Commuting, a choice or a necessity?
A system dynamic simulation model for intra urban migration

By

Tianhang Li

Supervisor David Wheat

Submitted in Partial Fulfillment of the Requirements for the Degree
of Master of Philosophy in System Dynamics

System Dynamics Group
Geography Department
UNIVERSITY OF BERGEN
BERGEN
NORWAY

Table of Contents

List of Figures & Tables

Acknowledgements

I would like to thank my supervisor David Wheat. He introduced this interesting modeling project to me. In the meeting with him, I was guided into new directions of modeling. He also helped me with the writing and communication techniques. Without his help, this thesis would not be carried out as now.

I would also like to thank Professors Pål Davidsen and Erling Moxnes for their lectures in my first year as System Dynamic student. Also thanks for their advices in my midterm presentation.

Thanks to Carolin and Torgeir, I really enjoy the group work with you two. I will remember the happy time of meetings. Also thanks to Erlend, your feedbacks for my model and writing are really useful.

Abstract

The number of commuters is one of the main determinants of traffic and pollution problems. In a growing city like Bergen, the number of commuters has been increasing dramatically. A simulation model has been built to explain the system structure behind commuter problem. This model includes intra-urban migration and job change process.

The structure of this model explains how the distribution of dwelling vacancies and job vacancies influence the number of commuters. It also shows that the first time buyers are the reason why land-use policies have little effect on reducing existing commuters.

This system dynamics model can be used as a tool for policy makers to understand the dynamic problem of commuter change. The simulation and projection function can also be used for policy testing when this model is joined together with a land-use model and a transportation model.

Keywords: intra-urban migration, job change, commuter, residential mobility, residential location choice, search matching.

1. Introduction

1.1. Commuters problem in Bergen

The traffic volume in Bergen has been increasing in the last 10 years (Vegvesen 2011). Moreover, the population projection express a 19% annual increase rate in Bergen population till 2025. With this increase, it is believed that the road system in Bergen will face serious challenges (Vegvesen 2007) .

There are both monetary and social consequences of high traffic volume. The increasing traffic volume might cause problem for road capacity and public transportation capacity. The emission of CO_2 by traffic is also been seen as a main cause of pollution in Bergen. It might also lead to traffic safety problem for Bergen. Solving the traffic volume by increasing road capacity also lead to huge monetary cost and occupying big land area. The solution might be hard to carry out. (Vegvesen 2007, 2011)

One main reason for traffic volume is commuting trips (Strand, Christiansen, and Engebretsen 2010, Meland 2009). And, the main concern of traffic volume and road capacity is mainly on highway system (Vegvesen 2007). So, we assume that commuters using highway system to commute is the cause of traffic volume.

Population increase might be considered to be the reason for commuter increase. While the number of commuters might be influenced by total population, the increase of population is not the direct reason of commuter change. If the new residents all live where they work, the number of commuters will not increase by population increase. The direct reason of commuter increase is new residents or non-commuters choose to live far away from their work place or choose to work far away from their resident place. So, it is the resident change and job change lead to commuter change.

1.2. Solving problem by transportation land-use model

To investigate this problem, a transportation land use model project has been conducted for Bergen municipality. This transportation land-using model using system dynamics approach comprises three master theses. The transportation model is made by Brandsar (2013). And the land-using model is made by Schulze (2013). This thesis models the migration part of this project. The model in this thesis combines residential mobility, residential location choice and search-matching theory to simulate the increase of commuter in Bergen. The basic model structure follows the questions of who is moving, where they want to move and where they can finally settle down.

With the simulation of commuter change in Bergen, this model can be used in policy evaluation for Bergen municipality.

1.3. Research question

The purpose of this thesis is to build a simulation model and discover the underlying structure causing the increase of commuting. It is also the goal of this thesis that an effective policy can be found to reduce the number of commuters.

So, the research questions for this thesis are list below:

1. Can land-use actions (job creation, dwelling construction) influence the number of commuters?
2. How can land-using actions reduce the number of commuters?

After answering the research questions, the contribution of this thesis can be policy suggestions for land-use. The land-use policy rules can also be tested by this thesis.

2. Literature review

This chapter is going to survey the literature which is relevant to this paper. Modeling for the number of commutes is always carried out as part of transport land-use model. The first part of this chapter is going to survey previous transport land-use model. The second part of this chapter is going to survey literature on intra-urban migration which consists of three parts, residential mobility residential location choice and search-matching theory.

2.1. Commuter in transport land-use model

Transport land-use models are widely used for policy making and academic research. Several frameworks of models have been reviewed by Hunt, Kriger, and Miller (2005). Since this thesis in only a migration model, we here only focus on the migration part in transport land-use model.

There are many frameworks that can be used for transport land-use model. Urbansim is one of the simulation model focusing on residential location and housing market (Waddell 2000). It use housing price to forecast the possibility for households` movement. LocSim is another simulation model (Hooimeijer and Oskamp 1996). Using event-driven micro simulation model, it simulated residential mobility rate and location choice. LocSim is used for local housing market research. And, a decision making model without empirical data test was made by Smith et al. (1979).

Through these frameworks use different techniques, their purposes are same. The main job for migration part in transport land-use model is to model who want to move and where they want to move. The inputs for model are dwelling vacancies and accessibility. The output of model is commuters.

So, this thesis is going to model who is going to move and who is going to take the job and dwelling vacancies.

2.2. Residential mobility

There is enormous research on residential mobility. Some reviews have been done by former researchers (Dieleman 2001, Li and Tu 2011, Mulder 1996).

The theoretical analysis starts from Brown and Rossi (Brown and Moore 1970, Rossi 1980(1955)). They stated that the reason of movement is the dissatisfaction of housing. With the change of life cycle, households` demands changed. Former dwelling can no longer serve their demand, therefore households need new dwellings. Resent researches combine households` demand with momentary facts. For example, Case stated the investment facts in housing choice (Case, Quigley, and Shiller 2001). Krainer studied the impact of transition cost (Krainer 2001). Kiel studied the impact of housing price (Kiel 1994). Speare's research shows that the change of household or environment triggers the dissatisfy for dwelling and lead to people's relocation (Speare 1974).

We can find that there are enormous facts influencing residential mobility. But a simulation model cannot include all these facts. So, a boundary should be set to limit the number of facts. Former modeling research on residential mobility divided household into different categories like life cycle (Speare 1970), space in dwellings (Clark, Deurloo, and Dieleman 1984). So, this thesis assumes that the only reason for household moving is a life cycle change which might change the space in dwellings.

2.3. Residential location choice

Mulder (1996) reviewed many residential relocation literatures and group three approaches to triggers and subsequent housing choice. Pagliara and Wilson (2010) also presented some running models with different theories.

Research shows that residential location choice might be very complex. There are many facts that influencing the location choice. Accessibility, infrastructure and social environment are most common facts (Mulder 1996).Other than these normal facts Phe also stated a new theory about the location choice, that is quality and other characters of the dwelling is more important for location choice (Huu Phe and Wakely 2000). Adams` research showed that location choice is also influence by seekers` "mental map" (Adams 1969).

Among the common facts, infrastructure is excluded by this thesis because we assume that infrastructure level is equal inside Bergen municipality. Social environment is also excluded because this fact only affect a small spatial area(Guo and Bhat 2007) while thesis have a relatively bigger area. Other facts like quality and "mental map" are also excluded because no such research is conducted in Bergen or Norway. So, accessibility is the only fact for residential location choice.

Six techniques for residential location choice are introduced by Clark, Van Lierop, and Peter (1987). Due to that accessibility is the only fact. Random utility model is chosen for this thesis.

2.4. Search-matching theory

Search theory is widely used by job change research (Rogerson and MacKinnon 1981, van Ommeren, Rietveld, and Nijkamp 1997, Rouwendal 1998, 1999, Van den Berg and Gorter 1997). The basic logic for search theory is that if a vacancy matches seeker's demand, this vacancy will be taken by the seeker. In this thesis, the seeker's demand is location. If there is a job or dwelling vacancy and there is a seeker seeking vacancy in this area, the vacancy will be taken.

3. The dynamic problem

3.1. Number of commuters

The number of commuters can be seen below (Figure3.1 1). It can be calculated that there are more than seventy thousand workers (65% of total workers) working and live in different urban districts. The total number of commuters is the first research problem in this thesis.

		Working place							
		Bergenhus	åstad	fana	ytebygda	åsane	fyllingsdalen	laksevåg	arna
	Bergenhus	11427	1564	954	1283	1053	787	751	149
	åstad	7788	3172	1443	1420	650	722	670	165
	fana	5157	1851	4708	2571	412	777	526	246
Residence	ytebygda	2914	978	1560	4190	249	730	411	91
	åsane	6213	1409	616	987	6478	654	701	443
	fyllingsdalen	4642	1321	919	1521	390	2957	887	86
	laksevåg	5584	1806	805	1602	606	1108	3965	112
	arna	1230	376	442	308	619	146	134	2025

Source from (Strand, Christiansen, and Engebretsen 2010).

Figure3.1-1: number of commuters in year 2008

3.2. Change of commuters

Other than the number of commuters, the increasing trend of commuters is also a problem. The table below (Figure3.2 1) shows the change of traffic between urban districts from 2002 to 2008. It can be found that the numbers of commuters in most places in Bergen are increasing.

		To							
		Bergenhus	åstad	fana	ytrebygda	åsane	fyllingsdalen	lakesvåg	arna
	Bergenhus	25%	13%	23%	13%	8%	-20%	19%	34%
	åstad	11%	8%	11%	11%	38%	-11%	18%	-46%
	fana	16%	11%	33%	40%	54%	20%	4%	-18%
From	yrebygda	5%	-1%	40%	29%	25%	4%	34%	36%
	åsane	13%	11%	60%	-5%	24%	11%	54%	21%
	fyllingsdalen	-18%	1%	10%	-9%	5%	10%	3%	84%
	laksevåg	23%	15%	0%	46%	85%	-3%	33%	41%
	arna	13%	-61%	-4%	36%	28%	41%	111%	26%

Figure3.2-1: change of traffic, source from (Meland 2009)

3.3. Reference mode

In this thesis, we focus more on the total number of commuters than the distribution of commuters. It is the total number of commuters that cause the total traffic volume and pollution. The total number of commuter is shown below. There is about 8.5% increase of commute in these 6 years. Due to the lack of data, only two points are known.

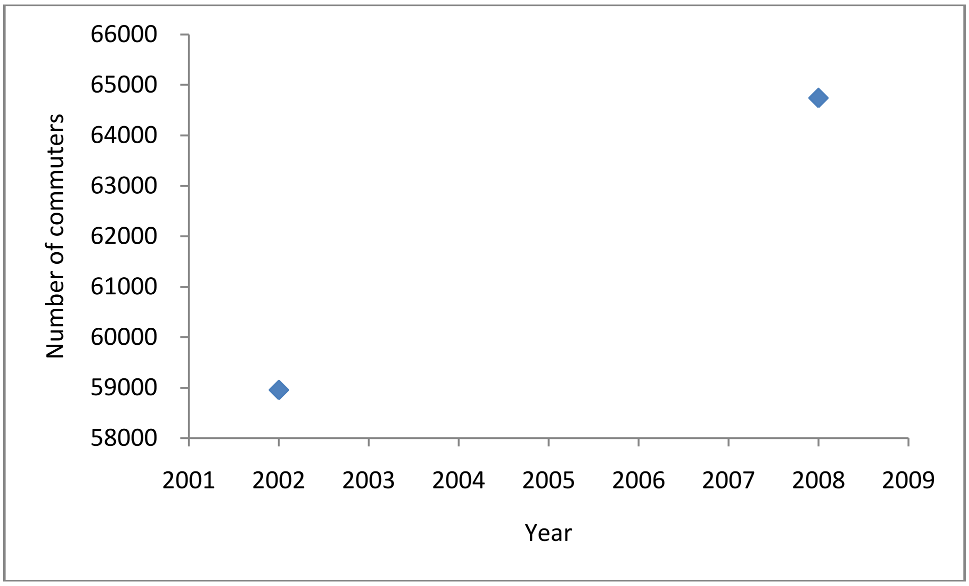


Figure3.3-1: reference mode, calculated by author.

4. Hypothesis

The system dynamics model described in this chapter offers one explanation for the increasing number of commuter and the influencing of land-use policy on the increase. The description will begin with some basic assumptions and definitions. Some cause loop diagrams are carried out secondly to illustrate the feedbacks within the boundary of intra-urban migration for commuters. Then the stock and flow structure of this model is explained in the next.

4.1. Definitions and basic assumptions

4.1.1. Spatial aggregated level

The purpose of this model is to model the commuters inside the Bergen municipality. For reasons of data acquisition and comparison, Askøy municipality is also included.

The whole area of Bergen municipality and Askøy municipality is then divided into six areas, Askøy, Bergen center, Bergen south, Bergen north, Bergen east and Bergen west. The table below (Figure4.1.1 1) shows the spatial division for these areas.

Askøy	Askøy municipality	A
Bergen center	Bergenhus and Åstad urban districts (bydeler) [1]	C
Bergen south	Ytrebygda and Fana urban districts (bydeler) [1]	S
Bergen north	Åsane urban district (bydel) [1]	N
Bergen east	Arna urban district (bydel) [1]	E
Bergen west	Laksevåg and Fyllingsdalen urban districts (bydeler) [1]	W

Figure4.1.1-1: spatial aggregated level.

In order to avoid confusion, we will use "area" for the six defined places from here on.

4.1.2. Job and workers

For simplification, unemployment is not considered in this model. This means that all residents in this model will be attributed to a job and work place. Even unemployed people will be given to a work place. The distribution of their working place follows the same distribution of employed people.

This assumption might cause some bias. In this assumption, homemakers and joblessness will have the same commuting behavior as workers. An average ratio for whole Bergen area is used to calculate the different between commuters in model and commuters in reality. If an area has more homemakers or joblessness than average, this model will generate more commuters than reality. If an area has less homemakers or joblessness than average, this model will generate fewer commuters than reality.

University and college students are not modeled separately. Their everyday trips to school are considered the same as workers' commuting trips. In that case, all people will try to find job from age 20.

4.1.3. Household and dwelling unit

We assume that there is only one household in one dwelling unit. In reality, there might be more than one household living in one dwelling unit. So, the total number of dwelling units is lower than total number of households. In this case, we multiply the dwelling unit with a ratio to fit household numbers.

And we also assume that one household only occupy one dwelling unit. In reality, one household might also have multiple dwellings. So, Proprietary rights and tenures are not considered in this model. And if one household only possess one place but does not live there or only live there during weekend or holiday, that dwelling unit is treated as vacancy.

These two assumptions might also lead to some bias if the distribution of multiply household dwellings and unused dwellings is not even.

An adult who is living with his/her parents is not treated as a separated household in this model. Those adults are treated the same as juveniles in this model. And we assume that once those adults got a job, they will leave their parents' home and live alone. In that way, they become new households.

We also divide households into two categories, single and couple. Single household are those people who live alone or single father/mother live with their children. Couple means two adults in one family (single parents with adult child are not treated as couples here).

4.2. Simulation model

4.2.1. Model structure

As mentioned above, this model is going to model the number of commuters. When the number of commuters in a certain year (year 2002) is given, the task for making a simulation model is to estimate the change of commuters every year. After accumulating the change, the number of commuters can be know in the following years.

The change of commuters is due to the change of residential place or working place. Therefore, this model contains two parts, residential move and job change.

The framework for this study of intra-urban migration is adapted from Brown's two stage choice approach (Brown and Moore 1970). The process of intra-urban migration is divided into two stage three phases (Figure4.2 1): Phase I, the decision to move and begin seeking for new residence place (residential mobility); Phase II, the decision to choose an optimal place (location choice). Phase III, dwelling seekers settle down and become residential population again.

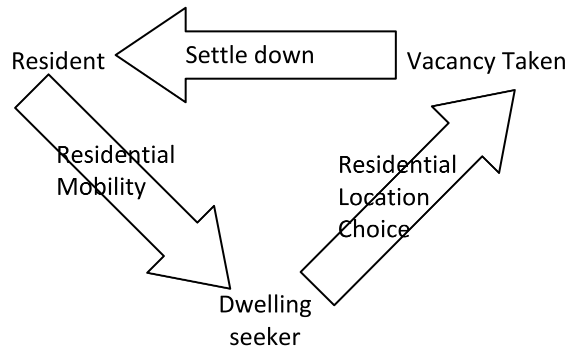


Figure4.2.1-1: Process map for two stage choice approach.

Job change model have the same structure as residential change. (Figure4.2 2) Workers might want to change job and become job seeker. And once a new job offer is accepted, the work place of this people changes.

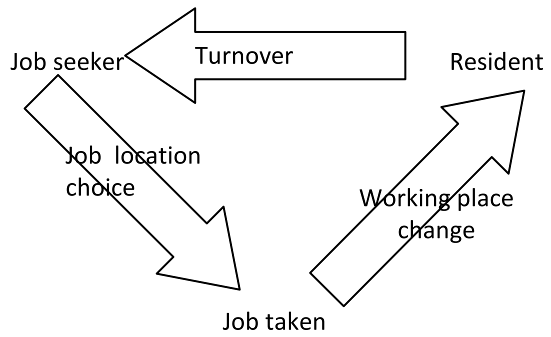


Figure4.2.1-2: Process map for job change.

4.3. Causal loop diagram

4.3.1. Residential mobility

Ever since an early classic research by Rossi (1980(1955)), intra-urban migration is considered to be a decision made by family rather than one individual. A family tends to move and live together if the destination and origin local in same City. So households are agents in this intra-urban migration model.

Brown and Moore (1970) defined a term of stress for intra-urban migration. Stress is generated by disparity between household's need and housing service. Households respond stress by adjusting their need, restructuring the housing service to satisfy their need or relocating themselves to seek alternatives. Households who adjust their need or restructuring the housing service are essentially choosing to stay in position. Other households who want to relocate themselves will seek a new residence.

To model the number of people who tend to move, we first should know who have the stress to move. They are those people whose current dwelling cannot fit their need. Mulder (1996)argued that some people tend to move not because of the problematic situation but simply because the desire of improving living condition. That indicated that individual households have individual need. There is hardly any standard need. Even a constant average need standard does exist. It is no possible for this thesis to use because the lack of data.

So rather than model residential mobility as a fraction, we consider the movement to be trigged by events. Mulder and Wagner (1993) stated that changing state of life course is one important motive for people to move. We assume that dwelling should fit household's need if household's need doesn't change. The dwelling only no fit the need when household's need has changed. And reason for changing of household's need is the change of life course. The relationship in diagram can be seen below (Figure4.3.1 1). In the figure, we use commuter who live in Bergen north and work in Bergen centre as example. We use N represent Bergen north and C represent Bergen centre.

Other facts that might influence the residential mobility are not considered in this model.

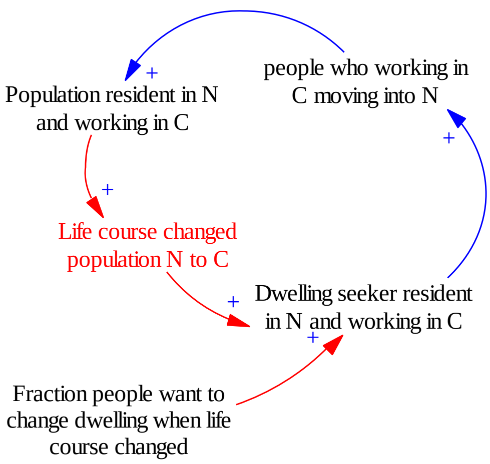


Figure4.3.1-1: CLD for residential mobility.

4.3.2. Location choice

A logit model is widely used in location choice model (Pagliara and Wilson 2010). In this project, logit model is partly adapted from Pagliara'sresearch (Pagliara et al. 2010). In this model, two facts are influencing household's residential location choice, accessibility to work and accessibility to city centre. This phenomena is studied by Hjorthol (2003) in Norway. Research shows that sub center develops after population exceeds 2.5 million (McMillen and Smith 2003). So we can assume that Bergen centre is the only center in Bergen. So accessibility to City center might be influencing all people`s decision.

So the travel time to work place and travel time to city centre influence household's residential location choice in this model. People will tend to choose a dwelling close to their work and city centre. The relationship in diagram can be seen below (Figure4.3.2 1).

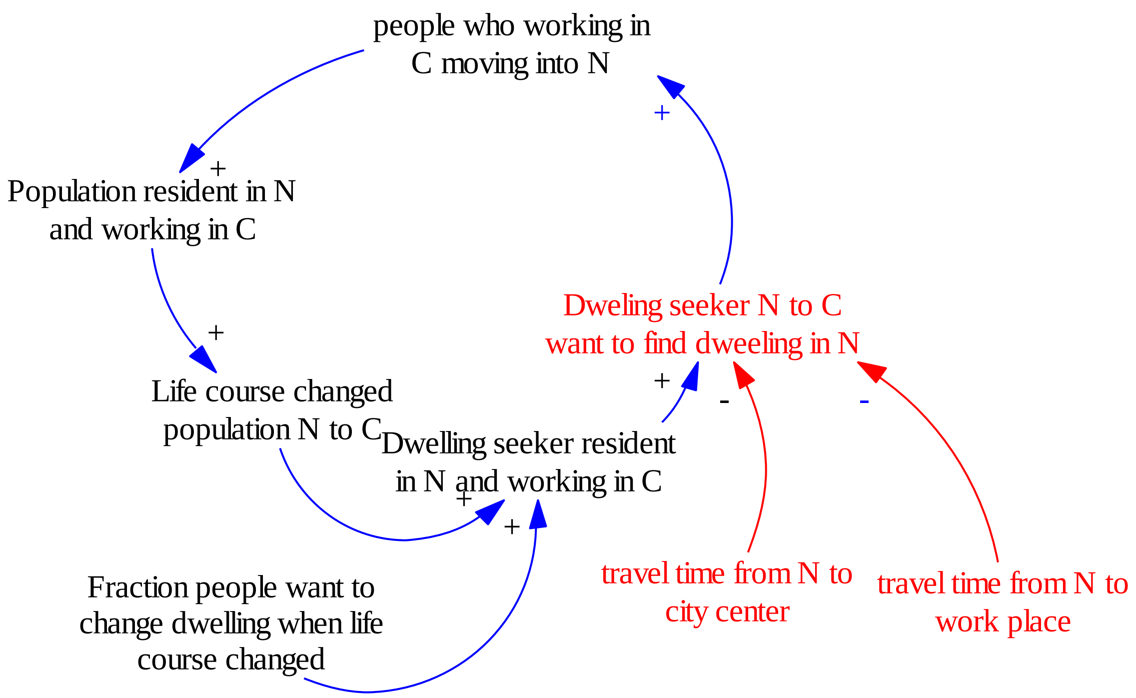


Figure4.3.2-1: CLD for Location choice

4.3.3. Search matching for dwelling vacancy

The location choice for dwelling seeker is seekers' choice. The choices conduct search actions in that area. And if dwelling vacancies match search action, the vacancy will be taken. On the other word, a movement of residence happens only when there are dwelling vacancies and seekers in the area in the same time. If there are more seekers than dwelling vacancies, some seekers cannot find vacancies. If there are more dwelling vacancies than seekers, some vacancies will not be occupied. And people who work or not in C all might want to live in N. They might search for vacancies in N together. The relationships in diagram can be seen below.

