 (k-core)**.** *Given a simple graph $G = (V, E)$ and a threshold k, the k-core of G is the subgraph of G where all the nodes that have a degree higher than k.*

However our graph is directed and weighted. Some modifications are therefore needed. First, to take the weights into account, we can replace the edges of cost n by n edges of cost 1. An edge (n_1, n_2) of weight 3 will give 3 edges from n_1 to n_2. Additionally, instead of pruning the nodes according to their total degree, we consider separately the in and the out degree.

Definition 4.3.2 (Directed k-core)**.** *Given a directed graph $G = (V, E)$ and a threshold k, the directed k-core of G is the subgraph of G where all the nodes have a weighted in-degree and an out-degree higher than k.*

The idea of the algorithm is to recursively prune all the nodes having their in-degree or their out-degree inferior than the threshold k. We iterate until there is no more node to prune[6] and the remaining nodes forms the k-core. Here is the algorithm that we used to compute the k-core :

```
1  // Input : G = Graph(V,E), threshold k
2  // Output : The k-core of G.
3  // The degrees consider the weight.
4
5  Kcore(G,k):
6     while(G.minInDegree <= k or G.minOutDegree <= k):
7        for all node n in G:
8           if(G.indegree(n) <= k or G.outDegree(n) <= k):
9              G.remove(n)
10    return G
```

The intuition behind the k-core is to obtain a subgraph of nodes who exchange at least a flow of value k. In our work of determining groups of countries, we use this algorithm by varying the threshold k in order to obtain different groups depending on the migration flow exchanged inside the members of the group.

About the complexity, we have the following decomposition :

- In the worst case[7], there is only one node removed from the graph at each iteration and there is finally no remaining node in the k-core, yielding $O(n)$ for this step.
- At each iteration of the while loop, we consider all nodes, giving $O(n)$.
- To compute the in-degree and the out-degree of every node, each edge has to be considered at most twice. We have $O(m)$ for this step with m the number of edges.

The time complexity of this algorithm is $O(n^2 \times m)$. According to Batagelj and Zaversnik [18], there exist algorithms with a better complexity, but in practice the algorithm that we described is sufficient and gives the results quickly enough. For the sake of simplicity, we keep this one. The Figure 4.4 illustrates the results obtained.

[6] It is verified when the in-degree and the out-degree of each node is superior than k.
[7] A bad combination of the graph and k

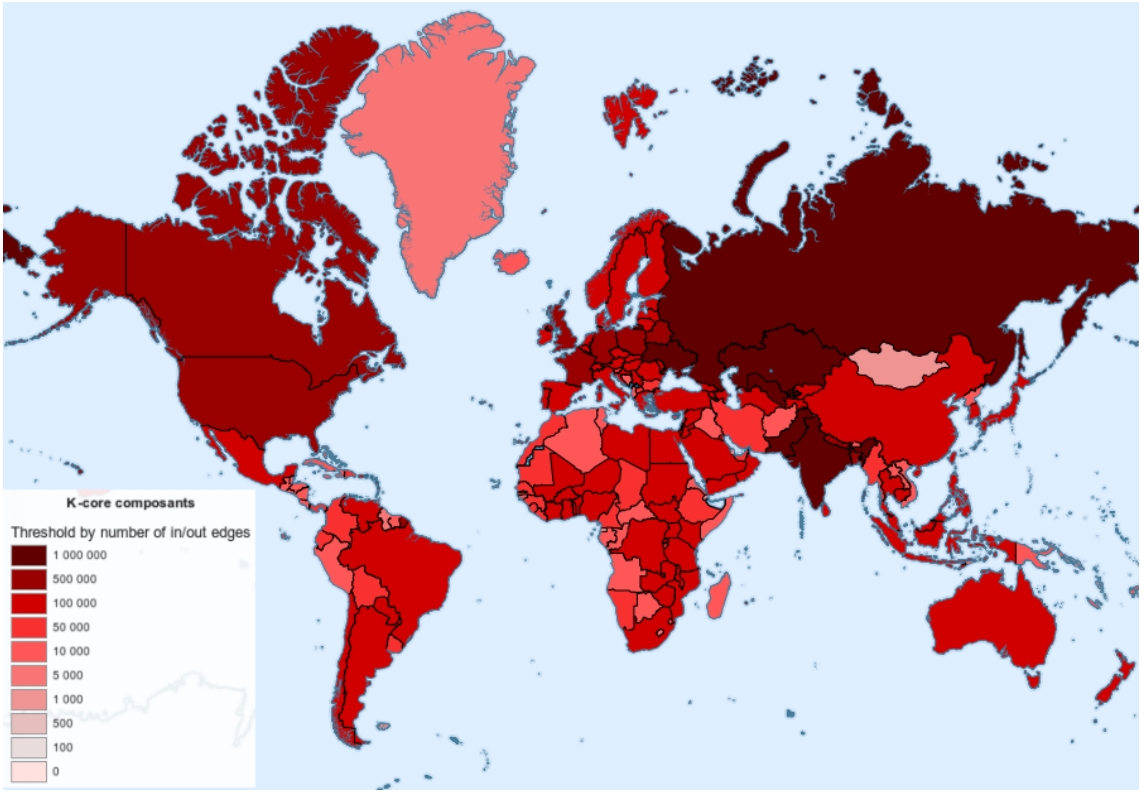


Figure 4.4: Map of k-cores for different k.

On this map, the darker is the color gradient, the higher is the k-core order and the stronger is the connexion between these countries. A preconceived idea may be to believe that the United states or Canada, present in the top rank of the PageRank in the analysis of Section 3.6, belong to the densest k-core. The map invalidate this idea. Instead, we observe a (1'000'000)-core with six countries : India, Russia, Kazakhstan, Pakistan, Russia, Ukraine, Uzbekistan.

Inside this group, it is interesting to see how the migrations are arranged. There are two possibilities, either the graph of the group is full, either it has more than one connected component. The detection of this arrangement is similar to find the strongly connected component in a graph with a fixed threshold for the edges. The threshold is necessary to prune the lowest edges of the graph. If we do not do it, they can parasite the computation of the strongly connected components : a migration of one hundred people is negligible comparing one of one million. However the threshold must be relatively low comparing to the highest edge of the group. If the value is too high, the components obtained can be irrelevant. The strongly connected components can be efficiently obtained with the Tarjan algorithm [146] with a complexity of $O(V+E)$ in space and time[8].

By proceeding like this, the possible subgroups can be easily obtained. the Figure 4.5 shows with the red edges the migrations inside the (1'000'000)-core by considering only the migrations bigger than 0.1% of the biggest flow [9] .

[8] With V the number of nodes and E the number of edges.
[9] With this pruning, we remove the small migrations.

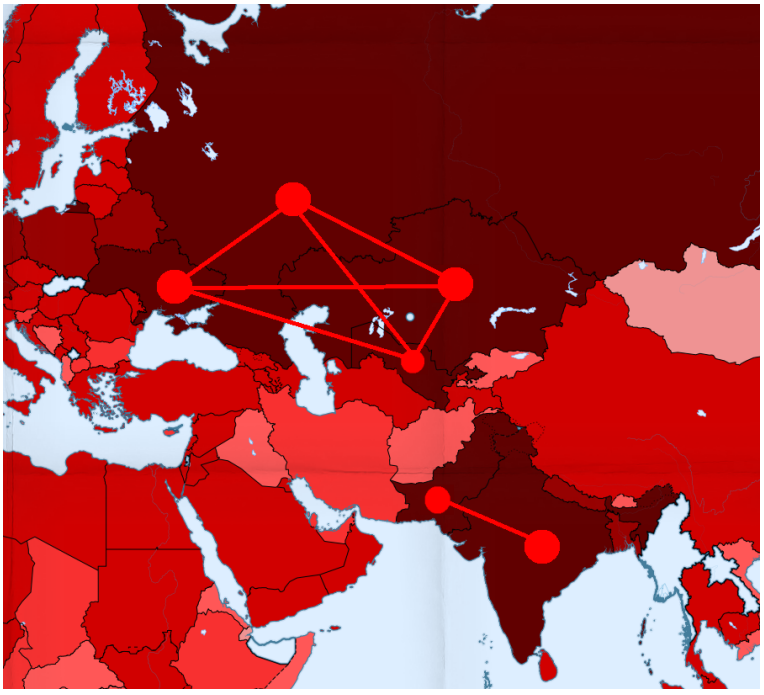
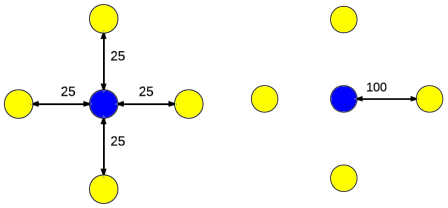


Figure 4.5: Migrations in the (1'000'000)-core.

The observation is direct, we have two strongly connected components, one with the India and the Pakistan and the other including Russia, Ukraine, Uzbekistan and Kazakhstan. Let us notice that there is no edge between the two groups, they are totally independent. On the one hand, it means that the populations exchange inside a group is huge : each country of the group has a strong import and export relation with the other members of the group. On the other hand, the exchange between the two groups is almost null. There is a plausible explanation of this configuration. Let us remember that the data corresponds to the migrations of the 1990-2000 decade, just during and after the dissolution of the USSR. Our analysis shows that during these years the ex-USSR countries kept an important relation. Many people are moving inside this group. Concerning the other group, India and Pakistan, we know that they are both populated countries and that their relation is important. Indeed, India and Pakistan were both part of the British Indian Empire and kept many relationship ever since [96].

Furthermore, one might ask why countries like the United States, known to be strong importer and exporter and well ranked, are not in the strongest k-core. The logic behind this is directly results from the structure of the k-core. Exchanging a large total flow to many countries is not sufficient to belong to a strong k-core. Let us illustrate this with the Figure 4.6 where we have two different configurations of exchange, each with the same total flow.



(a) distributed relation. (b) centralised relation.

Figure 4.6: Two configurations of exchange.

Let us consider the 100-core. On the Graph (a) each yellow countries are pruned because none of them have a sufficient weighted degree. Because these countries are not in the 100-core, the exchanges of the blue country with the yellows are not considered anymore. The blue country is thereby also pruned. On the Graph (b) even the total amount of exchanges is the same than the Graph (a), the exchanges are centralised between two countries allowing them to be on the 100-core. Thus, the difference between the United States and Russia is that the latter has its exchange relation more centralised with few countries while the exchanges of the former are more distributed. Following this idea, we understand the intuition behind the k-core and the differences between countries according to the k-core threshold.

Besides, we can observe interesting similarities between this map and the Map 3.7 obtained with the eigenvector centrality measure of the Section 3.5. Indeed, several countries being in the densest k-cores are the same than the ones having the best eigenvector values which means that these two concepts are related. Both of them have the purpose to locate the nodes having strong connections together. It is interesting to see that different ways to do the analysis can converge to similar results.

4.3.2 In-k-core

As we said previously, we can build a directed k-core only by considering the in-degree of each nodes.

Definition 4.3.3 (In-k-core)**.** *Given a directed graph $G = (V, E)$ and a threshold k, the directed in-k-core of G is the subgraph of G where all the nodes have a weighted in-degree higher than k. The out-degrees are not considered at all.*

The algorithm used is quite similar to the previous one. The only difference is instead considering the in-degree and the out-degree, we keep only the in-degree. Its complexity is also $O(n^2m)$.

```
1  // Input : G = Graph(V,E), threshold k
2  // Output : The in-k-core of G.
3  // The degrees consider the weight.
4
5  Kcore(G,k):
6     while(G.minInDegree <= k):
7        for all node n in G:
8           if(G.indegree(n) <= k):
9              G.remove(n)
10    return G
```

With the results obtained, we draw the Map 4.7.

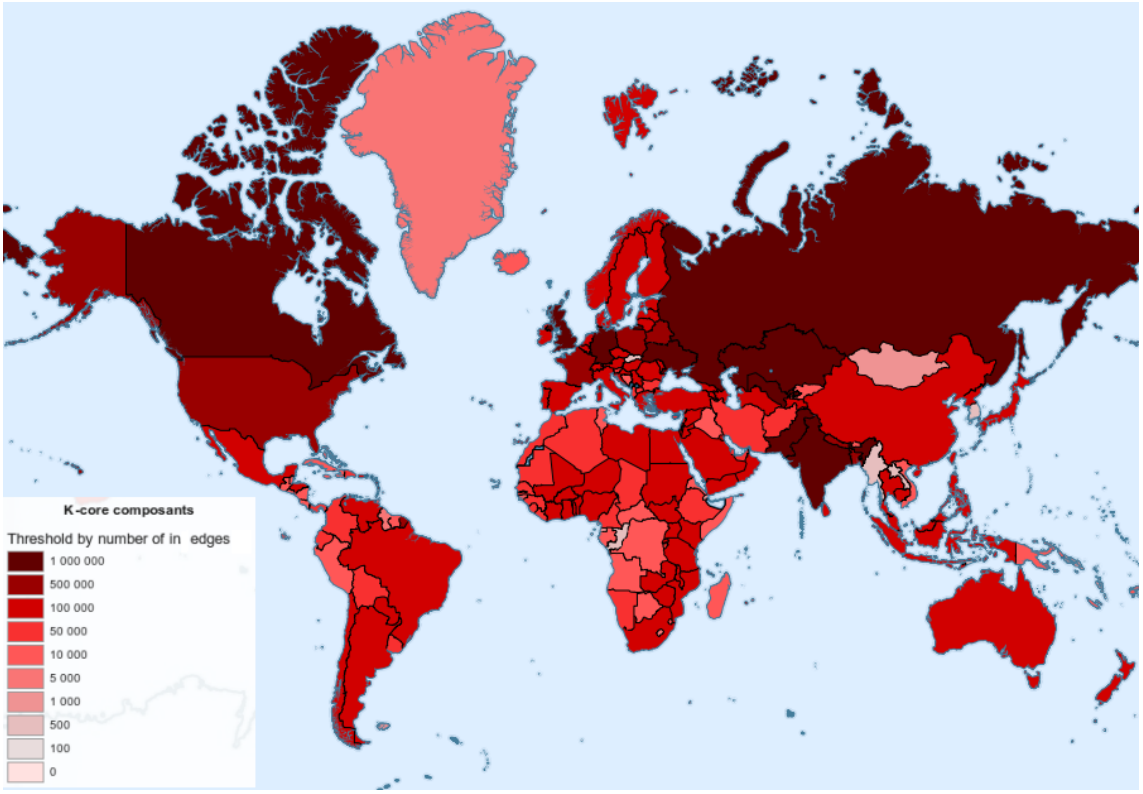


Figure 4.7: Map of in-k-core for different k.

The interpretation of the k-core behind this map is to highlight subgraphs of countries which receive at least a total migration flow of value k from countries in the same subgraph. Instead of considering together the in-degree and the out-degree, the condition is only based on the in-degree. Because the condition of belonging to the same k-core is less restrictive, we expect to have a denser map than the Map 4.4. The Map 4.7 where the color gradient is darker confirms this idea. For example, if we take the in-(1'000'000)-core, we have three more countries : Canada, Germany and United kingdom.

Analysing specifically the in-k-core enables to highlight such countries. All these countries have the particularity to have a strong in-relation with the other countries of the same core but the out-relation in it is not as dense. It is why they are pruned when we add the condition on the out-degree as showed on the Map 4.4.

4.3.3 Out-k-core

Like the in-k-core, we can do the same with the out-degree to obtain the out-k-core.

Definition 4.3.4 (Out-k-core)**.** *Given a directed graph $G = (V, E)$ and a threshold k, the directed out-k-core of G is the subgraph of G where all the nodes have a weighted out-degree higher than k. The in-degrees are not considered at all.*

Again, the algorithm used stays similar than previously, with the same time complexity $O(n^2m)$.

```
// Input : G = Graph(V,E), threshold k
// Output : The out-k-core of G.
// The degrees consider the weight.
```

```
Kcore(G,k):
    while(G.minOutDegree <= k):
        for all node n in G:
            if(G.outDegree(n) <= k):
                G.remove(n)
    return G
```

With the results obtained, we draw the Map 4.8.

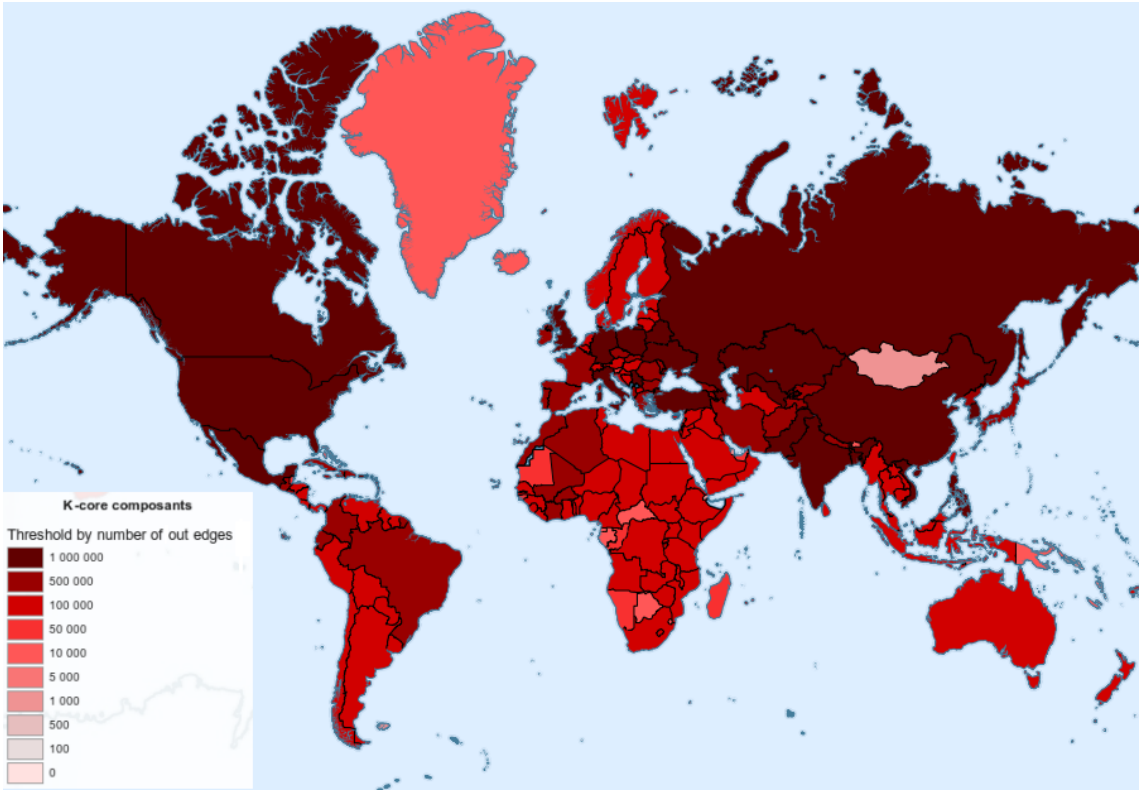


Figure 4.8: Map of out-k-core for different k.

Here, instead of exhibiting countries having a strong in-relation, we consider the ones that are connected according to their out degree. Comparing to the two previous (1'000'000)-core, we have many more countries that belong on it. From the nine of the in-(1'000'000)-core there are fourteen more : Azerbaijan, Bangladesh, Belarus, China, Italy, South Korea, Mexico, Philippines, Poland, Puerto Rico, Serbia, Turkey, United Stated and Vietnam. Moreover, by looking the maps, we can deduce that for the same threshold, the out-k-core is the densest core among the three considered on this section.

4.3.4 Global analysis

Now that the different k-cores are computed and illustrated, we can make a global analysis in order to highlight the differences between them.

Firstly, we can see that some countries belong to both in/out k-cores but not in the full k-core with the same threshold. At first sight, it can seem surprising given that the condition to be on the full k-core is based on the in and the out-degree. Let us take the example of Canada which is in both in/out-k-core but not in the full k-core. The problem is that even if this country respects

the two previous conditions, it is not necessarily the case of the countries with which Canada exchanges. If some of them countries are pruned, Canada loses a part of its score making it eligible to pruning. It is why the full k-core is not the simple intersection of the in and the out-k-core. Considering the full k-core provides thereby supplementary information.

Furthermore, we can identify different kinds of countries according to their different k-cores.

- Countries that belong to all strong k-core like Russia, Ukraine or Uzbekistan. Such countries have together a strong export and import relation, each receiving and sending a lot from/to others.
- Countries that belong to strong out-k-core but only to weak in-k-core like Slovakia. Such countries have a strong export relation with some countries but their import relation has a weaker amplitude.
- Countries that only belong to the small k-cores of the three maps like Mongolia. Such countries are quite independent. They do not have a strong export or import relation with a particular country.

Let us notice that there is no country that belongs to the strongest in-k-core but not to the out-k-core. Each of them are in the out-k-core as well. Furthermore, if we compare the general appearance of the maps, we can see that the out-k-core is the densest one. These two observations teach us a interesting general trend :

- Because the in-k-core map is less dense, the immigrants of a country come from diverse countries and were fairly distributed among these countries.
- Because the out-k-core map is denser, the emigrants of a country go to diverse countries and the distribution is more centralised to some countries.

For instance, it is the case of Mexico where 98% of emigrants go to only one country, the United states, while 98% of immigrants come from 35 different countries. It is important to mention that it is not a strict rule, just a general trend observed from the k-core maps.

4.4 Core - periphery

Another way to analyse and to separate countries into groups is the core-periphery model. In this model, the countries are split into two groups :

1. The core, containing the nodes that are tightly connected together.
2. The periphery, where the nodes have only few connections with the other periphery nodes.

The groups are formed only according to these two characteristics, the connections core-periphery or periphery-core do not matter. Typically a core-periphery structure has a "star shape" as shown on the Figure 4.9.

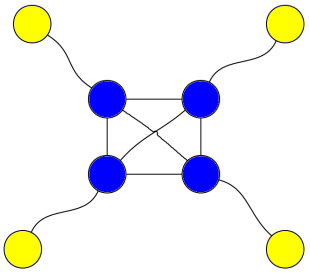


Figure 4.9: Star shape for a core(blue)-periphery(yellow) structure.

In contrast to the other methods, here we have two groups. In the literature different models of core-periphery have already been developed and mostly applied to unweighted undirected graphs [131, 136] and some to directed networks [46]. In our context, we would like to apply this method to the migration graph which is directed and weighted. To be best of our knowledge, the following model has not been developed in the litterature.

4.4.1 Application to unweighted undirected graph

Following the idea previously stated, the core-periphery structure encourages links between countries in the core and deters the links within the periphery. This problem can be formulated as an optimisation problem. To do so, let us define

$$\delta_{i,j} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are both in the core} \\ -1 & \text{if } i \text{ and } j \text{ are both in the periphery} \\ 0 & \text{otherwise} \end{cases}$$

Similarly, we can also define

$$v_i = \begin{cases} 1 & \text{if } i \text{ is in the core} \\ 0 & \text{if } i \text{ is in the periphery} \end{cases}$$

in order to have

$$\delta_{i,j} = v_i + v_j - 1.$$

From this expression we can raise the following objective function

$$\max_{\delta} \left(\sum_{i=1}^{N} \sum_{j=1}^{N} A_{ij} \delta_{ij} \right)$$

where N is the number of nodes in the graph and $\mathbf{A}$ the adjacency matrix of the graph. However, with this function we obtain a trivial solution where $\forall i \; v_i = 1$ which means that every country is put on the core and none in the periphery. The problem is that there is no penalty for a node to be in the core without being connected with other nodes in the core. Moreover, there is no gain to be in the periphery.

It is why this expression has to be refined. One way to do it is to use another matrix than the adjacency one :

$$\max_{\delta} \left(\sum_{i=1}^{N} \sum_{j=1}^{N} B_{ij} \delta_{ij} \right)$$

where $\mathbf{B}$ is a matrix related to $\mathbf{A}$ and expressed like this[10] :

$$\mathbf{B} = \mathbf{A} - \lambda$$

where $\lambda \in [0, 1]$. Intuitively λ discourages the situation where two core nodes are not connected and encourages two periphery nodes to do not be linked. For the extreme values of λ, we have the two following situations[11] :

1. $\lambda = \min \mathbf{A} = 0$: We fall in the previous situation.
2. $\lambda = \max \mathbf{A} = 1$: We find another uninteresting trivial solution where $\forall i \ v_i = 0$.

We determine two natural ways to define λ :

1. $\lambda = 0.5$: This case is natural because it gives a balanced table of gains as presented in the Table 4.3.

λ=0.5	Nodes linked	Nodes not linked
Both in the core	+0.5	-0.5
One in the core, one in the periphery	0	0
Both in the periphery	-0.5	+0.5

Table 4.3: Table of gains for λ=0.5.

2. $\lambda = \text{mean}(\mathbf{A})$: The intuition behind this value is, instead of having a constant penalty as previously, to have a penalty which depends on the density of the graph. Indeed, for an almost complete graph λ will be close to 1 and a single lack of edges can lead to do not be in the core. In the opposite case where the graph is almost empty, λ is close to 0 and any edges between two countries can lead them to be in the core.

