# Executive Summary

# Table of Contents

Executive Summary 1

Table of Contents 2

Introduction 3

Problem analysis 4

Narrative 5

Assumptions 6

Model logic 7

Household agent 7

Model setup 7

Verification & Validation 10

Model testing 11

Interpretation of results 12

Conclusions 13

Appendix A: Job graphs 14

Primary and secondary sector jobs 14

Service and non-profit jobs 14

Finance and IT jobs 15

Appendix B: Changing model input values 16

MODEL\_SETUP.nls 17

Households in City 17

MOVE\_CITIES.nls 17

Resistance to move based on Time since moving + Children + Age 17

Moving Distance modifiers for City Attractiveness 17

Attractiveness Jobs Countryside 18

# Introduction

Economic activity is geographically extremely concentrated in cities, and even more in large cities. At present, competition between countries to a large part amounts to competition between their cities. This notion is starting to be embedded in, for instance, Dutch economic policy thinking.

Most countries have a city that is by far the biggest and most dominant city. For instance, London accounts for about 20% of the UK’s GDP and the next biggest city is much smaller. Indeed, it appears that the distribution of city sizes within a country often follows the Zipfian distribution, that is, the size of any city is inversely proportional to its ranking in the list of city ranked by size (Gaujal et al.; 2014). So, competition between countries often boils down to competition between their biggest cities.

How this Zipfian distribution emerges however has not yet been agreed on. There are many ideas on how cities form such a distribution, many of which rely on economic theories. There are of course other ways to approach this phenomenon. The Netherlands Bureau for Economic Policy Analysis (CPB) for example is interested to find out if decisions made at household level can explain the emergence of the Zipf’s law. They wish to understand more on the emergence of the Zipf’s law and how it may be influenced by future policy making or how it will influence future policies. This research has been conducted in order to bring light to these matters. Another interesting phenomenon concerning the Zipf’s law is that for some countries the Zipfian distribution does not hold strictly, i.e. with a power coefficient of one. The Netherlands is one of those countries, with many cities of more equal size than the Zipfian distribution would predict. The CPB wonders if that is a problem or an asset? What can be the reason for this? Their first approach is focused on decisions at household levels, which leads to the following research question posed in this report:

*How do individual decisions at household level influence moving behaviour in cities to cause the emergence of the Zipf's law?*

In order to answer this question the method of Agent-Based Simulation has been proposed since it is especially made to study emergent behaviour over time and allows for decisions rules to be made at an agent’s level. The CPB is also interested to see what this method of research offers and how it might be of help to them in future research.

# Problem formulation

Within the field of statistics a Zipfian distribution can be generated by several stochastic processes. Abby Ostriker (2014) did some research in this field, but the results didn’t provide answers to the questions above.

Two studies (Mansury and Gulyas, 2007, Gaujal et al., 2014) showed that a Zipfian distribution of cities can also be generated with an agent based model. An agent based model tries to explain complex macro patterns, like city size, on the basis of individual decision making. These individuals take account of their preferences and their perceived environment and they possess some heterogeneity in decision making. In comparison with traditional economic methods, the agent based modeling approach is more simulation oriented and less analytical.

Identifying causal mechanisms for city formation is important for policy issues. The studies of and Gulyas, 2007, Gaujal et al., 2014 are based on the forces of agglomeration and disagglomeration, both specified as a function of current city size. Individuals are attracted by large cities, which is a well know phenomenon in urban economic literature, but with growing city size the disadvantages of size increase gradually. Possibly this model can be extended with other relevant causal mechanisms.

# System analysis

# Narrative

# Model logic and Assumptions

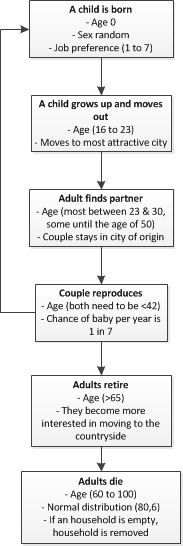
## Household agent

Agents in the ABM model represent households. A household has members, which is coded as a list of people. Every household member has the following attributes:

* Age (0 to 100)
* Job preference (1 to 7)
* Sex (Male / Female)

Next to household members, a household also has a location. This location can be in one of the cities or on a patch in the countryside.