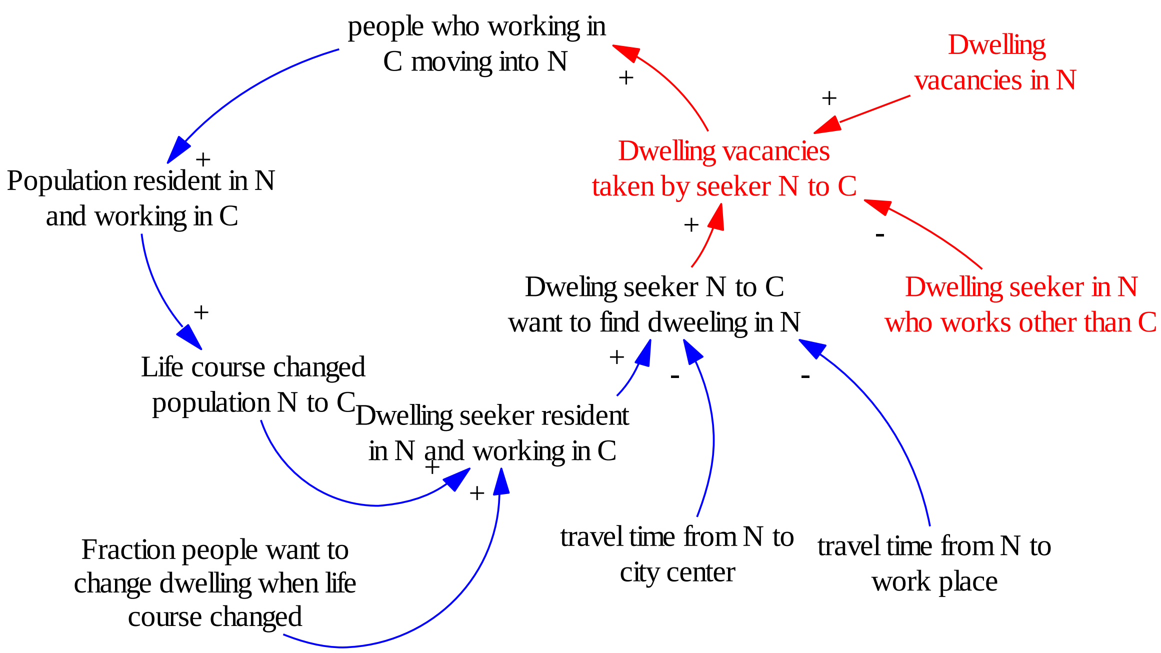


Figure4.3.3-1: CLD for Search matching of dwelling vacancy.

4.3.4. Balance of population and dwellings

When people finish a migration action, they occupied dwelling vacancy and become residence in a new area. In the mean time, they leave their previous dwelling and are no longer residents in the previous area.

There are three links created by these actions. First, the action of migration reduces the population in the area of emigration. Second, the vacancies which have been occupied are no longer vacancy anymore. Third, people emigrate and leave their former dwelling vacant which increases the dwelling vacancy in that area. Therefore, three new links are added into the cause loop diagram above. The result can be seen below.

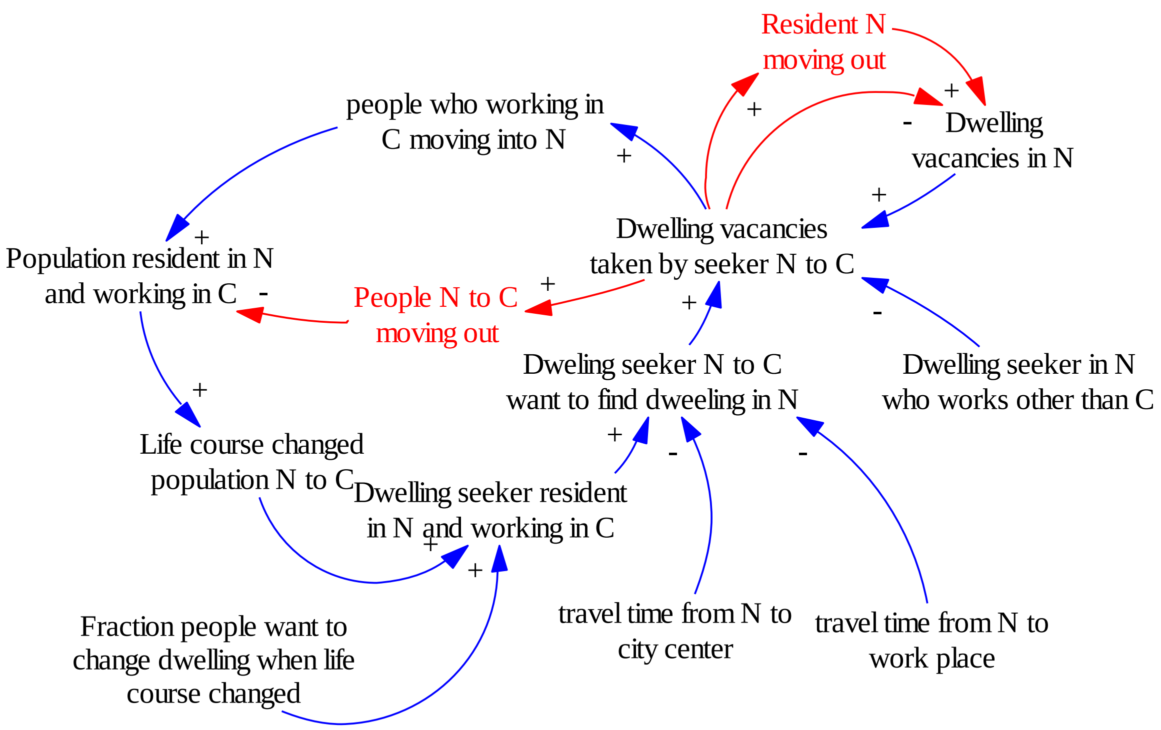


***Figure4.3.4-1: CLD for* Balance of population and dwellings.**

4.3.5. On job seeker and location choice

The job change process is similar to residence change process. Some workers will tend to find new jobs after working in the same firm for few years. Most Job seekers might want to find a job close to their residence place (Nebiyou and David 2010). So the travel time from residence to work place (N to C) influences people's job location choice.

As shown below, turnover rate and population determine the number of people wanting to find new job. And travel time determines the fraction of people want to find job in for example Bergen center. The total number of seeker and fraction want to find job in Bergen center determined the number of on job seeker who live in for example Bergen north and still want to work in Bergen center).

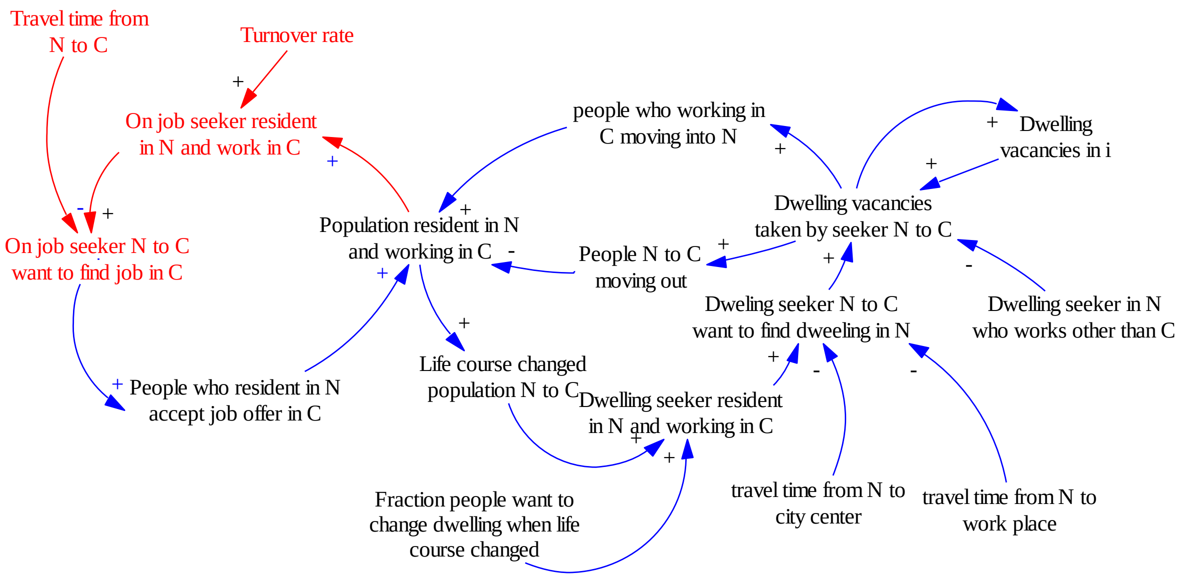


Figure4.3.5-1: CLD for on job seeker.

4.3.6. Search matching for job vacancy and balance of labor and vacancy

Job seekers search for jobs in an area. If there is a job vacancy available, the job vacancy can be taken.

Four links are explained here. First, once seeker accept job, he become commuter again. Second, accepting a new job also means quitting former job. So the number of commuter will decrease. Third, accepting a new job means the vacancy is taken. This will decrease the job vacancies. Fourth, quitting former job make the position available which increase the job vacancies.

These relations can be seen below (Figure4.3.6 1).

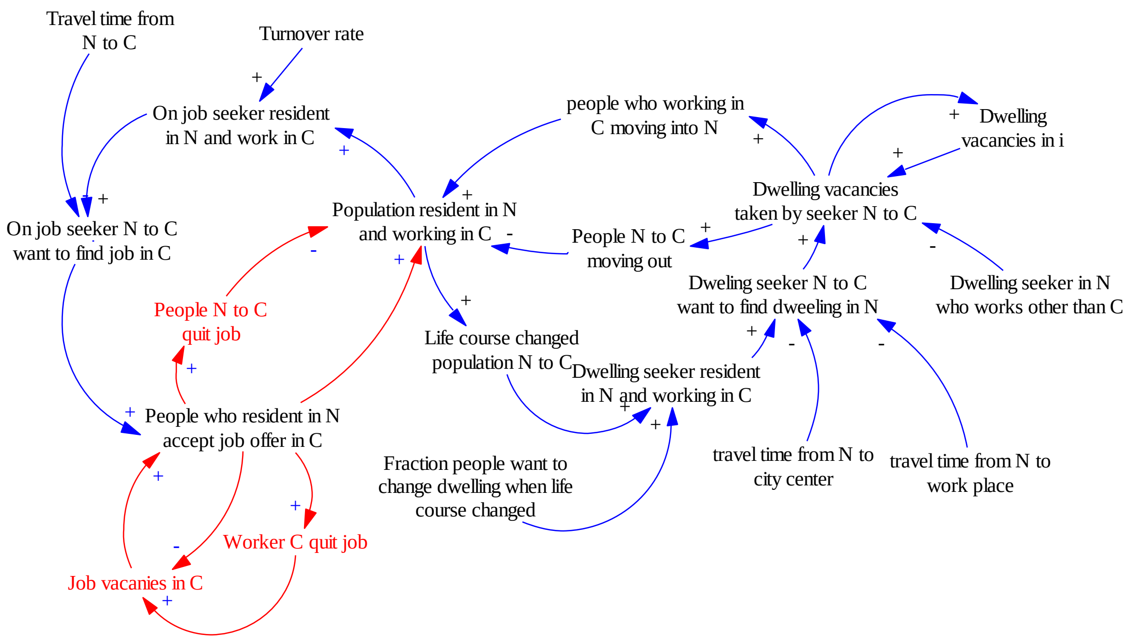


Figure4.3.6-1: CLD for job change

4.3.7. Land use

As a growing city, there are new jobs and new constructed dwellings to support the population growth. Dwelling construction will increase dwelling vacancies. And new jobs are created to increase the job vacancies. The result can be seen below.

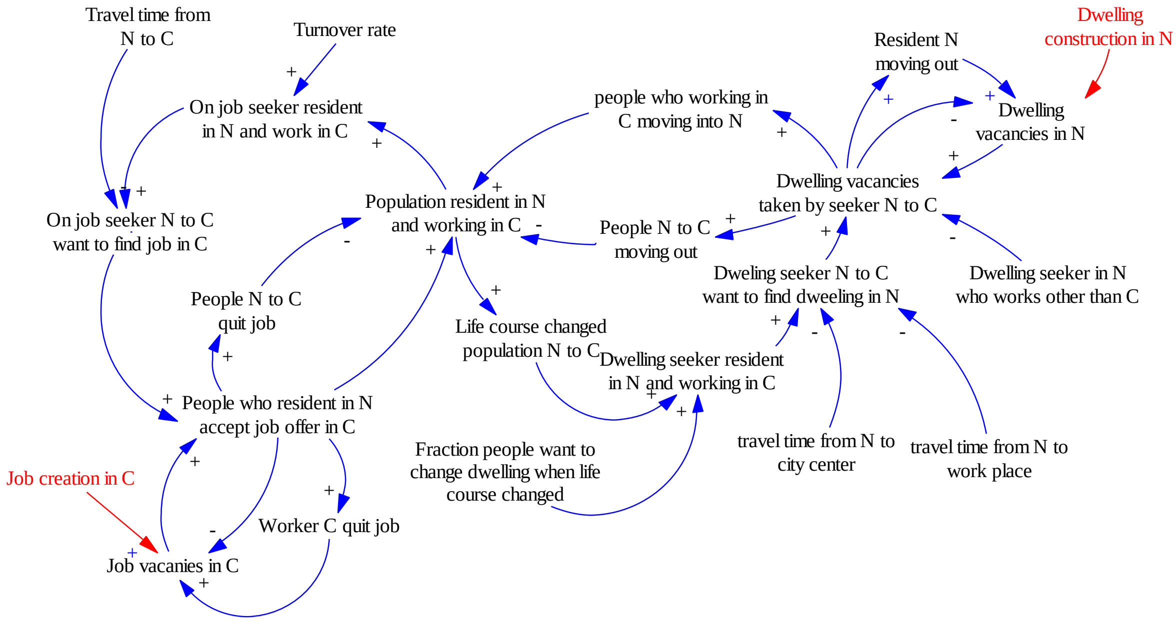


Figure4.3.7-1: CLD for resident change with land use.

4.3.8. Inter regional migration and job creation

The system above is the population change within urban area. But population does not only redistribute within city. People do not only move from one part of the city to another part. People also move from one region to another region. The total population is changed by inter regional migration and nature increase. Emigration and retired people (they do not have work anymore) will reduce the commutes. And immigrants and new young labors will increase the population.

In this thesis, we assume that the reason for migration can be seen as job search which have been tested before (Jackman and Savouri 1992, Rogerson and MacKinnon 1981, David, Inge, and ersa10p 2011). The process of immigration can be seen below.

The labor in one region is the working population. When workers move out from the region (emigrate), they no longer work in the region, this will decrease the labor in region. Retirement and death of workers also decrease the existing labor.

Meanwhile, with the growth of the economy, the total jobs in regions like Bergen will increase. And there might be a gap between job and labor. Which means the job needs laborers. The jobs will be first filled with new graduated student. And then, it will be filled with immigrants. In this thesis, we assume that working age population is fully employed. So there is no unemployed person.

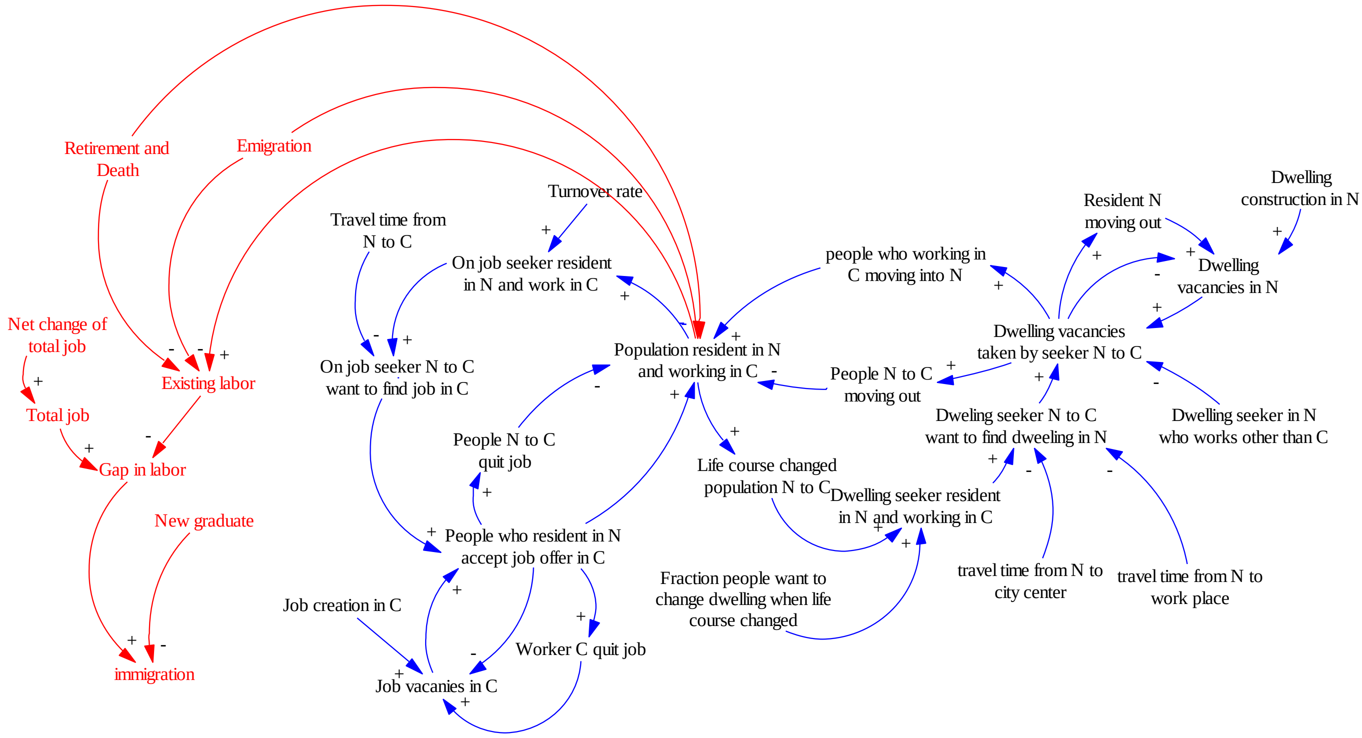


Figure4.3.8-1: CLD for immigration

4.3.9. Job and dwelling search for first time seekers

Immigrants and new graduated students have no former work place and resident place. Even new graduate student might be local and live previously in their parents` home. The locations of their parents` home have no influence on their work place preference or dwelling preference. Immigrants and new graduate students will seek job first (van Ommeren, Rietveld, and Nijkamp 2000). Whether they can get a job depends on whether job vacancies are available for them. And after first time seekers get jobs, they will search dwelling vacancies. Once they get a dwelling vacancy and move in, they become new residents. The relationship can be seen below.

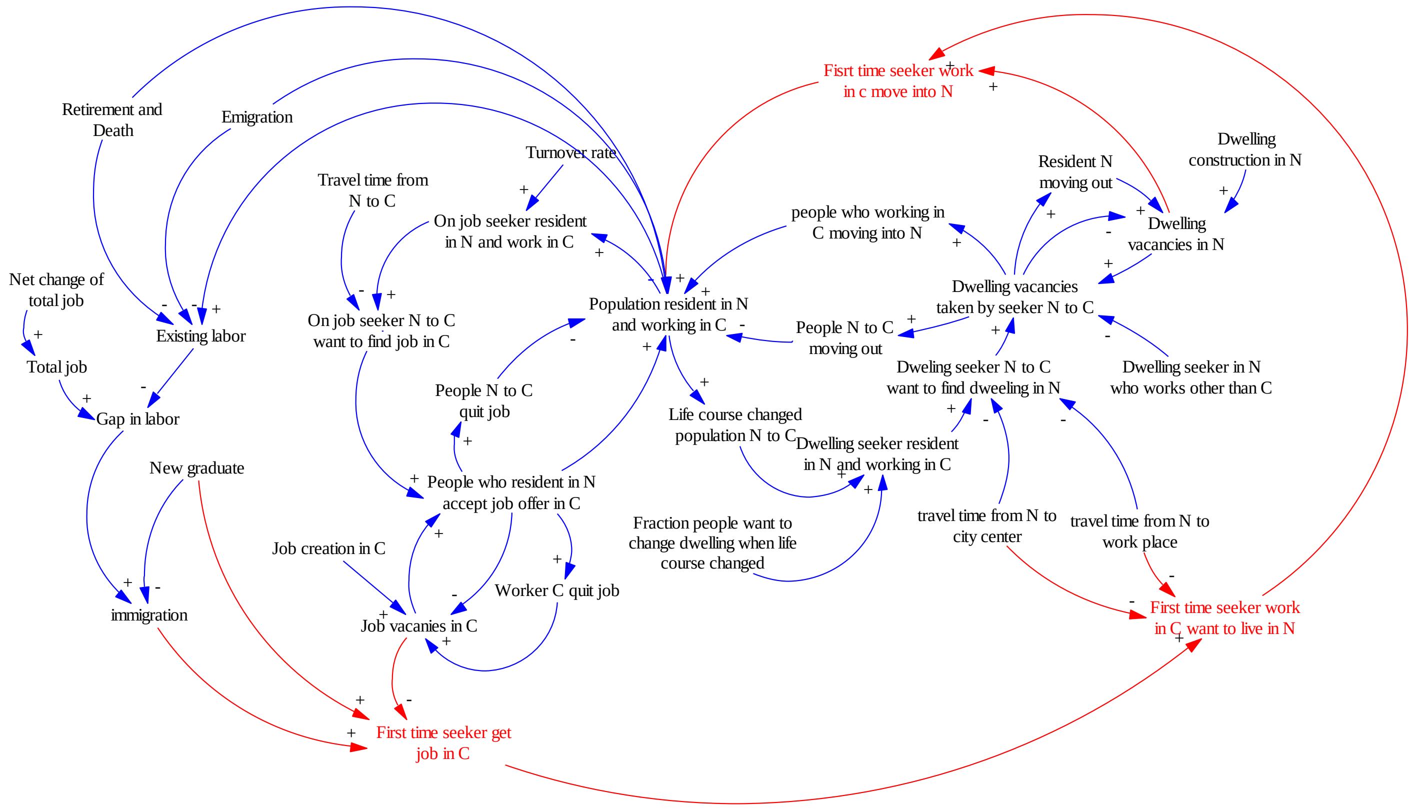


Figure4.3.9-1: CLD for immigration

4.3.10. Array function for this model

In this model, every person has two attributes, resident place and working place. So, the population is divided 6*6 matrixes. For demonstration, Figure4.3.10 1 shows a model structure map with 2*2 matrixes (C to C, C to N, N to C, N to N). This figure is used here to demonstrate the function of matrixes. There are four kinds of commuters (C to C, C to N, N to C, N to N). All of them might become dwelling seekers. So, commuter C to C and N to C become dwelling seeker working in C. And commuter C to N and N to N become dwelling seeker working in N. Then, for dwelling seeker working in C, some of them might want to live in C, and some of them might want to live in N. The same for dwelling seeker working in N, some of them take vacancy in C and some take vacancy in N. At last, if a vacancy in C is taken by worker in C, this worker becomes commuter C to C. If a vacancy in N is taken by worker in C, this worker becomes commuter C to N.