In the general case, we have the Table 4.4 :

	Nodes linked	Nodes not linked
Both in the core	1-λ	-λ
One in the core, one in the periphery	0	0
Both in the periphery	-λ	1-λ

Table 4.4: Table of gains according to λ.

Such optimisation problem can be solved using AMPL C.6 with a particular solver. However there is a more efficient way to solve this problem. Indeed, as described the following development, the problem can be reduced to consider λ as a degree threshold determining if a node will belong to the core or not.

To highlight this, we develop the problem mathematically. We start with the following objective function :

[10] λ is a scalar, the operation consist to apply the subtraction of λ on every entries of $\mathbf{B}$.
[11] Let us remember that the graph is unweighted. A thereby only contains 0 and 1.

$$\max_{\delta} \sum_{i=1}^{N} \sum_{j=1}^{N} B_{ij}\delta_{ij}$$

where we can express δ in function of v :

$$\max_{v} \sum_{i=1}^{N} \sum_{j=1}^{N} B_{ij}(v_i + v_j)$$

$$\max_{v} \sum_{i=1}^{N} \sum_{j=1}^{N} \Big(B_{ij}v_i + B_{ij}v_j \Big).$$

Having $B_{ij}v_j$ is equivalent to have $B_{ji}v_i$ because having an edge from a core node to a periphery node gives the same gain than the opposite. We have

$$\max_{v} \sum_{i=1}^{N} \sum_{j=1}^{N} \Big(B_{ij}v_i + B_{ji}v_i \Big).$$

By putting this expression in matrix form, we obtain the expression

$$\max_{v \in \{0,1\}^N} \mathbf{1}_{1\times N}(\mathbf{B} + \mathbf{B}^T)\mathbf{v} \tag{4.1}$$

With $\mathbf{v}$ a column-vector $(N \times 1)$ containing all the v. From this expression, we can deduce that, as $\mathbf{B}$ is known, the solution consists to the following attribution :

$$v_i = \begin{cases} 1 & \text{if} \quad \big(\mathbf{1}_{1\times N}\big(\mathbf{B} + \mathbf{B}^T\big)\big)_i > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

Let us notice that $\mathbf{1}_{1\times N}\big(\mathbf{B} + \mathbf{B}^T\big)$ is a row-vector $(1 \times N)$. According to the binary nature of v, if the entry i of the previous vector is strictly negative, the only way to maximise the expression is to assign $v_i = 0$ and similarly if the vector is positive.

Besides, the adjacency matrix of an undirected graph is symmetric which yields

$$\mathbf{B} = \mathbf{B}^T = \mathbf{A} - \lambda.$$

We can thereby simplify our previous expression

$$\begin{aligned} \big(\mathbf{1}_{1\times N}(\mathbf{B} + \mathbf{B}^T)\big)_i &> 0 \\ \big(\mathbf{1}_{1\times N}(2\mathbf{A} - 2\lambda)\big)_i &> 0 \\ 2 \times \big(\mathbf{1}_{1\times N}\mathbf{A}\big)_i &> 2\lambda \\ 2\sum_{j=1}^{N} A_{ij} &> 2\lambda \\ \sum_{j=1}^{N} A_{ij} &> \lambda \end{aligned}$$

Intuitively, the last expression means that a node is added in the core if and only if it has a degree higher than λ. On the case where $\lambda = \text{mean}(\mathbf{A})$, it corresponds to the average degree. With the expression obtained, we do not have to use AMPL anymore. A simple algorithm computing the equation (4.2) can solve the problem.

```
1  // Input : A the adjacency matric, lambda the density parameter
2  // Output : a vector v containing the assignations for the N nodes.
3
4  corePeriph(A,lambda):
5       B = A - lambda
6       C = ones(1,n)*(B+transposed(B))
7       for all i in C:
8            if C_i >= 0:
9                 v_i = 1
10           else:
11                v_i = 0
12      return v
```

`ones(1,n)` is an unit raw vector of size n. Concerning the time complexity, the addition of matrices, the multiplication vector-matrix and the transpose matrix can be done on $O(n^2)$ with n the length of the matrix[12]. We obtain $\mathbf{C}$ on $O(n^2)$. Furthermore, we iterate once on every entries of $\mathbf{C}$ which can be done on $O(n)$. It gives finally a complexity of $O(n^2)$.

4.4.2 Application to weighted directed graph

Up to now, the analysis only concerned the unweighted and undirected graph. However our graph is weighted and directed. We need thereby to refine our model. Concretely, there are two differences to consider :

1. The adjacency matrix is not symmetric anymore. We therefore cannot state $\mathbf{B} = \mathbf{B}^t$.
2. The adjacency matrix is not binary anymore. The entries correspond to the weight of the edges connecting the nodes.

As previously, we obtain the equation (4.1) but we cannot simplify it in the same way. Indeed, when we develop it we obtain

$$\begin{aligned}
\left(\mathbf{1}_{1\times N}(\mathbf{B}+\mathbf{B}^T)\right)_i &> 0 \\
\left(\mathbf{1}_{1\times N}(\mathbf{A}-\lambda+\mathbf{A^T}-\lambda)\right)_i &> 0 \\
\left(\mathbf{1}_{1\times N}(\mathbf{A}+\mathbf{A^T})\right)_i &> 2\lambda \\
\sum_{j=1}^{N}\left(A_{ij}+A_{ji}\right) &> 2\lambda
\end{aligned}$$

The term $\sum_{j=1}^{N}\left(A_{ij}+A_{ji}\right)$ is equal to the sum of weighted in and out-edges for the node i. To belong to the core, this sum must be higher than 2λ. Applied to the migration graph, it corresponds to the number of immigrants and emigrants of the country. And so, with this expression we can obtain a core-periphery model for the migration graph in $O(n^2)$ with n the number of countries.

Now that the method is operational, let us apply it on the migration graph. The most suitable value for λ parameter is in this case mean($\mathbf{A}$). Indeed, given that the graph is weighted, the

[12] Our matrix is square because it is an adjacency matrix.

entries of $\mathbf{A}$ are not binaries anymore, and thereby $\lambda = 0.5$ does not have a particular meaning. the distribution is illustrated on the Map 4.10.

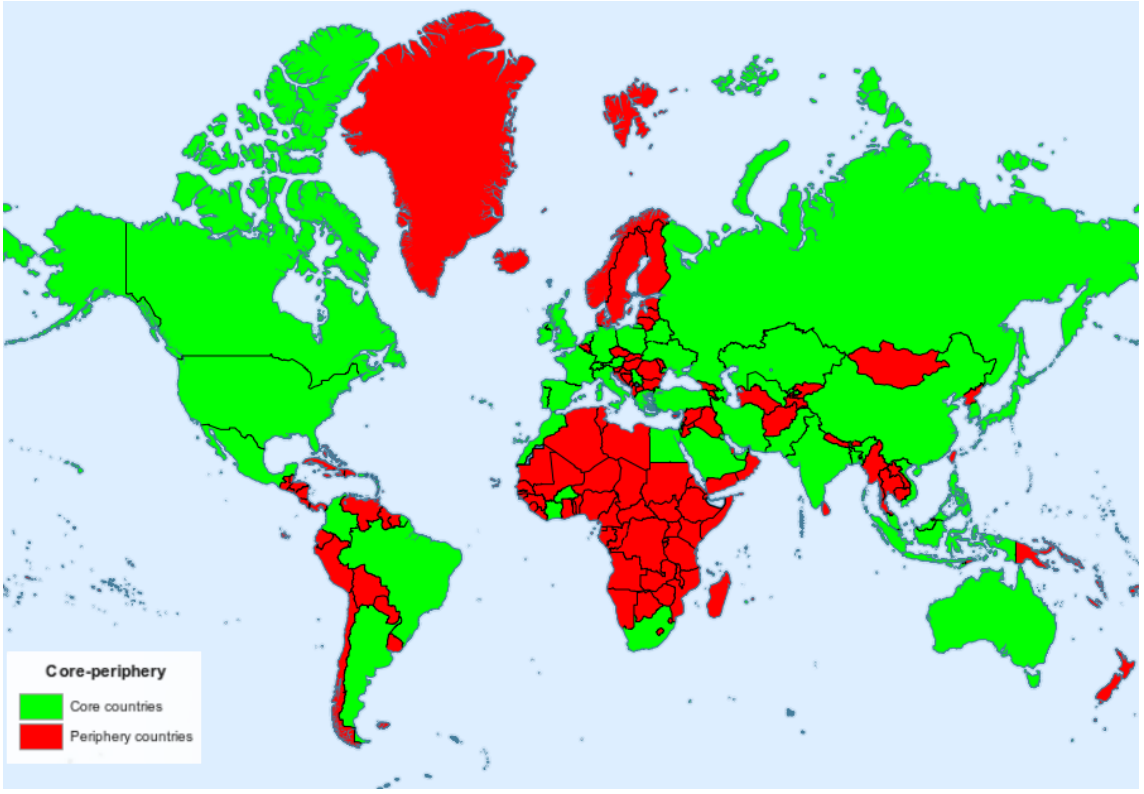


Figure 4.10: Core-periphery decomposition for $\lambda = mean(\mathbf{A})$.

Among the countries visible on the map, we can see that the core cover a large area and that the periphery is more localised. Indeed, we can identify some typical periphery areas like

- The main part of Africa, except some countries like Ivory Coast or South Africa.
- the Scandinavia.
- The main part of the Balkan peninsula except Greece and Serbia.
- The west of the south America.
- A set of countries at the east of India.

As expected, countries known to have important import of export flows like the United States or Russia are present on the core. Concerning the small countries not visible on the map, their flows are negligible compared the flows of other countries. For this reason all of them belong to the periphery. Generally speaking, because they have a lower degree compared to the populated countries, lowly populated countries are less likely to belong to the core.

Furthermore, to be sure about the correctness of our model, we compare it with an AMPL model (code available in Appendix B) which gave the same results.

4.5 Communities

As we said, there are different ways to group countries. A family of method is based on the concept of communities. To understand it, let us first informally define what a graph clustering is.

Definition 4.5.1 (Graph clustering)**.** *A graph clustering is a classification of the nodes of the graph into groups where the repartition of the nodes tends to optimise two things :*

- *The connections between nodes within the same group are strong.*
- *The connections between nodes of different groups are weak.*

That brings us to the definition of communities.

Definition 4.5.2 (Community)**.** *A community is a group of vertices found with a graph clustering method.*

Unlike the previous methods, this one has the following characteristics :

- Each edges are considered. There is no pruning.
- Every country will belong to one and only one group.
- The number of groups is not limited at two.

Besides its theoretical aspect, the community detection has diverse concrete applications in different fields like informatics, biology, sociology and many more [94, 107]. Our idea is to use it in the field of migrations.

First of all, we will study how we can efficiently detect communities in a general way and after that we will apply such methods on the migration graph.

4.5.1 Principles

As we said, the main task to form communities falls to form clusters where only the nodes inside a same cluster are tightly connected. We have to find an efficient way to do that.

There is a solution proposed by Newman and Girvan [120] which envisage the community problem with another point of view. They designed a metric, called the modularity, having the purpose of measuring the quality of graph partition into communities. The intuition behind this measure is to compare the density of the connections within a same community with the expected density for a same community partition [25]. The expected density means that we consider a randomised graph having the same number of nodes where every node keeps the same degree but where the edges are placed randomly. Mathematically it gives [103] :

$Q =$ `(Fraction of edges within the same community)` $-$ `(Expected fraction of such edges)`.

The higher is the modularity Q, The better is the partitioning. The problem of finding best communities turns thereby to maximise the modularity [69]. Firstly, we need to have a computable expression for Q. For a directed weighted graph such as the migration network, an expression of modularity is given by the formula [106] :

$$Q = \frac{1}{m} \sum_{i,j} \left[A_{ij} - \frac{k_i^{in} k_j^{out}}{m} \right] \delta(c_i, c_j) \tag{4.3}$$

with

- A_{ij} the weight of the edge from the node i to the node j.
- $m = \sum_{i,j} A_{ij}$, the sum of all the weighted edges.
- k_i^{in} the weighted in-degree of the node i.
- k_i^{out} the weighted out-degree of the node i.
- c_i the community of the node i.
- $\delta(c_i, c_j) = 1$ if i and j are in the same community, 0 otherwise.
- $\frac{k_i^{in} k_j^{out}}{m}$ correspond to the probability to have an edge from the node i to j in a random graph having the same configuration than ours.

All of these parameters are known. The task is now to find the partitioning producing the highest modularity Q.

A naive solution is to consider all the partitions and select the one having the highest Q. However, the problem of finding the optimal partitions is known to be NP-complete [31]. For this reason the solutions requiring to enumerate all the partitions are infeasible in practice.

4.5.2 Algorithm

The field of community detection has already been deeply studied. Several algorithms exist each with their own specificities. Here, we will use the algorithm proposed by Blondel et al. [25] called the Louvain method which is computationally efficient. Indeed, compared to the other algorithms, here the bottleneck is not the computation time but the size of the network which can exceed the storage capacity.

Let us see how the algorithm works. It is divided into two phases that are iteratively repeated until a maximum of modularity is reached. At the initialisation, we have a graph with N nodes where each node is in a different community. We begin thereby with as many communities as nodes. The following pseudo code shows the global structure of the algorithm.

```
1  // input : G = Graph(V,E)
2  // output : G with nodes assigned to a community
3
4  communityDetection(G):
5       Assign a different community to each node of G
6       while newQ > Q:
7            Q := G.modularity() // Compute the modularity of G
8            phase1(G)
9            G := phase2(G)
10           newQ := G.modularity() // Modularity of G after the iteration
11      return G
```

The idea is to apply the two phases until we obtain no higher modularity. The modularity is obtained with the formula (4.3). Let us describe the two phases.

Phase 1

The purpose of the first phase is to change the community of a node only if the modification results in a gain of modularity. Practically, it is done like this :

```
// input : G = Graph(V,E) where each node is in a community
// output : G with a partition with a better Q

phase1(G):
     while there is a community change improving the gain:
          select a node u of G
          consider all the neighbours v of u
          for all node v:
               compute the modularity gain by putting u into the community of v
          keep only the best gain obtained
          if the best gain is positive:
               change the community of v with the one of u
     return G
```

This phase stops thereby when there is no more way to improve the modularity only by assigning the community of a neighbour for every node. In other words we stop when a maximum value for Q has been found, which can be local.

Phase 2

Once the first phase is over, we directly enter into the second phase. Its purpose is to build a new graph where the nodes correspond to the communities found after the first phase. The process to do that is illustrated by the following pseudo code. But before, let us introduce the concept of supernode.

Definition 4.5.3 (supernode)**.** *Creating a supernode consists in including several nodes of a graph into a unique one called the supernode. Concerning the edges, if two nodes are in different supernodes, there is an edge of the same weight between the two supernodes. If they are in the same supernode, there is a self loop. Finally, every edge having the same source and destination are merged into one by taking the sum of each weight.*

In our case, we will have one supernode per community. The following pseudo code illustrates this idea:

```
// input : G = Graph(V,E) where each node is in a community
// output : An aggregated graph of G according the communities

phase2(G):
     for all nodes n in G:
          add n into the supernode corresponding to its community.
     create the graph G2 with all the supernodes.
     return G2
```

We obtain a new graph having less nodes than the previous one.

Considering the two phases together, the general principle of the Louvain algorithm is to iterate on increasingly smaller graphs in order to obtain a each iteration a more refined community partition.

Concerning the complexity, as Blondel et al. [25] said, this algorithm is extremely fast. Detect-

ing communities in a graph with 118 million nodes takes less than three hours. Moreover, given that the size of the graph decreases at each iteration, most of the computing time is localised on the first iterations.

Another advantage of this algorithm is that the communities are found incrementally. It gives a degree of freedom about the resolution desired. The resolution defines the number of communities obtained. To choose a good resolution, we firstly need to introduce new concepts in order to define exactly what is a good resolution.

4.5.3 Resolution choice

This issue has already been studied by Lambiotte, Delvenne and Barahona [102]. The resolution choice is tightly connected to the concept of Markov process. Let us assume that we have a random walker starting at any node of the graph and moving randomly only by following the edges connecting the nodes. The Markov process corresponds to the motion of the random walker. The period in which the walker can move is limited by the Markov time. In a graph it will be defined in function of the number of edges taken. Intuitively, the higher is the time, the further the walker can go on the graph. The Markov time corresponds to the value of the resolution. Besides, let us imagine that the walker can go far in the graph. If we want him to stay in the same community we need to have large communities. It is why a higher Markov time lead to finding larger communities.

The choice of resolution will thereby depend on this concept. Generally speaking, we want a robust and stable resolution. We have different factors for it :

- The number of communities remains identical for some close enough Markov times.
- The modifications on the graph structure, for example by adding a new country or changing the weight of an edge, does not generate important alteration into the partition. In other words, we want a low variation of information. The variation of information defines how the partitions are sensitive to such difference in the initial configuration. In practice [8] the Louvain algorithm [25] is ran several times and a check is done afterward to see if the partition obtained do not depend too much on the initial configuration.
- The stability of the partition for the resolution taken have a relative high value. Intuitively, a partition will have a high stability if the random walker remains in its starting community during the Markov time windows. The higher is the Markov time, the higher is the probability to leave the community and the stability will be thereby lower.

The implementation of an algorithm studying these factors has already been done by Delmotte, Schaub and Yaliraki [8] on Matlab and C++. the signature of the function is the following :

```
[S, N, VI, C] = STABILITY(G, T,'PARAM',VALUE)
```

This function computes the optimal partition of the graph `G` by optimising the stability at each Markov time in vector `T`. With more details, it returns a vector containing :

- `S`, a vector containing the values of the stability.
- `N`, a vector containing the numbers of communities.
- `VI`, a vector containing the variations of information. The more is this value, the more the partitions obtained are sensitive to the perturbation on the initial configuration.
- `C`, a vector containing the optimal partitions.

Moreover, this function uses the Louvain method [25] detailed previously. We apply it on our graph with the following function call :

```
> stability2(Graph2000,10.^[-2:0.01:2],'directed','plot')
```

Where `Graph2000` is the adjacency matrix of the migration graph. The parameter `directed` specifies that the graph is directed and `plot` forces a plot of the results. We obtain thereby the Graph 4.11.

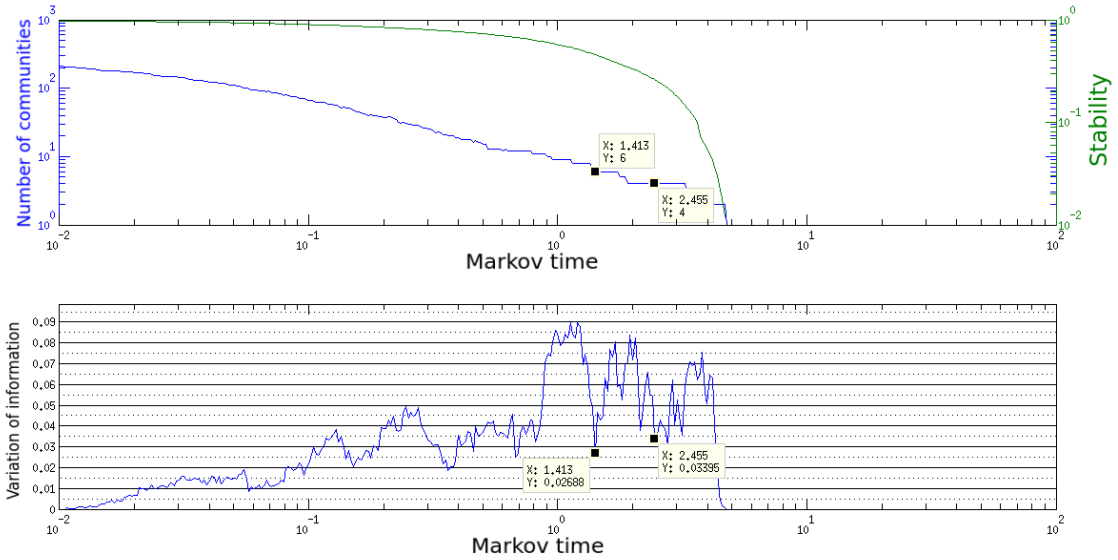


Figure 4.11: Analysis of the stability of the communities.

Thanks to these graphs, we have a more precise idea of which resolution to choose. Firstly, on the second graph we observe two falls of the variation of information for the Markov time 1.413 and 2.455. If we report these values on the first graph, we can see that the function forms a plateau for both values. The meaning behind these plateaus is that the number of communities, respectively 6 and 4, remains the same for some close Markov times. Moreover, the stability keeps a high value for both cases. Indeed, if we take a higher resolution, the stability begins to exponentially fall.

For all these reasons, we consider both resolutions for the study of communities in the migration graph.

4.5.4 Application

With the values obtained for the resolution, we apply twice the Louvain Method [25] with the Gephi (see Appendix C.10) implementation. It gives two partitions, one generating four communities and another yielding six. Let us analyse both cases.

4-community partition

First of all, we represent the different communities on a map, each with its own color. The Figure 4.12 shows the result obtained.

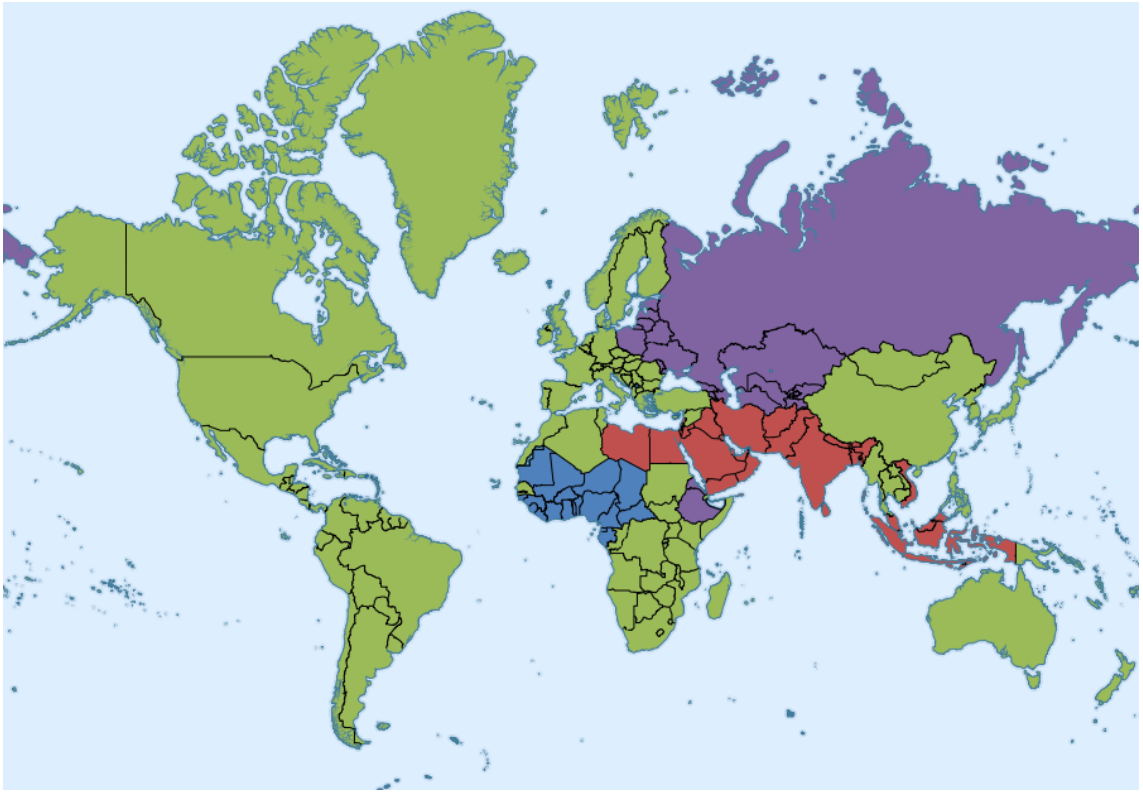


Figure 4.12: Communities map for a resolution of 2.455.

Just by looking this map, we can identify four groups :

- The so-called West Africa [85] with some neighbouring countries except the Gambia (D.2.29). Concerning this country, it has 60% of its immigration coming from the green community and more than 70% of its emigration to the green community. It explains why it is not in the blue community as its neighbours.

- A portion of the Middle East, the Indian subcontinent and the south of the Far East.

- The former USSR with Eritrea (D.2.23) and Ethiopia (D.2.24). The reason why these two countries are assigned to this community can be explained by some historic facts. Ethiopia has longstanding relations since the 17^{th} century with Russia[116, 129]. Eritrea was highly connected to Ethiopia during the 1990s (see Appendix D.2.23) and by this relation it is added to the purple community too.

- A last community taking over the rest of the world.

To be sure about the relevance of these communities, we check the fact that the countries within the same communities are tightly connected together and much less to the countries that belong to another community. The Graph 4.13 illustrates these results.