## Model setup

The model starts by choosing locations for the X number of cities the model is run. Then the X number of households are divided over these cities and the countryside. The first city is always placed in the middle of the map, after which the other cities are distributed randomly in a circle with a certain width around this first city. This width is minimally 250 patches and maximally 500 patches. The households are distributed over the cities and the countryside. 40% of the households are placed in the cities whereas 60% of the households are placed in the countryside at the start of the model run.

### Life stage progress

Every member in a household gets older each tick of the model. They progress through different life stages in the model according to the flowchart shown in figure 1.

When a child is born a member is added to the household of 2 or more members. A child is aged 0 and the sex is chosen randomly. At birth they are already given a job preference from 1 to 7:

1. Primary sector jobs (2% of people)
2. Secondary sector jobs (16% of people)
3. Tertiary sector jobs; Services (32% of people)
4. Tertiary sector jobs; Financial (4% of people)
5. Tertiary sector jobs; IT (4% of people)
6. Quartary sector jobs; Non-Profit (34% of people)
7. Jobless (8% of people)

Figure 1: Flowchart showing the life stage progressions

Each tick the child grows a year older. Between the ages of 16 and 23 the child will move out of its parents’ home and move to the city that is most attractive for him/her. This will spawn a new household in the model.

Between the ages of 23 and 30 adults will then find partners that live in the same location. When a couple is found, the households are merged into one household and they stay in the same location. It is possible for older people to be in search for a partner as well (<50 years old), because some people are not able to find a partner before they turn 30.

The couple then has a change to reproduce each year equal to 1 in 7. In order for a couple to reproduce they both have to be younger than 40. When a child is born, this child is added to the household and that child’s life stage progress starts at the beginning of the flowchart.

Adults retire when they are 65 years old. They then become more interested in moving to the countryside, thereby mimicking the moving behaviour of the elderly who move out of the cities. Adults die between the age of 60 and 100. This is coded using a normal distribution with u = 80 and s = 6. When the last member of a household dies, the household is removed from the system.

### Moving

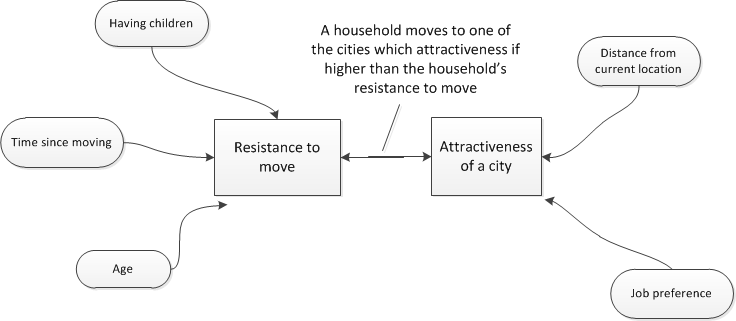
Each household has a resistance to move and a willingness to move which is actually the attractiveness of a city for each household. When the attractiveness of one or a number of cities is higher than the household’s resistance to move, the household will move randomly to one of the cities which attractiveness exceeds their resistance to move. Figure 2 shows the factors that influence the resistance to move and the attractiveness of a city.

Figure 2: Conceptualization of Moving system

#### Resistance to move

As you can see in figure 2, the resistance to move is influenced by three factors:

* Number of children in a household
* Time since moving
* Age of adult household members

The effects of these factors are summarized in a graph which is shown in figure 3. The x-axis show the number of years since the last time the household moved to another location. This resistance to move (y-axis) can then become higher when a household has children and the lowest point of the graph is moved to the right when a household is relatively old, so the elderly do not move as often as young people do. The resistance to move is a value between 0 and 1.