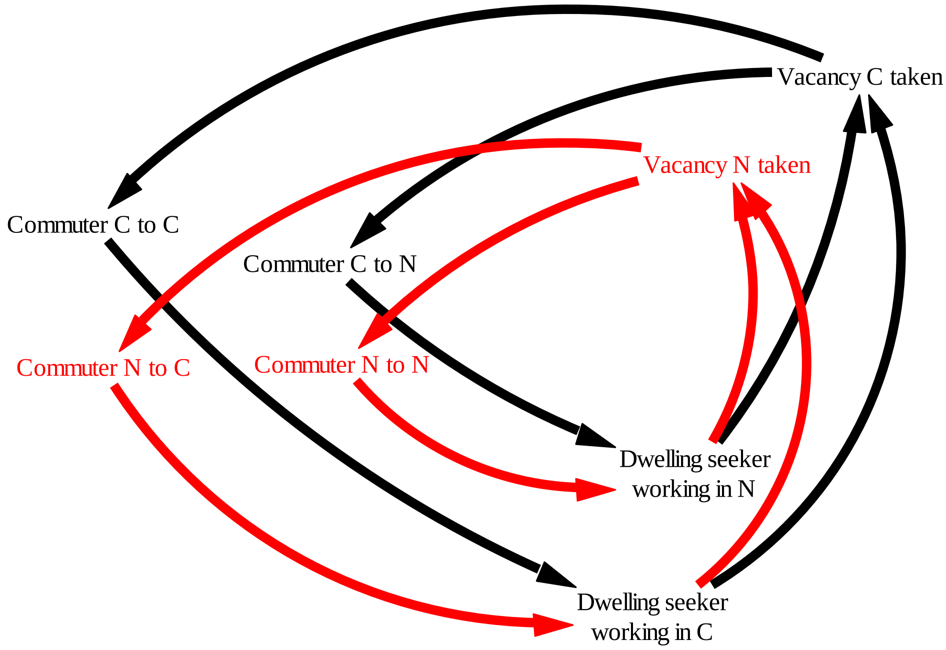


Figure4.3.10-1: model structure for matrix.

This is the process of migration change in matrix function. Through this function, the number of commuters is tracked by the process of migration.

So, all the variables in CLD can be variables for any area in these six areas. So, the job and dwelling vacancies in Bergen center can be taken by seeker coming from all six areas. It means that the dwellings in Beregen center can be taken by people working in Bergen center, it can also be taken by a people working in Askøy. Therefore, it is hard to conclude any result from this simple CLD. A simulation model is then been introduced. With the simulation model, the dynamic of interaction between areas can be learned. And policy results can be tested by this simulation model.

4.4. The system dynamic model

4.4.1. Population household and commuter

The first part of this model is used to link household number with population (see Figure4.4.1 1). The red part of the small diagraph shows the sector of this structure.

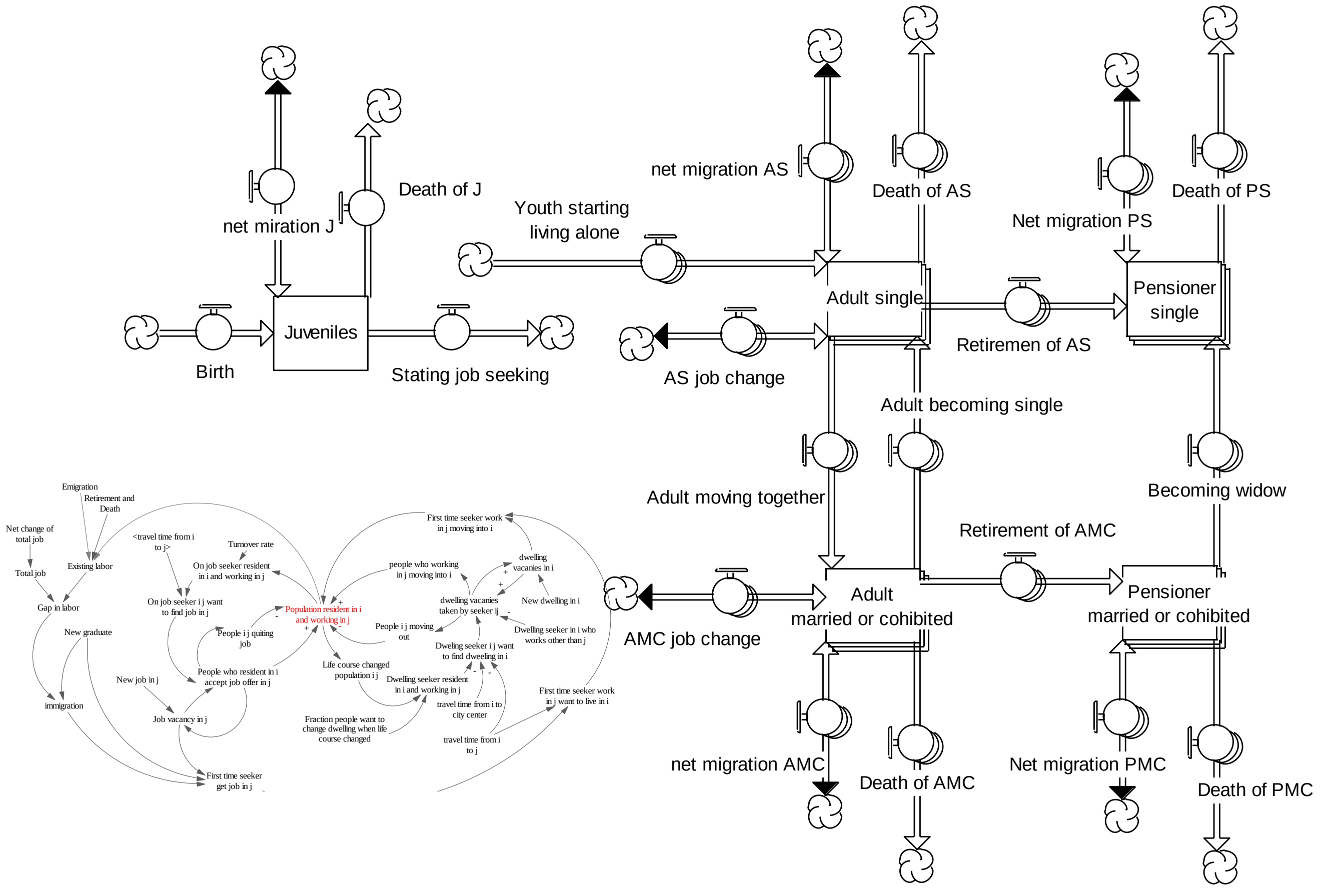


Figure4.4.1-1 model structure for population and household number

In this model, population was divided into four cohorts. The division is partly adapted from Ree’s life course patterns for Norway (Rees 1999).

Age 0 till age 19, the childhood and adolescent ages, in this cohort, people are assumed to live together with their parents..

Age 20 till age 66, adult ages, form this cohort, people leave their parents home to leave alone and start working. This cohort also represents labor force.

Age 67 and over, pensioner ages, people retire from this cohort.

People are divided into two categories, single and married or cohabited. Here, single means adult person living alone, or single mother or father with their children. Married or cohabited people mean two adult living together or parents living with their children. In other word, single means only one adult in one family. Married or cohabited stock means two adults in one family. Person who is over 19 but have no job and living in parents' home is not treat as adult in this model. The existing of children or not in a family does not influence the division of parents. Single parents with child are considered as single family. The relationship of the two adult, marriage cohabit or partnership, are not distinguished in this model. So, one single person represents one household. And, two married and cohabited people represent one household.

In this model, all people are attributed resident place and working place. So, array function is used for stocks. The four stocks, adult single adult married or cohabited pensioner single and pensioner married or cohabited, all represent a matrix (see Figure4.4.1 2). This matrix is similar with Figure3.1 1. For example, $P_{a,c}$ means population of people who resident in Askøy and working in Bergen center. The meaning of subscript can be seen in Figure4.1.1 1. The working place for pensioner is their previous work place before retirement.

$$\begin{bmatrix} P_{a,a} & P_{a,c} & P_{a,s} & P_{a,n} & P_{a,e} & P_{a,w} \\ P_{c,a} & P_{c,c} & P_{c,s} & P_{c,n} & P_{c,e} & P_{c,w} \\ P_{s,a} & P_{s,c} & P_{s,s} & P_{s,n} & P_{s,e} & P_{s,w} \\ P_{n,a} & P_{n,c} & P_{n,s} & P_{n,n} & P_{n,e} & P_{n,w} \\ P_{e,a} & P_{e,c} & P_{e,s} & P_{e,n} & P_{e,e} & P_{e,w} \\ P_{w,a} & P_{w,c} & P_{w,s} & P_{w,n} & P_{w,e} & P_{w,w} \end{bmatrix}$$

Figure4.4.1-2: Matrix for popoulaion.

For every population $P_{i,j}$, if i = j, it means people live and work in same area. They are not commuters. If $i \neq j$, it means people live in i and work in j. They commute from i to j every workday. Pensioners and children do not commute. For example, $P_{a,a}$ means population of people who live in Askøy and work in Askøy. They do not commute. $P_{a,c}$ means population of people who live in Askøy and work in Bergen center. They commute every day.

For all stocks, deaths and net migration will change the population. The intra-urban migration in this model is people move their resident place while working place remains. So it means people move from $P_{i1,j}$ to $P_{i2,j}$. If $i_1=i_2$, it means people move from one dwelling into another dwelling in the same area. For example, if one people who works in Bergen center move from Bergen center to Bergen north, the result of this movement is $P_{c,c} - 1$ and $P_{n,c} + 1$. If one people who live in Bergen center and work in Bergen center move to another dwelling in Bergen center, the result of this movement is$P_{c,c} - 1 + 1$. This means nothing changes in the model.

For adult stock, people might change job. It means people change working place while resident place remains. So it means people move from $P_{i,j1}$ to $P_{i,j2}$.

When people reach 19, they start looking for job. This action is captured by flow "starting job seeking". And after young people find job and dwelling, they leave their parents home and live alone. They become now adult in our model. This process is captured by flow "youth starting living alone". When people reach 67, they retire from their job. They become pensioners from adult. The flows of retirement represent these actions.

The flows between single and married or cohabited are moving together and become single. When two people who used to live alone decide to live together, both of them become married or cohabited from single. No matter they have registered as couple or not. When two related people decide to live separately, both of them become single again. If one related people died, another related people will become single. The marriage and divorce of pensioners is not considered in this model.

4.4.2. Residential mobility

"Square meters per person was a consistently good predictor of the propensity to move." (Clark, Deurloo, and Dieleman 1984) In this model, the change of space per person in household is used to simulate residential mobility.

The increase and decrease of member in one household relate to the living space in the family. With the change of people living in one dwelling, the room per person and space per person are changed. Clark found that average Sq. meter per person was highly associated with residential mobility(Clark, Deurloo, and Dieleman 1984).Since the change of member is the cause for space change, we use member change as cause for residential mobility.

There are two kinds of member change in household. The action of marriage, give birth of babies will increase the resident number in one dwelling. With increasing resident, space per person reduced. Divorce death and children leaving to live alone on the other hand will decrease the resident number in one dwelling and increase space per person.

The model structure can be seen below. When a household have new members, space per person in this household reduces. The dwelling becomes crowd. This causes stress in household. To solve this tense, some household might choose to ensure the crowd and stay in position. Others might decide to look for a new bigger dwelling to fit their space need. This behavior is researched by Speare (1970). It shows that most people move in the first year of marriage. When a household loss member, this household will have more space in dwelling. The extra space might cause extra expenditure. Some household might choose to find a smaller dwelling and some might like to stay in position. Moreover, it takes time for households to make decision. This process is described by Brown and Moore (1970).

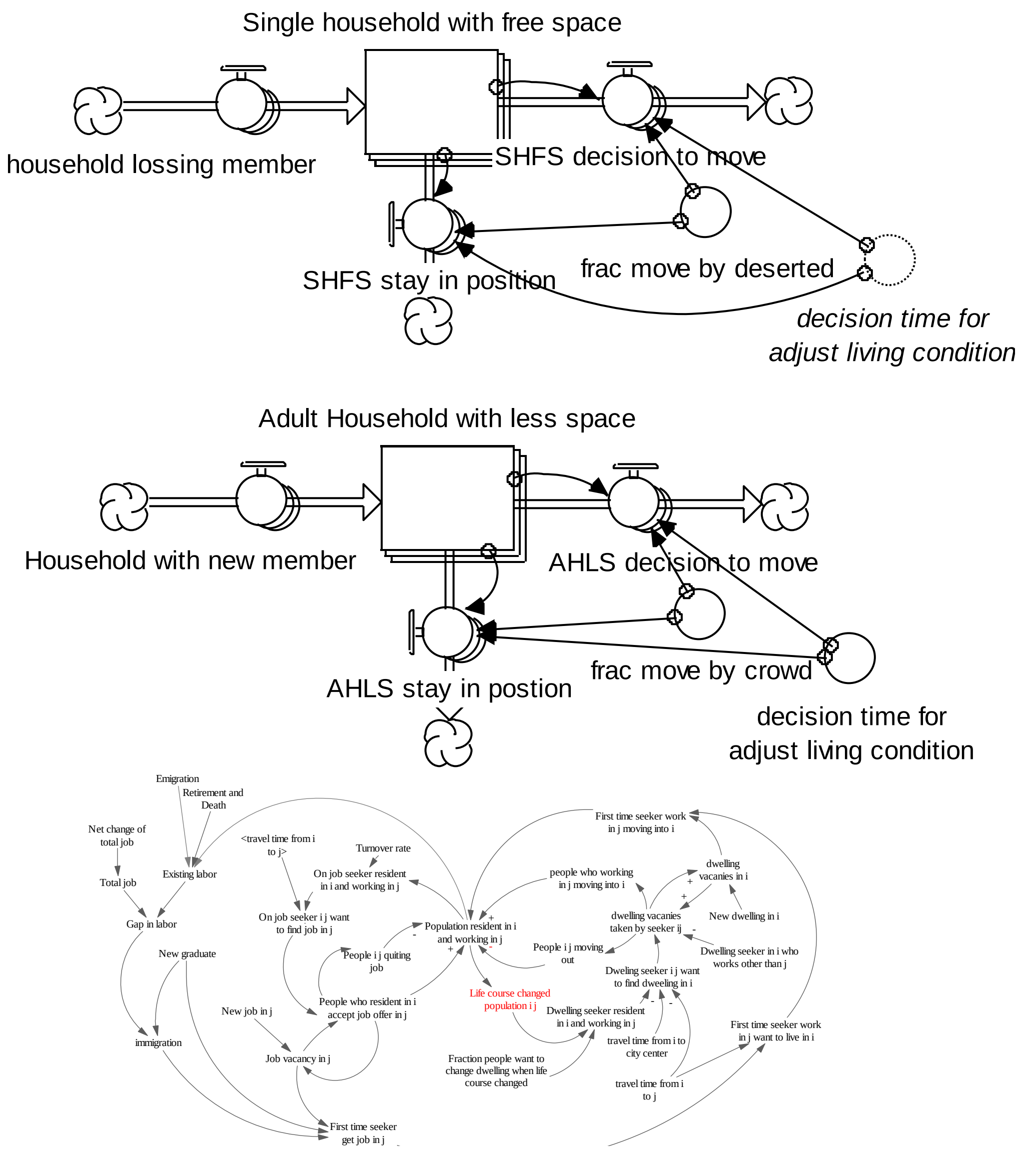


Figure4.4.2-1: model structure for resident mobility.

4.4.3. Dwelling seekers

Once households make their decision to move, they become dwelling seekers. But they are not the only dwelling seeker in the market. Immigrations and young people who just get job offer in the city also need a place to live. They are also dwelling seekers in the market. And, the action of divorce does not only create a household with more space. As one former household member should leave dwelling, person who leave the dwelling also need a new place to live. They also seek dwelling in the market. So, total dwelling seekers in the market are come from relocating households (people who are mentioned in 4.4.2), new households (divorced person who leave previous dwelling and new adult), immigrants and new graduated students.

The structure of dwelling seekers change can be seen below. When household decide to move, they become dwelling seekers. When dwelling seekers find an ideal dwelling vacancy, they will leave pervious dwelling and live in new place. And when they search for new living place for quite a long time and find no alternative to meet their demand, they might stop searching and choose to stay in position. Immigrations and new graduated students have no previous residence. So, they must occupy a dwelling for their residence. If there is no vacancies for them, they might become long distance commuters and live in other municipalities.

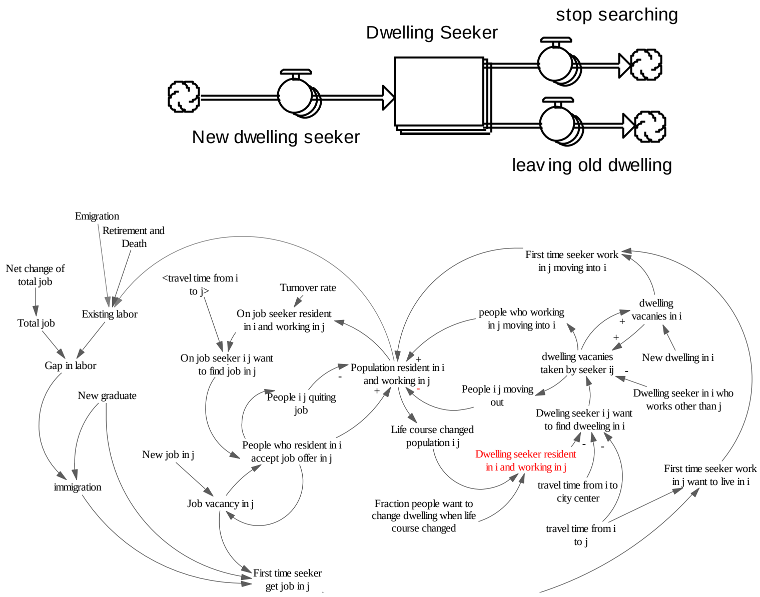


Figure4.4.3-1: model structure for dwelling seeker.

It is also worth mention that through dwelling seekers have decided to move but they are still living in the dwelling which they do not satisfied. The number of dwelling seeker will not influence the number of resident.

4.4.4. Residential location choice

The location choice is modeled using logit model adapted from Pagliara's research (Pagliara et al. 2010). People tend to live close to their work place. If travel time from dwelling to work place is near, the possibility household choosing this dwelling is high. The relationship between travel time and possibility is shown below. The equations can be found in appendix 3.

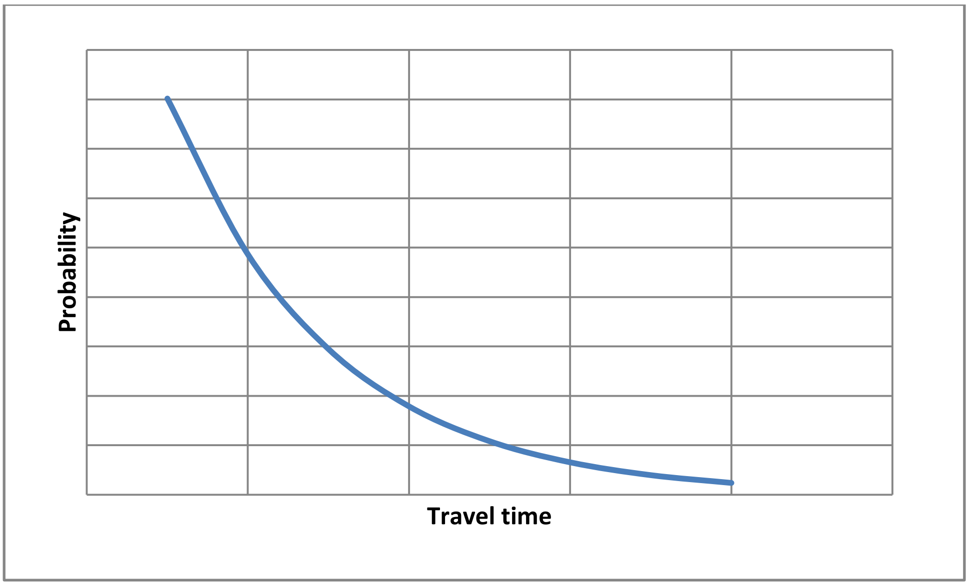
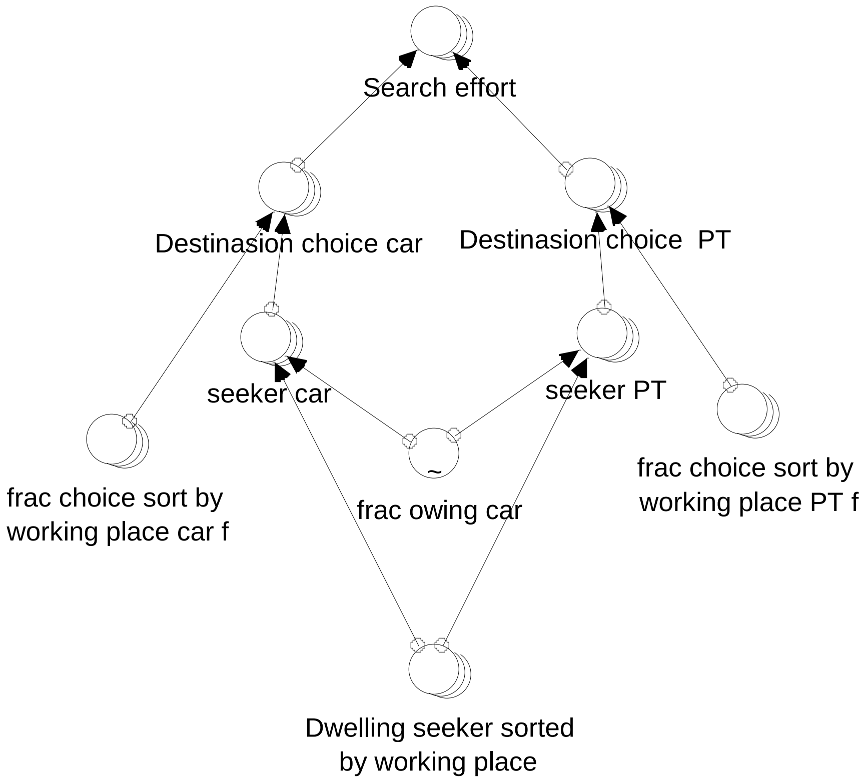


Figure4.4.4-1: relationship between travel time and possibility that location is chosen

Since some people use private car and some people use public transport to work, dwelling seekers are divided into two groups (car users and PT users) for location choice. The structure can be seen below (Figure4.4.4 2). Dwelling seekers are first sorted by their work place. It means that people working in same area have same preference on residential location. And then, seekers are divided into car users and PT users. For car users, the possibility that they choose an area is calculated by travel time with private car. For PT users, the possibility is calculated by a weighted travel time of bus, boat, tram and bicycle. All travel time used in this model is the research result of transportation model (Brandsar 2013). At the last, the choice for all dwelling seekers is sum up. The result is search effort of people want to live in a certain area.

The search effort means the effort of dwelling seekers on searching in one area. The more effort that seekers put in one area, the more likely that can get a vacancy. This function is used to insure that if the first choice location of seekers has no vacancy, seeker will try to find an alternative instead.