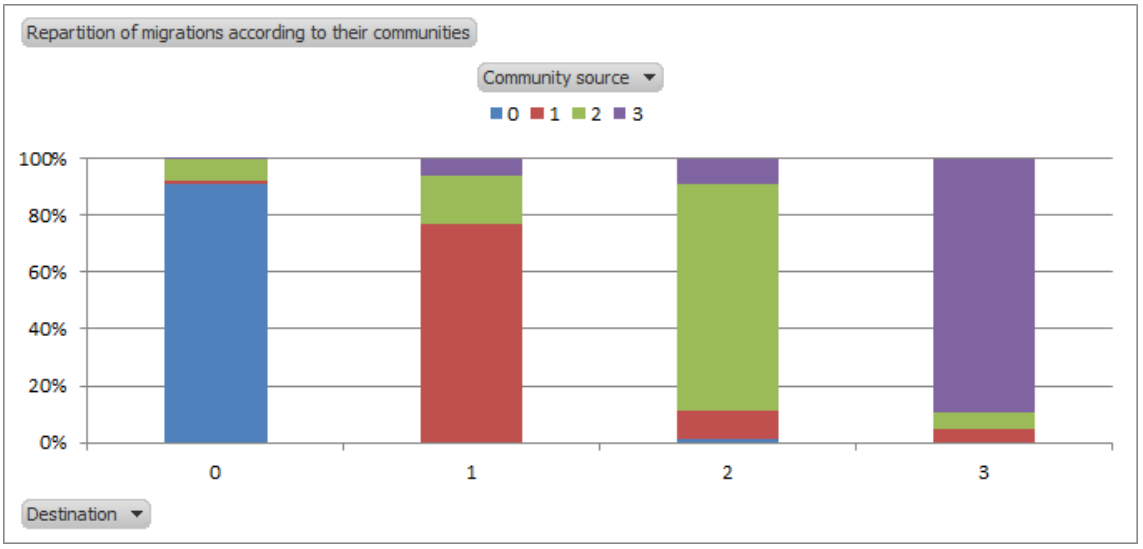


Figure 4.13: Repartition of total migration flow according to the 4-communities.

As expected, we can see that the main part of migrations for each community remains inside the community. This proves the strength and the relevance of the communities found.

6-community partition

Let us do the same process for the 6-communities. The Map 4.14 locates these communities.

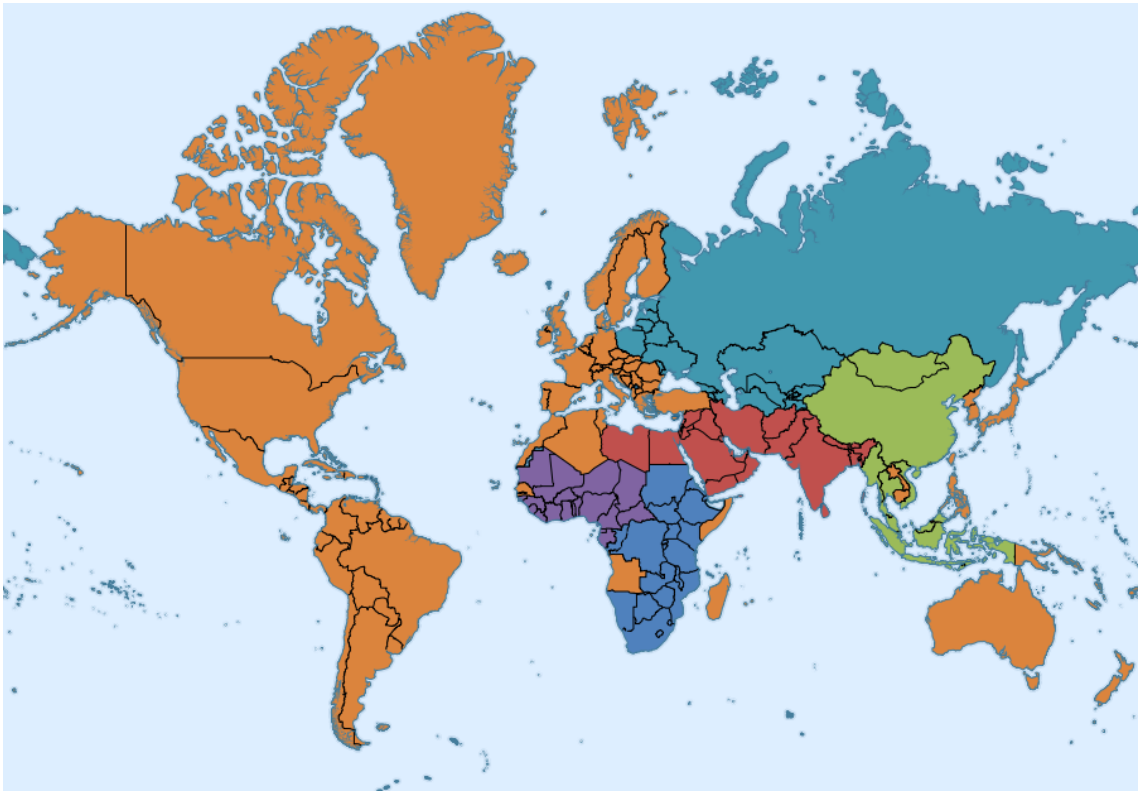


Figure 4.14: Communities map for a resolution of 2.455.

The six communities obtained are the following :

- The West Africa with some neighbouring countries.
- An important part of the sub-Saharan Africa without Madagascar, Angola and Somalia.
- A portion of the Middle East, the Indian subcontinent.
- The former USSR.
- Some countries on the Far East.
- A last community taking over the rest of the world.

As before, let us study the relevance of these communities. It is illustrated on Figure 4.15.

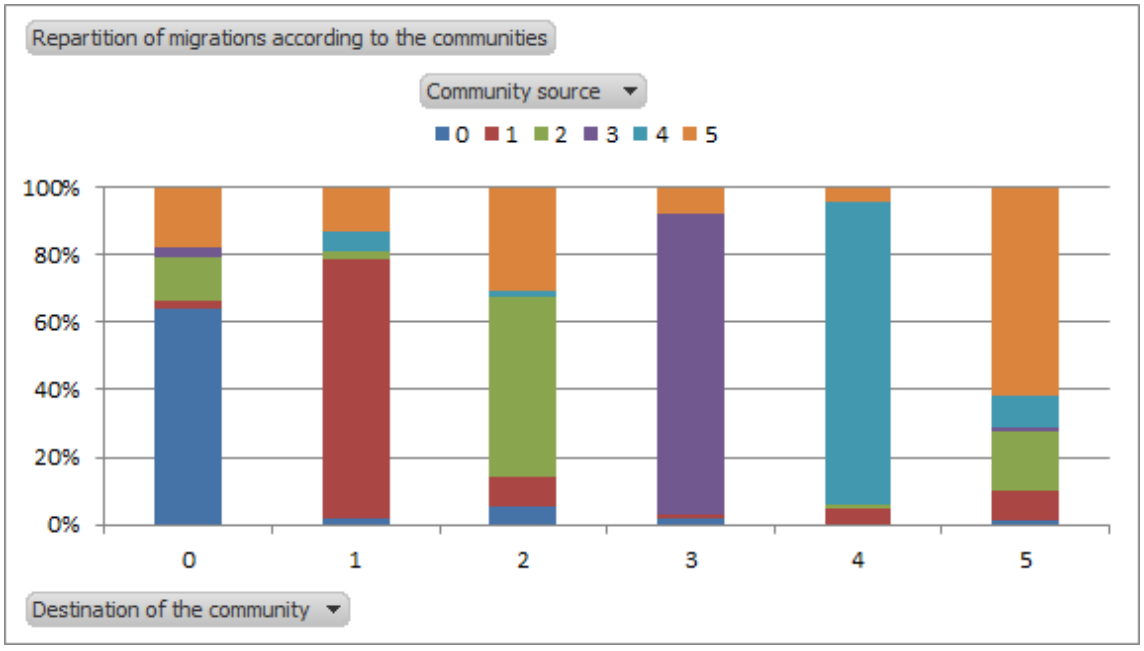


Figure 4.15: Repartition and importance of migrations according to their communities.

Again, we can see that the migrations intra-community remain the most important, proving the strength of the partition.

Comparison between communities

First of all, let us just compare the maps of the two communities. Starting from the 4-communities, we observe diverse things :

- The West Africa, the USSR and the Indian communities remain relatively similar. There are only some countries that are added or removed.
- The last community taking over the rest of the world is split into three communities, one located on sub-Saharan Africa, one the Far East and the last with the remaining countries.

It is interesting to see that, by passing from the 4 to the 6-community, it is the big community which is split and that the other remains stable. Another way to visualise it is to draw the

repartition of the migration flow of the 4-communities on the 6-communities as shown on the Graph 4.16.

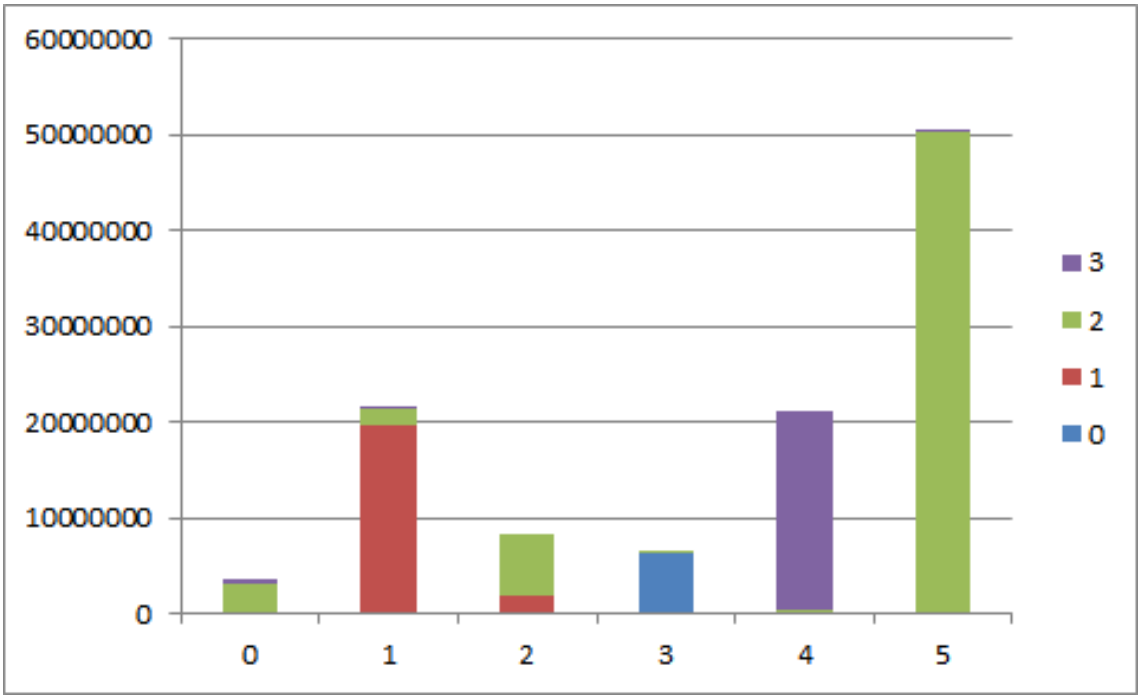


Figure 4.16: Repartition of the flows from the 6-communities according to the 4-communities.

As expected, the green flow, corresponding to the "rest of the world" is mainly scattered into three communities.

4.5.5 Relationship with the PageRank

On this section, we envisage the PageRank (Section 3.6) with a community point of view. But first, we have to introduce two new concepts.

Definition 4.5.4 (Local PageRank)**.** *The local PageRank of a country is the PageRank obtained by considering only the community of the country.*

Definition 4.5.5 (Community PageRank)**.** *The community PageRank is the PageRank obtained on the graph where all the nodes are merged into a supernode related to the community.*

As we just saw, the intra communities connections are strong compared to the extra connections. It will be interesting to see if the PageRank can be retrieved only by considering the local PageRank and the community PageRank. Our intuition tell us that, given the non-local effects are small compared to the local ones, the PageRank can be obtained by multiplying the community PageRank with the local PageRank. To do so, we need to compute these two PageRanks. We do it through the 4-communities.

Community PageRank

On this section we will compute the community PageRank. But before that, we represent the flows between the 4-communities on the Map 4.17.

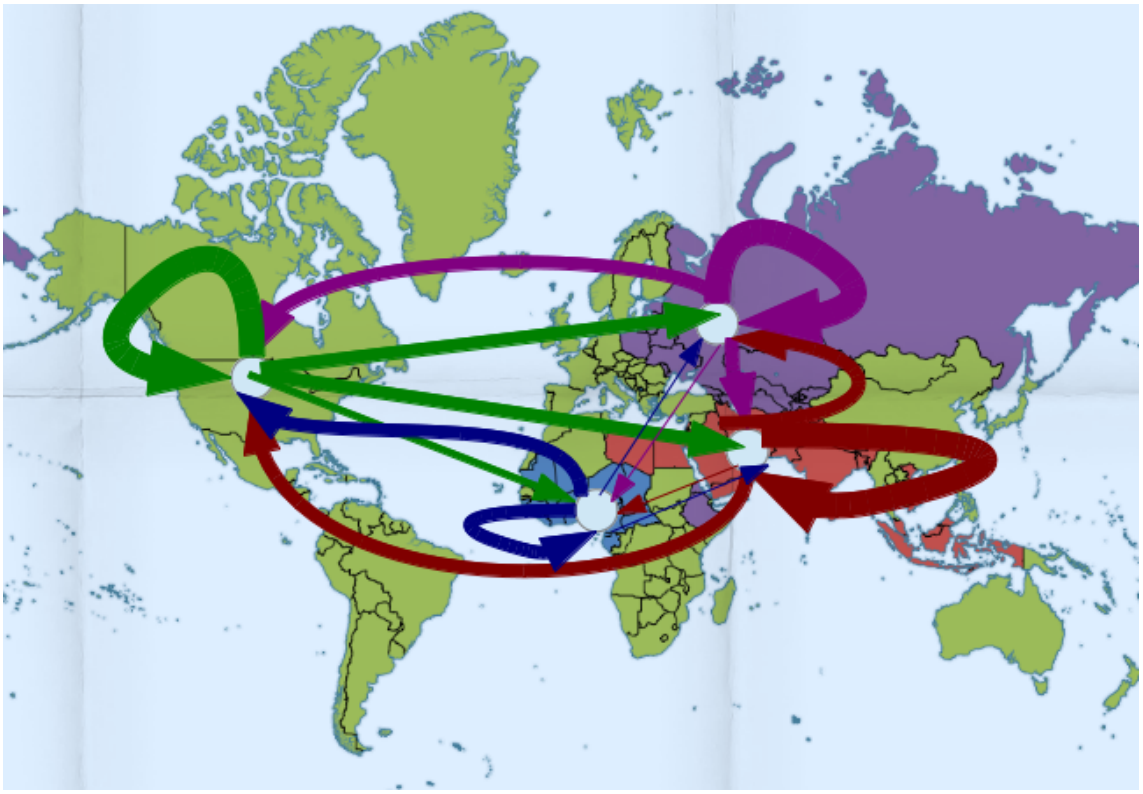


Figure 4.17: Map of flows between the 4-communities.

As expected, withing a same community, the self flow remains the most important. However, we can also see that the strength of a flow depends widely on the community. For example the strongest flow coming from the blue community is not as important as the flow of the green or the purple community.

By applying the same PageRank algorithm than in Section 3.6, we obtain the Ranking 4.5.

Rank	Community	Community PageRank
1	Rest of the world	0.6
2	Southern Asian	0.16
3	Ex-USSR	0.13
4	West Africa	0.11

Table 4.5: Ranking of the 4-communities according to their PageRank.

Local PageRank

Similarly, let us do the same for the local PageRank of each communities :

Green community (161 countries)			Purple community (18 countries)		
Rank	Country	Local PageRank	Rank	Country	Local PageRank
1	United States	0.215	1	Russian Federation	0.346
2	United Kingdom	0.059	2	Ukraine	0.183
3	Canada	0.059	3	Kazakhstan	0.103
4	Germany	0.048	4	Uzbekistan	0.052
5	Australia	0.046	5	Belarus	0.046
Red community (29 countries)			Blue community (19 countries)		
Rank	Country	Local PageRank	Rank	Country	Local PageRank
1	West Bank and Gaza	0.180	1	Ivory Coast	0.258
2	Saudi Arabia	0.112	2	Burkina Faso	0.203
3	Kuwait	0.094	3	Nigeria	0.078
4	India	0.078	4	Mali	0.066
5	United Arab Emirates	0.071	5	Cameroon	0.052

Table 4.6: Ranking of the local attractiveness according to the local PageRank.

Approximated PageRank

Now that we have the local and the community PageRank, we can compute the approximated PageRank. Let us first remember our goal. We want to obtain

$$\text{approximated PageRank} = \text{local PageRank} \times \text{community PageRank}.$$

To see if the value obtained is good, we can compute the error between the two PageRanks :

$$\Delta_{\text{PageRank}} = |\text{PageRank} - \text{approximated PageRank}|.$$

Ranking of the approximated PR	Country	Global PR Ranking
1	United States	1
2	Russian Federation	12
3	United Kingdom	3
4	Canada	2
5	Germany	4
6	Ivory Coast	14
7	West Bank and Gaza	7
8	Australia	5
9	France	6
10	Ukraine	32

Table 4.7: Comparison between the approximated and the real PageRank for the 4-community model.

By considering the Table 4.6, we can notice that this ranking gives strength to the important local attractor as Russian Federation, Ivory Coast or Ukraine. We can have an other point of view of global and local effects with the Table 4.8 below :

Ranking	Country	PageRank	Approximated PageRank	Δ_{PageRank}
1	United States	0.169	0.130	0.039
2	Germany	0.040	0.029	0.011
3	Canada	0.047	0.036	0.011
4	Syrian Arab Republic	0.011	0.001	0.010
5	United Kingdom	0.046	0.036	0.010
⋮	⋮	⋮	⋮	⋮
223	Kazakhstan	0.004	0.013	-0.010
224	Burkina Faso	0.009	0.022	-0.013
225	Ivory Coast	0.012	0.028	-0.016
226	Ukraine	0.006	0.024	-0.018
227	Russian Federation	0.013	0.045	-0.032

Table 4.8: Benchmark about the approximated PageRank, sorted decreasingly by Δ_{PageRank}.

In the table 4.8, $\Delta_{\text{PageRank}} > 0$ means that either the country is more attractive globally than locally, either the community the country belongs to is weak.

Similarly, $\Delta_{\text{PageRank}} < 0$, means that either the country is more attractive locally than globally, either the the country belongs to is strong. Given that the community PageRank of United States, Germany, Canada and United Kingdom is high (see Table 4.5), and as the community of the other countries in the Table 4.8 are low, we can make the following interpretation :

- United States, Germany, Canada and United Kingdom are better attractors globally than locally.
- Syrian Arab Republic combines both effects : its community is weak and the country is much more attractive globally than locally.
- Kazakhstan, Burkina Faso, Ivory Coast, Ukraine and Russian Federation are much more attractive locally than globally.

We can also develop the same idea with the 6-communities. As the partitioning is refined, we can hope to get better results.

Ranking of the approximated PR	Country	Global PR Ranking
1	United States	1
2	Canada	2
3	United Kingdom	3
4	Germany	4
5	France	6
6	West Bank and Gaza	7
7	Russian Federation	12
8	Australia	5
9	Mexico	8
10	Ivory Coast	14

Table 4.9: Comparison between the approximated and the real PageRank for the 6-community model.

As expected, the ranking is indeed better simulated than for the 4-communities. Local effects are still present but in a minor way. We can also check for the biggest global and local effects with the table below :

Ranking	Country	PageRank	Approximated PageRank	Δ_{PageRank}
1	United States	0.169	0.132	0.037
2	United Kingdom	0.046	0.033	0.013
3	Canada	0.047	0.035	0.012
4	Germany	0.040	0.031	0.009
5	Australia	0.036	0.028	0.008
⋮	⋮	⋮	⋮	⋮
223	Burkina Faso	0.009	0.017	-0.008
224	Ukraine	0.006	0.015	-0.008
225	Macao SAR	0.001	0.010	-0.009
226	Ivory Coast	0.012	0.022	-0.010
227	Russian Federation	0.013	0.028	-0.015

Table 4.10: Benchmark about the approximated PageRank, sorted decreasingly by Δ_{PageRank}

Macao SAR appears in the local attractors as it was part of the biggest community before and was not a local attractor in that group but it is a local attractor in its "new" community.

4.6 Conclusion

In this chapter we saw different methods to gather countries into particular groups having each their own particularities. We present here the main interesting results obtained :

- The balanced cycles method showed which countries have a balanced exchange of people. Through Maps 4.1 and 4.2 we saw that it is mainly the case for neighbour countries such as Ukraine, Poland and Belarus or the countries having a close PageRank ranking such as United Kingdom, Germany and Australia (Table 4.3).
- The k-core highlighted the countries having a strong relation of exchange together (Map 4.5). Furthermore, we learned also that the out-k-core is denser than the in-k-core which gives the intuition that the immigrants of a country came from diverse countries and were fairly distributed among these countries while the distribution of emigrants is more centralised to some countries. For example, 98% of the emigrants from Mexico go to only one country, the United States while 98% of immigrants come from 35 different countries.
- The core-periphery allows to see if the migration networks have a "star shape". Starting from an optimisation problem, the method we used to compute the core-periphery turned to be easier than expected. Indeed, the problem can be reduced to classify the countries according to a threshold on the number of emigrants and immigrants. The Map 4.10 shows that the periphery mainly consists on the one hand of countries relatively close together and on the other hand of countries lowly populated.
- The community model is the most detailed method on this chapter. Firstly, we analysed through the Markov time principle which resolutions lead us to have a stable and robust number of communities. We obtained two partitions, the first having four communities (Map 4.12) and the second six communities (Map 4.14).
- The purpose of the community model is to have groups where the nodes inside the same group are tightly connected to each other and where the connections between nodes in different groups are penalised. The Map 4.17 highlights this idea for the 4-community model.
- There is a way to link the community model with the PageRank ranking by approximating the PageRank by the PageRank within each community multiplied by the PageRank between the community.

- The approximated PageRank obtained with the 4-community models allows to give more importance to the countries which are good local attractors such as Ivory Coast by putting these countries higher in the ranking (Table 4.7).
- The approximated PageRank obtained with the 6-community models is a good approximation of the real PageRank as shown in the Table 4.9.

Chapter 5

Gravity model

"Gravity explains the motions of the planets, but it cannot explain who sets the planets in motion."

– Isaac Newton

5.1 Motivation

The last aspect considered in this work is about predicting migrations using different parameters and characteristics of countries. Unlike the previous aspect, this one has been highly studied in the migration literature and several methods already exist on this purpose. Among these methods, several are related to the gravity model [11] which uses an analogy with the Newton's law of Gravitation [121] to predict flows between entities. The gravity model is commonly used in various social sciences like economics through the national and international trades [58, 12, 65, 51] and also migrations [13, 135, 105].

There are two main families of gravity models :

- The bilateral gravity models [11] where only the origin and the attractiveness of the destination are considered.
- The multilateral gravity models [22, 33] where we do not take only the origin and the attractiveness of the destination into account but also the attractiveness of the alternative destinations.

On this report, we design bilateral gravity models. To elaborate such models, we thereby need to have an expression defining the attractiveness between entities. In the case of migration, it is mainly defined by some relevant characteristics of countries and the relations between them. Depending on the model, we can for example consider :

- The diasporas[1] present in the countries [20].
- The effect of colonial relations [80].
- The climatic factors [115].
- The genders [56].

[1] In the sense of a group of people coming from a country and living in another

- The education level [56].
- Economic factors [78].

Several gravity models using these kinds of parameters have already been developed [13, 105]. Such models have three mains purposes :

- Predicting migrations in the future.
- Applying it in the present to find missing migration data.
- Understanding the main factors that explain the migration flows.

This chapter is organised as follows :

1. Firstly we will explain the technical aspects to consider when building a gravity model.
2. Secondly, we will present some existing gravity models for migrations.
3. From the previous models, we will develop innovative gravity models having parameters that have never been used on this purpose such as the PageRank and a community factor.
4. Finally, we will use one of our previous model to try to predict future migrations but it will not be as succesful as excepted.

As for the previous chapters, we base our work on the migration data of the World Bank [169].

5.2 Mathematical concept of gravity model

As the gravity models generalises the Newton's formula, it can be written as :

$$F_{ij} = \alpha_0 A_{1_{ij}}^{\alpha_1} A_{2_{ij}}^{\alpha_2} A_{3_{ij}}^{\alpha_3} \dots A_{N_{ij}}^{\alpha_N} E_{ij}$$

with

- F_{ij} is the flow we want to analyse.
- α_0 a constant.
- $A_{i_{i=1:N}}$ the parameters used to explain the flow.
- $\alpha_{i_{i=1:N}}$ the exponents of the different parameters.
- E_{ij} is a term of error with an expectation of 1 ($\mathbb{E}(E_{ij}|A_{1_{ij}}, \dots, A_{N_{ij}}) = 1$).