Figure 3: The effect of the number of years since moving on the resistance to move of a household

#### Attractiveness of a city

The attractiveness of a city to a certain household is dependent on the job preferences in that household and the distance from their current location to a city. The attractiveness of a city also takes on values between 0 and 1. We assume that people prefer to stay at their current location when a more attractive city is very close by their current location. We also assume that people do not move to cities that are very far from their current location. This translates into lower (-0.1) city attractiveness for cities that are close by (<100 pacthes) or very far away from a household’s current location (>250 patches).

The attractiveness is also influenced by the household members’ job preference. Each city has a unique attractiveness score for each job based on the amount of people in the city with that job. The way in which this attractiveness is determined differs per job type:

Cities are more attractive to manufacturing and agriculture jobs if they are relatively small cities. The size of cities in this case is used as a proxy for land prices, which greatly influence the profitability for these types of jobs.

Jobs in the service and non-profit domain are not influenced by city size in general. Instead, jobs in these domains are expected to spread evenly over all cities. This means that cities with less service jobs than average is more attractive for service jobs.

There are also two groups of jobs which show a networking effect. Cities are more attractive for IT and finance jobs if more people are working in that domain in the city.

Finally, each city is evenly attractive for jobless people. In Appendix A attractiveness graphs are presented for the different job types.

#### Attractiveness of the countryside

The attractiveness of the countryside is a constant value for the different job types, so that for IT and finance job types the countryside is not very attractive, but for primary and secondary jobs the countryside is very attractive. This results in the following values from job type 1 to 7: 0.55; 0.55; 0.50; 0.45; 0.45; 0.5; 0.5.

# Verification & Validation

# Model testing

# Interpretation of results

# Conclusions

# Appendix A: Job graphs

This appendix shows under which circumstances certain jobs are attractive. Different job types rely on different formulas to calculate the attractiveness of a city for this job type, which results in different ranges in x-axis. Why they differ will be explained for each job type.

## Primary and secondary sector jobs

Job 1 and 2 are primary and secondary sector jobs, which attractiveness for a city is determined by the fraction of households in a city. The graph showing the attractiveness of a city for that job type for different fractions is shown in figure 4. The y-axis is the city attractiveness for that job type and the x-axis is the fraction of households in that city. These jobs are coded in such a way that cities with low density are more attractive for these job types.

The x-axis ranges from 0 to 0.5 because it depicts the fraction of households in city X from the total number of households in the country. In the base case of the model there are 5 cities, therefore if all households were distributed equally, each city would have this fraction be equal to 0.2 and a city with a fraction equal to 0.5 would be an extremely large city. Therefore a range from 0 to 0.5 takes into account all possibilities.

Figure 4: City attractiveness values for job types 1 and 2

## Service and non-profit jobs

Job 3 and 6 are service and non-profit jobs, which attractiveness for a city is determined by the fraction people in service / non-profit. These jobs are coded in such a way that all cities need a certain percentage of people performing these jobs compared to the total number of households in that city. The graphs in which these effects are taken into account is shown in figure 5 and 6 respectively. The x-axis shows the fraction of people in service and the y-axis shows the city attractiveness for that job type.

The x-axis has a range from 0 to 1, which is different from the graph in figure 4, because the fraction used to make this graph is different. Here the fraction is not the number of households in city X compared to the total number of households, but the number of people in service or non-profit jobs compared to the total number of households in that city. When this fraction is 0, the attractiveness for this job should be very high as a percentage of the total households in that city is always required to be filled for this job. When this fraction is equal to 1, the city attractiveness for this job type should be equal to 0 so that no more people take on such jobs than is necessary. Therefore the range of the x-axis is from 0 to 1.

Figure 5: City attractiveness values for job type 3

Figure 6: City attractiveness values for job type 6

## Finance and IT jobs

Finally, job 4 and 5 are finance and IT jobs, which attractiveness grows the more finance and IT jobs are taken in a certain city. This effect is shown in figure 7 where the x-axis presents the fraction of people in finance/IT and the y-axis shows the attractiveness of a city for these job types. The attractiveness is bound at 0.6 when the fraction is high so that these job types don’t cause exponential growth.