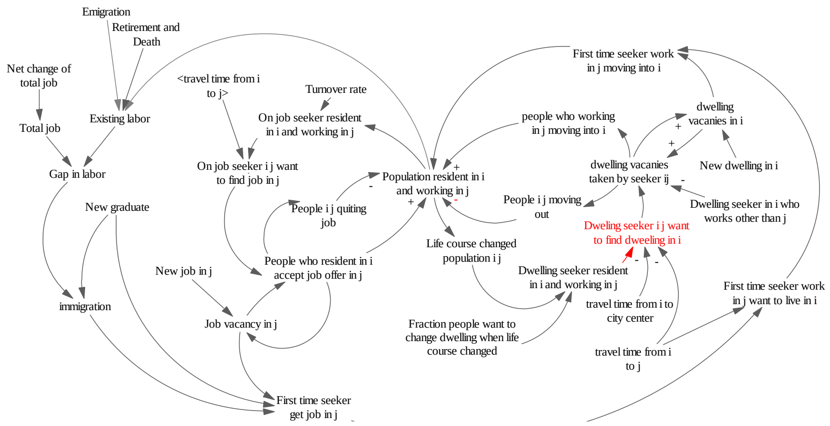


Figure4.4.4-2: model structure for residential location choice.

4.4.5. Search matching for dwelling vacancy.

The result of location choice calculation in this model is defined as search effort. Because after made a decision about residential location, dwelling seekers should search in that area for dwelling vacancy. This process can be seen below (Figure4.4.5 1). For a certain area, total seeker means the total number of dwelling seekers who want to live in that area. For dwelling vacancy, there will be an average transit time. Because signing contract, pervious resident moving out and other event in property transit all take times. So the vacancy taken is the minimum of dwelling seekers and available dwelling vacancies. And the number of vacancy taken divided by dwelling seekers is the fraction of dwelling seekers can get dwelling in that area.

The number of people will moving in a certain area is then the fraction of dwelling seekers can get vacancy in that area multiple the number of dwelling seekers who are searching in that area.

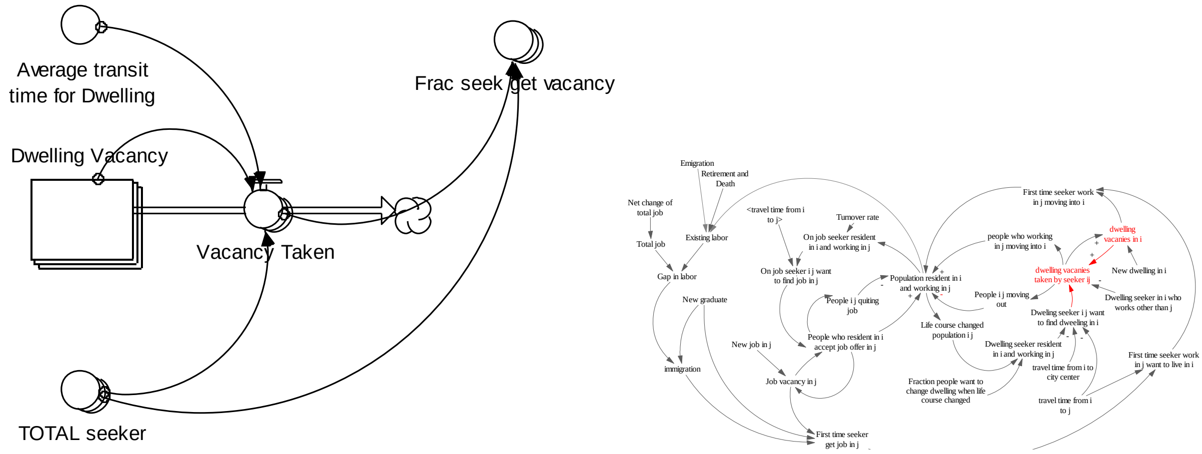


Figure4.4.5-1: model structure for dwelling search matching.

4.4.6. Balance of dwelling vacancy.

Vacancy taken is the outflow for stock of dwelling vacancy. In this chapter, we structure the inflow of dwelling vacancy. Several actions will create new dwelling vacancies. Figure below (Figure4.4.6 1) shows the source of new dwelling vacancy.

New dwelling: new dwellings are constructed in a developing city like Bergen. Once a construction project is finished, new dwelling will enter into market.

Household diminish: if all people in one household die, the household will diminish. The dwelling will become vacant. In this model, only single persons' death influence household diminishes, if one single person died, the dwelling will also become empty. If only one person in a related family died, the other person will still live in the dwelling. So the death of related people will not affect household diminish. We do not model the action that two members in one household die at the same time

Emigration: most household moving together when they are moving to another city. So, if one household migrate to another city, their dwelling become vacant.

Marriage: the action of marriage means two single adults move into one dwelling. They used to occupy two dwellings and occupy only one dwelling after marriage. So one of the dwellings they occupied become vacant.

Relocation: for intra-urban migration households, they move into new dwelling and leave their pervious dwelling. After they move out, the pervious dwelling becomes vacant. Figure 4.4.6 2 shows the structure for seekers leaving former dwellings. In the process of residence change, people do not change work place. So, we first calculate the number of households get new vacancies by their work place. And assume that households can get vacancies evenly no matter their former residence place. This might cause some bias in the model. Because, the selling price of former dwelling might influence the seekers available fund for new dwellings. But we do not include housing price in the model. So, we assume that households' purchasing powers are the same in every area.

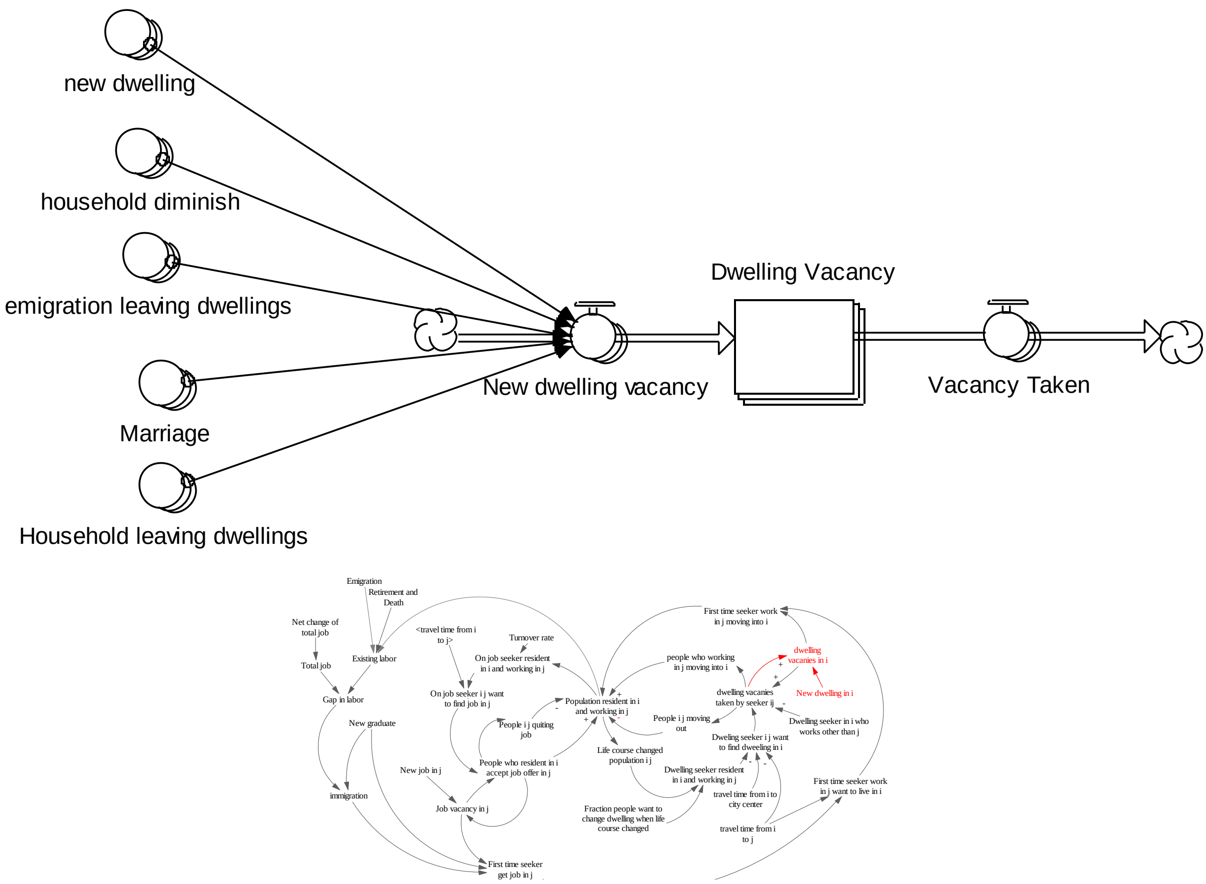


Figure4.4.6-1: model structure for balance of dwelling vacancy.

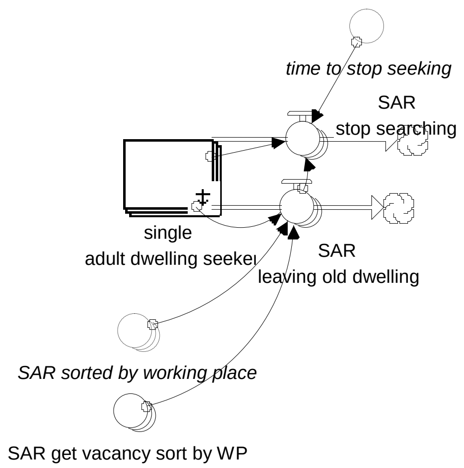


Figure4.4.6-2: model structure for relocate leaving old dwelling.

4.4.7. On-Job change

The job change model (Figure4.4.7 1) has similar structure with resident change. The turnover rate represent fraction of workers might who want to change job. Once people decide to change job, they become on job seeker.

Since job seekers might want to find a job near to their resident place (Nebiyou and David 2010, Rouwendal 1999, van Ommeren, Rietveld, and Nijkamp 1997). In this model, commuting time is used to simulate job seekers location choice. Job seekers are divided into two groups, car users and public transport users. And random utility theory is used to simulate the location choice base on travel time of private car or public transport. The relationship between travel time and possibility to seek job in one area have the same behavior as residential location choice (see Figure4.4.4 1). The equations can be found in appendix 3

The location choices of job seekers then are summed up. The number of job offers is then determined by the minimum number of job vacancy and job seekers in one area. And once job seekers receive job offers, they quit their previous jobs. The jobs which were taken by them become vacancy.

And emigrations, death of workers and retirements all generate job vacancies.

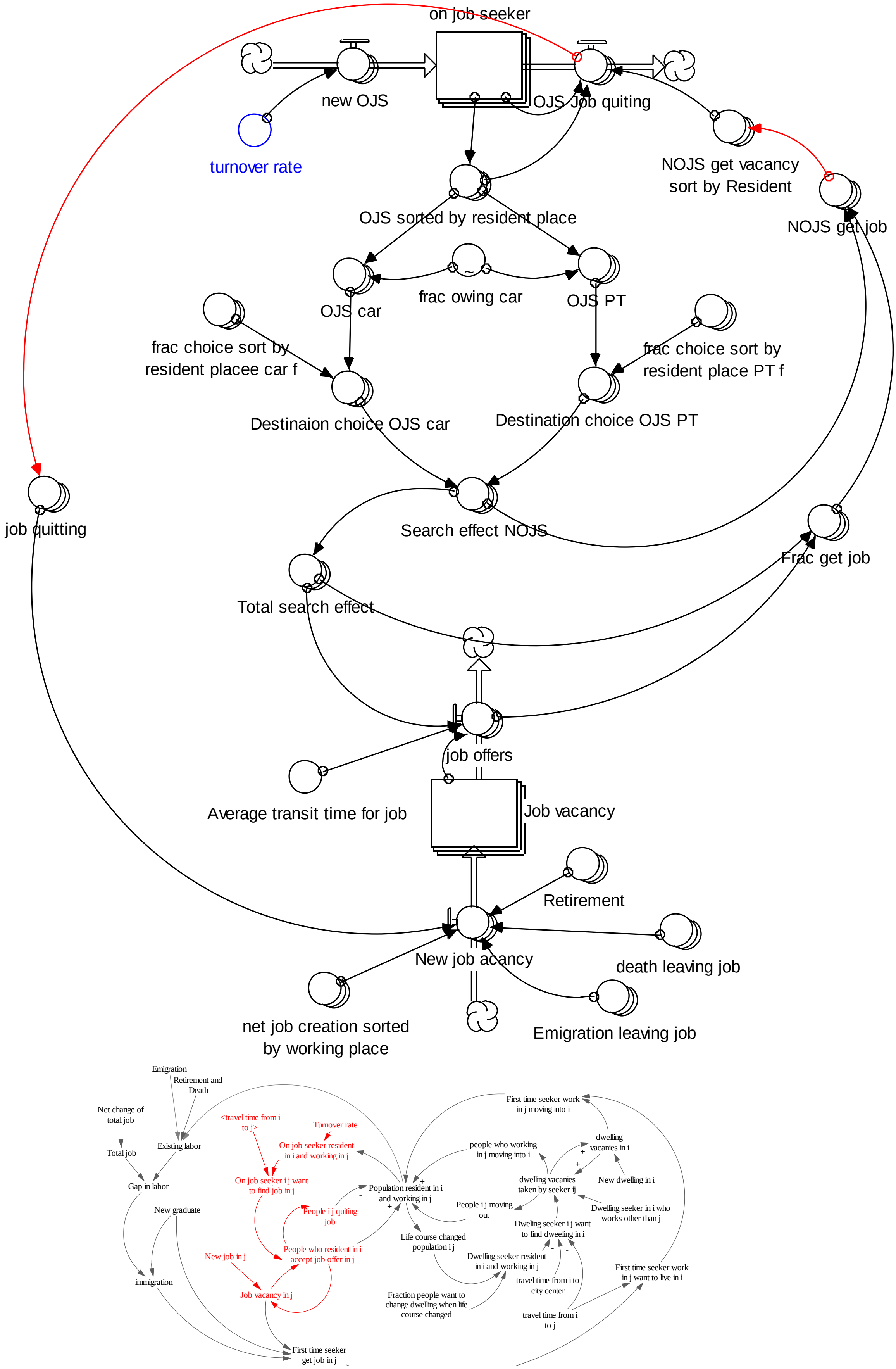


Figure4.4.7-1: model structure for on job change.

4.4.8. First-time Job seekers

Other than on job seeker, immigrations and students who just graduated from school are first time job seekers. They do not have a resident place when they are searching for a job. So they do not have preference on job location. They will search where there are more jobs. The fraction of first time job seekers seeking in one area equals the fraction of job vacancy in that area.

5. Data and parameters

5.1. Sources of data and parameters

The data and parameters used in this thesis are mainly come from three sources.

Most demographic data are from Statistics Norway (SSB) database. Residents are obliged to report resident place in Norway. Even they are moving within a municipality, the changes of address are going to be record by population registry. All these data exist in Statistics Norway database (Skiri 1995).

The travel time, job creation and dwelling construction are import from other two simulation models in this project. Simulation result of travel time is imported from Brandsar (2013). Job creation and dwelling construction from 2000 to 2010 is calculation result made by Schulze. From 2010, these data is using a simulation result of Schulze (2013).

The last source of parameters is research result in literatures. Because there are very few relevant research conducted in Bergen. Research results in other country like Netherlands and US are used in this model.

5.2. Data and parameters in residential mobility

Due to the SSB (Statistisk sentralbyrå) fee for processing raw data, the Details in commuter number cannot be accessed. An estimated commuter number is calculated by commuter matrix in year 2008 (Strand, Christiansen, and Engebretsen 2010) and fraction change in traffic from 2002 to 2008 (Meland 2009). It should be know that Meland (2009) only list the number of trips between different urban districts. Since a certain fraction of trips is conducted by commuter (Meland 2009, Strand, Christiansen, and Engebretsen 2010), the fraction change of trips can be used for fraction change of commuters.

The Marriage rate[3], divorce rate[4], death rate[5], emigration[6] is calculated from SSB (Statistisk sentralbyrå) data. It is assume that death, marriage and emigration happens evenly among all areas. We also assume that all marriage happens in adult cohort, which means pensioners do not marry to each other anymore. According data from SSB average number of child[7] is 2.34 for women who was born 1960s. The number is 2.6 for women who was born 1940s. So, I assume that the average number of child is decreasing slowly. Distribution of emigration among different cohorts is adapted from distribution in Hordaland country (fylke). The fraction of people decide to move in the next year after family structure change is from Spare's researched (Speare 1970).

5.3. Data and parameters in residential and work location choice

In logit model for residential and work location choice, to simulate location choice from 2000 to 2010, travel times from 2000 to 201 are needed. Such records are not available. So a simulation result of travel time matrix from Brandsar (2013) is used in this thesis. The travel times are shown in appendix 2.

Through there are research result of people`s willingness to commute in other country (Pagliara et al. 2010), the calibration process shows that people in Bergen are less willing to commute. They put more weight on travel time than other place. So, the parameters for people`s willingness to commute is calibrated to fit better simulation result in this thesis.

5.4. Data and parameters in dwelling and job vacancy

The data for net job creation and dwelling construction from 2000 to 2011 is import from Schulze (2013) calculation based on data from statistic Hordaland (Hordaland fylkeskommune statistikkteneste statistikk.ivest.no) and SSB. The data from 2012 to 2030 is feed with Schulze`s simulation (2013).

Due to that all population in this thesis is given to working place, the net job creation is divided by employment rate before imported into model. The employment rate is calculated by number of worker divided by number of adult resident.

Due to the fact that some dwellings are occupied by multiple households while in the model one dwelling can only accommodate by one household. A coefficient is used for dwelling construction. This coefficient is calibrated to fit better simulation result.

5.5. Data and parameters in job change

For job turnover rate, research shows that total job relocation is around 12%-15% every year (Klette and Mathiassen 1996).

6. Discussion

6.1. Discussion about model structure

Researches on residential mobility and residential location choice have long history and tremendous output. There are many different theories and techniques for making such migration model. In this part, we discuss the reasons for choosing theories and techniques

6.1.1. Discussion about Residential mobility

There are two main kinds of measure to make a simulation model for residential mobility. One is model movement by trigger(Brown and Moore 1970, Rossi 1980(1955)). Or mobility can be modeled as lifetime or fraction (Clark, Deurloo, and Dieleman 1984, Kim, Pagliara, and Preston 2005, Weinberg, Friedman, and Mayo 1981).

There are two reasons to renounce modeling technique of lifetime or fraction. The first reason is the lack of relevant research in Bergen. If modeling by lifetime or fraction, population should be divided into several groups, and every group has a certain lifetime or fraction for moving. The lifetime and fraction can be researched by statistics or census analysis. But for this particular case, no such research has been done before for Bergen municipality. Statistics or census data are also not available. Analysis result from other country might not fit Bergen area since mobility might different from different culture. The second reason is behavior match. Research shows that the longer duration of residence , the smaller mobility rates (Speare 1970). This phenomenon is different from system dynamic model behavior with a lifetime. If there is a constant lifetime in system dynamic model, the longer duration of residence, the larger mobility rates should be. So, this technique is not used in this model.

If modeling by trigger, the model should capture the number of people had the action might trigger movement, like change of family structure change or change of income. With available family structure change data exist from Statistisk sentralbyrå (SSB) for Bergen area, this measure is chosen for this model.

In this model, family structure change is selected as the only reason for moving. While Job change is not selected as reasons for moving because job change might be no significant for residential moving (van Ommeren, Rietveld, and Nijkamp 1999). Or job change only affect household who rent dwellings but not for households who own dwellings(Clark and Davies Withers 1999). As only 26% households renting dwellings in Bergen2, the effect of job change on residential move is ignored.

The movement for reducing commute time is also excluded because it might not be a obvious reason to move. Some research shows commuting distance has effect on moving (Kim, Pagliara, and Preston 2005, van Ommeren, Rietveld, and Nijkamp 1999). Some shows not (Clark, Huang, and Withers 2003). Or it only have effect in inter regional migration (Deding, Filges, and Van Ommeren 2009)

Social facts like population density, quality of school, detached house, number of bedrooms, age, household income and transactions cost have been proved to be strong influence on migration (Kim, Pagliara, and Preston 2005, Weinberg, Friedman, and Mayo 1981). But social-economic and demographic composition variables have small spatial extent of influence (Guo and Bhat 2007). When this model have a relatively large aggregated level, it is not necessary to include these facts.

6.1.2. Discussion about Residential location choice

Like residential mobility, residential location choice is also influenced by enormous facts. For the same spatial reason, all social facts are excluded in this model. But dwelling type and dwelling state are still very important facts (Hjorthol 2003). To simplify the model, we should assume that market supply can perfect fit household's demand about dwelling type and dwelling state. Therefore, we exclude these facts in resident location choice model.

6.1.3. Discussion about Job change

For job location choice many researches show that commuting distance have negative effect on job location choice (Nebiyou and David 2010, Rouwendal 1999, van Ommeren, Rietveld, and Nijkamp 1997). The reason for this phenomenon is people willing to accept lower pay for shorter commuting distance (Van den Berg and Gorter 1997).

But, it is not clear that a job seeker can find a lower par job for shorter commuting distance. And in reality, there are varieties of job positions, people won`t or cannot accept a position in other field to have shorter commuting distance. So, the job seekers` logit model for location choice might overestimate the effect of travel time on search effect. This part of model need further local research`s support.

6.2. Limitation and Weaknesses

There are great limitations in this thesis. The first limitation is the relevant research in Bergen. The most part of this thesis is simulation model for people`s decision making. The methodology of simulation model can be adapted from research for other country. But parameters might not be the same for different countries. The perceived values for time, distance are different in different place. So, there is a great uncertainty for parameters in this thesis.

The second limitation is the lack of data. Most data needed in this model is unavailable for model maker. Three kinds of reason cause the lack of data. First, funding problem, in Norway, most published data is organized in municipality level. Through data in detailed level (urban district) do exist, this thesis have no authority and enough funding for access. Second, data collection is done by statistic department. The purpose of data collecting is not for simulation model making. There are many differences between statistic data and data for simulation. For example, data for dwelling unit is collected from government register department. Multiple household can live in one dwelling unit in real world and statistic data. But when making a simulation model, dwelling unit should fit the number of households. Third, many data needed for model making have never been collected. Due to these limitations, there are several weaknesses in this thesis.

6.2.1. Division of occupation

In this model, occupation is not included. But for commuters, different occupation might influence the behavior of commuting. For example, some casher in supermarket might not travel on rush hours. That means travel time on rush hours might not influence their residential location choice. Also, in this thesis, we assume that it is possible for people to get a job in every part of city. This assumption might be wrong for some occupation like doctors or academics that have fewer job location choices.

6.2.2. Housing price

Housing price is excluded in this model. There are two reasons for this. One reason is lack of data. The other reason is that the price is more or less determined by market demand. When market demand is endogenous in this model, it is reasonable to exclude the price. But it might be better to include price since price have strong feedback to demand.

6.2.3. Area bias

Through in a relatively large aggregated special level, most social fact can be ignored. Some area bias still exists like the influence from climate and landscape. mental map might also influence people`s location choice (Adams 1969). So some alternative specific constants are definitely needed for residential location choice. But without previous research, it is not yet possible to do so.