This expression is commonly written on a log-log base [141] in order to have a linear form to use the LS (Least Squares regression 5.3.1). We have therefore :

$$\ln F_{ij} = \ln(\alpha_0) + \alpha_1 \ln(A_{1_{ij}}) + \alpha_2 \ln(A_{2_{ij}}) + \alpha_3 \ln(A_{3_{ij}}) + \dots + \alpha_N \ln(A_{N_{ij}}) + \ln(\epsilon_{ij}).$$

An alternative expression of the equation is :

$$F_{ij} = \exp\Big(\ln(\alpha_0) + \alpha_1 \ln(A_{1_{ij}}) + \alpha_2 \ln(A_{2_{ij}}) + \alpha_3 \ln(A_{3_{ij}}) + \dots + \alpha_N \ln(A_{N_{ij}}) + \ln(\epsilon_{ij}) \Big).$$

We will come back later on the advantages of each of the expressions.

5.3 Concept of regression analysis

The issue is to obtain a unique gravity equation from the whole data set that we have. To overcome it, we can use the concept of regression.

Definition 5.3.1 (Regression analysis)**.** *"Statistical approach to forecasting change in a dependent variable (sales revenue, for example) on the basis of change in one or more independent variables (population and income, for example). Known also as curve fitting or line fitting because a regression analysis equation can be used in fitting a curve or line to data points, in a manner such that the differences in the distances of data points from the curve or line are minimised."*[2]

In the case of migrations, we want to predict a flow between two countries according to a set of information about the countries involved.

$$F_{ij} \approx f_{A_1,\dots,A_N}(I_i, I_j),$$

where f is a function depending on a few parameters and I_i is the set of information about country i.

$$\begin{cases} F_{1,1} = & f_{A_1,\dots,A_N}(I_1, I_1) + \epsilon_{1,1} \\ & \vdots \\ F_{1,n} = & f_{A_1,\dots,A_N}(I_1, I_n) + \epsilon_{1,n} \\ & \vdots \\ F_{n,n} = & f_{A_1,\dots,A_N}(I_n, I_n) + \epsilon_{n,n}. \end{cases}$$

The purpose of the regression is to minimise $\|\epsilon\|$, the norm of the errors. Several norms can be used according to the kind of regression we are using. We present here two main regression methods.

5.3.1 Least Squares regression of a linear equation

In the case of linear regression, f is a linear function of the information :

$$F_{ij} \approx \alpha_0 + \alpha_1 A_1(I_i, I_j) + \cdots + \alpha_N A_N(I_i, I_j)$$

We obtain therefore, the following system :

$$\begin{cases} F_{1,1} = & \alpha_0 + \alpha_1 A_1(I_1, I_1) + \cdots + \alpha_N A_N(I_1, I_1) + \epsilon_{1,1} \\ & \vdots \\ F_{1,n} = & \alpha_0 + \alpha_1 A_1(I_1, I_n) + \cdots + \alpha_N A_N(I_1, I_n) + \epsilon_{1,n} \\ & \vdots \\ F_{n,n} = & \alpha_0 + \alpha_1 A_1(I_n, I_n) + \cdots + \alpha_N A_N(I_n, I_n) + \epsilon_{n,n}. \end{cases}$$

The euclidian norm $\|e\|_2$ is commonly used in this situation because ordinary Least Squares estimator is strongly consistent under minimal assumptions [101] and because it is the best unbiased linear estimator [3] [132].

As we want to build a linear gravity model, the formula is :

$$\ln F_{ij} = \ln(\alpha_0) + \alpha_1 \ln(A_{1_{ij}}) + \alpha_2 \ln(A_{2_{ij}}) + \alpha_3 \ln(A_{3_{ij}}) + \cdots + \alpha_N \ln(A_{N_{ij}}) + \ln(\epsilon_{ij}).$$

[2] Definition from the Business Dictionary [34].
[3] With the lowest variance of the estimate.

5.3.2 The zero-value and heteroscedasticity problems

The problem of this expression is that we cannot consider the zero-value of F_{ij}. Indeed, the logarithm of zero is not a finite value but we would like our regression model to take account of as many observations as possible. In the migration data set, half of the migrations have a zero-value. Moreover, a problem that we can not solve concerns the zero-values in the parameters. This can happen when there is a data missing for a factor or a past zero-migration between two countries. As most of the expressions are logarithmic, the observations are not taken into account in these situations as well.

Another major issue in regression is the presence of heteroscedasticity, the non-homogeneity of the variance of the predicted values[4]. The problem is that it is hard to predict and to guess the form of this heteroscedasticity. Heteroscedasticity does not change the bias of the regression but it biases the standard errors [52]. Silva and Tenreyro [141] proposed several estimators based on different forms of the heteroscedasticity. They compared them in different cases and showed that the PPML (Poisson pseudo-maximum likelihood) gives the best results, also compared to LS (Least Squares) regression. Furthermore, they proved that the data do not have to be Poisson at all and that the model naturally deals with the zero-value problem of the dependent variable.

This is the reason why we use this method as well.

5.3.3 Poisson Regression - Maximum likelihood

The Poisson regression is initially based on the idea that the dependent variable (the variable we want to explain) follows a Poisson distribution and that its expected value can be modeled by a linear combination of parameters (the independent variables) :

$$\ln(\mathbb{E}(Y|\mathbf{X})) = \alpha + \beta'\mathbf{X} = \theta'\mathbf{X}.$$

It is sometimes also called the log-linear model. The Poisson model can be written also as :

$$\mathbb{E}(Y|\mathbf{X}) = \exp(\theta'\mathbf{X}).$$

As previously said, this formulation avoids the problem of the zero-values and the parameters θ can be estimated by maximum likelihood which must be computed by numerical methods (see [45] for more details).

5.4 Existing models

Several gravity models have been developed in the literature. We are going to present some of them to see which are the common criteria in the several models.

The convention we use is that if a data is on a log-base, the variable is written in lower case [5]. Otherwise the name of the variable is written in capital case [6].

[4] This can be formally shown through methods such as the BreuschPagan test [52].
[5] For the sake of consistency, even if the names of the variables differ in the literature, we use the same notation.
[6] An exception is made for com_lang and com_bound which are used in our model.

5.4.1 Lewer and Van den Berg[105]

This first model takes mainly socio-economics factor into account : the population of the two countries, the ratio of per capita incomes, the distance between the countries, a criteria of diaspora but also if they share a common language, border or a historical colony, ... It is expressed like this :

$$\begin{aligned} \text{lmig}_{ij} = \quad & a_0 + a_1(\text{lpop}_i\text{lpop}_j) + a_2\text{rely}_{ij} + a_3\text{ldist}_{ij} + a_4\text{stock}_{ij} + a_5\text{com\_lang}_{ij} + \\ & a_6\text{com\_bound}_{ij} + a_7\text{LINK}_{ij} + a_8\text{rlaw}_{ij} + a_9\text{property}_{ij} + u_{ij}. \end{aligned}$$

where :

- lmig_{ij} is the (log of the) number of migrants between country i and j.
- a_i are the coefficient of the different parameters.
- lpop_i is the (log of the) population of country i.
- rely_{ij} is the (log of the) ratio of destination to source country per capita incomes.
- ldist_{ij} is the (log of the) distance between country i and j.
- stock_{ij} is the number of source country natives already living in the destination country.
- com_lang_{ij} is a dummy variable for common languages. It is equal to 1 if countries i and j share a common official language and 0 otherwise.
- com_bound_{ij} is a dummy variable for common borders. It is equal to 1 if there is a common border between i and j and 0 otherwise.
- LINK_{ij} is a dummy variable for colonial histories. It is equal to 1 if countries i or j was a colony of the other one and 0 otherwise.
- rlaw_{ij} is the (log of the) ratio of indices quantifying how well destination and source countries adhere to the rule of law.
- property_{ij} is the (log of the) ratio of indices quantifying how well destination and source countries protect property rights.
- u_{ij} denotes a random error term that we want to minimise.

Another model developed by Lewer and Van den Berg [105] contains the gross secondary education enrollment ratio :

$$\begin{aligned} \text{lmig}_{ij} = \quad & a_0 \ + a_1 \ (\text{lpop}_i\text{lpop}_j) + a_2 \ \text{rely}_{ij} + a_3 \ \text{ldist}_{ij} + a_4 \ \text{stock}_{ij} + a_5 \ \text{com\_lang}_{ij} + \\ & a_6 \ \text{com\_bound}_{ij} + a_7 \ \text{LINK}_{ij} + a_8 \ \text{human}_j + u_{ij}. \end{aligned}$$

where human_j is the gross secondary education enrollment ratio in the source country.

For 2710 observations of migrations [7] and with respectively these 10 and 8 parameters, they obtain a R-squared (see Appendix A.1 for more details about the R-squared.) of 0.663 and 0.662 through a LS regression.

[7] Database : from all countries to 16 OCDE countries between 1991 and 2000.

5.4.2 Ramos and Surinach [133]

In this model, Ramos and Surinach distinguish the effect of the two populations, they take the areas of the two countries into account and they divide different aspects concerning the historical colonies and the spoken languages.

$$\begin{aligned}\text{lmig}_{ijt} = \; & a_1\ \text{lpop}_{it} + a_2\ \text{lpop}_{jt} + a_3\ \text{ldist}_{ij} + a_4\ \text{area}_i + a_5\ \text{area}_j + \\ & a_6\ \text{com\_bound}_{ij} + a_7\ \text{com\_lang}_{ij} + a_8\ \text{LANGETHNO}_{ij} + a_9\ \text{LINK}_{ij} + \\ & a_{10}\ \text{COMCOL}_{ij} + a_{11}\ \text{LINK45}_{ij} + a_{12}\ \text{gdp}_{jit} + \text{fixed effects} + u_{ijt}.\end{aligned}$$

where :

- area_i is the (log of the) area of country i.
- LANGETHNO_{ij} is a dummy variable for common languages. It is equal to 1 if countries i and j share a common language spoken by at least 9% of the population in both countries and 0 otherwise.
- COMCOL_{ij} is a dummy variable for colonial histories. It is equal to 1 if countries i or j have had a common coloniser after 1945 and 0 otherwise.
- LINK45_{ij} is a dummy variable for colonial histories. It is equal to 1 if countries i or j have had a colonial relationship after 1945 and 0 otherwise.
- gdp_{ji} is the (log of the) ratio of GDP per capita of countries j and i.
- 'fixed effects' :
  - year fixed effects to control for common time shocks.
  - country fixed effects to account for time-invariant unobserved heterogeneity[8].

They model the data between 1960 and 2010 through time series. The purpose is to build a model with evolving parameters (in this case, the populations and the GDP change from one year to another). With the 'fixed effect', they obtain a R-squared of 0.634 but only a R-squared of 0.438 without.

Besides they also developed a similar model adding a country effect and considering only intra-EU migrations with which they obtained a R-squared of 0.834.

5.4.3 Artuc, Docquier, Ozden and Parsons [13]

In this model, Artuc, Docquier, Ozden and Parsons consider the past migrations, the labor force participation rate, the fact that English is an official language but also arbitrary criteria such as the fact that the military service is compulsory or not. A particularity of this model is that they divide the population according to their gender and according to their educational level[9] which allows them to add other variables.

$$\begin{aligned}\text{lmig}_{ij} = \; & a_0 + a_1\ \text{LANG}_{ij} + a_2\ \text{com\_bound}_{ij} + a_3\ \text{ldist}_{ij} + a_4\ \text{LINK}_{ij} + a_5\ \text{diaspora}_{ij} + \\ & a_6\ \text{ENGLISH}_j + a_7\ \text{gdp}_j + a_8\ \text{fertility}_j + a_9\ \text{skillworkforce-male-or-female}_j + \\ & a_{10}\ \text{laborforce-male-or-female}_j + a_{11}\ \text{LABORPARTICIPATION-MALE}_j + \\ & a_{12}\ \text{LABORPARTICIPATION-FEMALE}_j + a_{13}\ \text{MILITARYSERVICE}_j + \\ & a_{14}\ \text{POLYGAMY}_j + a_{15}\ \text{GCC}_j.\end{aligned}$$

where :

[8] This is commonly used in panel data analysis for time series regressions [111].

[9] Low or high skilled.

- diaspora$_{ij}$ is the number of migrants from i to j 30 years before.
- ENGLISH$_j$ is a dummy variable equal to 1 if the destination country speaks English.
- gdp$_j$ is the per capita income of the destination country in PPP.
- fertility$_j$ is the total fertility rate in the destination country.
- skillworkforce$_j$ is the share of the destination country workforce that is tertiary educated (by gender).
- laborforce$_j$ is the population aged 25 and over in the destination country (by gender).
- LABORPARTICIPATION$_j$ is the labor force participation rate in the destination country (by gender).
- MILITARYSERVICE$_j$ is a dummy variable equal to 1 if military service is compulsory in the destination country.
- POLYGAMY$_j$ is a dummy variable equal to 1 if polygamy is legally or socially accepted in the destination country.
- GCC$_j$ is a dummy variable equal to 1 if the destination country belongs to GCC (Gulf Cooperation Council).

With fifteen explaining factors, a separation between genders and qualification of people, they obtain a R-squared between 0.872 and 0.898.

5.5 Our gravity model

Our motivation is to build a gravity model which is general and easy to explain. For example, taking account the polygamy into account, the military service rate or the fertility seems more arbitrary than taking account of the GDP or the population. We do not want to target particular parts of the world (such as the GCC) either.

To test the different explaining factors, we develop a script to test all the combinations as there may be correlations between the different parameters. To have a robust model and to avoid problems presented in Section 5.3.2, we use the heteroscedasticity-consistent standard error option of Stata.

First of all, we test the Least Squares regression[10] A.2.2. Then, we take the best combinations of parameters with different thresholds of p-value[10]. We develop a model containing the past migrations but also a model without them. Indeed, even if taking the past migrations gives better results, it is interesting to do not take them into account to show the importance of the different factors that push people to migrate. After that, we do the same reasoning with the Poisson RegressionA.2.3. The different models we developed are in the Appendix A.

Here follows the list of variables we preselected before any analysis that do not appear in any other model. The complete list of the 23 variables we use is available in the Appendix A.2.1.

- com_orig_l~g$_{ij}$: dummy variable equal to 1 if destination and source country share official languages which have the same root, 0 otherwise.
- lpr1$_i$: PageRank (log) of the source country - attractiveness.

[10] A recap of the statistical concepts are available in Appendix A.1.

- lprinv1_i : Inverted PageRank (log) of the source country - repulsiveness.
- lpr2_j : PageRank (log) of the destination country - attractiveness.
- lprinv2_j : Inverted PageRank (log) of the destination country - repulsiveness.
- ldensity1_i : Density (log) of the source country $= \frac{\text{population}}{\text{area of the country}}$.
- ldensity2_j : Density (log) of the destination country $= \frac{\text{population}}{\text{area of the country}}$.
- lidh1_i : HDI (log) of source country.
- lidh2_j : HDI (log) of destination country.
- issamecom6_{ij} : dummy variable equal to 1 if destination and source country belong to the same 6-community, 0 otherwise.
- issamecom4_{ij} : dummy variable equal to 1 if destination and source country belong to the same 4-community, 0 otherwise.

Let us notice that we do not split the genders of the people in our model. Indeed, if we look at the proportion of males and females moving to each country (Figure 5.1 where each line represents a different country), it is globally equivalent and we do not want to take particular effects into account.

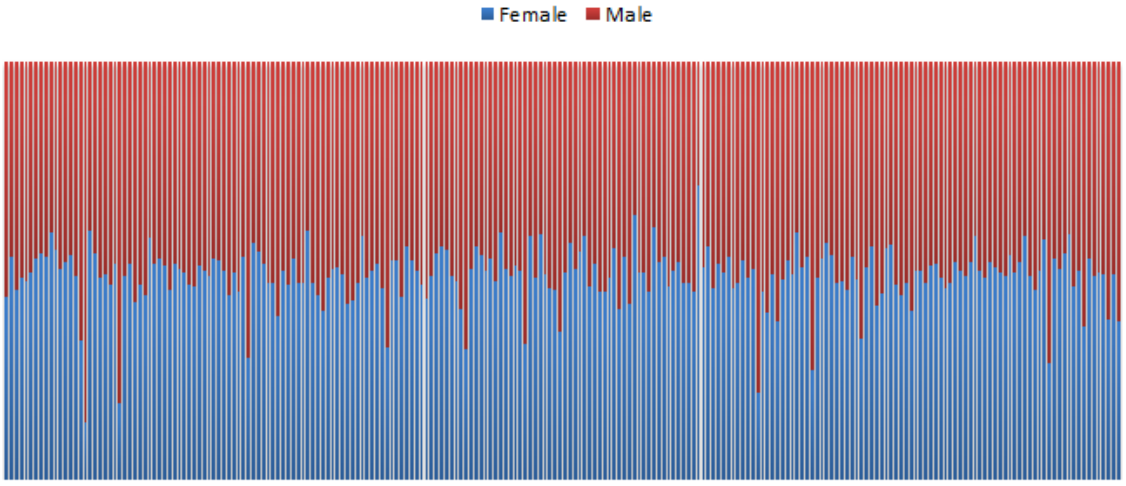


Figure 5.1: Percentage of both genders by destination country

5.5.1 Analysis of results

As said before, the Poisson Regression should give the best results. For this reason, we will highlight two final models : one that takes the past migrations into account (Figure A.8) and one that does not (Figure A.11).

Comparison with other models

The pseudo R-squared[11], which is a measure of the quality of the regression is equal to 0.9565 for the model with past migrations and 0.7457 for the other one.

[11]This factor is explained among others in Appendix A.1.

Let us compare our results with existing models, first with past migrations then without :

Author	# observations	# parameters	R-squared	Specificity ?
Artuc	30'419	15	0.872→0.898	By gender and qualification
Our model	17'094	7	0.9565	

Figure 5.2: Comparison between our model and existing model with past migrations.

Author	# observations	# parameters	R-squared	Specificity ?
Lewer	2'710	10	0.663	
Ramos	142'112	13	0.634	Time series
Ramos	3'356	13	0.834	Time series, only for EU-countries
Our model	34'089	10	0.7457	

Figure 5.3: Comparison between our model and existing models without past migrations.

With the past migrations (Figure 5.2), the compared model distinguishs genders and the qualification of the migrants and uses more parameters. Even though, our model gives better results.

Without the past migrations (Figure 5.3) in the general case, our model gives better results. However, when the model from Ramos and Surinach is applied to a much smaller set of countries, its results are excellent. We did not apply our model on smaller sets so it is impossible to predict if our model is efficient in such situations.

Analysis of our model

The list of the considered factors and their coefficient [12] follows in Figure 5.4 :

Parameter	Coefficient with old migrations	Coefficient without old migrations
lmig1990	.8392448***	Not used
ldist	-.0917612***	-.0531668**
lpop2	.0454992***	.2855455***
lpop1	.086***	.3204015***
com_lang	Not used	.7973999***
com_bound	.1796087**	2.059786***
lprinv1	Not used	.6856506***
lpr2	.128293***	.7462983***
isenglish1	Not used	-.4842202***
lpib	.0941667***	.1669608***
issamecom6	Not used	1.835347***

Figure 5.4: Considered factors of our gravity models and their coefficient. Superscripts ***,**,* denote statistical signifiance at 1, 5 and 10 % respectively.

As the coefficients of the models are hard to interpret, we will try to measure their importance. To do so, we build the same models with, each time, a factor missing. Then we can see how less is the pseudo R-squared in each of these situations.

[12] An explanation of each factor is available in the Appendix A.2.1.

Firstly, we do it for the model containing the past migrations.

Parameter missing	Loss for the R-squared value
lmig1990	-0.3154
lpib	-0.0029
lpr2	-0.0020
lpop1	-0.0019
com_bound	-0.0006
ldist	-0.0006
lpop2	-0.0004

Figure 5.5: Loss for the R-squared value if the parameter is withdrawn.

The Table 5.5 shows how important are the past migrations in the model. Indeed, any other parameter could be removed without changing the quality of the regression sensibly. Only with the migrations between 1980 and 1990, we obtain a R-squared value of 0.9468 (Figure A.9) for the migrations between 1990 and 2000. This factor only gives a better model than the ones developed in the literature [13]. However, these seven factors appear in each of our twenty best models when we did all the combinations possible. Let us notice that with only one or two pieces of information for each country, we can build the whole matrix I_{ij}. For instance, if we have the GDP per capita of country i and country j, we can build lpib_{ij}. If we have the coordinates of the two countries, we can build the distance ldist_{ij} It is not the case for the past migrations where we need to have information for each pair of countries. This factor contains thus much more information than the other ones.

That validates the idea to interest ourselves in a model without past migrations. It is important to point out that past migrations are sometimes considered as a proxy variable for the diasporas [13] which are known to have a major impact on the migrations [21].

If we consider the model without past migrations, we obtain the table 5.6.

Parameter	Loss for the R-squared value
lpr2	-0.0835
com_bound	-0.0752
issamecom6	-0.0545
lprinv1	-0.0257
lpop1	-0.0169
lpop2	-0.0148
com_lang	-0.0124
ldist	-0.0062
lpib	-0.0055
isenglish1	-0.0051

Figure 5.6: Loss for the R-squared value if the parameter is withdrawn

The Table 5.6 gives us a better idea of the importance of the different factors. It is interesting to notice that three of the four best factors are developed for the first time in this report :

- The ranking of attractiveness of the destination country (lpr2).
- The ranking of repulsiveness of the source country (lprinv1).

[13] To the best of our knowledge at least.

- The fact to belong or not in the same community (issamecom6).

Past migrations excluded, the following seven factors appear in the twenty best models of migrations : lpop2, lpop1, com_lang, com_bound, lprinv1, lpr2, issamecom6.
For this reason, we also develop in the Appendix A.2.3 a model with these core parameters which is a good trade-off between the quality of the R-Squared and the number of factors considered (Figure A.12). Again, let us notice that commonly used factors such as the GDP per capita do not appear.

This confirms the validity of our previous chapters and analysis. Furthermore, our parameters are built on the past migrations and are stable with variations as proved before while the economic factors are not always easy to obtain and vary more. To the best of our knowledge, even with more specific database and with more parameters than what we are using, no model of international migrations competes with our best one.

5.6 Predicting future migrations

As said before, another goal of a gravity model is to predict migrations. Now that we have designed several models, we use one to predict the future migrations. For this purpose, we take the model without past migrations containing what we call the core parameters (the ones that appear in each of the twenty best models). We have thereby the following parameters to consider :

- The population of the origin country.
- The population of the destination country.
- The PageRank of the destination country.
- The inverted PageRank of the origin country.
- The common language between the two countries.
- The common border between the two countries.
- The common 6-community between the two countries.

However, as the migrations, these parameters can also evolve in time. There is thereby a need to define their own evolution law. This task is trivial for the common languages and the common boundaries which remains the same through years.

For the PageRank and the 6-community, they are both mathematical measures directly obtained from the migrations. We cannot thereby use the PageRank of the year t to model the migration of the same year. However, as previously showed, the PageRank is stable in time 3.6. For this reason, we keep the same value of PageRank over time. We do the same with the community.

Concerning the population of a country, on the one hand it depends on the birth and death rate and on the other hand on the immigration and the emigration [164]. To have the population of the country c at the year t, we thereby use the following formula for each countries c which belongs to the set C of the countries considered :

$$\begin{aligned} \text{pop}(c)_t &= \text{pop}(c)_{t-1} \\ &+ \Big(\text{pop}(c)_{t-1} \times \text{birthRate}(c) + \sum_{i=c_1}^{C} \text{migration}(i,c)_{t-1}\Big) \\ &- \Big(\text{pop}(c)_{t-1} \times \text{deathRate}(c) + \sum_{i=c_1}^{C} \text{emigration}(c,i)_{t-1}\Big). \end{aligned}$$

Because the birth and death rate stay relatively constant through close years for most countries [158, 159], we only take the value of the year 2000. Furthermore, when one value is missing, we assign it to zero.

One issue encountered is that our model is adapted for migrations occurring during a whole decade and we want to predict migrations for each year. To model the yearly migrations we uniformly distribute the values obtained for the decennial migrations. We finally have a model the evolution of the population.

Let us see how good are the predictions obtained. To do so, we use our model to predict the migrations in 2012 and, for each country, we compare the real population with the one obtained. The relative error for the population of a country is thereby :

$$\Delta_{\%} = \frac{|\text{Real population} - \text{predicted population}|}{\text{Real population}} \times 100.$$

By taking this error for each country, we obtain the following results :

Criterion	$\Delta_{\%}$
Min value	0.03
Max value	64.37
Mean value	6.33
Variance	68.46
Standard deviation	8.27

Table 5.1: Statistical tests for $\Delta_{\%}$.

With a mean of 6.33% and a standard deviation of 8.27% we can believe that our model is accurate despite of the approximations done. The max value of 64.37%, obtained by Qatar, is however high. There are only eight countries having an error superior than 20% : Qatar, United Arab Emirates, Micronesia, Bahrain, Kuwait, Samoa, Albania and Tonga. As seen previously through this paper, most of them are particular countries. These big differences can also be explained through particular events occurring in or involving the countries.