The x-axis for this graph shows values from 0 to 0.1. This is because the x-axis represents the fraction of people in finance or IT jobs compared to the number of households in a city. Because the number of IT and finance jobs in total is very low, only 4% of all people, this value in a single city will range from 0 to very high. Because this maximum value is entirely dependent on the type of people that life in a city (more IT/finance people in a city will exponentially bring even more IT and finance people), we have capped the attractiveness at 0.6 as explained before. The x-axis is then capped at 0.1 since the y value is stable anyway.

Figure 7: City attractiveness values for job types 4 and 5

# Appendix B: Changing model input values

There are several input values and graphs that can be changed to test different scenarios. This appendix will summarize all the values that can be changed.

*BASE INPUT VALUES*

* Number of cities
* Number of households
* Runtime
* Warmup time
* Percentage initial households in and out of cities
* Minimum distance between cities
* Maximum distance between cities

*JOB GRAPHS (all job graphs can be altered)*

All job graphs can be altered, including the attractiveness of the jobless category which is the same for all cities.

*RESISTANCE TO MOVE*

The effect of ‘time since moving’, the number of children in a household and the age of the adult members in a household can be changed.

*ATTRACTIVENESS JOBS COUNTRYSIDE*

The attractiveness of all the job categories in the countryside can be changed.

*MOVING DISTANCE*

The range in which cities are close to the current location of a household and when they are too far away can be changed. The effect of these close by or far away cities on the city attractiveness can also be changed.

# MODEL\_SETUP.nls

## Households in City

*“to setup-households*

*let noOfHouseholdsInCity round (PercHouseholdsInitialInCity \* noOfHouseholds)*

*let noOfHouseholdsOutCity round ((1 - PercHouseholdsInitialInCity) \* noOfHouseholds)”*

This piece of code divides all the households between households in cities and out of cities. The percentage of households that are in cities can now be changed using a slider. This percentage is names “PercHouseholdsInitialInCity”. Initially the value was 0.4 (40%). The slider is can be changed from 0.0 to 1.0.

# MOVE\_CITIES.nls

## Resistance to move based on Time since moving + Children + Age

See Excel sheet named: RTM\_TimeSinceMoving

## Moving Distance modifiers for City Attractiveness

Here we alter the attractiveness of the city based upon the distance from the current location to that city. For cities very closeby (less than 100 patches) people will move less likely, since it is easily commuted daily. For cities very far away (more than 250 patches) people will move less likely since it is too far (from family etc.)

*“let minMovingDistance MinimalMovingDistance*

*let maxMovingDistance MaximumMovingDistance”*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Default** | **Min** | **Increment** | **Max** |
| **MinimalMovingDistance** | 100 | 0 | 10 | 200 |
| **MaximumMovingDistance** | 250 | 200 | 10 | 400 |

*“if distToCity < minMovingDistance [*

*set cityAverage cityAverage - MinDistCityAttractiveness*

*]*

*if distToCity > maxMovingDistance[*

*set cityAverage cityAverage - MaxDistCityAttractiveness*

*]”*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Default** | **Min** | **Increment** | **Max** |
| **MinDistCityAttractiveness** | 0.1 | 0 | 0.05 | 0.3 |
| **MaxDistCityAttractiveness** | 0.1 | 0 | 0.05 | 0.3 |

## Attractiveness Jobs Countryside

*“set jobAttractivenessList lput (list Job1Attractiveness Job2Attractiveness Job3Attractiveness Job4Attractiveness Job5Attractiveness Job6Attractiveness Job7Attractiveness) jobAttractivenessList”*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Default** | **Min** | **Increment** | **Max** |
| Job1Attractiveness | 0.55 | 0.4 | 0.05 | 0.6 |
| Job2Attractiveness | 0.55 | 0.4 | 0.05 | 0.6 |
| Job3Attractiveness | 0.50 | 0.4 | 0.05 | 0.6 |
| Job4Attractiveness | 0.45 | 0.4 | 0.05 | 0.6 |
| Job5Attractiveness | 0.45 | 0.4 | 0.05 | 0.6 |
| Job6Attractiveness | 0.5 | 0.4 | 0.05 | 0.6 |
| Job7Attractiveness | 0.5 | 0.4 | 0.05 | 0.6 |