Through many weaknesses do exist, it does not mean that this model is valueless. With model testing in the next chapter, it can be found that this model can still capture the main underlying structure and behavior of the system.

7. Model validation and testing

This chapter follows the instruction of model validation and testing from Sterman (2000).

7.1. Behavior reproduction

This thesis presents a model of the commuter pattern over time. So, we test whether the model can reproduce that pattern. As mention above, only two data points (year 2002 and year 2008) are available in historical data. Data point in year 2002 is used as initial data. So, only data in year 2008 can be compared.

First, total commuter is compared (Figure7.1 1). The total commuter is the sum of all population who resident and work in different area. Simulation is from 2002 to 2010. So in year 2002, Data and simulation have same value. I year 2008, simulation result (64657 person) is a little below data (64626 person). The trend of simulation shows the increase is exponentially. This behavior also fit the exponential increase trend of traffic (Brandsar 2013).

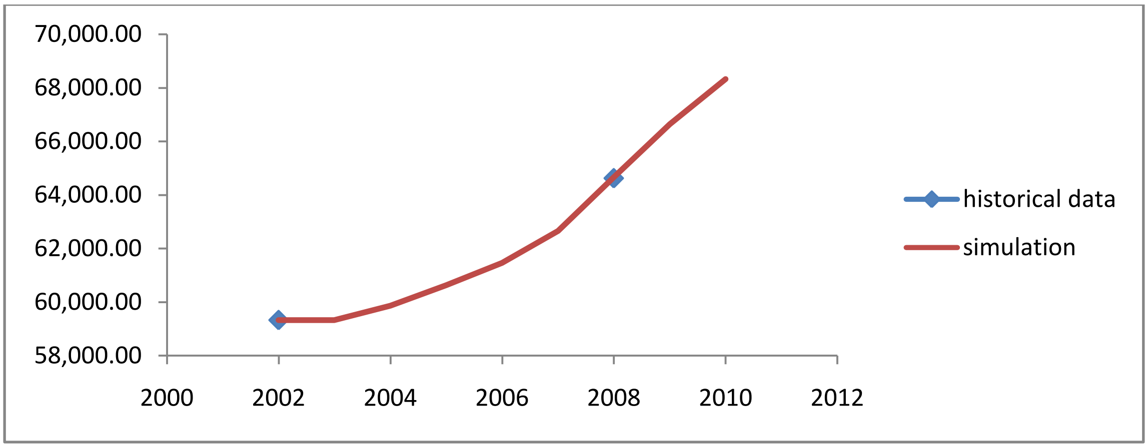


Figure7.1-1: model behavior reproduction for total commuter.

Second, the commuter matrix is compared. Simulation result is shown below (Figure7.1 2). This table shows the number of people with living and work place in year 2008. The dimensionless error ($(\text{Simulation result} - \text{Historical data})/\text{Historical data}$) of simulation is shown in the next (Figure7.1 3). The average dimensionless absolute error is 31.85%.

2008		Working place					
		Askoy	Center	South	North	East	West
Residence	Askoy	5321	2080	713	65	19	1293
	Center	168	24739	4334	2073	316	3297
	South	84	13352	9781	744	511	4615
	North	32	7598	1042	6384	704	767
	East	9	1700	1191	870	1596	227
	West	363	6842	9105	376	167	6934

Figure7.1-2: model behavior for commuter matrix.

2008		Working place					
		Askoy	Center	South	North	East	West
Resident place	Askoy	-2.25%	-22.86%	-27.45%	-68.09%	-52.88%	-28.60%
	Center	13.32%	3.29%	-15.02%	21.73%	0.77%	12.52%
	South	-34.22%	22.49%	-24.93%	12.61%	51.54%	88.83%
	North	-55.15%	-0.31%	-35.00%	-1.45%	58.94%	-43.39%
	East	-44.88%	5.82%	58.77%	40.48%	-21.20%	-18.92%
	West	-22.08%	-48.76%	87.85%	-62.24%	-15.45%	-22.24%

Figure7.1-3: model behavior bias for commuter matrix.

The error of simulation might be caused by several reasons. They are listed below.

1. Error of travel time.
2. Error of spatial gravity center.
3. Error of calibrated parameters.
4. Error of model boundary.

The travel time used in this model is result of another model by Brandsar (2013). In that transportation model, the inputs are commuter matrix and other transportation user. And the output is travel time matrix. But due to the same problem of lacking data, the commuter matrixes are estimated as a linear increase from 2002 to 2008 which might cause some error in travel time. Plus, other transportation users are also from estimation. The error of input might cause error of output. So the travel time used in this model might have error itself which might cause error of simulation.

In transportation model (Brandsar 2013), the travel time is the average travel time for residents in the area. Travel time for every origin destination pair is unique. Travel time is same if origin destination pair is same. It means people who live in the far end of the area have the same travel time with people who live in the center of area. To do so, spatial gravity center is needed. The gravity center should be the gravity center of distribution of population. But to acquire such data, detailed population is needed. As the same problem as other data, detailed population distribution is no available. So, only estimated gravity centers are used. The error in choosing gravity centers might cause error in travel time and furthermore cause error in simulation.

Other than error of input, the parameters in model might also cause error of simulation. The parameter in random utility calculation of residential location choice is calibrated in the model. Without support from previous research on people's preference in Bergen, the parameter is only a rough estimation. This might cause simulation errors.

Also, only travel time is considered in this model. Research shows that transportation should not be the only fact for residential location preferences (Glen Weisbrod 1980). So simulation result is better if the model can capture other facts. But without support of data and previous research, all other facts must be excluded from this model's boundary. The small model boundary might also cause errors.

7.2. Extreme condition test

Some extreme condition tests are conducted to test the model structure. The first extreme condition test is high travel time within Bergen center. In this condition, we assume that travel time within Bergen center is doubled while other travel time remains. The simulation result shows the people who live and work in center Bergen decreases (Figure7.2 1). The red curve (curve 2) shows simulation without change. The blue curve (curve 1) shows the simulation with extreme condition. Simulation result satisfies the assumption that if one area is too crowd. Travel time become too high within the area. Most people might want to move out.

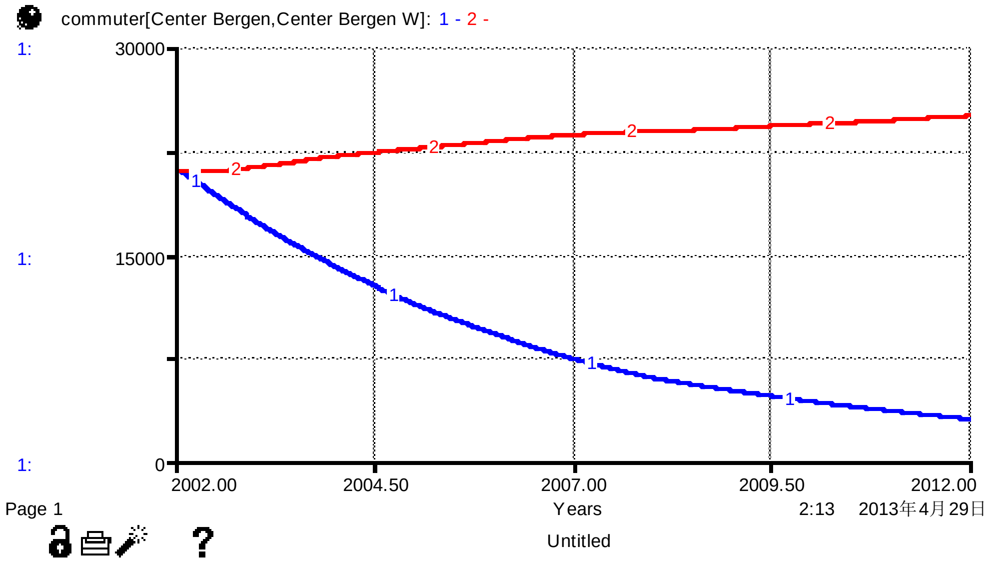


Figure7.2-1: Extreme condition test for high travel time inside Bergen center.

The next test is Extreme condition test for no job creation. It is assumed that no new job is created during the simulation. The total job remains constant. The simulation result is showed below (Figure7.2 2). The blue curve (curve 1) shows simulation without change. The red curve (curve 2) shows the simulation with extreme condition. When there is no increase in job, the total workers also remain constant; even there are more new graduated job seekers than new retired workers.

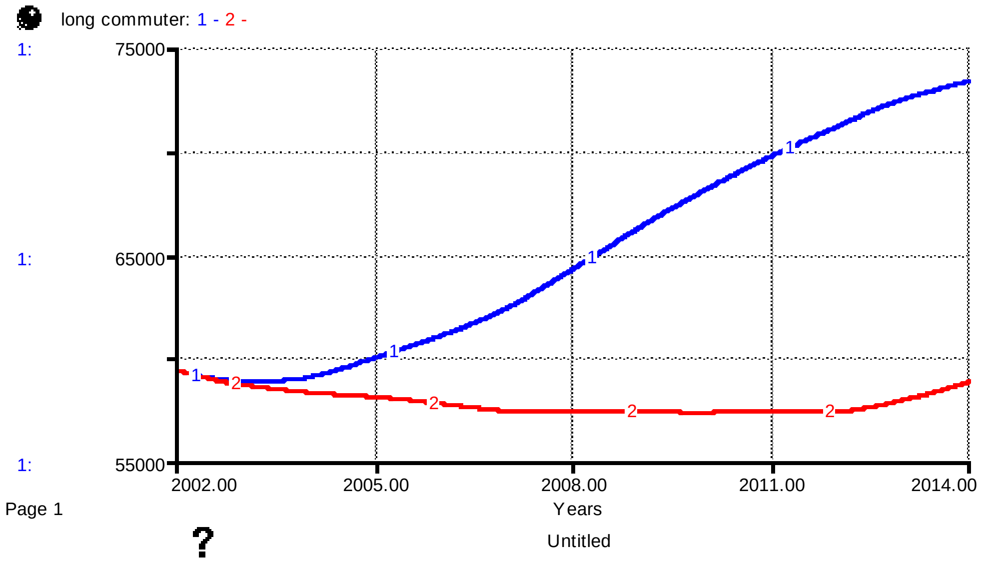


Figure7.2-2: Extreme condition test for no job creation

The nest extreme condition test is no dwelling construction test. This test assumes that no new dwelling is constructed. The simulation result is showed below (Figure7.2 3). The red curve shows simulation without change. The blue curve shows the simulation with extreme condition. The simulation shows that without new dwellings, population continues increase for several years and stop growing after that. This phenomenon can be explained as below. Without new dwellings, new residents first consume the existing vacancy in the market. When all vacancies are occupied, they cannot find dwelling vacancies anymore. In this model, these people hold in a state as dwelling seeker. In reality, they might choose to live outside Bergen and become commuter between municipalities.

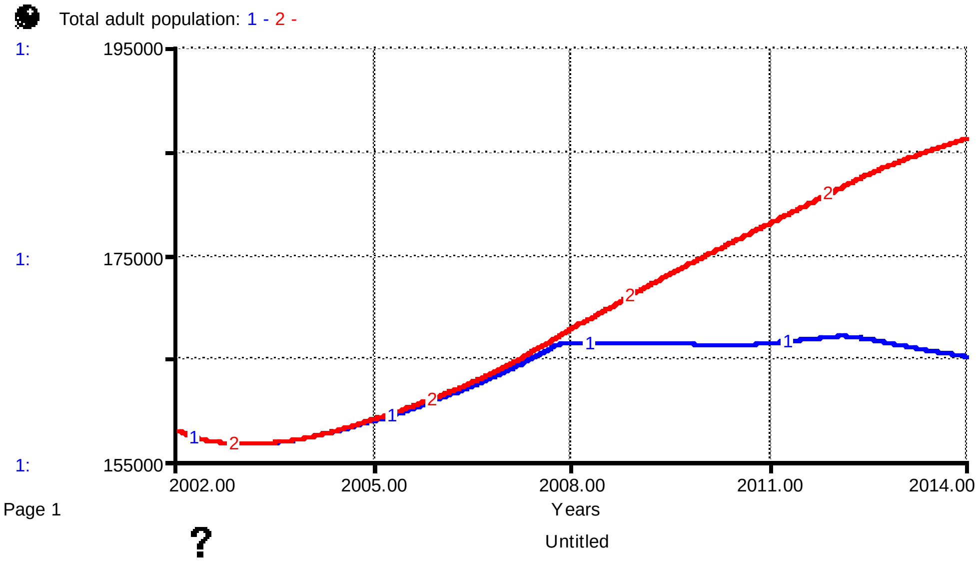


Figure7.2-3: Extreme condition test for no dwelling construction

We also tested array function. In this test, we test if the change in Bergen South can influence the population in Bergen North. The test condition is no dwelling construction in Bergen South.

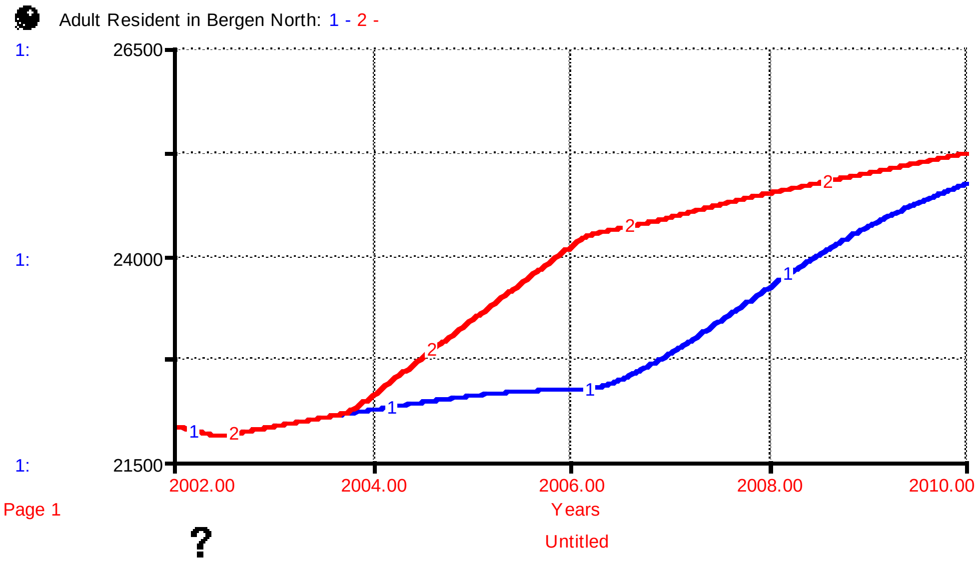


Figure7.2-4: Extreme condition test for no dwelling construction, resident in North

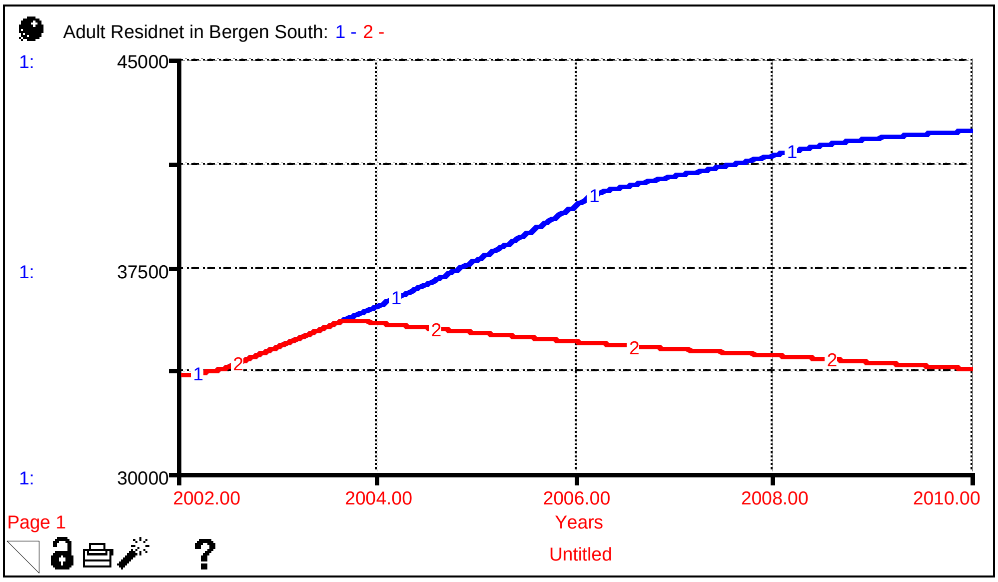


Figure7.2-5: Extreme condition test for no dwelling construction, resident in South

Figure7.2 4 and Figure7.2 5 show the simulation result. When there are no new dwellings in Bergen South, population in Bergen North increase while population in Bergen South decreases. This is because people cannot find new dwellings in South. So the housing demand for North increases. The population in Bergen South decreases rather than keeps constant. This is because more single households live in South. Figure7.2 6 shows that the number of household keeps constant in Bergen South.

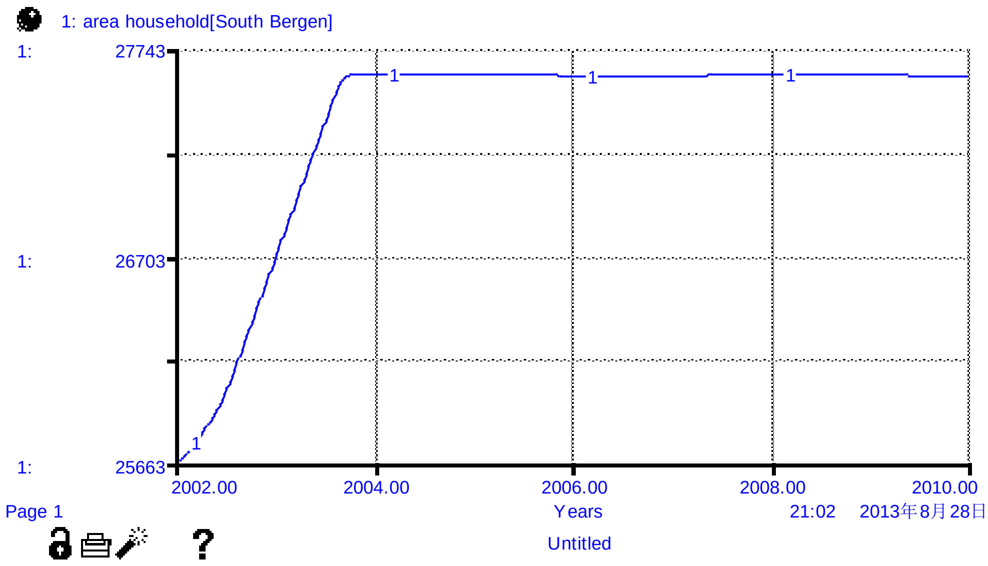


Figure7.2-6: Extreme condition test for no dwelling construction, household

7.3. Integration error test

The time step for this model is one fortieth year. And Euler`s method is chosen for integration. Model behaviors do not change when changing time step into smaller one or changing integration method.

7.4. Sensitivity Analysis

Behavior mode sensitivity is test by changing important parameters in the model. By sensitivity analysis, the importance of parameter can be found.

The first analysis is the test for travel time. Travel times for all areas are test with normal value, double travel times and half travel times. The simulation is shown below (Figure7.4 1). When travel time is very high (red curve 2), much fewer people want to commute. So the total commuters decrease significantly at the first. After some year, the total commuters increase again with the increase of population. And if travel time was shorten (green curve 4); more people might decide to commute. The total commuters have a dramatic increase during the simulation.

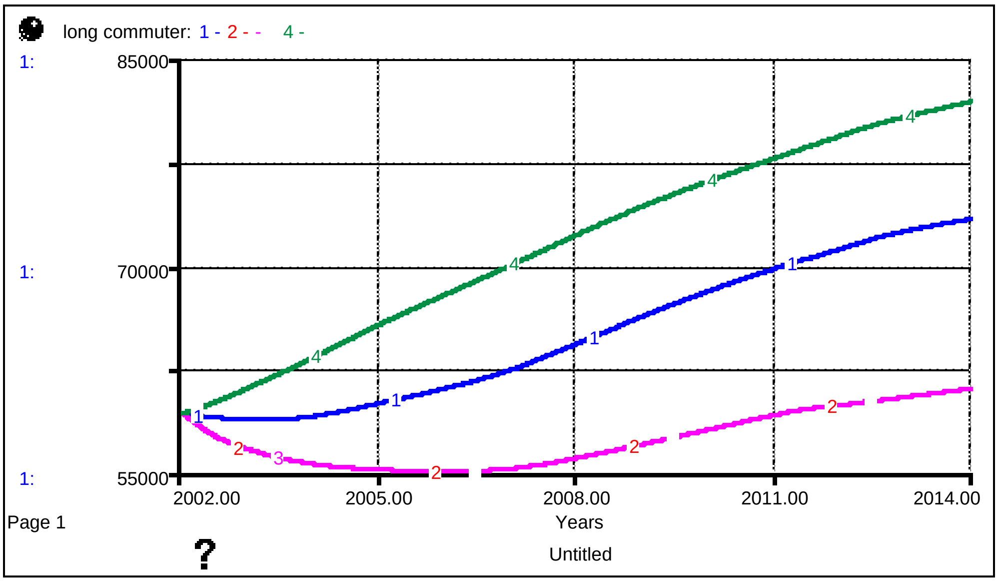


Figure7.4-1: Sensitivity analysis for travel time

The next analysis is the test for parameter of people`s willingness to commute. The simulations (Figure7.4 2) have the same result as travel time. This indicated that both travel time and parameter in logit function are equally important in this model.

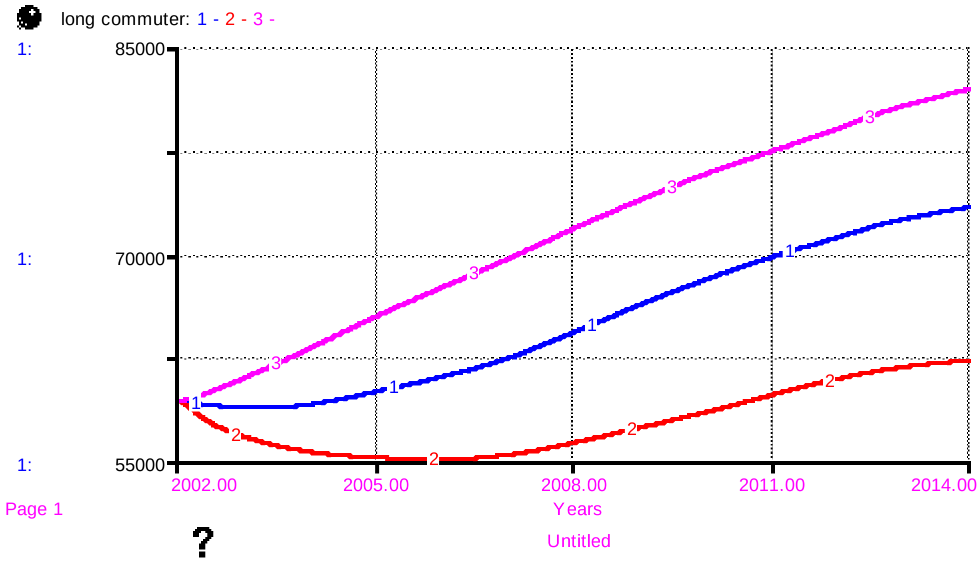


Figure7.4-2: Sensitivity analysis for parameter of willingness commuting

The Sensitivity test result for parameters in residential mobility is shown below (Figure7.4 3). The total commuters do not have significant change when relocatee numbers is doubled (curve 2) or halved (curve 3). This phenomenon shows that relocatee number have relatively small effect on change of commuter numbers. From the model structure study, we can explain this as the relocatee`s movement is their second choice. Their preferences do not change, so their movements do not change the commuting state. It also shows that the main reason for people to move might be need for space rather than need for changing commuting time.