To realise if our model is truly efficient to predict future migrations, we can compare the results with a simple model where only the birth and death rates are considered :

$$\begin{aligned} \text{pop}(c)_t &= \text{pop}(c)_{t-1} \\ &+ \Big(\text{pop}(c)_{t-1} \times \text{birthRate}(c)\Big) \\ &- \Big(\text{pop}(c)_{t-1} \times \text{deathRate}(c)\Big). \end{aligned}$$

That gives the following results :

Criterion	$\Delta_{\%}$
Min value	0.03
Max value	64.38
Mean value	6.34
Variance	68.43
Standard deviation	8.27

Table 5.2: Statistical tests for $\Delta_{\%}$.

Our basic model to predict populations thanks to migrations is eventually disappointing. Indeed, the statistics are really similar taking the migrations into account or not.

This shows that unpredicted effects are much more important than general trends of migrations. A model to predict population should thus consider more specific effects.

5.7 Conclusion

In this chapter we saw what a gravity model is and how to apply it to migrations. We also presented which criteria of regressions are used in the literature. We added three efficient explaining factors to the existing models :

- the PageRank which corresponds to the attractiveness of the countries - *does the destination country attracts more than the other one ?*
- the inverted PageRank which corresponds to the repulsiveness of the countries - *does the origin country repulses more than the other one ?*
- the fact to belong or not to the same community as the other country - *are the two countries strongly connected ?*

With these parameters and some others, we obtained two final models :

- an accurate one containing the past migrations (Figure A.8). This model gives the best prediction according to the pseudo R-squared metric. To the best of our knowledge, this is more efficient than any model developed in the literature.
- an economic one in term of information used without the past migrations (Figure A.11). This model is still efficient even with much less information. Furthermore, to the best of our knowledge, this is the best international model developed without past migrations.

We tried to develop a basic model to predict future populations but the results were inconclusive.

Finally, it would have been interesting to develop a refined model based on time series regressions because it would probably give better results. Indeed, time series regressions consider more observations and are thus better models to predict future events. This remains an open topic in this master's thesis and it would be interesting to analyse it in the future.

Chapter 6

Conclusion

"In literature and in life we ultimately pursue, not conclusions, but beginnings."

– Sam Tanenhaus, *Literature Unbound*

Throughout this report, we used and modified several well-known mathematical concepts of graph theory to analyse the migration flows and we proved their validity by applying them to econometric models which gave us excellent results.

We developed three major points in the field of migrations : ranking countries according to several criteria, grouping them into consistent groups and elaborating a gravity model for migrations.

- The goal of the ranking was to develop a ranking of attractiveness and repulsiveness but also to find a hierarchy between countries where several countries can belong to the same level. Our best results were obtained with the PageRank metric which can give at the same time a stable and reliable ranking of attractiveness and repulsiveness. The final result of this analysis is represented in the Figure 6.1.

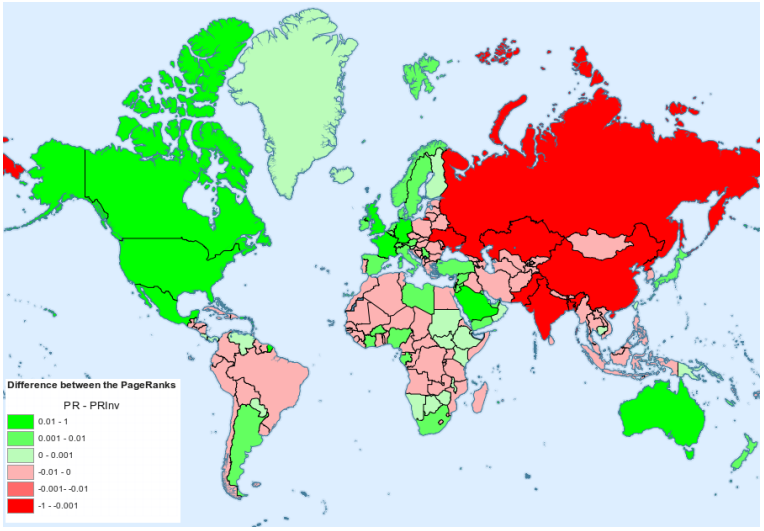


Figure 6.1: Global score of attractiveness/repulsiveness for each country.

Besides, using a topological ordering with a simple spanning tree heuristic, we obtained a representation (Figure 6.2) of the most important hubs in the migration network.

Figure 6.2: Spanning tree score of attractiveness/repulsiveness for each country.

- In the group detection chapter, we saw several methods that can be used to form groups of countries by using a graph theory approach. The community method was adapted to find several sets of countries strongly connected to each other and weakly connected to the other groups. The Figure 6.3 shows a community partition.

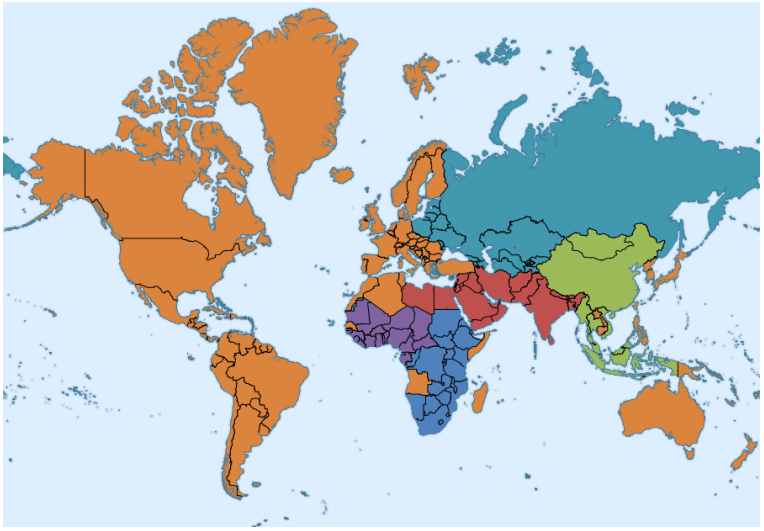


Figure 6.3: Communities of countries exchanging a lot of migrants.

  Moreover, combined with the PageRank, this method appeared to be efficient to find out if a country is locally or globally attractive, or not attractive at all.

- Concerning the gravity model, our goal was to develop a new formula of gravity either giving better results than the litterature or using less parameters. We managed to realise both these objectives at the same time applying the concepts of attractivity, repulsivity and community developed in the other chapters.
  A way to go beyond is to study the evolution of migrations to predict future migrations and thus the future population of each country.

To go further it would be interesting to distinguish the migrations of highly or lowly educated people. This would take the 'brain leak' effect into account and improve our models. Besides, using time series regressions could improve our model to predict future migrations and populations.

Bibliography

[1] Geoname. http://download.geonames.org/export/dump/countryInfo.txt.

[2] JUnit. http://junit.org/.

[3] Mendeley. http://www.mendeley.com/en/1/1/.

[4] Stata. http://www.stata.com/.

[5] Working with Excel Files in Python. http://www.python-excel.org/.

[6] Warsaw pact. 1955.

[7] 1990 Census of Population and Housing and Census 2000. Vietnam foreign-born population, for the United States: 1990 and 2000.

[8] M. B. A. Delmotte, M. Schaub, S. Yaliraki. Community Detection using the stability of a graph partition, 2012.

[9] S. Agarwal. Ranking on Graph Data. 2006.

[10] J. I. Alvarez-Hamelin, L. Dall'Asta, A. Barrat, and A. Vespignani. K-core decomposition of internet graphs : Hierarchies, self-similarity and measuremnt biases. 3(2):371–393, 2008.

[11] J. E. Anderson. The Gravity Model. Annual Review of Economics, 3(1):133–160, Sept. 2011. http://www.annualreviews.org/doi/abs/10.1146/annurev-economics-111809-125114.

[12] J. E. Anderson and E. Van Wincoop. Gravity with Gravitas: A Solution to the Border Puzzle. http://www.nber.org/papers/w8079.pdf.

[13] E. Artuc, F. Docquier, C. Ozden, and C. R. Parsons. A Global Assessment of Human Capital Mobility : the Role of non-OECD Destinations. http://perso.uclouvain.be/frederic.docquier/oxlight.htm.

[14] R. S. Avi-Yonah. Statement to Congress. Technical report, 2012.

[15] R. Bagnara. A Unified Proof for the Convergence of Jacobi and Gauss–Seidel Methods. SIAM Review, 37(1):93–97, Mar. 1995. http://epubs.siam.org/doi/abs/10.1137/1037008.

[16] H. Balakrishnan. Wide-Area Internet Routing. (January), 2009.

[17] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst. Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods. Jan. 1994. http://epubs.siam.org/doi/book/10.1137/1.9781611971538.

[18] V. Batagelj and M. Zaversnik. An O (m) algorithm for cores decomposition of networks. arXiv preprint cs/0310049, pages 1–9, 2002. http://arxiv.org/abs/cs/0310049.

[19] R. Becker, A. Walks, R. Brownrigg, and T. Minka. Package 'maps', 2013.

[20] M. Beine, F. Docquier, and c. Ozden. Diasporas. (July), 2009.

[21] M. Beine, F. Docquier, and C. Ozden. Diaspora effects in international migration : key questions and methodological issues. 2010.

[22] S. Bertoli and J. F.-h. Moraga. Multilateral Resistance to Migration. (5958), 2011.

[23] R. Biswas. International Tax Competition: A Developing Country Perspective (Commonwealth Secretariat). 2002.

[24] V. Blondel. INMA2111 - Mathématiques discrètes II: Algorithmes et complexité.

[25] V. Blondel, J.-l. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks.

[26] P. Bonacich. Power and Centrality : A Family of Measures '. 92(5):1170–1182, 2012.

[27] P. Bonacich and P. Lloyd. Eigenvector-like measures of centrality for asymmetric relations. Social Networks, 23(3):191–201, July 2001. http://linkinghub.elsevier.com/retrieve/pii/S0378873301000387.

[28] S. P. Borgatti. 2-Mode Concepts in Social Network Analysis.

[29] S. P. Borgatti and M. G. Everett. Models of core r periphery structures. pages 375–395, 1999.

[30] S. P. Boyd. Course EE363: Linear Dynamical Systems : Perron-Frobenius Theory, 2008.

[31] U. Brandes, D. Delling, M. Gaertler, M. Hoefer, Z. Nikoloski, and D. Wagner. On Modularity - NP-Completeness and Beyond. (001907).

[32] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, Sridhar Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. (1):1–15.

[33] K. D. Bruyne, G. Magerman, and J. V. Hove. Multilateral Gravity–A network approach. cirano.qc.ca, pages 1–12. http://www.cirano.qc.ca/conferences/public/pdf/networks2012/16-Glenn_Magerman_De_Bruyne_Van_Hove-Multilateral_gravity.pdf.

[34] Business Dictionary. Definition of regression analysis. http://www.businessdictionary.com/definition/regression-analysis-RA.html.

[35] Business Dictionary. Test statistic. http://www.businessdictionary.com/definition/test-statistic.html.

[36] W. M. Campbell, C. K. Dagli, and C. J. Weinstein. Social Network Analysis with Content and Graphs. 20(1), 2013.

[37] Central Intelligence Agency. The World Factbook : GDP - composition, by sector of origin. https://www.cia.gov/library/publications/the-world-factbook/fields/2012.html.

[38] Central Intelligence Agency. The World Factbook : Net migration rate. https://www.cia.gov/library/publications/the-world-factbook/rankorder/2112rank.html.

[39] P. Charbit, S. Thomassé, and A. Yeo. The Minimum Feedback Arc Set Problem is NP-Hard for Tournaments. Combinatorics, Probability and Computing, 16(01):1, Sept. 2006. http://www.journals.cambridge.org/abstract_S0963548306007887.

[40] M. Chiang. Networked Life : 20 Questions and Answers. (April), 2012.

[41] H. H. Chuong, M. H. Ta, E. Lai, D. Arguelles, AsianWeek Magazine, and UCLA Asian American Studies Center. Vietnamese Americans, 2003. http://www.asian-nation.org/vietnamese.shtml.

[42] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. pages 1–6, 2004.

[43] CMAJ. Health consequences of Cuba's Special Period. Canadian Medical Association Journal, 179(3):257, July 2008. http://www.cmaj.ca/cgi/doi/10.1503/cmaj.080976.

[44] D. Coppersmith, L. K. Fleischer, and A. Rurda. Ordering by weighted number of wins gives a good ranking for weighted tournaments. ACM Transactions on Algorithms, 6(3):1–13, June 2010. http://portal.acm.org/citation.cfm?doid=1798596.1798608.

[45] Course Number MRG 217. Maximum Likelihood Estimation (MLE). New York University, pages 1–22, 2009.

[46] P. Csermely. Structure and dynamics of core/periphery networks. Journal of Complex Networks, pages 93–123, 2013.

[47] J. P. de Cuellar. Letter dated 20 march 1991 from the secretary general addressed to the president of the security council.

[48] H. de Haas. Maroc : Préparer le terrain pour devenir un pays de transition migratoire ? 2014.

[49] A. De Montis, M. Barthélemy, A. Chessa, and A. Vespignani. The structure of interurban traffic: a weighted network analysis. Environment and Planning B: Planning and Design, 34(5):905–924, 2007. http://www.envplan.com/abstract.cgi?id=b32128.

[50] Y.-A. de Montjoye and L. Rocher. in preparation.

[51] A. Deardorff and J. A. Frankel. The Regionalization of the World Economy. University of Chicago Press, (January):7–32, 1998.

[52] M. Dejemeppe. Econométrie appliquée : microeconométrie. 2010.

[53] A. Di Bartolomeo, T. Jaulin, and D. Perrin. Palestine. (July), 2011.

[54] I. Dinur and S. Safra. The Importance of Being Biased. 104(104):1–36, 2001.

[55] F. Docquier. Comprehensive migration matrices by eductation level and by gender (1990-2000). 2013. http://perso.uclouvain.be/frederic.docquier/filePDF/ADOP_AgeOfEntry.xlsx.

[56] F. Docquier, A. Marfouk, C. Özden, and C. Parsons. Geographic, gender and skill structure of international migration. 2011. http://mpra.ub.uni-muenchen.de/47917/.

[57] N. Dudarenko and J. Rana. Ranking Algorithm by Contacts Priority for Social Communication Systems. pages 38–49, 2010.

[58] M. Duenas and G. Fagiolo. Modeling the International-Trade Network: a gravity approach. Journal of Economic Interaction and Coordination, 2011. http://link.springer.com/article/10.1007/s11403-013-0108-y.

[59] J. a. Dunne, R. J. Williams, and N. D. Martinez. Food-web structure and network theory: The role of connectance and size. Proceedings of the National Academy of Sciences of the United States of America, 99(20):12917–22, Oct. 2002. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=130560&tool=pmcentrez&rendertype=abstract.

[60] C. Dustmann and A. Glitz. Migration and Education. 2011.

[61] F. Eisenbrand and F. Grandoni. On the complexity of fixed parameter clique and dominating set. Theoretical Computer Science, 326(1-3):57–67, Oct. 2004. http://linkinghub.elsevier.com/retrieve/pii/S030439750400372X.

[62] L. Eldén. A Note on the Eigenvalues of the Google Matrix. pages 1–3, 2003.

[63] G. Fagiolo and M. Mastrorillo. International migration network: Topology and modeling. Physical Review E, 88(1):012812, July 2013. http://link.aps.org/doi/10.1103/PhysRevE.88.012812.

[64] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On Power-Law Relationships of the Internet Topology.

[65] R. C. Feenstra, J. R. Markusen, and A. K. Rose. Using the gravity equation to differentiate among alternative theories of trade. 34(2):430–447, 2014.

[66] A. Finlan. The Collapse of Yugoslavia 1991-1999. Osprey Publishing, 2004.

[67] J. A. Fisher. Trace Scheduling: A Technique for Global Microcode Compaction. IEEE Transactions on computers, C(7):478–490, 1981.

[68] S. Fortunato. Community detection in graphs. 2010.

[69] S. Fortunato and Marc Barthélemy. Resolution limit in community detection. pages 1–8, 2008.

[70] R. Fourer, D. M. Gay, and B. W. Kernighan. AMPL. http://ampl.com/.

[71] GeoHive. Human Development Index. http://www.geohive.com/earth/gen_hdi.aspx.

[72] Geohive. Human development index between 1980 and 2011. http://www.geohive.com/earth/gen_hdi.aspx.

[73] Geohive. Past and and future population 1950-2050. http://www.geohive.com/earth/his_proj.aspx.

[74] Global Finance. GDP-GNI-Definitions. http://www.gfmag.com/tools/glossary/gdp-gni-definitions.html#axzz30HXmL1Dt.

[75] Global Property Guide. The CNMI's real estate crisis continues. http://www.globalpropertyguide.com/Pacific/Commonwealth-of-Northern-Mariana-Islands.

[76] M. T. Goodrich and R. Tamassia. Data Structures and Algorithms in Java. 4th editio edition.

[77] A. Greenwald and J. Wicks. QuickRank: A Recursive Ranking Algorithm. 1910.

[78] J. Grogger and G. Hanson. Income maximization and the selection and sorting of international migrants. http://www.nber.org/papers/w13821http://www.sciencedirect.com/science/article/pii/S0304387810000647.

[79] R. A. Hanneman and M. Riddle. Introduction to Social Network Methods.

[80] K. Head, T. Mayer, and J. Ries. The erosion of colonial trade linkages after independence. Journal of International Economics, 81(1):1–14, May 2010. http://linkinghub.elsevier.com/retrieve/pii/S0022199610000036.

[81] R. Hijmans, E. Williams, and C. Vernes. Package 'geosphere', 2014.

[82] In Motion. Haitian migration: 20th century. http://www.inmotionaame.org/migrations/topic.cfm?migration=12&topic=3.

[83] Index Mundi. Country comparison - GDP per capita (PPP). http://www.indexmundi.com/g/r.aspx?v=67.

[84] Index Mundi. Country comparison - Oil production. http://www.indexmundi.com/g/r.aspx?v=88.

[85] M. indicator. Composition of macro geographical (continental) regions, geographical sub-regions, and selected economic and other groupings, 2013. http://millenniumindicators.un.org/unsd/methods/m49/m49regin.htm.

[86] Institude for digital research and education - UCLA. Stata Annotated Output - Poisson Regression. http://www.ats.ucla.edu/stat/stata/output/stata_poisson_output.htm.

[87] Institude for digital research and education - UCLA. Stata Annotated Output Regression Analysis. http://www.ats.ucla.edu/stat/stata/output/reg_output.htm.

[88] Institude for digital research and education - UCLA. What are pseudo R-squared? http://www.ats.ucla.edu/stat/mult_pkg/faq/general/Psuedo_RSquareds.htm.

[89] Integraledesmaths. Distance entre deux points à la surface d'une sphère. http://integraledesmaths.free.fr/idm/GeoAEDistSph.htm.

[90] International Monetary Fund. GDP by country 2000. http://www.imf.org/external/pubs/ft/weo/2014/01/weodata/index.aspx.

[91] International Monetary Fund. Offshore Financial Centers (OFCs): IMF Staff Assessments. http://www.imf.org/external/NP/ofca/OFCA.aspx.

[92] International Monetary Fund. Unemployment rate. http://www.imf.org/external/pubs/ft/weo/2012/01/weodata/weoselgr.aspx.

[93] I. Ipsen. Analysis and Computation of Google ' s PageRank.

[94] G. Jia, Z. Cai, M. Musolesi, Y. Wang, D. A. Tennant, J. Ralf, M. Weber, J. K. Heath, and S. He. Community Detection in Social and Biological Networks using Differential Evolution. 2012.

[95] C. Kenyon-Mathieu and W. Schudy. How to rank with few errors. Proceedings of the thirty-ninth annual ACM symposium on Theory of computing - STOC '07, page 95, 2007. http://portal.acm.org/citation.cfm?doid=1250790.1250806.

[96] Y. Khan. The Great Partition : The Making of India and Pakistan. Yale University Press, 2007.

[97] A. E. Krause, K. a. Frank, D. M. Mason, R. E. Ulanowicz, and W. W. Taylor. Compartments revealed in food-web structure. Nature, 426(6964):282–5, Nov. 2003. http://www.ncbi.nlm.nih.gov/pubmed/14628050.

[98] M. Kristiansen, A. Mygind, and A. Krasnik. Health effects of migration. Danish medical bulletin, 54, 2007.

[99] Kushnirs. Gross Domestic Product (GDP). http://kushnirs.org/macroeconomics/gdp/gdp_world.html.

[100] W. Lafeber. Inevitable Revolutions: The United States in Central America. 1993.

[101] T. L. Lai, H. Robbins, and C. Z. Wei. Strong consistency of least squares estimates in multiple regression. Proceedings of the National Academy of Sciences of the United States of America, 75(7):3034–3036, July 1978. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=392707&tool=pmcentrez&rendertype=abstract.

[102] R. Lambiotte, J. Delvenne, and M. Barahona. Laplacian Dynamics and Multiscale Modular Structure in Networks. 2009.

[103] E. A. Leicht and M. E. J. Newman. Community structure in directed networks. pages 1–5, 2007.

[104] L. Leroux. Carte de Flux sous R, 2014. http://www.gis-blog.fr/2014/03/31/carte-de-flux-sous-r/.

[105] J. J. Lewer and H. Van den Berg. A gravity model of immigration. Economics Letters, 99(1):164–167, Apr. 2008. http://linkinghub.elsevier.com/retrieve/pii/S0165176507002455.

[106] Y. Liu, Q. Liu, and Z. Qin. Community Detecting and Feature Analysis in Real Directed Weighted Social Networks. Journal of Networks, 8(6):1432–1439, June 2013. http://ojs.academypublisher.com/index.php/jnw/article/view/10238.

[107] S. Lozano, J. Duch, and A. Arenas. Community detection in a large social dataset of European Projects. 2003.

[108] Lucid Software Inc. LucidChart. https://www.lucidchart.com/.

[109] J. Luo and C. Magee. Detecting Evolving Patterns of Self Organizing Networks by Flow Hierarchy Measurement. Complexity, 2011.

[110] D. Lusseau and M. E. J. Newman. Identifying the role that animals play in their social networks. Proceedings. Biological sciences / The Royal Society, 271 Suppl(December):S477–81, Dec. 2004. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1810112&tool=pmcentrez&rendertype=abstract.

[111] G. Madala. Introduction to econometrics, second edition.

[112] Mapbox. TileMill. https://www.mapbox.com/tilemill/.

[113] Mathworks. Matlab.

[114] A. Maus. Bonacich's Centrality. 2011. http://a-ma.us/wp/2011/03/bonacichs-centrality/.

[115] B. Michel and C. Parsons. Climatic factors as determinants of International Migration. pages 1–36, 2012. http://econpapers.repec.org/paper/ctllouvir/2012002.htm.

[116] Ministry of Foreign Affairs of Ethiopia. Ethiopia-Russia relations. http://www.mfa.gov.et/BilateralMore.php?pg=25.

[117] R. J. Mokken. Cliques, Clubs and Clans. 13:161–173, 1979.

[118] Monetary and Capital Markets Department (International Monetary Fund) and Legal Department (International Monetary Fund). A report on the Assessment Program and Proposal for Integration with the Financial Sector Assessment Program. Technical report, 2008.

[119] B. Naveh. JGraphT. http://jgrapht.org/.

[120] M. E. J. Newman. The mathematics of networks. pages 1–12.

[121] I. Newton. Philosophiae Naturalis Principia Mathematica. 1687.

[122] OECD. Towards Global Tax Co-operation. page 17, 2000.

[123] OECD UNDESA. World Migration in Figures. (October):1–6, 2013.

[124] Open Data. Country List ISO 3166 Codes Latitude Longitude. https://opendata.socrata.com/dataset/Country-List-ISO-3166-Codes-Latitude-Longitude/mnkm-8ram.

[125] M. Orozco. Remitting Back Home and Supporting the Homeland: The Guyanese Community in the U.S. Inter American Dialogue, 2003.

[126] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking : Bringing Order to the Web. 1998.

[127] Paho. French Guiana, Guadeloupe and Martinique. http://www.paho.org/saludenlasamericas/index.php?id=41&option=com_content&Itemid=&lang=pt.

[128] A. Panhaleux. Recherche de cycles dans les graphes. Technical report, 2007. http://perso.ens-lyon.fr/eric.thierry/Graphes2007/adrien-panhaleux.pdf.

[129] R. G. Patman. Soviet Ethiopian Relations: The Horn of Dilemma. Margot Light, ed., Troubled Friendships: Moscow's Third World Ventures, 1993.

[130] J. Pattillo, N. Youssef, and S. Butenko. Clique relaxation models in social network analysis. pages 1–20.

[131] D. Persitz. Power and Core-Periphery Networks. SSRN Electronic Journal, 2010. http://www.ssrn.com/abstract=1579634.

[132] D. S. G. Pollock. Gauss-Markov Theorem. (3):1–2, Aug. 2006.