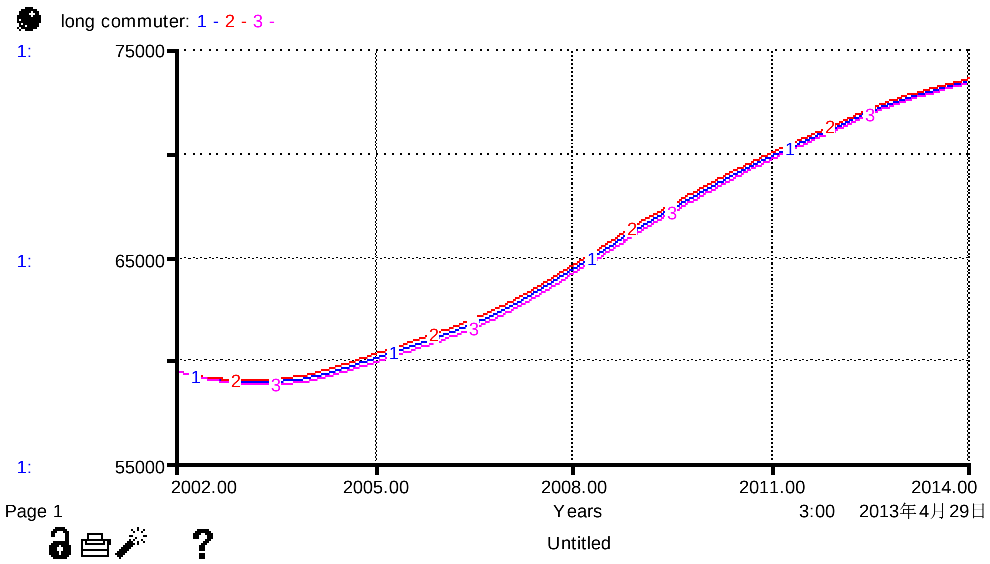


Figure7.4-3: Sensitivity analysis for parameter of residential mobility

The Sensitivity test result for job turnover rate is shown below (Figure7.4 4). Compared with sensitivity test of resident mobility, the changes of turnover rate have bigger influence on number of commuters. This might due to the reason that job turnover rate (around 10%) is bigger than residential mobility rate (around 2%). So, then change of on job seekers have bigger influence.

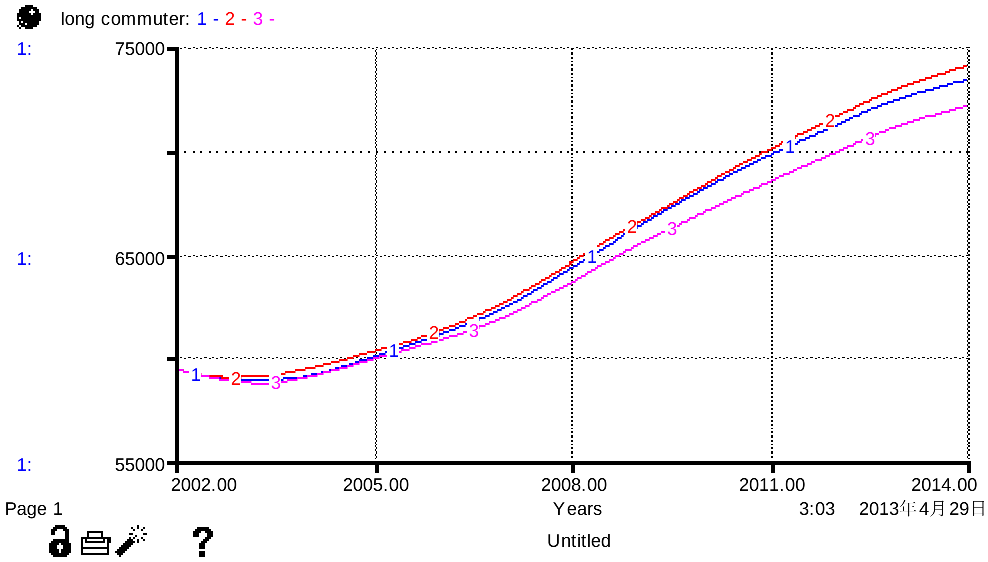


Figure7.4-4: Sensitivity analysis for turnover rate

7.5. Structure behavior test

Structure behavior test is used to test the strength of feedback loop. Testing is conducted by cancelling part of the model.

We first test no resident relocation. Result (Figure7.5 1) shows that there are little different (curve 1: base run, Curve 2 test run). This means the local resident relocation have little effect on commuters. This is because we assume that people's willingness to commute is constant.

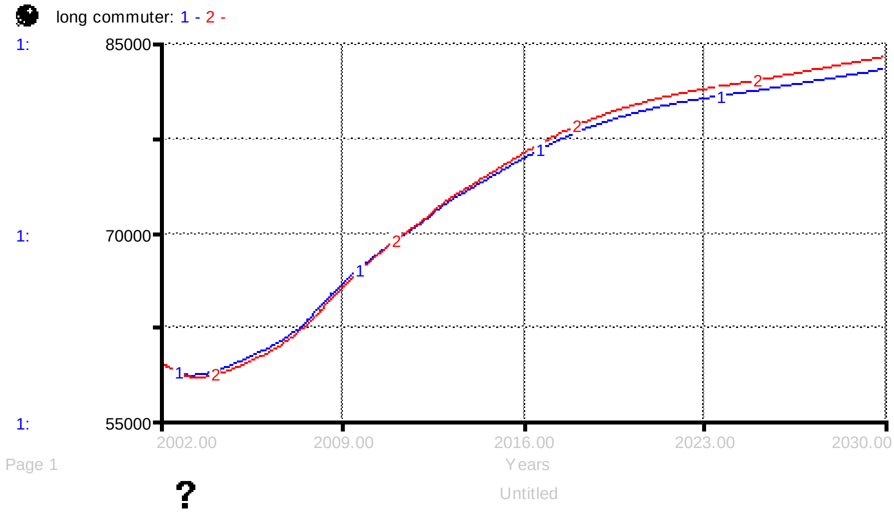


Figure7.5-1: structure behavior test for residential mobility

We now test no job change. Result (Figure7.5 2) shows a little drop (curve 1: base run, Curve 2 test run) on total commuters. This means that the job change process is increasing the number of commuters.

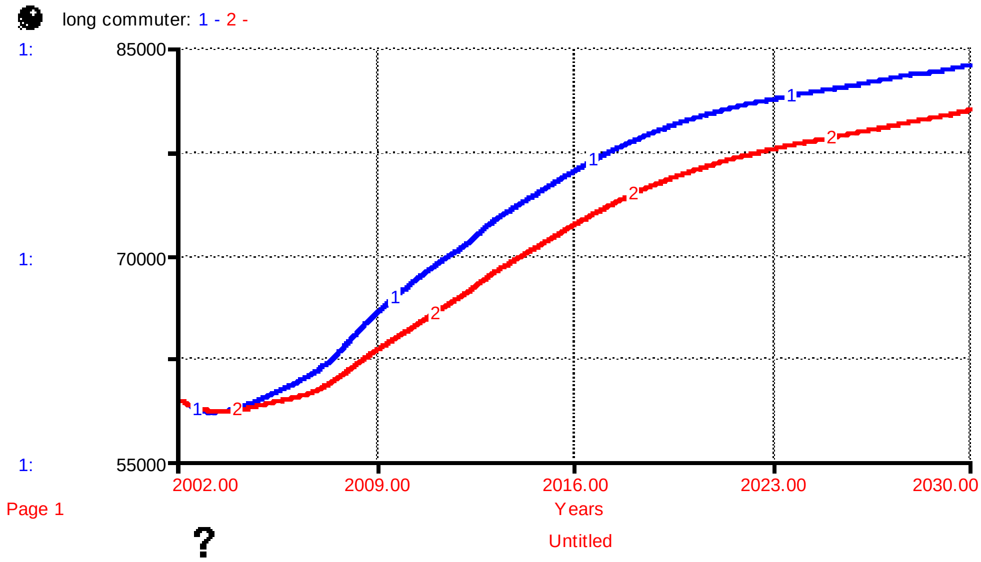


Figure7.5-2: structure behavior test for job change

The next test is population increase. Result (Figure7.5 3) shows total commuters keep constant (Curve 2).

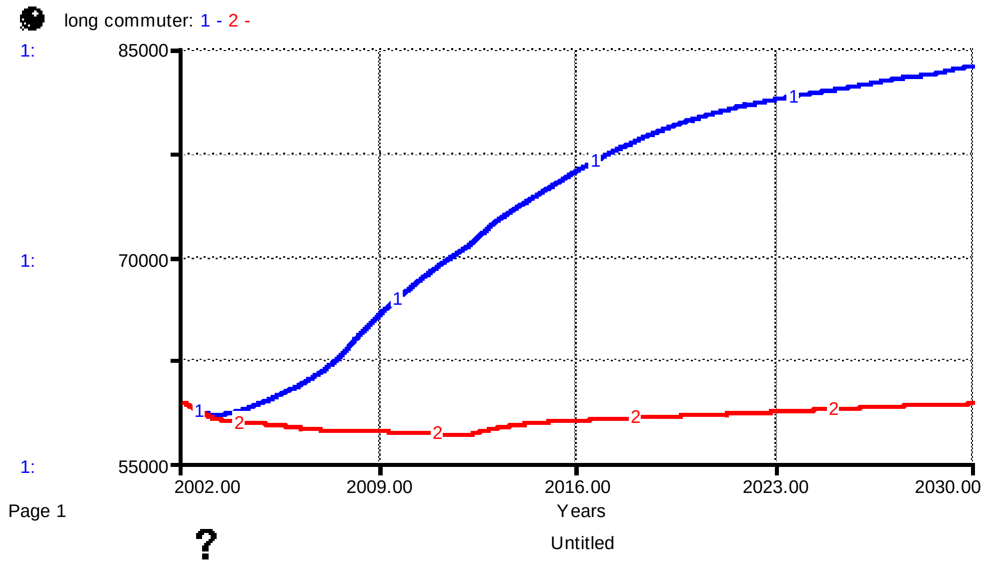


Figure7.5-3: structure behavior test for no population increase

8. Result and policy

This model is only migration part of transportation and land-use model. Before all models are joined together, many feedback loops are still open circuits. So it is not possible to test specific policy or conduct any cost benefit analysis. In this chapter, we only test some rules for land use policies.

Three land use policies are tested in different conditions. The first land use policy is free market. That means there is no land use policy, market will generate freely. The second policy is *equal construct policy*, this policy means that business land use and dwelling land use develops simultaneously and distribute equally. In a simple way, in every new zoned land, half of the place is used for business and the other half is used for dwelling. The third policy is *aggressive zoning policy*. In the policy, new dwelling are constructed where most job locates. The distribution of new dwellings follows the distribution of jobs. The goal for this policy is decreasing job housing imbalance.

8.1. Projection for Policy test

We test policies by implement policy from year 2012 to year 2030. To simulate model from year 2012 to year 2030, some forecasts are used. The total population is exogenous input from medium national growth population projection from SSB[8]. The marriage rate and other parameters in population sectors keep constant after year 2011. Travel time matrix keep constant after year 2011. The policy test is based Different land use policy is tested in this model. The base run for policy test is free market simulation. The distribution of new business and new dwellings is input from Schulze (2013).

8.2. Policy test for all areas

The simulation result for three policies shown below. The first curve is base run (free market). The total commuters increase with decreasing acceleration. This behavior is mainly driven by the increase of total population.

The second curve shows the simulation result of *equal construct policy*. The total commuters also increase with decreasing acceleration. The number of commuters is always lower than base run. This means if in every construction project, offices and dwellings are built together. The total number of commuters will increase much slower than free market. Figure8.2 2 shows the increase of commuters from 2012. At the end of year 2030, with equal construct policy, there are only 67% of new commuters comparing with base run.

The third curve shows the simulation result of *aggressive zoning policy*. In this policy, dwellings are built in business areas. That means most dwellings are built in Bergen center. This policy shows an incredible result. The number of commuters nearly remains constant while the total population is increasing (Figure8.2 1). The increase of number of commuter (Figure8.2 2) is only 19% of base run.

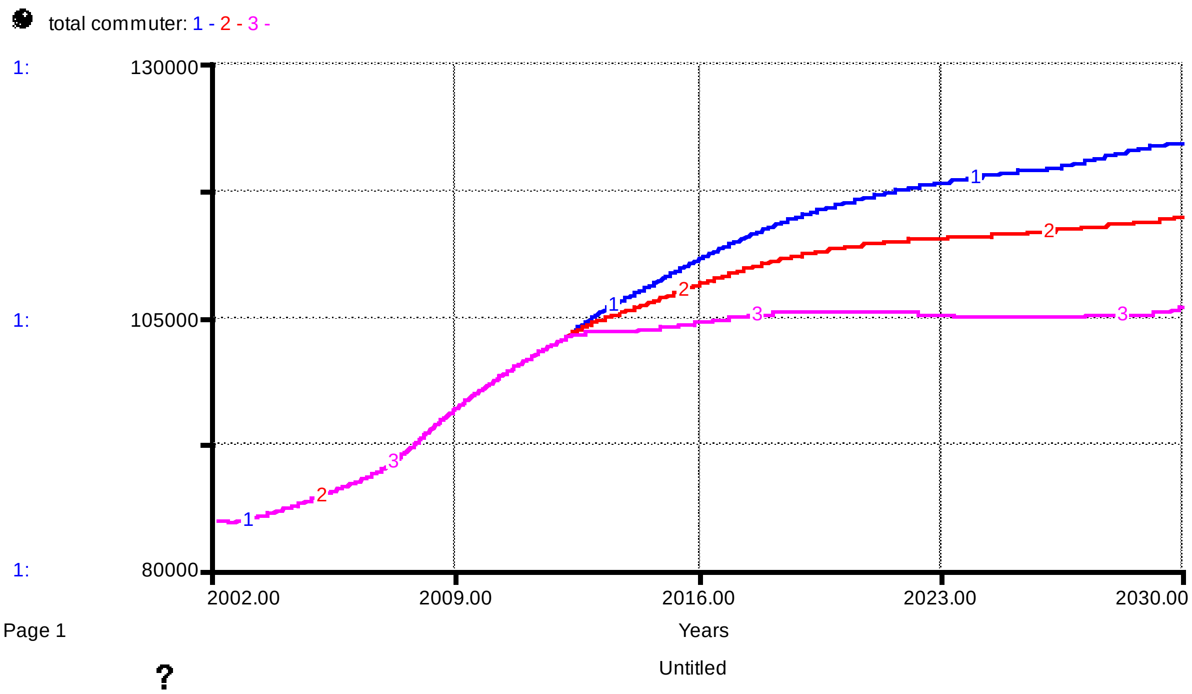


Figure8.2-1: Total number of commutes for policy in all areas.

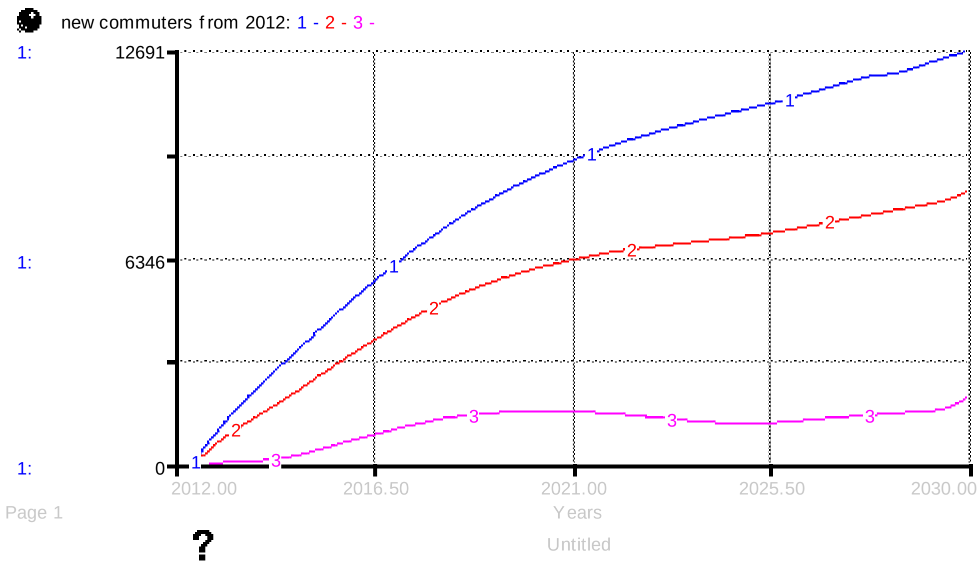


Figure8.2-2: Increase of commutes for policy in all areas.

The third policy builds most dwelling in Bergen center which reduce the job housing imbalance. The policy test shows that the increase of commuters can be reduced if job housing imbalance was reduced. But, most new business and new dwellings are placed in Bergen center in the *aggressive zoning policy*. This policy might be impossible to implement. Because Bergen center like other city center is an area with limited available land. It is not possible to continue develop city center while leave all other area undeveloped.

So, we test again the policy with a new assumption. For second and third policy, the policies only influence the land use for area other than city center. The development of Bergen center will remains as base run in policy run.

8.3. Policy test for areas other than Bergen center

The simulation results for three policies are shown below. The first curve is base run (free market). It is the same simulation with policy test for all area. Figure8.3 1 and Figure8.3 2 shows the simulation result of total commuters and new commuters from 2012. Figure8.3 3 shows the new commuters in Bergen except into centre commuters.

The second curve shows the simulation result of *simultaneously policy*. The increase of total commuters is 92% of base run (Figure8.3 2). If we only compare new commuters other than into centre commuters, it is 59% of base run (Figure8.3 3).

The third curve shows the simulation result of *aggressive policy*. The increase of number of commuter is only 94% of base run. It is 67% of base run if we count only commuters other than into centre commuters.

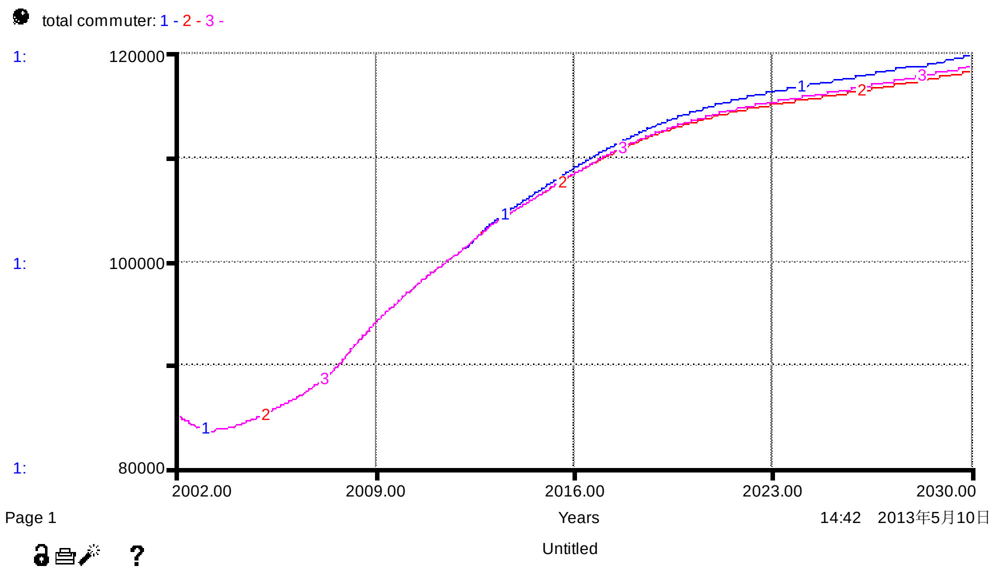


Figure8.3-1: land used in aggressive policy for areas other than Bergen center (total commuter).

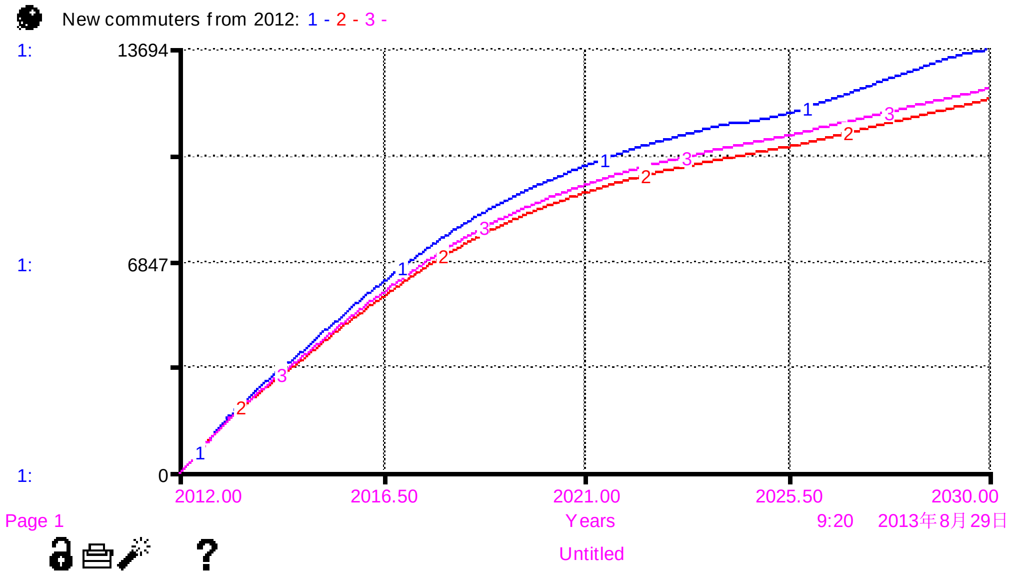


Figure8.3-2: land used in aggressive policy for areas other than Bergen center (new commuter).

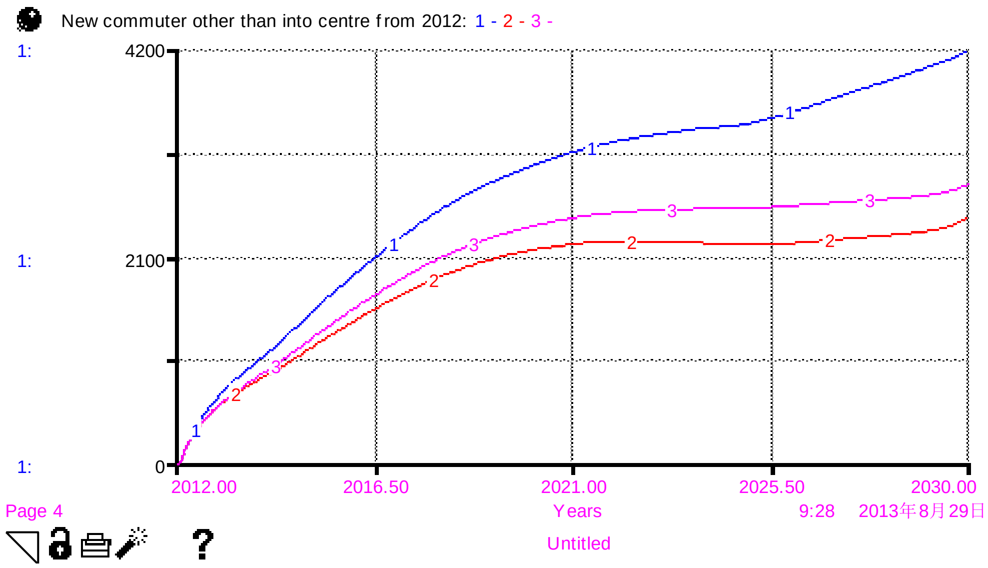


Figure8.3-3: New commuters other than into Centre for policy test in areas other than Bergen center.

The simulation shows two results.

1. The policies have very little result on number of commuters. The main reason for this is that most commuters are in-commuters for Bergen center. Without change job housing ratio for Bergen center, it is hard to change total number of commuters.
2. The *equal construct policy* has slightly better result than *aggressive zoning policy*. This means that the policy tends to decrease existing job housing imbalance have less result than the policy tends to provide balance new job and new housing. So, in next chapter, we are going to investigate the reason for these patterns.

8.4. Policy discussion

To explore the reason behind policy test phenomenon, a simplified version of model is used and shown below. This version use Åsane (Bergen north) as an example area. The area of Åsane has more dwellings than job, so most commuters for this area are out-commuters. The policy to solve the commuter problem in this area might be introducing more work place while not increasing too much residential place. On another word, new jobs should be placed in this area while new dwellings should be placed in other areas.

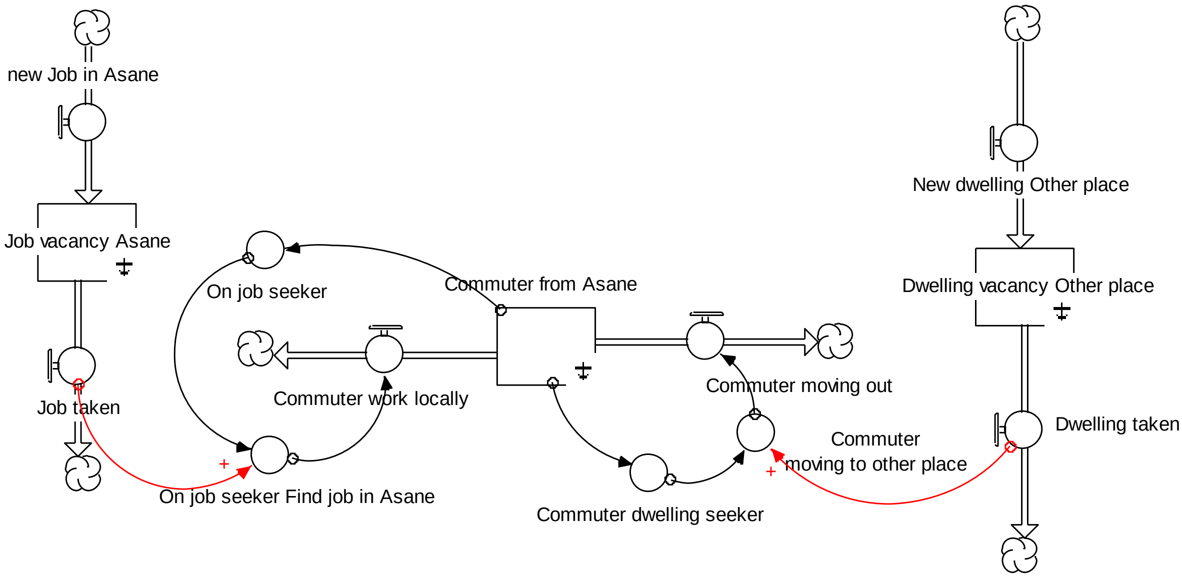


Figure8.4-1: simplified model structure for commuter Åsane.

As shown in Figure8.4 1, the number of commuter from Åsane can be reduced by commuter moving out to other place or find local job. In the process of residential relocation, out-commuters from Åsane might seek dwelling either in Åsane or in other place. When dwelling in other areas increases, out-commuters from Åsane might have better chance to find a dwelling in other areas. So, more commuters might move out. And the number of out-commuters might decrease.

The job change process has the same structure with residential relocation process. Some out-commuters from Åsane will seek new jobs. If more job vacancies are provided in local area, Out-commuters might have better chance to get local job and stop commuting. So, more local job in Åsane will decrease the number of out-commuters from Åsane.

But, with a bigger map of the model (see below), immigrations and new graduate are also participating in job seeking and dwelling seeking. The actions of increasing job in Åsane might also increase the number of new workers in Åsane. And the actions of increasing dwellings in other place might also increase chance for them to live in other area. The result is then more new workers become in-commuters into Åsane.

Since only about half of the gross entries to job is contributed by local job to job in Bergen (Stambøl 2005). The flow of new workers can surely influence the land-use policy result

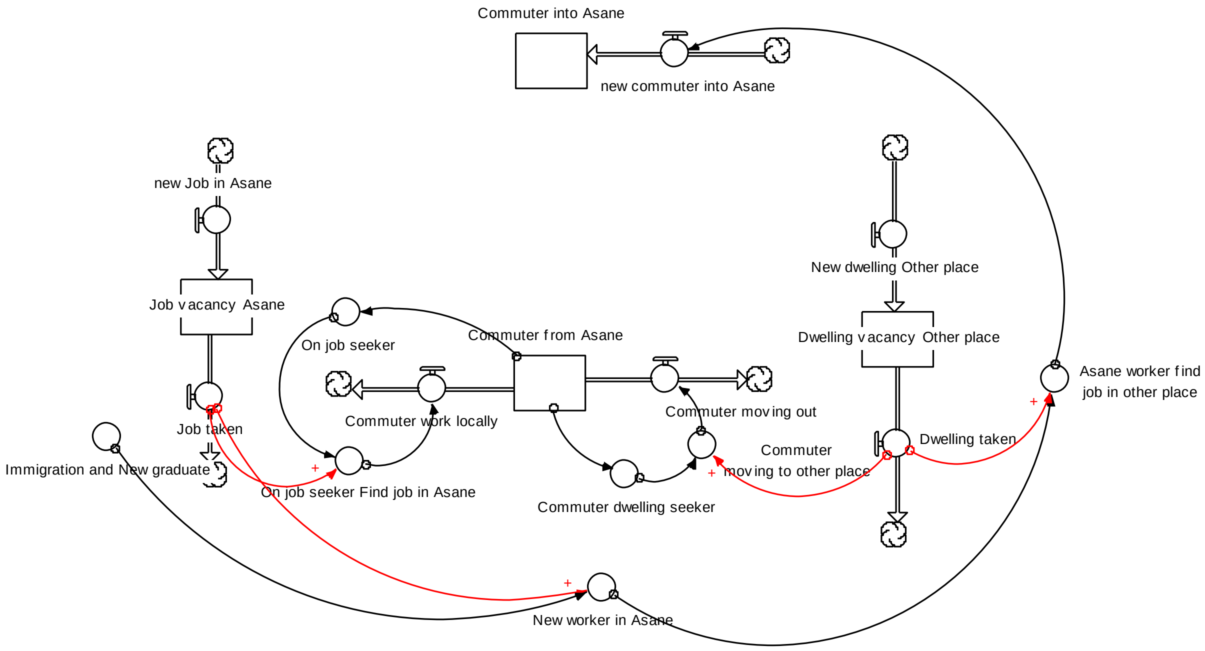


Figure8.4-2: simplified model structure for commuter Åsane.

So, the land-use actions that reduce the number out-commuters for an area might also lead to an increase of in-commuter in this area. This is the reason the *aggressive zoning policy* have little effects.

To test this result, another test is made here. Bergen north is an area with more resident than jobs. To reduce this imbalance, we assume that three thousand new jobs are placed in Bergen north in year 2015 (total job in Bergen north is sixteen thousand in year 2015). The figure below is the simulation result of this action. From the Figure8.4 3 (curve 1: base run. Curve 2 test run), it can be seen that with more jobs are available in Bergen north, more out-commuters can get job within area. So the number of out-commuters decreases. But Figure8.4 4 shows that those jobs are also taken by resident in other areas. The number of in-commuters increases in Bergen north. The total commuters (Figure8.4 5) do not change much. This simulation demonstrates the analysis above.

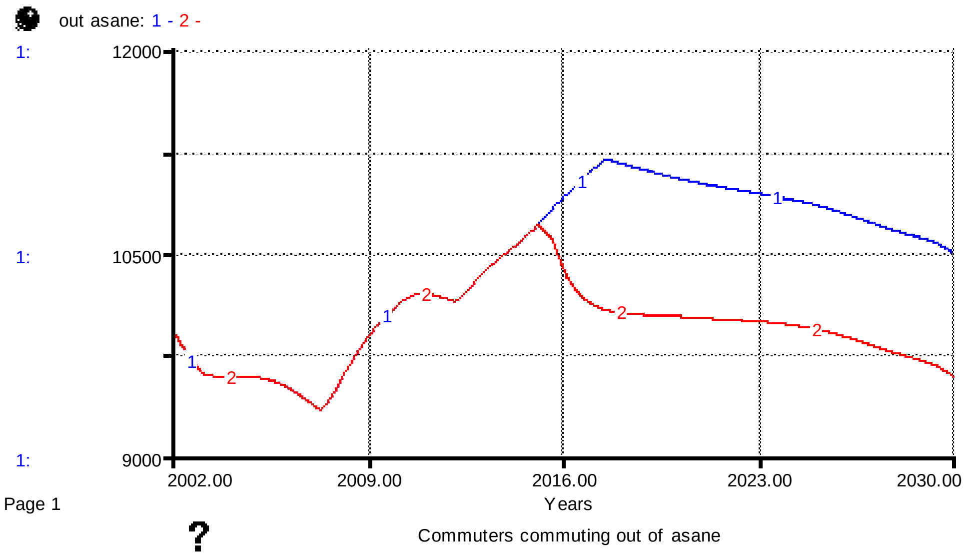


Figure8.4-3: Test for pulse new jobs in åsane. Out commuters

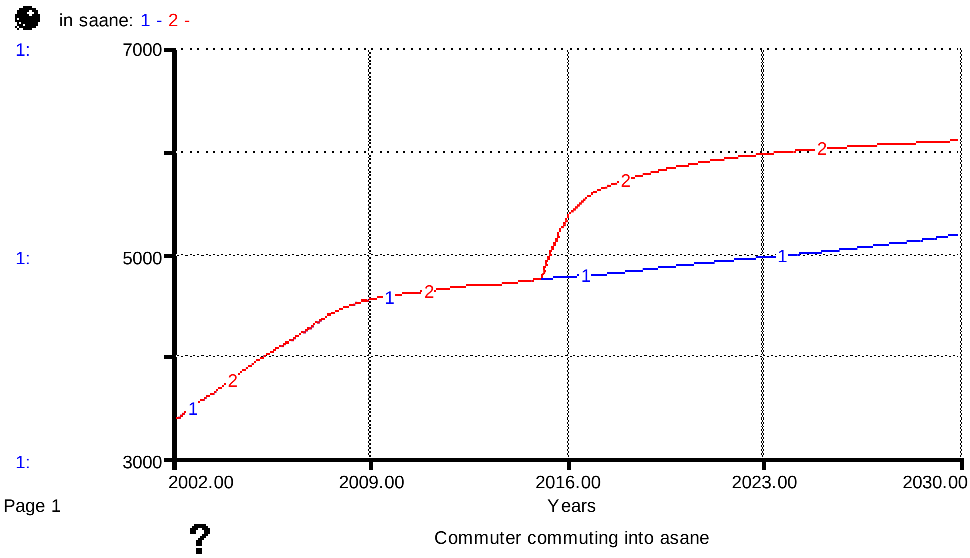


Figure8.4-4: Test for pulse new jobs in åsane. In commuters

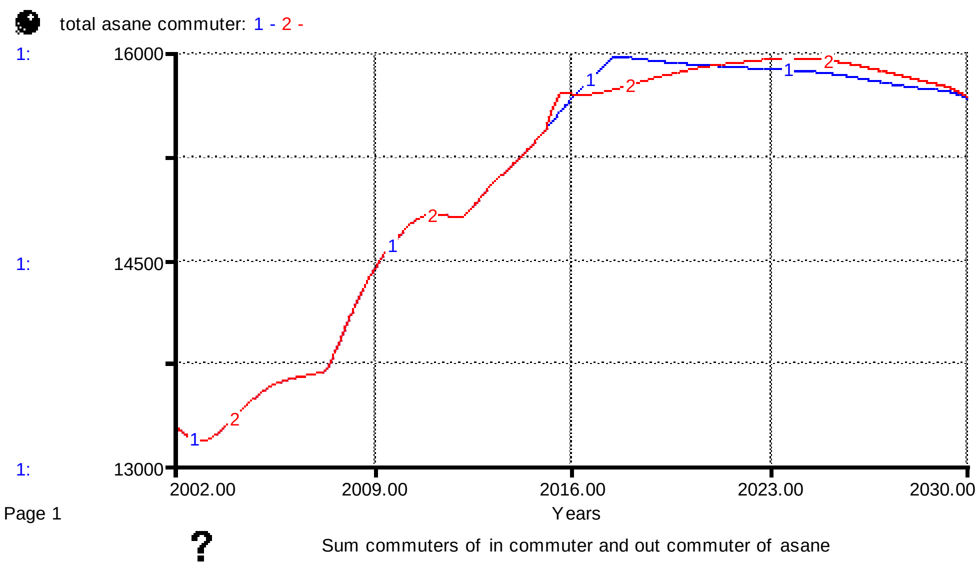


Figure8.4-5: Test for pulse new jobs in åsane. Total commuter in and out Åsane

8.5. Policy sensitively test

As mentioned above, there are many uncertainties in this model. So, sensitively test is conducted for policy test. The test can be seen in appendix. In most case, the policy result does not change. The policy result only changes on the condition that no people want to commute.

8.6. Result

Comparing the different policy test result, we can conclude some policy rules. First, in area where most commuters are in-commuters like Bergen center, increasing new dwellings might be effective to reduce the number of in-commuters. Second, in areas where most commuters are out-commuters like Bergen north, increasing new working place might not be effective to reduce the number of total commuters. The ratio of on-job seeker with first time job seeker will also influence the result of land-use policy.

Base on the test and analysis above, the result for research question can be concluded as below:

1. The land-use actions (job creation, dwelling construction) can influence the number of commuters.
2. With limited available land in city center, land-use policies have little influence on increase of commuters.

9. Conclusion

From this research, the simulation model for intra-urban migration and job change revealed the insight of commuter change. People's willingness to commute, job location distribution and dwelling distribution all donate important effect on commuter change. Moreover, travel time which is influenced by commuting traffic and road capacity also has important influence on commuter change.

The simulation and model structure shows that commuter problem has a very complex system. To reduce the number of commuters, a large number of dwellings are needed in Bergen center. So, high-density dwelling buildings are needed in Bergen center. If the land uses policy only affect construction in areas other than Bergen center. The effect will be very little. The simulation also shows that creating new work places outside Bergen center does not have strong effect on reducing total number of commuters.

This research also highlights the importance of first time buyer (imsmigrant and new graduate students). A growing urban area has enormous population increase. The first time buyers have different location choice process with relocatees. These difference might be one important fact that reducing the efficiency of land-use policy.

This research also provides some clue for further research. First, some local data collections and research experiments are necessarily needed if people want to have better understanding on commuter problem in Bergen municipality. Second, problem solvers for commuters are facing a complex dynamic system. It is highly recommend that policy makers should have a simulation model with less aggregated special level and bigger boundary. Third, the population increase might have strong impact on commuter problem. The researches on immigration and new graduate might have strong benefit for commuter problem solving.

Endnotes

1. The version of classification of urban district in this model is Urban district 2004 (Bydel 2004).
2. Data from SSB Table: 09708: Private households, by tenure status (per cent) (M) (UD)
3. Data from SSB Table: 09659: Marriages contracted (M)
4. Data from SSB Table: 09660: Divorces and separations (M)
5. Data from SSB Table: 06913: Population 1 January and population changes during the calendar year (M)
6. Data from SSB Table: 05426: Immigration, emigration and net migration (M)
7. Data from SSB Table: 07870: Number of children distributed, by age and cohort of births of woman/man (per cent))
8. Data from SSB Table: 09482: Population projections, by sex and age, in 9 variants (M) (UD)

Reference

Adams, J. S. 1969. "DIRECTIONAL BIAS IN INTRA-URBAN MIGRATION." *Economic Geography* no. 45 (4):302-323. doi: 10.2307/142667.

Brandsar, Torgeir. 2013. "The logistics of a city's human resources, A simulation model of Bergen's transportation system for strategic management of peak hour congestion Master Thesis. System Dynamics Group, Department of Geography. University of Bergen. Unpublished Manuscript.".

Brown, Lawrence A., and Eric G. Moore. 1970. "The Intra-Urban Migration Process: A Perspective." *Geografiska Annaler. Series B, Human Geography* no. 52 (1):1-13.

Case, Karl E, John M Quigley, and Robert J Shiller. 2001. "Comparing Wealth Effects: The Stock Market Versus the Housing Market."

Clark, W. A. V., M. C. Deurloo, and F. M. Dieleman. 1984. "Housing Consumption and Residential-Mobility." *Annals of the Association of American Geographers* no. 74 (1):29-43.

Clark, W. A. V., W. F. J. Van Lierop, and Nijkamp Peter. 1987. "Chapter 3 Residential mobility and household location modelling." In *Handbook of Regional and Urban Economics*, 97-132. Elsevier.

Clark, William A. V., and Suzanne Davies Withers. 1999. "Changing Jobs and Changing Houses: Mobility Outcomes of Employment Transitions." *Journal of Regional Science* no. 39 (4):653-673. doi: 10.1111/0022-4146.00154.

Clark, William A. V., Youqin Huang, and Suzanne Withers. 2003. "Does commuting distance matter?: Commuting tolerance and residential change." *Regional Science and Urban Economics* no. 33 (2):199-221.

David, McArthur, Thorsen Inge, and ersa10p. 2011. Determinants of internal migration in Norway. European Regional Science Association.

Deding, Mette, Trine Filges, and Jos Van Ommeren. 2009. "SPATIAL MOBILITY AND COMMUTING: THE CASE OF TWO-EARNER HOUSEHOLDS*." *Journal of Regional Science* no. 49 (1):113-147. doi: 10.1111/j.1467-9787.2008.00595.x.

Dieleman, FransM. 2001. "Modelling residential mobility; a review of recent trends in research." *Journal of Housing and the Built Environment* no. 16 (3-4):249-265. doi: 10.1023/A:1012515709292.

Glen Weisbrod, Moshe Ben-Akiva, Steven Lerman. 1980. "Tradeoffs in residential location decisions: Transportation versus other factors." *Transportation Policy and Decision-Making,* no. 1 (1).

Guo, Jessica Y., and Chandra R. Bhat. 2007. "Operationalizing the concept of neighborhood: Application to residential location choice analysis." *Journal of Transport Geography* no. 15 (1):31-45.

Hjorthol, Randi. 2003. *Byidealer, bostedspreferanser og aktivitetsmønstre i Oslo, Bergen og Trondheim*. Vol. 672/2003. Oslo: TØI.

Hooimeijer, Pieter, and Anton Oskamp. 1996. "A simulation model of residential mobility and housing choice." *Journal of Housing and the Built Environment* no. 11 (3):313-336. doi: 10.1007/bf02496594.

Hunt, John D, David S Kriger, and EJ Miller. 2005. "Current operational urban land ‐ use – transport modelling frameworks: A review." *Transport Reviews* no. 25 (3):329-376.

Huu Phe, Hoang, and Patrick Wakely. 2000. "Status, Quality and the Other Trade-off: Towards a New Theory of Urban Residential Location." *Urban Studies* no. 37 (1):7-35. doi: 10.1080/0042098002276.
Jackman, R., and S. Savouri. 1992. "REGIONAL MIGRATION IN BRITAIN - AN ANALYSIS OF GROSS FLOWS USING NHS CENTRAL REGISTER DATA." *Economic Journal* no. 102 (415):1433-1450. doi: 10.2307/2234799.
Kiel, Katherine A. 1994. "The impact of house price appreciation on household mobility." *Journal of Housing Economics* no. 3 (2):92-108.
Kim, J. H., F. Pagliara, and J. Preston. 2005. "The intention to move and residential location choice behaviour." *Urban Studies* no. 42 (9):1621-1636. doi: Doi 10.1080/00420980500185611.
Klette, Tor Jakob, and Astrid Mathiassen. 1996. "Job Creation, Job Destruction and Plant Turnover in Norvegian Manufacturing." *Annals of Economics and Statistics / Annales d'Économie et de Statistique* (41/42):97-125.
Krainer, John. 2001. "A theory of liquidity in residential real estate markets." *Journal of Urban Economics* no. 49 (1):32-53.
Li, pei, and Yong Tu. 2011. "behaviors on intra urban residential mobility a review and implications to the future research."
McMillen, Daniel P., and Stefani C. Smith. 2003. "The number of subcenters in large urban areas." *Journal of Urban Economics* no. 53 (3):321-338.
Meland, Solveig. 2009. "Reisevaneundersøkelse for Bergensområdet 2008 Datagrunnlag og hovedresultater."
Mulder, C. H., and M. Wagner. 1993. "Migration and Marriage in the Life-Course - a Method for Studying Synchronized Events." *European Journal of Population-Revue Europeenne De Demographie* no. 9 (1):55-76.
Mulder, Clara. 1996. "Housing choice: Assumptions and approaches." *Journal of Housing and the Built Environment* no. 11 (3):209-232. doi: 10.1007/bf02496589.
Nebiyou, Tilahun, and Levinson David. 2010. An Agent-Based Model of Worker and Job Matching. University of Minnesota: Nexus Research Group.
Pagliara, Francesca, John Preston, David Simmonds, and J. D. Hunt. 2010. "Stated Preference Examination of Factors Influencing Residential Attraction." In *Residential Location Choice*, 21-59. Springer Berlin Heidelberg.
Pagliara, Francesca, and Alan Wilson. 2010. "The State-of-the-Art in Building Residential Location Models." In *Residential Location Choice*, edited by Francesca Pagliara, John Preston and David Simmonds, 1-20. Springer Berlin Heidelberg.
Rees, Philip. 1999. *Internal migration and regional population dynamics in Europe : Norway case study*. Vol. 99/19. Oslo: SSB.
Rogerson, Peter, and Ross D. MacKinnon. 1981. "A GEOGRAPHICAL MODEL OF JOB SEARCH, MIGRATION AND UNEMPLOYMENT." *Papers in Regional Science* no. 48 (1):89-102. doi: 10.1111/j.1435-5597.1981.tb01148.x.
Rossi, P. H. 1980(1955). *Why families move*: Sage Publications.
Rouwendal, Jan. 1998. "Search Theory, Spatial Labor Markets, and Commuting." *Journal of Urban Economics* no. 43 (1):1-22. doi: http://dx.doi.org/10.1006/juec.1996.2033.
Rouwendal, Jan. 1999. "Spatial job search and commuting distances." *Regional Science and Urban Economics* no. 29 (4):491-517. doi: http://dx.doi.org/10.1016/S0166-0462(99)00002-2.