[133] R. Ramos and J. Suriñach. A gravity model of migration between ENC and EU. 2013.

[134] T. M. Rempel. Palestinian Refugees in the West Bank and the Gaza Strip, 2006. http://www.forcedmigration.org/research-resources/expert-guides/palestinian-refugees-in-the-west-bank-and-the-gaza/alldocuments.

[135] J.-P. Rodrigue, C. Comtois, and B. Slack. The Geography of Transport Systems., 2009. http://people.hofstra.edu/geotrans/eng/methods/ch5m1en.html.

[136] M. P. Rombach, M. A. Porter, J. H. Fowler, and P. J. Mucha. Core-Periphery Structure in Networks.

[137] M. Ruhs and C. Vargas-Silva. The Labour Market Effects of Immigration. The migration observatory, 2014.

[138] B. Sabella. Russian Jewish Immigration and the Future of the Israeli-Palestinian Conflict. MER 182 - Jerusalem and the Peace Agenda, 23. http://www.merip.org/mer/mer182/russian-jewish-immigration-future-israeli-palestinian-conflict.

[139] A. Shah. Conflict between Ethiopia and Eritrea. Social, Political, Economic and Environmental Issues That Affect Us All, 2000. http://www.globalissues.org/article/89/conflict-between-ethiopia-and-eritrea.

[140] B. Siliverstovs and D. Schumacher. To log or not to log , or How to estimate a gravity model ? 2006.

[141] J. M. C. S. Silva and S. Tenreyro. The log of gravity. 88(November):641–658, 2006.

[142] I. Simonsen, K. Astrup Eriksen, S. Maslov, and K. Sneppen. Diffusion on complex networks: a way to probe their large-scale topological structures. Physica A: Statistical Mechanics and its Applications, 336(1-2):163–173, May 2004. http://linkinghub.elsevier.com/retrieve/pii/S0378437104000445.

[143] W. Stanley. The Protection Racket State: Elite Politics, Military Extortion, and Civil War in El Salvador. 1996.

[144] Statsdirect.com. P-values. http://www.statsdirect.com/help/default.htm#basics/p_values.htm.

[145] S. H. Strogatz. Exploring complex networks. Insight review articles, 410(March), 2001.

[146] R. Tarjan. Depth-First Search and Linear Graph Algorithms. SIAM Journal on Computing, 1(2):146–160, June 1972. http://epubs.siam.org/doi/abs/10.1137/0201010.

[147] W. tax havens. Hong Kong. http://worldstaxhavens.com/taxhaven/hong-kong/.

[148] W. tax havens. Macau. http://worldstaxhavens.com/taxhaven/macau/.

[149] Tax Havens Guide. List of tax havens. http://www.taxhavensguide.com/list-of-tax-havens.php.

[150] The Levin Institute - The State University of New York. Globalization101 : Cultural effects of migration, 2014. http://www.globalization101.org/cultural-effects-of-migration/.

[151] The Levin Institute - The State University of New York. Globalization101 : Economic effects of migration, 2014. http://www.globalization101.org/economic-effects-of-migration/.

[152] E. Thomas-Hope, Pauline Knight, and C. Noel. Migration in Jamaica. International Organization for Migration, 2010.

[153] United Nations Department of Economic and Social Affairs Population Division. International Migration 2013. 2013. www.unmigration.org.

[154] United Nations Development Programme. Human Development Report 2011. United Nations Development Programme, 2011.

[155] E. Vincent. Catastrophe migratoire à Mayotte. http://www.lemonde.fr/societe/article/2012/12/27/catastrophe-migratoire-a-mayotte_1810793_3224.html.

[156] T. Walmsley, C. Ozden, C. Parsons, and M. Schiff. Where on Earth is Everybody? The Evolution of Global Bilateral Migration 1960–2000.

[157] J. Westerlund and F. Wilhemsson. Estimating the gravity model without gravity using panel data. Applied Economics, 6, 2009.

[158] World Bank Group. Birth rate, crude (per 1,000 people). http://data.worldbank.org/indicator/SP.DYN.CBRT.IN/countries?page=2.

[159] World Bank Group. Death rate, crude (per 1,000 people). http://data.worldbank.org/indicator/SP.DYN.CDRT.IN/countries.

[160] World Bank Group. GDP (current US$). http://data.worldbank.org/indicator/NY.GDP.MKTP.CD?page=2.

[161] World Bank Group. GDP (current US dollar). http://data.worldbank.org/indicator/NY.GDP.MKTP.CD.

[162] World Bank Group. Global bilateral migration database. http://data.worldbank.org/data-catalog/global-bilateral-migration-database.

[163] World Bank Group. Population (Total). http://data.worldbank.org/indicator/SP.POP.TOTL.

[164] World Bank Group. World Population Growth.

[165] World Bank Group. Natural Gas Rents (Percentage of GDP), 2011. http://data.worldbank.org/indicator/NY.GDP.NGAS.RT.ZS?page=1.

[166] World Bank Group. Oil rents (Percentage of GDP), 2011. http://data.worldbank.org/indicator/NY.GDP.PETR.RT.ZS?page=1.

[167] World Bank Group. Total natural resources rents (Percentage of GDP), 2011. http://data.worldbank.org/indicator/NY.GDP.TOTL.RT.ZS?page=1.

[168] World Bank Group. World Development Indicators : Contribution of natural resources to gross domestic product. pages 0–4, 2014.

[169] World Bank Group, C. Ozden, C. R. Parsons, M. Schiff, and T. L. Walmsley. World Bank Economic Review : Where on Earth is Everybody? The Evolution of Global Bilateral Migration 1960-2000. Technical report. http://data.worldbank.org/data-catalog/global-bilateral-migration-database.

[170] World Health Organization. Global Health Observatory Data Repository. http://apps.who.int/gho/data/node.main.688?lang=en.

Appendix A

Stata analysis

A.1 How to analyse the quality of a result ?

A.1.1 Least Squares Regression

We used the logiciel Stata C.2 to compute the best regression (minimum Least Squares error). Stata provides much information to analyse the quality of a solution. Let's show an example of its use :

```
. reg lmig2000 lmig1990 lmig1980 lmig1970 lmig1960
```

Source	SS	df	MS
Model	82645.8448	4	20661.4612
Residual	17274.2998	11840	1.45897802
Total	99920.1446	11844	8.43635128

Number of obs	= 11845
F(4, 11840)	=14161.60
Prob > F	= 0.0000
R-squared	= 0.8271
Adj R-squared	= 0.8271
Root MSE	= 1.2079

lmig2000	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
lmig1990	.7664604	.0093545	81.93	0.000	.748124	.7847968
lmig1980	.1428804	.012004	11.90	0.000	.1193505	.1664103
lmig1970	.0437213	.0125194	3.49	0.000	.0191812	.0682615
lmig1960	-.0038186	.0092504	-0.41	0.680	-.0219508	.0143136
_cons	.6186965	.022587	27.39	0.000	.5744223	.6629707

Figure A.1: Example of Stata regression

In the figure A.1, we can decrypt the information given by Stata. Most of the following information directly come from annotated output of regression analysis [87] applied in our situation.

- Source : The Total variance is divided into the variance which can be explained by the independent variables (Model) and the variance which is not explained by them (Residual).
- SS : Sum of the Squares of the difference between each of the scores and their mean.
- df : Degree of freedom of the sources associated with the variables.
- MS : Mean Squares = $\frac{\text{SS}}{\text{df}}$.
- Number of obs : Number of observations used in the regression analysis.

- F(df_{model},$df_{residual}$) : This is the F-statistic $= \frac{\text{explained variance}}{\text{unexplained variance}}$ or in practice the Mean Square Model (20661.4612) divided by the Mean Square Residual (1.45897802).
- Prob>F : This is the p-value associated with the above F-statistic. It is used in testing the null hypothesis that all of the model coefficients are 0.
- R-squared : It is the proportion of variance in dependent variable (lmig2000) which can be explained by the independent variables (lmig1990, lmig1980, lmig1970, lmig1960). This is an overall measure of the strength of association.
- Adj R-squared : Adjustement of the R-squared that penalizes the addition of extraneous predictors to the model ($= 1 - \frac{(1-Rsq)(N-1)}{N-k-1}$) where k is the number of predictors.
- Root MSE : It is the standard deviation of the error term and is the square root of the Mean Square Residual.
- lmig2000 : This column shows the dependent variable at the top (lmig2000) with the predictor variables below it (lmig1990, lmig1980, lmig1970, lmig1960). The last variable (_cons) represents the constant or intercept that we called α_0.
- Coef. : These are the coefficients $\alpha_0, \ldots, \alpha_N$ of the independent variables to predict the dependent variable.
- Std. Err. : These are the standard errors associated with the coefficients.
- t : These are the t-statistics[1] used in testing whether a given coefficient is significantly different from zero.
- P>|t| : This column shows the 2-tailed p-values used in testing the null hypothethis that the coefficient is zero. The threshold of acceptable variables most commonly used are 0.1, 0.05, 0.001 [144]. In any case, in this situation, lmig1990, lmig1980 and lmig1970 are significantly different from zero while lmig1960 is not.
- [95% Conf. Interval] : These are the 95 % confidence intervals for the coefficients. These confidence intervals can help you to put the estimate from the coefficient into perspective by seeing how much the value could vary.

A.1.2 Poisson Regression - Maximum likelihood

Analysis of quality with Stata

```
. poisson mig2000 lprinv1 lpr2 issamecom6

Iteration 0:   log likelihood = -4.504e+08
Iteration 1:   log likelihood = -3.441e+08
Iteration 2:   log likelihood = -3.022e+08
Iteration 3:   log likelihood = -3.022e+08
Iteration 4:   log likelihood = -3.022e+08

Poisson regression                              Number of obs   =      38025
                                                LR chi2(3)      =   7.61e+08
                                                Prob > chi2     =     0.0000
Log likelihood = -3.022e+08                     Pseudo R2       =     0.5572

------------------------------------------------------------------------------
     mig2000 |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     lprinv1 |   1.202752   .0000731  1.6e+04   0.000     1.202608    1.202895
        lpr2 |   .9822272   .0000503  2.0e+04   0.000     .9821286    .9823259
  issamecom6 |   2.315882   .0002034  1.1e+04   0.000     2.315483    2.316281
       _cons |   18.39381   .0004277  4.3e+04   0.000     18.39298    18.39465
------------------------------------------------------------------------------
```

[1]The t-statistic is a "sample used to determine whether a hypothesis will be accepted or rejected" [35].

Again, let's decrypt the information [86].

- Iteration : This is a listing of the log likelihood at each iteration.
- Log likelihood : This is the log likelihood of the fitted model. It is used in calculation of the Likelihood Ratio (LR) chi-square test of whether all predictor variables regression coefficients are simultaneously zero.
- LR chi2(df) : This is the LR test statistic for the omnibus test that at least one predictor variable regression coefficient is not equal to zero in the model.
- Pseudo R2 : This is McFadden's pseudo R-squared [88] $= 1 - \frac{\text{ll(model)}}{\text{ll(null)}}$. There is not any true equivalent between the R-square in Least Squares regression and pseudo R-squares in Poisson regression, so we can not directly compare the two values.
- z and P>|Z| : equivalent to t and P>|t| in Least Squares regressions.

A.2 Our models of gravity

A.2.1 List of preselected variables

- lmig1990_{ij} : number of migrants (log) between countries i and j between 1980 and 1990.
- lmig1980_{ij} : number of migrants (log) between countries i and j between 1970 and 1980.
- lmig1970_{ij} : number of migrants (log) between countries i and j between 1960 and 1970.
- lmig1960_{ij} : number of migrants (log) between countries i and j between 1950 and 1960.
- ldist_{ij} : distance (log) between source and destination countries.
- lpop1_{i} : population (log) of source country.
- lpop2_{j} : population (log) of destination country.
- com_lang_{ij} : dummy variable equal to 1 if destination and source country share a common language, 0 otherwise.
- com_bound_{ij} : dummy variable equal to 1 if destination and source country share a common land border, 0 otherwise.
- $\text{com\_orig\_l\~g}_{ij}$: dummy equal to 1 if destination and source country share official languages which have the same root, 0 otherwise.
- lpr1_{i} : PageRank (log) of the source country - attractiveness.
- lprinv1_{i} : Inverted PageRank (log) of the source country - repulsiveness.
- lpr2_{j} : PageRank (log) of the destination country - attractiveness.
- lprinv2_{j} : Inverted PageRank (log) of the destination country - repulsiveness.
- ldensity1_{i} : Density (log) of the source country $= \frac{\text{population}}{\text{area of the country}}$.
- ldensity2_{j} : Density (log) of the destination country $= \frac{\text{population}}{\text{area of the country}}$.
- lidh1_{i} : HDI (log) of source country.
- lidh2_{j} : HDI (log) of destination country.

- isenglish1$_i$: dummy variable equal to 1 if English is one of the official languages of the source country, 0 otherwise.

- isenglish2$_j$: dummy variable equal to 1 if English is one of the official languages of the destination country, 0 otherwise.

- lpib$_{ij}$: log of the ratio of the GDP per capita of destination and source country.

- issamecom6$_{ij}$: dummy variable equal to 1 if destination and source country belong to the same 6-community, 0 otherwise.

- issamecom4$_{ij}$: dummy variable equal to 1 if destination and source country belong to the same 4-community, 0 otherwise.

A.2.2 Least Squares Regression

To read and interpret the following results, a detailed example is available above A.1.

```
Linear regression                               Number of obs =   11105
                                                F( 23, 11081) = 4362.20
                                                Prob > F      =  0.0000
                                                R-squared     =  0.8564
                                                Root MSE      =   1.101
```

lmig2000	Coef.	Robust Std. Err.	t	P>|t|	[95% Conf.	Interval]
lmig1990	.6461543	.0137339	47.05	0.000	.6192334	.6730752
lmig1980	.0953508	.0176754	5.39	0.000	.0607038	.1299978
lmig1970	.0756814	.0184733	4.10	0.000	.0394704	.1118924
lmig1960	.008622	.0112864	0.76	0.445	-.0135015	.0307454
ldist	-.2011221	.0157716	-12.75	0.000	-.2320373	-.170207
lpop2	.098326	.0101301	9.71	0.000	.0784692	.1181827
lpop1	.0905919	.0105933	8.55	0.000	.0698272	.1113566
com_lang	.0013087	.0278266	0.05	0.962	-.0532364	.0558538
com_bound	.2852008	.0545257	5.23	0.000	.1783207	.3920808
com_orig_l~g	.1012131	.037073	2.73	0.006	.0285433	.1738829
lprinv1	.0337218	.0190714	1.77	0.077	-.0036616	.0711052
lprinv2	-.1382715	.0203496	-6.79	0.000	-.1781603	-.0983827
lpr1	.0384395	.0129564	2.97	0.003	.0130427	.0638364
lpr2	.2509597	.0160432	15.64	0.000	.2195122	.2824073
ldensity1	-.0048981	.0090205	-0.54	0.587	-.02258	.0127838
ldensity2	.0546265	.0072499	7.53	0.000	.0404154	.0688377
lidh1	-.1640209	.0647535	-2.53	0.011	-.2909492	-.0370925
lidh2	.8074844	.0639516	12.63	0.000	.6821278	.932841
isenglish2	-.0123256	.0235146	-0.52	0.600	-.0584185	.0337673
isenglish1	.0928701	.0247094	3.76	0.000	.0444353	.1413049
lpib	.0140348	.0146334	0.96	0.338	-.0146492	.0427189
issamecom6	-.0359164	.0306805	-1.17	0.242	-.0960557	.0242228
issamecom4	.1523479	.0284968	5.35	0.000	.096489	.2082068
_cons	.8613804	.4046795	2.13	0.033	.0681364	1.654624

Figure A.2: Least Squares Regression with our 23 parameters.

Linear regression

Number of obs	15973
F(11, 15961)	11590.72
Prob > F	0.0000
R-squared	0.8547
Root MSE	1.1208

lmig2000	Coef.	Robust Std. Err.	t	P>|t|	[95% Conf.	Interval]
lmig1990	.8162846	.0049897	163.59	0.000	.8065043	.826065
ldist	-.1849974	.0132539	-13.96	0.000	-.2109765	-.1590183
lpop2	.0355828	.0056469	6.30	0.000	.0245142	.0466514
lpop1	.0984294	.0077867	12.64	0.000	.0831666	.1136922
com_bound	.3419198	.0497928	6.87	0.000	.2443204	.4395193
lprinv1	.031037	.0148305	2.09	0.036	.0019675	.0601065
lpr2	.2836476	.0134761	21.05	0.000	.2572328	.3100623
ldensity2	.0442414	.0054105	8.18	0.000	.0336362	.0548466
isenglish1	.0824686	.018226	4.52	0.000	.0467437	.1181936
lpib	.074243	.0046615	15.93	0.000	.0651059	.0833801
issamecom6	.2024367	.0211722	9.56	0.000	.1609368	.2439366
_cons	2.020337	.2626426	7.69	0.000	1.505528	2.535146

Figure A.3: Best Least Squares Regression (p-value $< 5\%$).

Linear regression

Number of obs	20088
F(9, 20078)	2367.35
Prob > F	0.0000
R-squared	0.5198
Root MSE	2.0439

lmig2000	Coef.	Robust Std. Err.	t	P>|t|	[95% Conf.	Interval]
ldist	-1.103299	.0190721	-57.85	0.000	-1.140682	-1.065916
lpop2	.1869568	.0079092	23.64	0.000	.1714541	.2024595
lpop1	.4363622	.0111866	39.01	0.000	.4144354	.4582889
com_lang	.8043912	.0319681	25.16	0.000	.7417312	.8670512
com_bound	1.805189	.1003421	17.99	0.000	1.60851	2.001868
lprinv1	.392925	.0237584	16.54	0.000	.3463566	.4394934
lpr2	1.105356	.0159892	69.13	0.000	1.074016	1.136696
ldensity1	.0238726	.0107652	2.22	0.027	.002772	.0449733
issamecom6	1.351501	.0315206	42.88	0.000	1.289718	1.413284
_cons	11.17707	.381974	29.26	0.000	10.42837	11.92577

Figure A.4: Best Least Squares Regression without any diaspora parameter (p-value <5 %).

Linear regression

Number of obs	=	20088			
F(8, 20079)	=	2663.80			
Prob > F	=	0.0000			
R-squared	=	0.5197			
Root MSE	=	2.0441			

lmig2000	Coef.	Robust Std. Err.	t	P>|t|	[95% Conf.	Interval]
ldist	-1.105161	.0190413	-58.04	0.000	-1.142483	-1.067838
lpop2	.1868576	.0079113	23.62	0.000	.1713508	.2023644
lpop1	.4324272	.0110215	39.23	0.000	.4108242	.4540302
com_lang	.8049534	.0319675	25.18	0.000	.7422944	.8676123
com_bound	1.792602	.0999116	17.94	0.000	1.596767	1.988437
lprinv1	.4033702	.0231424	17.43	0.000	.3580093	.4487312
lpr2	1.105062	.0159946	69.09	0.000	1.073711	1.136413
issamecom6	1.351787	.0315151	42.89	0.000	1.290015	1.41356
_cons	11.41418	.3658533	31.20	0.000	10.69707	12.13128

Figure A.5: Best Least Squares Regression without any diaspora parameter (p-value <1 %)

A.2.3 Poisson Regression

To read and interpret the following results, a detailed example is available above A.1.

Poisson regression

Number of obs	=	11502			
Wald chi2(23)	=	17895.93			
Prob > chi2	=	0.0000			
Log pseudolikelihood = -21595722	Pseudo R2	=	0.9548		

mig2000	Coef.	Robust Std. Err.	z	P>|z|	[95% Conf.	Interval]
lmig1990	.8739704	.0370938	23.56	0.000	.8012679	.9466729
lmig1980	-.0467014	.0285163	-1.64	0.101	-.1025922	.0091895
lmig1970	.031318	.0179963	1.74	0.082	-.003954	.0665901
lmig1960	-.0207747	.0132361	-1.57	0.117	-.046717	.0051676
ldist	-.0780884	.0331956	-2.35	0.019	-.1431506	-.0130262
lpop2	.0618755	.0259249	2.39	0.017	.0110636	.1126874
lpop1	.0937775	.025416	3.69	0.000	.0439629	.143592
com_lang	-.0482548	.0554169	-0.87	0.384	-.1568698	.0603602
com_bound	.2019361	.0807968	2.50	0.012	.0435773	.360295
com_orig_l~g	.1704435	.0806701	2.11	0.035	.012333	.328554
lprinv1	.0302094	.0419354	0.72	0.471	-.0519825	.1124012
lprinv2	-.0424536	.0388971	-1.09	0.275	-.1186904	.0337832
lpr1	-.0235392	.0291892	-0.81	0.420	-.080749	.0336705
lpr2	.1319168	.0300269	4.39	0.000	.0730653	.1907684
ldensity1	-.048965	.020152	-2.43	0.015	-.0884622	-.0094677
ldensity2	.0246791	.0180276	1.37	0.171	-.0106544	.0600126
lidh1	.3481901	.1584742	2.20	0.028	.0375864	.6587939
lidh2	-.3584704	.183265	-1.96	0.050	-.7176633	.0007224
isenglish2	.0882065	.0583674	1.51	0.131	-.0261915	.2026046
isenglish1	-.0531723	.0640499	-0.83	0.406	-.1787077	.0723631
lpib	.1441598	.0364165	3.96	0.000	.0727847	.2155349
issamecom6	-.1060243	.0512931	-2.07	0.039	-.2065569	-.0054917
issamecom4	.1851292	.0474998	3.90	0.000	.0920314	.278227
_cons	.0898426	.9508149	0.09	0.925	-1.77372	1.953406

Figure A.6: Poisson Regression with our 23 parameters.

```
Iteration 0:   log pseudolikelihood = -6.951e+08
Iteration 1:   log pseudolikelihood = -2.589e+08  (backed up)
Iteration 2:   log pseudolikelihood =  -71300909
Iteration 3:   log pseudolikelihood =  -25785609
Iteration 4:   log pseudolikelihood =  -23662448
Iteration 5:   log pseudolikelihood =  -23661184
Iteration 6:   log pseudolikelihood =  -23661184

Poisson regression                              Number of obs   =      17094
                                                Wald chi2(8)    =   10486.97
                                                Prob > chi2     =     0.0000
Log pseudolikelihood =  -23661184               Pseudo R2       =     0.9567
```

mig2000	Coef.	Robust Std. Err.	z	P>|z|	[95% Conf.	Interval]
lmig1990	.8377809	.0119743	69.97	0.000	.8143118	.86125
ldist	-.1052675	.0336426	-3.13	0.002	-.1712059	-.0393291
lpop2	.0555464	.0158551	3.50	0.000	.024471	.0866219
lpop1	.0928607	.0173311	5.36	0.000	.0588924	.1268291
com_bound	.1513121	.08222	1.84	0.066	-.0098362	.3124604
lpr2	.1248717	.0344098	3.63	0.000	.0574297	.1923136
ldensity1	-.032368	.0177332	-1.83	0.068	-.0671245	.0023885
lpib	.0995725	.0118077	8.43	0.000	.0764298	.1227151
_cons	.8710784	.3813873	2.28	0.022	.1235731	1.618584

Figure A.7: Best Poisson Regression.

```
Iteration 0:   log pseudolikelihood = -7.012e+08
Iteration 1:   log pseudolikelihood = -2.331e+08  (backed up)
Iteration 2:   log pseudolikelihood = -2.218e+08
Iteration 3:   log pseudolikelihood =  -26371308
Iteration 4:   log pseudolikelihood =  -23816285
Iteration 5:   log pseudolikelihood =  -23763118
Iteration 6:   log pseudolikelihood =  -23763097
Iteration 7:   log pseudolikelihood =  -23763097

Poisson regression                              Number of obs   =      17094
                                                Wald chi2(7)    =    8334.92
                                                Prob > chi2     =     0.0000
Log pseudolikelihood =  -23763097               Pseudo R2       =     0.9565
```

mig2000	Coef.	Robust Std. Err.	z	P>|z|	[95% Conf.	Interval]
lmig1990	.8392448	.0123062	68.20	0.000	.8151251	.8633646
ldist	-.0917612	.0316546	-2.90	0.004	-.1538031	-.0297193
lpop2	.0454992	.0154566	2.94	0.003	.0152048	.0757935
lpop1	.086	.0167975	5.12	0.000	.0530775	.1189225
com_bound	.1796087	.0859299	2.09	0.037	.0111893	.3480282
lpr2	.128293	.0364429	3.52	0.000	.0568662	.1997197
lpib	.0941667	.0115019	8.19	0.000	.0716233	.1167101
_cons	.912956	.3975424	2.30	0.022	.1337873	1.692125

Figure A.8: Best Poisson Regression (p-value <5%).

```
Iteration 0:   log pseudolikelihood =  -30166630
Iteration 1:   log pseudolikelihood =  -30154613
Iteration 2:   log pseudolikelihood =  -30154611

Poisson regression                              Number of obs   =      18418
                                                Wald chi2(1)    =    1627.48
                                                Prob > chi2     =     0.0000
Log pseudolikelihood = -30154611                Pseudo R2       =     0.9468

------------------------------------------------------------------------------
             |               Robust
     mig2000 |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
     lmig1990 |   .9169029   .0227282    40.34   0.000     .8723563    .9614494
        _cons |   1.190753   .2454759     4.85   0.000     .7096293    1.671877
------------------------------------------------------------------------------
```

Figure A.9: Poisson Regression only with past migrations.

```
Iteration 0:   log pseudolikelihood = -1.938e+09
Iteration 1:   log pseudolikelihood = -1.168e+09  (backed up)
Iteration 2:   log pseudolikelihood = -2.669e+08
Iteration 3:   log pseudolikelihood = -1.672e+08
Iteration 4:   log pseudolikelihood = -1.657e+08
Iteration 5:   log pseudolikelihood = -1.657e+08
Iteration 6:   log pseudolikelihood = -1.657e+08

Poisson regression                              Number of obs   =      34089
                                                Wald chi2(11)   =    2829.15
                                                Prob > chi2     =     0.0000
Log pseudolikelihood = -1.657e+08               Pseudo R2       =     0.7461

------------------------------------------------------------------------------
             |               Robust
     mig2000 |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       ldist |  -.0478366   .0231287    -2.07   0.039    -.093168   -.0025052
        lpop2 |   .2776738   .0329882     8.42   0.000     .2130182    .3423295
        lpop1 |   .3191779   .0530447     6.02   0.000     .2152123    .4231435
     com_lang |   .8146983   .1376385     5.92   0.000     .5449318    1.084465
   com_bound |   2.064284    .156836    13.16   0.000     1.756891    2.371677
     lprinv1 |    .684335   .0804804     8.50   0.000     .5265964    .8420737
        lpr2 |   .7609689   .0462133    16.47   0.000     .6703925    .8515453
    ldensity2 |   .0431833   .0529703     0.82   0.415    -.0606365    .1470031
  isenglish1 |  -.492785   .1309172    -3.76   0.000    -.7493781   -.236192
        lpib |   .1655796   .0336864     4.92   0.000     .0995556    .2316037
  issamecom6 |   1.828639   .1557701    11.74   0.000     1.523335    2.133943
       _cons |   4.566758   1.492827     3.06   0.002     1.640872    7.492644
------------------------------------------------------------------------------
```

Figure A.10: Best Poisson Regression without any diaspora parameter.

```
Iteration 0:   log pseudolikelihood = -1.940e+09
Iteration 1:   log pseudolikelihood = -7.428e+08  (backed up)
Iteration 2:   log pseudolikelihood = -2.711e+08
Iteration 3:   log pseudolikelihood = -1.671e+08
Iteration 4:   log pseudolikelihood = -1.660e+08
Iteration 5:   log pseudolikelihood = -1.660e+08
Iteration 6:   log pseudolikelihood = -1.660e+08

Poisson regression                              Number of obs   =      34089
                                                Wald chi2(10)   =    2631.54
                                                Prob > chi2     =     0.0000
Log pseudolikelihood = -1.660e+08               Pseudo R2       =     0.7457
```

mig2000	Coef.	Robust Std. Err.	z	P>|z|	[95% Conf.	Interval]
ldist	-.0531668	.0221615	-2.40	0.016	-.0966026	-.009731
lpop2	.2855455	.0325026	8.79	0.000	.2218416	.3492494
lpop1	.3204015	.0538949	5.94	0.000	.2147694	.4260337
com_lang	.7973999	.1334803	5.97	0.000	.5357834	1.059016
com_bound	2.059786	.1579722	13.04	0.000	1.750166	2.369405
lprinv1	.6856506	.0793349	8.64	0.000	.5301572	.8411441
lpr2	.7462983	.0482121	15.48	0.000	.6518042	.8407923
isenglish1	-.4842202	.127058	-3.81	0.000	-.7332493	-.2351912
lpib	.1669608	.0343071	4.87	0.000	.0997201	.2342015
issamecom6	1.835347	.156362	11.74	0.000	1.528883	2.141811
_cons	4.568886	1.487171	3.07	0.002	1.654085	7.483687

Figure A.11: Best Poisson Regression without any diaspora parameter (p-value <5%).

```
Iteration 0:   log pseudolikelihood = -2.100e+09
Iteration 1:   log pseudolikelihood = -4.774e+08  (backed up)
Iteration 2:   log pseudolikelihood = -2.524e+08
Iteration 3:   log pseudolikelihood = -1.873e+08
Iteration 4:   log pseudolikelihood = -1.854e+08
Iteration 5:   log pseudolikelihood = -1.854e+08
Iteration 6:   log pseudolikelihood = -1.854e+08

Poisson regression                              Number of obs   =      38025
                                                Wald chi2(7)    =    1840.92
                                                Prob > chi2     =     0.0000
Log pseudolikelihood = -1.854e+08               Pseudo R2       =     0.7284
```

mig2000	Coef.	Robust Std. Err.	z	P>|z|	[95% Conf.	Interval]
lpop2	.2290409	.032169	7.12	0.000	.1659908	.2920909
lpop1	.2733527	.0511482	5.34	0.000	.173104	.3736014
com_lang	.6540568	.117681	5.56	0.000	.4234062	.8847073
com_bound	2.227188	.1694689	13.14	0.000	1.895035	2.559341
lprinv1	.6879909	.0802147	8.58	0.000	.530773	.8452088
lpr2	.8460789	.0454398	18.62	0.000	.7570185	.9351393
issamecom6	1.701253	.1399017	12.16	0.000	1.42705	1.975455
_cons	6.45357	1.247309	5.17	0.000	4.008889	8.89825

Figure A.12: Best Poisson Regression without any diaspora parameter with the 7 'core' criteria.

Appendix B

Code AMPL

```
1  set Countries;
2  param lambda;
3  param AdjacencyMatrix{Countries,Countries};
4  var delta{Countries,Countries} integer;
5  var isIn{Countries} integer>=0;
6  subject to boolean{c in Countries}:
7  isIn[c]<=1;
8  subject to constructionDelta{c in Countries,d in Countries}: delta[c,d]=isIn[c]+
   isIn[d]-1;
9  maximize TotalCost: sum{c in Countries,d in Countries} (AdjacencyMatrix[c,d]-lambda
   )*delta[c,d]
10 data Core.dat;
11 option solver cplex;
12 solve;
```

Appendix C

Resources used

On this appendix we will present every software or programming language used through our work.

C.1 TileMill

Description

- Software used to create map [112].

Usage

- Drawing the different colored maps.

C.2 Stata

Description

- Integrated statistical software package that provides tools for data analysis, data management, and graphics [4].

Usage

- Elaborating gravity models on chapter 5.

C.3 Java

Description

- Oriented-object programming language.

Libraries

- `JgraphT`[119] : Provides mathematical graph-theory objects and algorithms.
- `Junit`[2] : A programmer-oriented testing framework.

Usage

- Implementation of algorithms and metrics presented on section 3.10.

C.4 Python

Description

- Interpreted programming language.

Libraries

- `xlrd/xlwt` : Used to manipulate Excel file with Python [5].

Usage

- Implementation of scripts for creating the database (data retrieval, normalization, etc.) as explained in the chapter 2.
- Implementation of scripts for the automation of the creation of gravity model with Stata for different combinations of parameters as presented on chapter 5.
- Implementation of scripts for generating maps with TillMill.

C.5 Matlab

Description

- High-level language and interactive environment for numerical computation, visualization, and programming [113].

Libraries

- Stability toolbox [8] : Implements the stability method as introduced in the article [102].

Usage

- Computation of the resolutions presented on section 4.5.
- Implementation of algorithms to find the 2 and 3-cycles on section 4.2.

C.6 AMPL

Description

- Algebraic modeling language for describing and solving high-complexity problems for large-scale mathematical computation like optimization[70].

Usage

- Computation of core and periphery components on section 4.4.

C.7 Microsoft Excel

Description

- Spreadsheet application developed by Microsoft.

Usage

- Representation, classification and storage of the data.
- Drawing diverse graph for the communities (4.5) analysis.

C.8 R

Description

- Free software environment for statistical computing and graphics.

Libraries

- `maps`[19] : Draw geographical maps.
- `geosphere`[81] : Spherical trigonometry for geographic applications.

Usage

- Drawing the maps 3.15, 4.1 and 4.2 (Code inspired by [104])

C.9 LucidChart

Description

- Web-based diagramming software [108].

Usage

- Drawing the graphs 3.12, 3.14 and 4.6.

C.10 Gephi

Description

- Free software environment for statistical computing and graphics.

Usage

- Compute some values used for the ranking (chapter 3) like the pageRank, the weighted degrees and the eigenvector centrality.
- Compute the community partition.

C.11 Mendeley

Description

- Free reference manager for students and researchers [3].

Usage

- Help to make the bibliography.

Appendix D

Details about data

D.1 List of root languages

- Afro-Asiatic
- Algonquian
- Austroasiatic
- Austronesian
- Aymaran
- Creole
- Dené–Yeniseian
- Dravidian
- Eskimo–Aleut
- Indo-European
- Japonic
- Koreanic
- Mongolic
- Niger–Congo
- Nilo-Saharan
- Northeast Caucasian
- Quechuan
- Sino-Tibetan
- South Caucasian
- Tai–Kadai
- Tupian
- Turkic
- Uralic

D.2 Information about countries

The purpose of this appendix is to make a short summary of information about the countries mentionned in this paper.
Here follows the different criteria we used and their source :

Population(2000) : Geohive [73]

GDP(2000) : World Bank [160] completed with International Monetary Fund [90] if data were missing.

HDI(2000) : Geohive [72]

Contribution of natural ressources in the GDP : World Bank [168]

Total emigration(1990→2000) : Own computation from the data of the WorldBank [169]

Total immigration(1990→2000) : Own computation from the data of the WorldBank [169]

Extra information :

Offshore Financial Centers : International Monetary Fund [91]

Tax haven : There is no unified list of tax havens, as each country, entity or organization has its own criteria for evaluation, often with a significant degree of subjectivity. We decided to use the report of the Organisation for Economic Co-operation and Development [122]

D.2.1 Albania

Population(2000) : 3'305'000
GDP(2000) : 3.64 B$
HDI(2000) : 0.691
Contribution of natural ressources in the GDP :

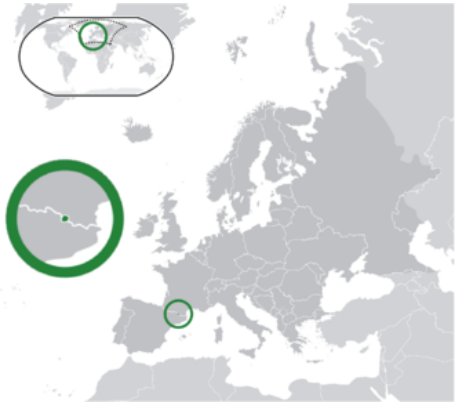
Total	Oil	Natural Gas	Coal	Mineral	Forest
5.5%	4.6%	0%	0%	0.2%	0.7%

Total emigration (1990→2000) : 1'024'015
Total immigration (1990→2000) : 74'454

Extra information :

D.2.2 Andorra

Population(2000) : 65'000
GDP(2000) : 1'134 B$
HDI(2000) : 0.80 (computed from its value in 2010)
Contribution of natural ressources in the GDP :
Missing value
Total emigration (1990→2000) : 4'208
Total immigration (1990→2000) : 42'123
Extra information : Tax haven and offshore financial center



D.2.3 Angola

Population(2000) : 13'925'000
GDP(2000) : 9.130 B$
HDI(2000) : 0.384
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
42.9%	42.6%	0%	0%	0%	0.3%

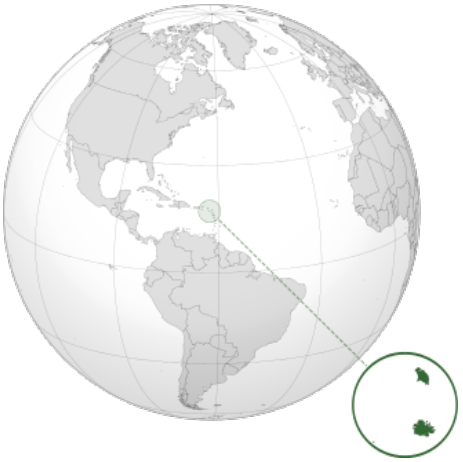
Total emigration (1990→2000) : 375'013
Total immigration (1990→2000) : 33'501
Extra information :

D.2.4 Antigua and Barbuda

Population(2000) : 78'000
GDP(2000) : 788 M$
HDI(2000) : 0.721 (computed from its value in 2010)
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
0%	0%	0 %	0%	0%	0%

Total emigration (1990→2000) : 41'789
Total immigration (1990→2000) : 15'816
Extra information : Offshore financial center



D.2.5 Armenia

Population(2000) : 3'043'000
GDP(2000) : 1.912 B$
HDI(2000) : 0.643
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
5.2%	0%	0%	0%	3.7%	1.5%

Total emigration (1990→2000) : 853'833
Total immigration (1990→2000) : 291'228
Extra information : Former part of the USSR

D.2.6 Australia

Population(2000) : 19'259'000
GDP(2000) : 415.208 B$
HDI(2000) : 0.906
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
8%	0.8%	0.6%	0.9%	5.5%	0.1%

Total emigration (1990→2000) : 461'557
Total immigration (1990→2000) : 4'027'347
Extra information :

D.2.7 Azerbaijan

Population(2000) : 8'463'000
GDP(2000) : 5.273 B$
HDI(2000) : 0.657 (computed from its value in 2010)
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
39.8%	36.9%	2.7%	0%	0.2%	0%

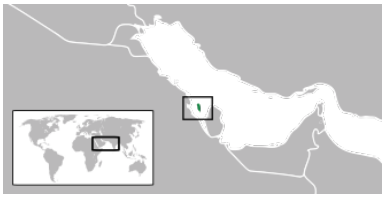
Total emigration (1990→2000) : 1'508'195
Total immigration (1990→2000) : 261'257
Extra information : Former part of the USSR

D.2.8 Bahrain

Population(2000) : 655'000
GDP(2000) : 9.063 B$
HDI(2000) : 0.773
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
23.7%	19.4%	4.3%	0%	0%	0%

Total emigration (1990→2000) : 35'018
Total immigration (1990→2000) : 239'348
Extra information : Tax haven and Offshore financial center



D.2.9 Bangladesh

Population(2000) : 132'151'000
GDP(2000) : 47.125 B$
HDI(2000) : 0.422
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
4.2%	0%	2.7%	0%	0%	1.4%

Total emigration (1990→2000) : 4'987'708
Total immigration (1990→2000) : 965'907

Extra information :

D.2.10 Belarus

Population(2000) : 10'034'000
GDP(2000) : 12.737 B$
HDI(2000) : 0.702 (computed from its value in 2005)
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
2.5%	1.5%	0%	0%	0%	1%

Total emigration (1990→2000) : 1'746'885
Total immigration (1990→2000) : 1'139'891
Extra information : Former part of the USSR

D.2.11 Belgium

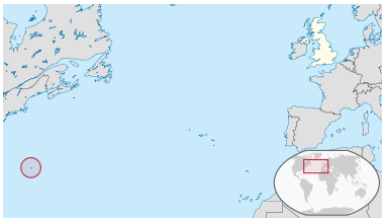
Population(2000) : 10'264'000
GDP(2000) : 232.673 B$
HDI(2000) : 0.876
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
0.1%	0%	0%	0%	0%	0.1%

Total emigration (1990→2000) : 340'734
Total immigration (1990→2000) : 879'008

Extra information :

D.2.12 Bermuda



Population(2000) : 63'000
GDP(2000) : 3.480 B$
HDI(2000) : Missing value
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
0%	0%	0%	0%	0%	0%

Total emigration (1990→2000) : 18'261
Total immigration (1990→2000) : 17'647
Extra information : Offshore financial center

D.2.13 Bosnia and Herzegovina

Population(2000) : 4'035'000
GDP(2000) : 5.506 B$
HDI(2000) : 0.696 (computed from its value in 2005)
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
2.3%	0%	0%	0.1%	0.9%	1.3%

Total emigration (1990→2000) : 1'293'856
Total immigration (1990→2000) : 44'058
Extra information : The country has been at War between 1992 and 1995 during the collapse of Yugoslavia [66].

D.2.14 Burkina Faso

Population(2000) : 11'588'000
GDP(2000) : 2.611 B$
HDI(2000) : 0.281 (computed from its value in 2005)
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
22.1%	0%	0%	0%	14.9%	7.3%

Total emigration (1990→2000) : 1'389'678
Total immigration (1990→2000) : 572'113

Extra information :

D.2.15 Cameroon

Population(2000) : 15'343'000
GDP(2000) : 9.287 B$
HDI(2000) : 0.427
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
11.6%	8.3%	0.2%	0%	0.3%	2.9%

Total emigration (1990→2000) : 138'138
Total immigration (1990→2000) : 181'914
Extra information :

D.2.16 Canada

Population(2000) : 31'100'000
GDP(2000) : 724.919 B$
HDI(2000) : 0.879
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
4.5%	3.2%	0%	0.1%	0.8%	0.4%

Total emigration (1990→2000) : 1'255'255
Total immigration (1990→2000) : 5'554'485
Extra information :

D.2.17 Cayman Islands

Population(2000) : 38'000
GDP(2000) : Missing value
HDI(2000) : Missing value
Contribution of natural ressources in the GDP :
Missing value
Total emigration (1990→2000) : 1'502
Total immigration (1990→2000) : 23'868
Extra information : Offshore financial center and often considered as a tax haven. [149]

D.2.18 China

Population(2000) : 1'263'638'000
GDP(2000) : 1'198.475 B$
HDI(2000) : 0.588
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
5.8%	1.4%	0.1%	0%	15.3%	1.9%

Total emigration (1990→2000) : 5'814'483
Total immigration (1990→2000) : 214'306
Unemployement rate (2000) : 3.1%
Composition of GDP by sector :

Agriculture	Industry	Services
9.7%	45.3%	45%

Extra information :

D.2.19 Cook Islands

Population(2000) : 18'000
GDP(2000) : Missing value
HDI(2000) : Missing value
Contribution of natural ressources in the GDP : Missing value
Total emigration (1990→2000) : 21'102
Total immigration (1990→2000) : 2'762
Extra information : Tax haven and Offshore financial center

D.2.20 Cuba

Population(2000) : 11'072'000
GDP(2000) : 30.565 B$
HDI(2000) : 0.681
Contribution of natural ressources in the GDP : Missing value
Total emigration (1990→2000) : 1'052'296
Total immigration (1990→2000) : 16'797
Extra information : Cuba has known a socioeconomic collapse in 1990-1995, when the country lost the funding from the Soviet Union [43].

D.2.21 Djibouti

Population(2000) : 669'000
GDP(2000) : 551 M$
HDI(2000) : 0.381 (computed from its value in 2005)
Contribution of natural ressources in the GDP :
Total emigration (1990→2000) : 4'502
Total immigration (1990→2000) : 86'920
Extra information :

D.2.22 El Salvador

Population(2000) : 5'850'000
GDP(2000) : 13.134 B$
HDI(2000) : 0.619
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
1.5%	0%	0%	0%	0%	1.5%

Total emigration (1990→2000) : 935'250
Total immigration (1990→2000) : 31'657
Extra information : Between 1979 and 1992, there's been a civil war in El Salvador [143]. This resulted in a huge migration to the United States [100].

D.2.23 Eritrea

Population(2000) : 4'197'000
GDP(2000) : 706 M$
HDI(2000) : 0.303 (computed from its value in 2005)
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
15.5%	0%	0%	0%	13.5%	2%

Total emigration (1990→2000) : 359'218
Total immigration (1990→2000) : 10'451
Extra information : Huge migration of refugees from Eritrea to Ethiopia due to the civil war occuring during this period [139].

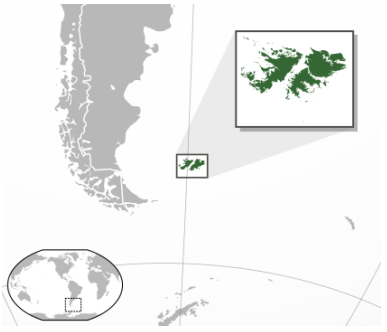
D.2.24 Ethiopia

Population(2000) : 66'024'000
GDP(2000) : 8.091 B$
HDI(2000) : 0.274
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
14.2%	0%	0%	0%	1.1%	13.1%

Total emigration (1990→2000) : 291'240
Total immigration (1990→2000) : 434'594
Extra information :

D.2.25 Falkland Islands



Population(2000) : 3'000
GDP(2000) : Missing value
HDI(2000) : Missing value
Contribution of natural ressources in the GDP :
Missing value
Total emigration (1990→2000) : 1'513
Total immigration (1990→2000) : 1'558
Extra information :

D.2.26 France

Population(2000) : 61'137'000
GDP(2000) : 1'326.334 B$
HDI(2000) : 0.846
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
0.2%	0%	0%	0%	0%	0.2%

Total emigration (1990→2000) : 1'766'513
Total immigration (1990→2000) : 6'259'369
Extra information :

D.2.27 French Guiana

Population(2000) : 165'000
GDP(2000) : Missing value
HDI(2000) : Missing value
Contribution of natural ressources in the GDP :
Missing value
Total emigration (1990→2000) : 7'853
Total immigration (1990→2000) : 80'029
Extra information :

D.2.28 Gabon

Population(2000) : 1'236'000
GDP(2000) : 5.068 B$
HDI(2000) : 0.621
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
45.3%	42.8%	0.1%	0%	0.1%	2.3%

Total emigration (1990→2000) : 15'054
Total immigration (1990→2000) : 193'940
Extra information :

D.2.29 Gambia

Population(2000) : 1'357'000
GDP(2000) : 783 M$
HDI(2000) : 0.36
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
5.3%	0%	0%	0%	0%	5.3%

Total emigration (1990→2000) : 35'522
Total immigration (1990→2000) : 170'474
Extra information :

D.2.30 Georgia

Population(2000) : 4'777'000
GDP(2000) : 3.057 B$
HDI(2000) : 0.686 (computed from its value in 2005)
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
0.9%	0.2%	0%	0%	0.3%	0.4%

Total emigration (1990→2000) : 1'139'643
Total immigration (1990→2000) : 219'036
Extra information :

D.2.31 Germany

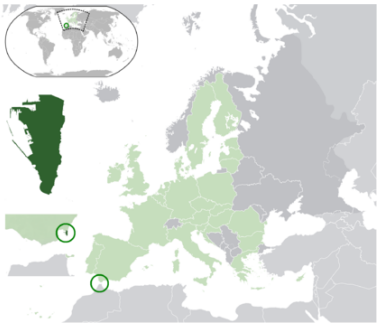
Population(2000) : 82'184'000
GDP(2000) : 1'886.401 B$
HDI(2000) : 0.864
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
0.2%	0%	0%	0%	0%	0.1%

Total emigration (1990→2000) : 3'602'063
Total immigration (1990→2000) : 11'134'583
Extra information :

D.2.32 Gibraltar

Population(2000) : 27'000
GDP(2000) : Missing value
HDI(2000) : Missing value
Contribution of natural ressources in the GDP :
Missing value
Total emigration (1990→2000) : 4'564
Total immigration (1990→2000) : 8'100
Extra information : Tax haven and Offshore financial center



D.2.33 Grenada

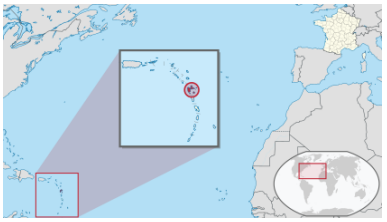
Population(2000) : 102'000
GDP(2000) : 523 M$
HDI(2000) : 0.704 (computed from its value in 2010)
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
0%	0%	0%	0%	0%	0%

Total emigration (1990→2000) : 59'717
Total immigration (1990→2000) : 7'904
Extra information : Tax haven and Offshore financial center

D.2.34 Guadeloupe

Population(2000) : 155'000
GDP(2000) : Missing value
HDI(2000) : Missing value
Contribution of natural ressources in the GDP :
Missing value
Total emigration (1990→2000) : 244'696
Total immigration (1990→2000) : 80'913
Extra information :



D.2.35 Guyana

Population(2000) : 786'000
GDP(2000) : 713 M$
HDI(2000) : 0.579
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
20.4%	0%	0%	0%	16%	4.4%

Total emigration (1990→2000) : 353'917
Total immigration (1990→2000) : 7'960
Extra information : Due to its history, Guyana is inhabited by people coming from diverses countries forming diasporas. However, The political and economic conditions of the country and the symbolic linkages to the homeland led to many emigrations [125].

D.2.36 Haiti

Population(2000) : 8'413'000
GDP(2000) : 3.665 B$
HDI(2000) : 0.421
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
1.9%	0%	0%	0%	0%	1.9%

Total emigration (1990→2000) : 774'643
Total immigration (1990→2000) : 25'788
Extra information : Refugee : After the coup d'etat in 1991 where the Haitian president Aristide was overthrown, a lot of Haitian were seeking refuge in the United States due to the human rights violations. [82]

D.2.37 India

Population(2000) : 1'006'300'000
GDP(2000) : 476.609 B$
HDI(2000) : 0.461
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
5.6%	1.2%	0.3%	1.4%	1.6%	1.2%

Total emigration (1990→2000) : 9'516'765
Total immigration (1990→2000) : 6'235'768
Extra information : Tax haven

D.2.38 Iraq

Population(2000) : 22'679'000
GDP(2000) : 16.9 B$
HDI(2000) : 0.531 (computed from its value in 2005)
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
46%	45.5%	0.4%	0%	0%	0%

Total emigration (1990→2000) : 1'030'084
Total immigration (1990→2000) : 18'559
Extra information : Iraq has been at War against the coalition forces during the Gulf War (2 August 1990 → 28 February 1991). This induced a post-crisis environment which lead people to leave the country [47].

D.2.39 Israel

Population(2000) : 6'115'000
GDP(2000) : 124.895 B$
HDI(2000) : 0.856
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
0.3%	0%	0.1%	0%	0.1%	0%

Total emigration (1990→2000) : 712'009
Total immigration (1990→2000) : 2'254'116
Extra information : Huge migration from ex-URSS countries on the early 1990. [138]

D.2.40 Italy

Population(2000) : 57'784'000
GDP(2000) : 1'104.009 B$
HDI(2000) : 0.825
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
0.2%	0.2%	0%	0%	0%	0%

Total emigration (1990→2000) : 3'136'248
Total immigration (1990→2000) : 2'122'478
Extra information :

D.2.41 Ivory Coast

Population(2000) : 16'885'000
GDP(2000) : 10.417 B$
HDI(2000) : 0.374
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
8.8%	4.1%	1%	0%	1.9%	1.8%

Total emigration (1990→2000) : 549'079
Total immigration (1990→2000) : 2'206'780
Extra information :

D.2.42 Jamaica

Population(2000) : 2'616'000
GDP(2000) : 9.009 B$
HDI(2000) : 0.68
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
2.7%	0%	0%	0%	2.3%	0.4%

Total emigration (1990→2000) : 928'578
Total immigration (1990→2000) : 24'503
Extra information : Huge migration of educated and skilled people in 1990s and after. It is especially due to an unstable economy and to a mismatch between the skills needed in the country [152].

D.2.43 Kazakhstan

Population(2000) : 15'687'000
GDP(2000) : 18.292 B$
HDI(2000) : 0.657
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
32.1%	24.9%	2.2%	2.6%	2.5%	0%

Total emigration (1990→2000) : 3'382'369
Total immigration (1990→2000) : 2'835'254
Extra information : Former part of the USSR

D.2.44 Korea, Dem. Rep

Population(2000) : 22'840'000
GDP(2000) : Missing value
HDI(2000) : Missing value
Contribution of natural ressources in the GDP :
Missing value
Total emigration (1990→2000) : 544'190
Total immigration (1990→2000) : 36'143
Extra information : I do not want to get into any trouble ;-)

D.2.45 Korea, Rep

Population(2000) : 46'839'000
GDP(2000) : 533'384 B$
HDI(2000) : 0.83
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
0.1%	0%	0%	0%	0%	0%

Total emigration (1990→2000) : 1'896'741
Total immigration (1990→2000) : 568'054
Extra information :

D.2.46 Kuwait

Population(2000) : 1'972'000
GDP(2000) : 37.718 B$
HDI(2000) : 0.754
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
55.1%	53.8%	1.3%	0%	0%	0%

Total emigration (1990→2000) : 362'943
Total immigration (1990→2000) : 1'496'866
Extra information :

D.2.47 Kyrgyz Republic

Population(2000) : 4'851'000
GDP(2000) : 1.370 B$
HDI(2000) : 0.577
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
15%	0.8%	0%	0.3%	13.9%	0.1%

Total emigration (1990→2000) : 694'967
Total immigration (1990→2000) : 397'507
Extra information : Former part of the USSR

D.2.48 Liechtenstein

Population(2000) : 33'000
GDP(2000) : 2.484 B$
HDI(2000) : 0.862 (computed from its value in 2010)
Contribution of natural ressources in the GDP :
Missing value
Total emigration (1990→2000) : 4'497
Total immigration (1990→2000) : 11'204
Extra information : Tax haven and Offshore financial center

D.2.49 Luxembourg

Population(2000) : 439'000
GDP(2000) : 20.268 B$
HDI(2000) : 0.854
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
0.2%	0%	0%	0%	0.1%	0.1%

Total emigration (1990→2000) : 24'383
Total immigration (1990→2000) : 140'796
Extra information : Offshore financial center

D.2.50 Macao SAR, China

Population(2000) : 432'000
GDP(2000) : 6.102 B$
HDI(2000) : Missing value
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
0%	0%	0%	0%	0%	0%

Total emigration (1990→2000) : 96'871
Total immigration (1990→2000) : 240'286
Extra information : Offshore financial center, often considered as a tax haven [148]

D.2.51 Madagascar

Population(2000) : 15'742'000
GDP(2000) : 3.878 B$
HDI(2000) : 0.427
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
8.8%	0%	0%	0%	2.2%	6.6%

Total emigration (1990→2000) : 66'684
Total immigration (1990→2000) : 41'669
Extra information : Tax haven

D.2.52 Maldives

Population(2000) : 273'000
GDP(2000) : 624 M$
HDI(2000) : 0.576
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
0%	0%	0%	0%	0%	0%

Total emigration (1990→2000) : 737
Total immigration (1990→2000) : 3'009
Extra information :

D.2.53 Mali

Population(2000) : 10'621'000
GDP(2000) : 2.422 B$
HDI(2000) : 0.275
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
16.1%	0%	0%	0%	12.9%	3.2%

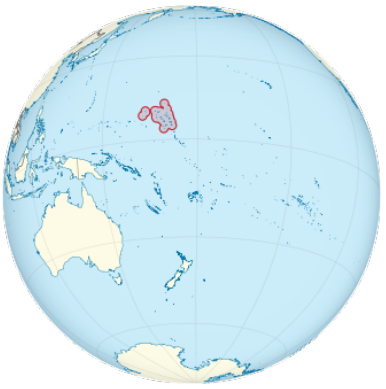
Total emigration (1990→2000) : 773'801
Total immigration (1990→2000) : 155'619
Extra information :

D.2.54 Marshall Islands

Population(2000) : 53'000
GDP(2000) : 111 M$
HDI(2000) : Missing value
Contribution of natural ressources in the GDP :

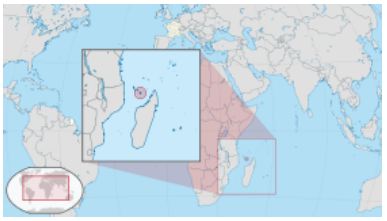
Total	Oil	Natural Gas	Coal	Mineral	Forest
0%	0%	0%	0%	0%	0%

Total emigration (1990→2000) : 5'963
Total immigration (1990→2000) : 1'616
Extra information : Tax haven and Offshore financial center



D.2.55 Mayotte

Population(2000) : 149'000
GDP(2000) : Missing value
HDI(2000) : Missing value
Contribution of natural ressources in the GDP :
Missing value
Total emigration (1990→2000) : 1'116
Total immigration (1990→2000) : 40'139
Extra information : Tax haven and Offshore financial center



D.2.56 Mexico

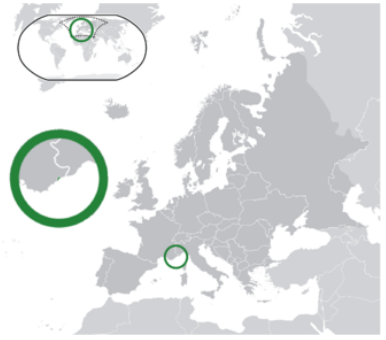
Population(2000) : 99'927'000
GDP(2000) : 692.178 B$
HDI(2000) : 0.718
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
8.6%	6.8%	0.5%	0%	1%	0.2%

Total emigration (1990→2000) : 9'550'582
Total immigration (1990→2000) : 499'216
Extra information :

D.2.57 Monaco

Population(2000) : 32'000
GDP(2000) : 2.648 B$
HDI(2000) : Missing value
Contribution of natural ressources in the GDP :
Missing value
Total emigration (1990→2000) : 2'744
Total immigration (1990→2000) : 21'683
Extra information : Tax haven and Offshore financial center



D.2.58 Mongolia

Population(2000) : 2'664'000
GDP(2000) : 1.137 B$
HDI(2000) : 0.555
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
28.7%	2.1%	0%	14.1%	12%	0.6%

Total emigration (1990→2000) : 4'863
Total immigration (1990→2000) : 8'200
Extra information :

D.2.59 Montserrat

Population(2000) : 5'000
GDP(2000) : Missing value
HDI(2000) : Missing value
Contribution of natural ressources in the GDP :
Missing value
Total emigration (1990→2000) : 4'487
Total immigration (1990→2000) : 163
Extra information : Tax haven and Offshore financial center

D.2.60 Morocco

Population(2000) : 28'113'000
GDP(2000) : 37.021 B$
HDI(2000) : 0.507
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
5%	0%	0%	0%	4.4%	0.5%

Total emigration (1990→2000) : 1'615'047
Total immigration (1990→2000) : 54'888
Extra information : During the 1960s, the strong economic growth in Western Europe required an high demand for low-skilled labor yielding a migration from Morocco to Europe. The consequence of this event was a continuous huge family migration the next years forming by snowball effect a diaspora in Western Europe. [48]

D.2.61 Nepal

Population(2000) : 23'184'000
GDP(2000) : 5.494 B$
HDI(2000) : 0.398
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
5%	0%	0%	0%	0%	5%

Total emigration (1990→2000) : 768'652
Total immigration (1990→2000) : 589'141
Extra information :

D.2.62 Netherlands

Population(2000) : 15'930'000
GDP(2000) : 385.075 B$
HDI(2000) : 0.882
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
1%	0.1%	0.9%	0%	0%	0%

Total emigration (1990→2000) : 767'609
Total immigration (1990→2000) : 1'501'879
Extra information :

D.2.63 Nigeria

Population(2000) : 124'207'000
GDP(2000) : 48.386 B$
HDI(2000) : 0.408 (computed from its value in 2005)
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
30.3%	26.8%	1.9%	0%	0%	1.6%

Total emigration (1990→2000) : 659'209
Total immigration (1990→2000) : 744'018
Extra information :

D.2.64 Niue

Population(2000) : 2'000
GDP(2000) : Missing value
HDI(2000) : Missing value
Contribution of natural ressources in the GDP :
Missing value
Total emigration (1990→2000) : 6'176
Total immigration (1990→2000) : 389
Extra information : Tax haven and Offshore financial center

D.2.65 Norfolk Islands

Population(2000) : 2'000
GDP(2000) : Missing value
HDI(2000) : Missing value
Contribution of natural ressources in the GDP :
Missing value
Total emigration (1990→2000) : 502
Total immigration (1990→2000) : 958
Extra information :

D.2.66 Northern Mariana Islands

Population(2000) : 70'000
GDP(2000) : Missing value
HDI(2000) : Missing value
Contribution of natural ressources in the GDP :
Missing value
Total emigration (1990→2000) : 10'750
Total immigration (1990→2000) : 44'843
Extra information :

D.2.67 Pakistan

Population(2000) : 152'429'000
GDP(2000) : 73.952 B$
HDI(2000) : 0.436
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
3.7%	0.9%	1.8%	0%	0.1%	1%

Total emigration (1990→2000) : 3'812'231
Total immigration (1990→2000) : 2'640'929
Extra information :

D.2.68 Palau

Population(2000) : 19'000
GDP(2000) : 155 M$
HDI(2000) : Missing value
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
0%	0%	0%	0%	0%	0%

Total emigration (1990→2000) : 18'178
Total immigration (1990→2000) : 6'276
Extra information :

D.2.69 Philipinnes

Population(2000) : 81'222'000
GDP(2000) : 81.026 B$
HDI(2000) : 0.602
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
3.5%	0.9%	1.8%	0%	0.1%	1%

Total emigration (1990→2000) : 3'083'089
Total immigration (1990→2000) : 322'483
Extra information :

D.2.70 Poland

Population(2000) : 38'654'000
GDP(2000) : 171.276 B$
HDI(2000) : 0.77
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
1.9%	0.1%	0.1%	0.7%	0.6%	0.4%

Total emigration (1990→2000) : 5'146'963
Total immigration (1990→2000) : 822'338
Extra information : Member of the Warsaw Pact [6] with the rest of the east bloc.

D.2.71 Puerto Rico

Population(2000) : 3'814'000
GDP(2000) : 61.702 B$
HDI(2000) : Missing value
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
0%	0%	0%	0%	0%	0%

Total emigration (1990→2000) : 1'490'469
Total immigration (1990→2000) : 354'204
Extra information : Commonly labelled as tax haven [14]

D.2.72 Qatar

Population(2000) : 640'000
GDP(2000) : 17.76 B$
HDI(2000) : 0.784
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
24.2%	11.9%	12.3%	0%	0%	0%

Total emigration (1990→2000) : 4'898
Total immigration (1990→2000) : 470'684
Extra information :

D.2.73 Russian Federation

Population(2000) : 146'710'000
GDP(2000) : 259.708 B$
HDI(2000) : 0.691
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
18.7%	13.9%	2.3%	0.9%	1.3%	0.3%

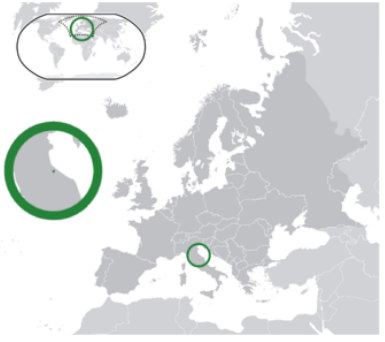
Total emigration (1990→2000) : 10'375'654
Total immigration (1990→2000) : 12'050'715
Extra information : Former part of the USSR

D.2.74 Saint Pierre and Miquelon

Population(2000) : 6'000
GDP(2000) : Missing value
HDI(2000) : Missing value
Contribution of natural ressources in the GDP :
Missing value
Total emigration (1990→2000) : 169
Total immigration (1990→2000) : 1'416
Extra information :

D.2.75 San Marino

Population(2000) : 27'000
GDP(2000) : 774 M$
HDI(2000) : Missing value
Contribution of natural ressources in the GDP :
Missing value
Total emigration (1990→2000) : 3'630
Total immigration (1990→2000) : 9'190
Extra information : Often considered as a tax haven. [149]



D.2.76 Saudi Arabia

Population(2000) : 21'312'000
GDP(2000) : 188.442 B$
HDI(2000) : 0.726
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
49.7%	47.2%	2.5%	0%	0%	0%

Total emigration (1990→2000) : 213'383
Total immigration (1990→2000) : 5'130'955
Extra information :

D.2.77 Serbia

Population(2000) : 10'272'000
GDP(2000) : 6.083 B$
HDI(2000) : 0.719
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
4.3%	1.5%	0.1%	0.6%	0.5%	1.6%

Total emigration (1990→2000) : 1'922'243
Total immigration (1990→2000) : 363'797
Extra information :

D.2.78 Somalia

Population(2000) : 7'501'000
GDP(2000) : 2.1 B$
HDI(2000) : Missing value
Contribution of natural ressources in the GDP :
Missing value
Total emigration (1990→2000) : 376'456
Total immigration (1990→2000) : 19'501
Extra information :

D.2.79 South Africa

Population(2000) : 45'064'000
GDP(2000) : 132.878 B$
HDI(2000) : 0.616
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
7.9%	0%	0%	3.4%	3.9%	0.4%

Total emigration (1990→2000) : 691'282
Total immigration (1990→2000) : 999'206
Extra information :

D.2.80 Spain

Population(2000) : 40'589'000
GDP(2000) : 580.345 B$
HDI(2000) : 0.839
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
0.2%	0%	0%	0%	0%	0.1 %

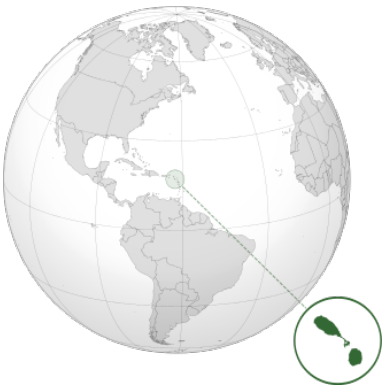
Total emigration (1990→2000) : 1'110'832
Total immigration (1990→2000) : 1'752'868
Extra information :

D.2.81 St. Kitts and Nevis

Population(2000) : 46'000
GDP(2000) : 417 M$
HDI(2000) : 0.693 (computed from its value in 2010)
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
0%	0%	0%	0%	0%	0%

Total emigration (1990→2000) : 28'267
Total immigration (1990→2000) : 3'834
Extra information : Offshore financial center



D.2.82 Switzerland

Population(2000) : 7'267'000
GDP(2000) : 256.043 B$
HDI(2000) : 0.873
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
0%	0%	0%	0%	0%	0%

Total emigration (1990→2000) : 330'364
Total immigration (1990→2000) : 1'562'598
Extra information : Offshore financial center

D.2.83 Syrian Arab Republic

Population(2000) : 16'471'000
GDP(2000) : 19.326 B$
HDI(2000) : 0.583
Contribution of natural ressources in the GDP (2004) :

Total	Oil	Natural Gas	Coal	Mineral	Forest
25.3 [167] %	21.1 [166] %	4.2 [165]%	0%	0%	0%

Total emigration (1990→2000) : 587'512
Total immigration (1990→2000) : 535'900
Extra information :

D.2.84 Tokelau

Population(2000) : 2'000
GDP(2000) : Missing value
HDI(2000) : Missing value
Contribution of natural ressources in the GDP :
Missing value
Total emigration (1990→2000) : 2'403
Total immigration (1990→2000) : 246
Extra information :

D.2.85 Turkey

Population(2000) : 67'329'000
GDP(2000) : 266.568 B$
HDI(2000) : 0.634
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
0.7%	0.2%	0%	0%	0.3%	0.2%

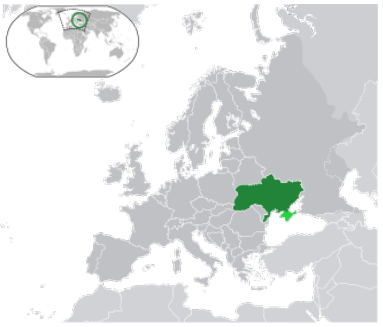
Total emigration (1990→2000) : 3'001'362
Total immigration (1990→2000) : 1'260'165
Extra information :

D.2.86 Ukraine

Population(2000) : 49'005'000
GDP(2000) : 31.262 B$
HDI(2000) : 0.669
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
4.6%	0.8%	1.1%	2.1%	0%	0.6%

Total emigration (1990→2000) : 5'915'949
Total immigration (1990→2000) : 5'206'151
Extra information : Former part of the USSR



D.2.87 United Arab Emirates

Population(2000) : 3'219'000
GDP(2000) : 10.337 B$
HDI(2000) : 0.753
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
23.8%	21.9%	1.9%	0%	0%	0%

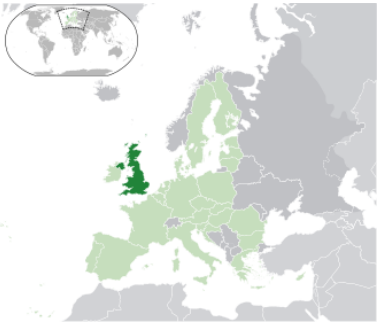
Total emigration (1990→2000) : 72'970
Total immigration (1990→2000) : 2'285'611
Extra information : Offshore financial center

D.2.88 United Kingdom

Population(2000) : 59'140'000
GDP(2000) : 1'493.628 B$
HDI(2000) : 0.833
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
1.2%	1%	0.2%	0%	0%	0%

Total emigration (1990→2000) : 4'061'454
Total immigration (1990→2000) : 4'865'946
Extra information :



D.2.89 United States

Population(2000) : 282'172'000
GDP(2000) : 10'289.7 B$
HDI(2000) : 0.897
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
1.3%	0.9%	-0.1%	0.2%	0.2%	0.1%

Total emigration (1990→2000) : 2'181'691
Total immigration (1990→2000) : 34'811'958
Extra information :

D.2.90 Uzbekistan

Population(2000) : 25'042'000
GDP(2000) : 13.760 B$
HDI(2000) : 0.59 (computed from its value in 2005)
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
20.9%	3.4%	8.5%	0.1%	8.9%	0%

Total emigration (1990→2000) : 1'665'468
Total immigration (1990→2000) : 1'363'986
Extra information : Former part of the USSR

D.2.91 Vietnam

Population(2000) : 79'178'000
GDP(2000) : 33.64 B$
HDI(2000) : 0.528
Contribution of natural ressources in the GDP :

Total	Oil	Natural Gas	Coal	Mineral	Forest
11.8%	7.6%	0.9%	1.3%	0.5%	1.5%

Total emigration (1990→2000) : 1'748'815
Total immigration (1990→2000) : 40'599
Extra information : Since the Vietnam War, there's been a huge migration from Vietnam to the United States [41]. The fall of the Soviet bloc induced a raise of migrations in the early 1990*s* (see Figure D.1).

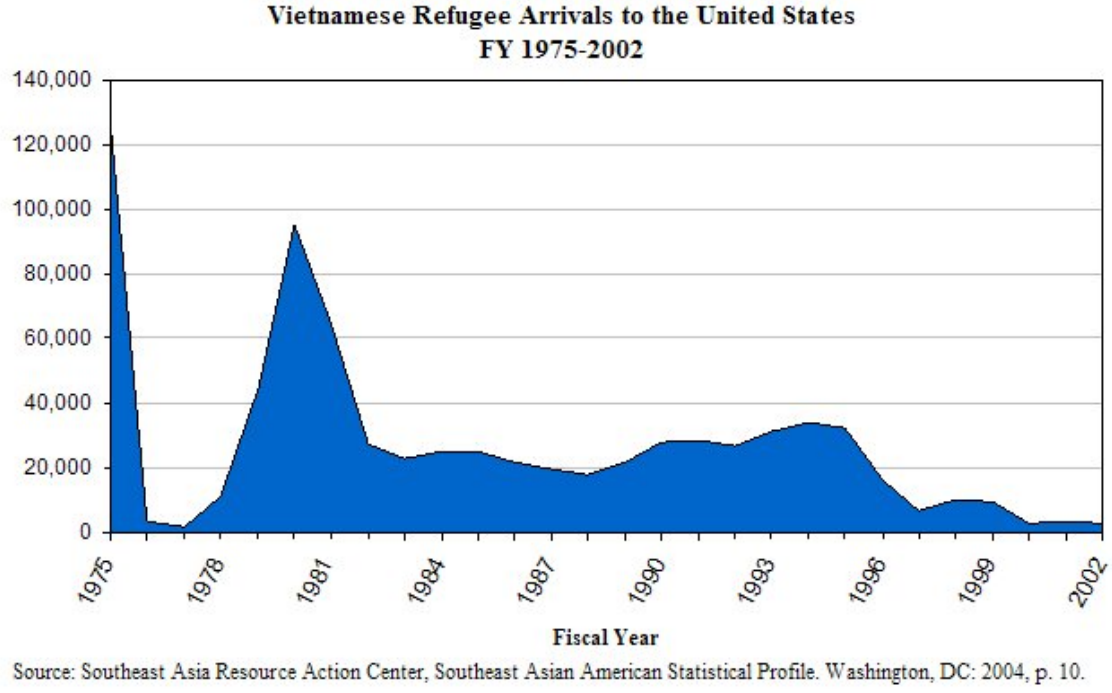
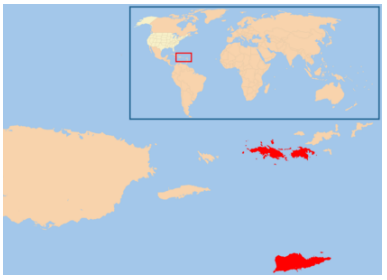


Figure D.1: Vietnamese refugee arrivals to the United States

D.2.92 Virgin Islands (U.S.)



Population(2000) : 109'000
GDP(2000) : Missing value
HDI(2000) : Missing value
Contribution of natural ressources in the GDP :
Missing value
Total emigration (1990→2000) : 91'050
Total immigration (1990→2000) : 56'576
Extra information : Tax haven

D.2.93 West Bank and Gaza - Occupied Palestinian Territory

Population(2000) : 1'130'000
GDP(2000) : 4.113 B$
HDI(2000) : 0.598 (computed from its value in 2011)
Contribution of natural ressources in the GDP : Missing value
Total emigration (1990→2000) : 965'868
Total immigration (1990→2000) : 1'407'615
Extra information : Palestinians and Israel made peace in 1993[1] and set conditions for a future Palestinian state[134]. This induced a large immigration from neighbor countries [53].