Schulze, Carolin. 2013. "Land use in the Bergen Region, A System Dynamics Analysis of the Jobs-Housing Imbalance. Master Thesis. System Dynamics Group, Department of Geography. University of Bergen. Unpublished Manuscript.".

Skiri, Halvard. 1995. "Role and Statue of Civil Registration (Population Registration) and Vital Statistics Systems in Norway."

Smith, Terence R., W. A. V. Clark, James O. Huff, and Perry Shapiro. 1979. "A Decision-Making and Search Model For Intraurban Migration." *Geographical Analysis* no. 11 (1):1-22. doi: 10.1111/j.1538-4632.1979.tb00669.x.

Speare, Alden. 1970. "Home ownership, life cycle stage, and residential mobility." *Demography* no. 7 (4):449-458. doi: 10.2307/2060237.

Speare, Alden. 1974. "Residential satisfaction as an intervening variable in residential mobility." *Demography* no. 11 (2):173-188. doi: 10.2307/2060556.

Stambøl, Lasse Sigbjørn. 2005. "Urban and Regional Labour Market Mobility in Norway."

Sterman, John. 2000. *Business dynamics : systems thinking and modeling for a complex world*. Boston: Irwin/McGraw-Hill.

Strand, Arvid, Petter Christiansen, and Øystein Engebretsen. 2010. "Bybanen i Bergen - Førundersøkelse av arbeidspendling og reisevaner."

Van den Berg, Gerard J., and Cees Gorter. 1997. "Job Search and Commuting Time." *Journal of Business & Economic Statistics* no. 15 (2):269-281. doi: 10.1080/07350015.1997.10524705.

van Ommeren, Jos, Piet Rietveld, and Peter Nijkamp. 1997. "Commuting: In Search of Jobs and Residences." *Journal of Urban Economics* no. 42 (3):402-421. doi: http://dx.doi.org/10.1006/juec.1996.2029.

van Ommeren, Jos, Piet Rietveld, and Peter Nijkamp. 1999. "Job Moving, Residential Moving, and Commuting: A Search Perspective." *Journal of Urban Economics* no. 46 (2):230-253. doi: http://dx.doi.org/10.1006/juec.1998.2120.

van Ommeren, Jos, Piet Rietveld, and Peter Nijkamp. 2000. "Job mobility, residential mobility and commuting: A theoretical analysis using search theory." *The Annals of Regional Science* no. 34 (2):213-232. doi: 10.1007/s001689900004.

Vegvesen, Statens. 2007. "Transportanalyse for Bergensområdet ".

Vegvesen, Statens. 2011. ""Kjuagutt og stril - mindre bil" Konseptvalgutredning(KVU) for transportsystemet i Bergensområdet."

Waddell, P. 2000. "A behavioral simulation model for metropolitan policy analysis and planning: residential location and housing market components of UrbanSim." *Environment and Planning B: Planning and Design* no. 27 (2):247-263.

Weinberg, Daniel H., Joseph Friedman, and Stephen K. Mayo. 1981. "Intraurban residential mobility: The role of transactions costs, market imperfections, and household disequilibrium." *Journal of Urban Economics* no. 9 (3):332-348. doi: http://dx.doi.org/10.1016/0094-1190(81)90031-0.

Appendix 1. Reference mode

Year 2002	Askøy	center bergen	south bergen	norh bergen	east bergen	west bergen
askoy	4872	2226	761	125	23	1345
center bergen	91	21082	4601	1493	379	2971
South bergen	67	10052	10273	496	357	2213
North Bergen	33	6941	1444	5444	379	1090
East Bergen	6	1855	694	507	1674	181
West Bergen	240	13050	4461	736	136	7885

Year 2008	askøy	center bergen	south bergen	norh bergen	east bergen	west bergen
askoy	5605	2747	997	205	41	1835
center bergen	157	23951	5100	1703	314	2930
South bergen	131	10900	13029	661	337	2444
North Bergen	72	7622	1603	6478	443	1355
East Bergen	15	1606	750	619	2025	280
West Bergen	477	13353	4847	996	198	8917

The commuter number of calculated from *(Strand, Christiansen, and Engebretsen 2010) and (Meland 2009)* .

Appendix 2. Travel time

Unit: hours Years	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
car[BergenC to BergenN]	0.17	0.32	0.37	0.40	0.41	0.41	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42
car[BergenC to BergenE]	0.36	0.51	0.56	0.58	0.59	0.60	0.60	0.60	0.61	0.61	0.61	0.61	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60
car[BergenC to BergenS]	0.12	0.27	0.32	0.35	0.36	0.36	0.37	0.37	0.37	0.38	0.38	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37
car[BergenC to BergenW]	0.12	0.27	0.32	0.34	0.35	0.36	0.36	0.37	0.37	0.37	0.37	0.37	0.37	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36
car[BergenC to Askoy]	0.19	0.33	0.39	0.41	0.42	0.43	0.43	0.43	0.43	0.44	0.44	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43
car[BergenN to BergenC]	0.17	0.33	0.39	0.43	0.44	0.45	0.46	0.47	0.49	0.51	0.52	0.53	0.54	0.54	0.55	0.55	0.55	0.55	0.55	0.55	0.55
car[BergenN to BergenE]	0.19	0.33	0.39	0.41	0.42	0.43	0.43	0.43	0.43	0.44	0.44	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43
car[BergenN to BergenS]	0.29	0.45	0.52	0.55	0.57	0.58	0.59	0.60	0.62	0.63	0.65	0.66	0.67	0.67	0.67	0.68	0.68	0.68	0.68	0.68	0.68
car[BergenN to BergenW]	0.29	0.45	0.51	0.55	0.56	0.58	0.58	0.59	0.61	0.63	0.64	0.65	0.66	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.68
car[BergenN to Askoy]	0.36	0.52	0.58	0.61	0.63	0.64	0.65	0.66	0.68	0.69	0.71	0.72	0.73	0.73	0.74	0.74	0.74	0.74	0.74	0.74	0.74
car[BergenE to BergenC]	0.36	0.52	0.58	0.61	0.63	0.64	0.65	0.66	0.68	0.69	0.71	0.72	0.73	0.73	0.74	0.74	0.74	0.74	0.74	0.74	0.74
car[BergenE to BergenN]	0.19	0.33	0.39	0.41	0.42	0.43	0.43	0.43	0.44	0.44	0.44	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43
car[BergenE to BergenS]	0.25	0.40	0.45	0.48	0.49	0.49	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
car[BergenE to BergenW]	0.46	0.61	0.66	0.69	0.70	0.70	0.71	0.71	0.71	0.71	0.71	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70
car[BergenE to Askoy]	0.53	0.68	0.74	0.76	0.77	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77
car[BergenS to BergenC]	0.12	0.27	0.32	0.35	0.36	0.37	0.37	0.37	0.38	0.38	0.38	0.38	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37
car[BergenS to BergenN]	0.29	0.44	0.50	0.52	0.53	0.54	0.54	0.55	0.55	0.56	0.56	0.55	0.55	0.55	0.54	0.54	0.54	0.54	0.54	0.54	0.54
car[BergenS to BergenE]	0.25	0.40	0.45	0.48	0.49	0.49	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
car[BergenS to BergenW]	0.24	0.39	0.44	0.47	0.48	0.49	0.49	0.49	0.50	0.50	0.51	0.50	0.50	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49
car[BergenS to Askoy]	0.28	0.42	0.48	0.50	0.51	0.52	0.52	0.52	0.52	0.53	0.53	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52
car[BergenW to BergenC]	0.12	0.43	0.56	0.60	0.62	0.61	0.60	0.60	0.62	0.63	0.64	0.64	0.64	0.64	0.63	0.63	0.62	0.61	0.61	0.61	0.61
car[BergenW to BergenN]	0.29	0.60	0.73	0.78	0.79	0.79	0.78	0.78	0.79	0.80	0.81	0.81	0.81	0.81	0.81	0.80	0.79	0.79	0.79	0.78	0.78
car[BergenW to BergenE]	0.46	0.61	0.66	0.68	0.69	0.69	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70
car[BergenW to BergenS]	0.24	0.55	0.68	0.73	0.74	0.74	0.73	0.73	0.74	0.76	0.77	0.77	0.77	0.77	0.76	0.75	0.75	0.74	0.74	0.74	0.74
car[BergenW to Askoy]	0.07	0.21	0.27	0.29	0.30	0.31	0.31	0.31	0.31	0.32	0.32	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31
car[Askoy to BergenC]	0.19	0.50	0.62	0.67	0.69	0.68	0.67	0.67	0.69	0.71	0.72	0.72	0.73	0.73	0.72	0.71	0.71	0.70	0.70	0.70	0.70
car[Askoy to BergenN]	0.36	0.67	0.80	0.85	0.86	0.86	0.85	0.85	0.86	0.88	0.89	0.90	0.90	0.90	0.89	0.89	0.88	0.88	0.87	0.87	0.87
car[Askoy to BergenE]	0.53	0.68	0.73	0.75	0.76	0.76	0.77	0.77	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.79
car[Askoy to BergenS]	0.28	0.42	0.47	0.49	0.50	0.51	0.51	0.52	0.52	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53
car[Askoy to BergenW]	0.07	0.21	0.27	0.29	0.30	0.31	0.31	0.32	0.32	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33

Unit: hours Years	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
PT[BergenC to BergenN]	0.67	0.64	0.62	0.61	0.61	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60
PT[BergenC to BergenE]	0.90	0.86	0.83	0.82	0.81	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80
PT[BergenC to BergenS]	0.60	0.58	0.58	0.57	0.57	0.57	0.57	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56
PT[BergenC to BergenW]	0.60	0.58	0.57	0.57	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56
PT[BergenC to Askoy]	0.68	0.66	0.64	0.64	0.63	0.63	0.63	0.63	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62
PT[BergenN to BergenC]	0.69	0.67	0.65	0.65	0.65	0.65	0.65	0.65	0.66	0.67	0.68	0.69	0.69	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.71
PT[BergenN to BergenE]	0.68	0.67	0.66	0.66	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65
PT[BergenN to BergenS]	0.84	0.80	0.78	0.77	0.76	0.76	0.76	0.77	0.77	0.78	0.79	0.80	0.81	0.81	0.81	0.81	0.81	0.82	0.82	0.82	0.82
PT[BergenN to BergenW]	0.84	0.80	0.77	0.76	0.76	0.76	0.76	0.76	0.77	0.78	0.79	0.79	0.80	0.80	0.81	0.81	0.81	0.81	0.81	0.81	0.81
PT[BergenN to Askoy]	0.92	0.87	0.85	0.83	0.83	0.82	0.82	0.83	0.83	0.84	0.85	0.86	0.87	0.87	0.87	0.87	0.88	0.88	0.88	0.88	0.88
PT[BergenE to BergenC]	0.92	0.88	0.86	0.86	0.85	0.85	0.85	0.85	0.86	0.87	0.88	0.89	0.89	0.90	0.90	0.90	0.90	0.90	0.91	0.91	0.91
PT[BergenE to BergenN]	0.68	0.67	0.66	0.66	0.66	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65
PT[BergenE to BergenS]	0.77	0.73	0.70	0.68	0.68	0.67	0.67	0.67	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
PT[BergenE to BergenW]	1.02	0.94	0.90	0.87	0.85	0.84	0.84	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.82	0.82	0.82	0.82	0.82	0.82
PT[BergenE to Askoy]	1.10	1.02	0.97	0.94	0.92	0.91	0.90	0.90	0.90	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89
PT[BergenS to BergenC]	0.60	0.59	0.58	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57
PT[BergenS to BergenN]	0.82	0.78	0.75	0.73	0.73	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71
PT[BergenS to BergenE]	0.77	0.73	0.70	0.68	0.68	0.67	0.67	0.67	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
PT[BergenS to BergenW]	0.42	0.39	0.37	0.36	0.35	0.35	0.34	0.34	0.34	0.35	0.35	0.35	0.35	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36
PT[BergenS to Askoy]	0.79	0.75	0.72	0.71	0.70	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69
PT[BergenW to BergenC]	0.85	0.80	0.76	0.74	0.73	0.72	0.71	0.72	0.73	0.74	0.74	0.74	0.74	0.73	0.72	0.71	0.70	0.70	0.69	0.68	0.68
PT[BergenW to BergenN]	1.07	0.99	0.93	0.90	0.88	0.87	0.86	0.86	0.87	0.88	0.89	0.89	0.89	0.88	0.87	0.86	0.85	0.84	0.84	0.83	0.83
PT[BergenW to BergenE]	1.04	0.95	0.90	0.87	0.85	0.84	0.84	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83
PT[BergenW to BergenS]	0.51	0.41	0.35	0.32	0.30	0.29	0.28	0.28	0.28	0.27	0.27	0.27	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26
PT[BergenW to Askoy]	0.53	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52
PT[Askoy to BergenC]	0.94	0.88	0.84	0.82	0.80	0.80	0.79	0.79	0.81	0.82	0.83	0.83	0.83	0.83	0.82	0.81	0.80	0.79	0.79	0.78	0.78
PT[Askoy to BergenN]	1.15	1.06	1.01	0.98	0.96	0.95	0.94	0.94	0.96	0.97	0.98	0.98	0.98	0.98	0.97	0.96	0.95	0.94	0.93	0.93	0.92
PT[Askoy to BergenE]	1.12	1.03	0.97	0.94	0.92	0.91	0.91	0.91	0.91	0.92	0.92	0.92	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93
PT[Askoy to BergenS]	0.82	0.76	0.73	0.71	0.70	0.69	0.69	0.69	0.70	0.70	0.71	0.71	0.71	0.71	0.72	0.71	0.71	0.71	0.71	0.71	0.72
PT[Askoy to BergenW]	0.53	0.53	0.53	0.52	0.52	0.52	0.52	0.53	0.53	0.54	0.54	0.54	0.55	0.55	0.55	0.55	0.55	0.55	0.55	0.55	0.55

These travel time tables are input from transportation model made by Torgeir Brandsar

Transportation model did not provided travel time within area. So, travel time without highway travel time is used to calculate travel time within areas.

Car travel time unit:hours Years	2000.00	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
WITHOUT HIGHWAY[BergenC to BergenN]	0.00	0.15	0.20	0.22	0.23	0.24	0.24	0.24	0.25	0.25	0.25	0.25	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24
WITHOUT HIGHWAY[BergenC to BergenE]	0.00	0.15	0.20	0.22	0.23	0.24	0.24	0.24	0.25	0.25	0.25	0.25	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24
WITHOUT HIGHWAY[BergenC to BergenS]	0.00	0.15	0.20	0.22	0.23	0.24	0.24	0.24	0.25	0.25	0.25	0.25	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24
WITHOUT HIGHWAY[BergenC to BergenW]	0.00	0.15	0.20	0.22	0.23	0.24	0.24	0.24	0.25	0.25	0.25	0.25	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24
WITHOUT HIGHWAY[BergenC to Askoy]	0.00	0.15	0.20	0.22	0.23	0.24	0.24	0.24	0.25	0.25	0.25	0.25	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24
WITHOUT HIGHWAY[BergenN to BergenC]	-0.03	0.13	0.19	0.22	0.23	0.24	0.24	0.24	0.25	0.25	0.25	0.25	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24
WITHOUT HIGHWAY[BergenN to BergenE]	0.00	0.15	0.20	0.22	0.23	0.24	0.24	0.24	0.25	0.25	0.25	0.25	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24
WITHOUT HIGHWAY[BergenN to BergenS]	-0.03	0.13	0.19	0.22	0.23	0.24	0.24	0.24	0.25	0.25	0.25	0.25	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24
WITHOUT HIGHWAY[BergenN to BergenW]	-0.03	0.13	0.19	0.22	0.23	0.24	0.24	0.24	0.25	0.25	0.25	0.25	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24
WITHOUT HIGHWAY[BergenN to Askoy]	-0.03	0.13	0.19	0.22	0.23	0.24	0.24	0.24	0.25	0.25	0.25	0.25	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24
WITHOUT HIGHWAY[BergenE to BergenC]	-0.03	0.13	0.19	0.22	0.23	0.24	0.24	0.24	0.25	0.25	0.25	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24
WITHOUT HIGHWAY[BergenE to BergenN]	0.00	0.15	0.20	0.22	0.23	0.24	0.24	0.24	0.25	0.25	0.25	0.25	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24
WITHOUT HIGHWAY[BergenE to BergenS]	0.00	0.15	0.20	0.22	0.23	0.24	0.24	0.24	0.25	0.25	0.25	0.25	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24
WITHOUT HIGHWAY[BergenE to BergenW]	0.03	0.19	0.24	0.27	0.29	0.30	0.30	0.31	0.31	0.32	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.34	0.34	0.34	0.34
WITHOUT HIGHWAY[BergenE to Askoy]	0.03	0.19	0.24	0.27	0.29	0.30	0.30	0.31	0.31	0.32	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.34	0.34	0.34	0.34
WITHOUT HIGHWAY[BergenS to BergenC]	0.00	0.15	0.20	0.22	0.23	0.24	0.24	0.24	0.25	0.25	0.25	0.25	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24
WITHOUT HIGHWAY[BergenS to BergenN]	0.00	0.15	0.20	0.22	0.23	0.24	0.24	0.24	0.25	0.25	0.25	0.25	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24
WITHOUT HIGHWAY[BergenS to BergenE]	0.00	0.15	0.20	0.22	0.23	0.24	0.24	0.24	0.25	0.25	0.25	0.25	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24
WITHOUT HIGHWAY[BergenS to BergenW]	0.11	0.25	0.30	0.33	0.34	0.34	0.34	0.34	0.35	0.35	0.35	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34	0.34
WITHOUT HIGHWAY[BergenS to Askoy]	0.02	0.17	0.22	0.25	0.26	0.26	0.26	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27
WITHOUT HIGHWAY[BergenW to BergenC]	-0.16	0.05	0.14	0.19	0.21	0.23	0.23	0.24	0.24	0.25	0.25	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24
WITHOUT HIGHWAY[BergenW to BergenN]	-0.16	0.05	0.14	0.19	0.21	0.23	0.23	0.24	0.24	0.25	0.25	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24
WITHOUT HIGHWAY[BergenW to BergenE]	0.02	0.20	0.28	0.32	0.34	0.35	0.35	0.36	0.37	0.37	0.38	0.37	0.37	0.37	0.36	0.36	0.35	0.35	0.35	0.34	0.34
WITHOUT HIGHWAY[BergenW to BergenS]	0.09	0.28	0.35	0.39	0.41	0.42	0.43	0.44	0.45	0.45	0.46	0.46	0.46	0.45	0.45	0.44	0.44	0.44	0.43	0.43	0.43
WITHOUT HIGHWAY[BergenW to Askoy]	0.00	0.15	0.20	0.22	0.23	0.24	0.24	0.24	0.25	0.25	0.25	0.25	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24
WITHOUT HIGHWAY[Askoy to BergenC]	-0.17	0.05	0.14	0.19	0.21	0.23	0.23	0.24	0.24	0.25	0.25	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24
WITHOUT HIGHWAY[Askoy to BergenN]	-0.17	0.05	0.14	0.19	0.21	0.23	0.23	0.24	0.24	0.25	0.25	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24
WITHOUT HIGHWAY[Askoy to BergenE]	0.01	0.20	0.28	0.31	0.33	0.35	0.35	0.36	0.37	0.37	0.38	0.37	0.37	0.37	0.36	0.36	0.35	0.35	0.35	0.34	0.34
WITHOUT HIGHWAY[Askoy to BergenS]	-0.01	0.18	0.26	0.30	0.32	0.33	0.34	0.34	0.35	0.36	0.36	0.36	0.36	0.36	0.35	0.35	0.34	0.34	0.34	0.33	0.33
WITHOUT HIGHWAY[Askoy to BergenW]	-0.01	0.14	0.20	0.22	0.23	0.24	0.24	0.24	0.25	0.25	0.25	0.25	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24

PT without highway	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
[BergenC to BergenN]	0.52	0.49	0.48	0.46	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
[BergenC to BergenE]	0.55	0.51	0.49	0.47	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
[BergenC to BergenS]	0.49	0.47	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
[BergenC to BergenW]	0.49	0.48	0.46	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
[BergenC to Askoy]	0.51	0.48	0.47	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
[BergenN to BergenC]	0.51	0.49	0.47	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
[BergenN to BergenE]	0.48	0.47	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
[BergenN to BergenS]	0.55	0.51	0.48	0.47	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
[BergenN to BergenW]	0.55	0.51	0.49	0.47	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
[BergenN to Askoy]	0.57	0.52	0.49	0.47	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
[BergenE to BergenC]	0.54	0.50	0.48	0.47	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
[BergenE to BergenN]	0.48	0.47	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
[BergenE to BergenS]	0.56	0.51	0.49	0.47	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
[BergenE to BergenW]	0.64	0.56	0.51	0.49	0.47	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
[BergenE to Askoy]	0.65	0.57	0.52	0.49	0.47	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
[BergenS to BergenC]	0.48	0.47	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
[BergenS to BergenN]	0.56	0.51	0.49	0.47	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
[BergenS to BergenE]	0.56	0.51	0.49	0.47	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
[BergenS to BergenW]	0.34	0.31	0.28	0.27	0.26	0.26	0.25	0.25	0.25	0.25	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26
[BergenS to Askoy]	0.55	0.51	0.49	0.47	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
[BergenW to BergenC]	0.58	0.53	0.49	0.48	0.47	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
[BergenW to BergenN]	0.65	0.57	0.52	0.49	0.47	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
[BergenW to BergenE]	0.65	0.56	0.51	0.48	0.47	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
[BergenW to BergenS]	0.41	0.31	0.26	0.24	0.22	0.21	0.21	0.20	0.20	0.20	0.20	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19	0.19
[BergenW to Askoy]	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
[Askoy to BergenC]	0.60	0.54	0.50	0.48	0.47	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
[Askoy to BergenN]	0.66	0.57	0.52	0.49	0.47	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
[Askoy to BergenE]	0.66	0.56	0.51	0.48	0.47	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
[Askoy to BergenS]	0.56	0.50	0.47	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
[Askoy to BergenW]	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45

The travel time within areas is calculated as the mean travel time without highway. So the input for travel time is shown below.

Travel time	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
A car to A	-0.07	0.12	0.20	0.24	0.26	0.27	0.28	0.28	0.29	0.29	0.30	0.29	0.29	0.29	0.29	0.29	0.28	0.28	0.28	0.28	0.28
A car to C	0.19	0.50	0.62	0.67	0.69	0.68	0.67	0.67	0.69	0.71	0.72	0.72	0.73	0.73	0.72	0.71	0.71	0.70	0.70	0.70	0.70
A car to S	0.28	0.42	0.47	0.49	0.50	0.51	0.51	0.52	0.52	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53	0.53
A car to N	0.36	0.67	0.80	0.85	0.86	0.86	0.85	0.85	0.86	0.88	0.89	0.90	0.90	0.90	0.89	0.89	0.88	0.88	0.87	0.87	0.87
A car to E	0.53	0.68	0.73	0.75	0.76	0.76	0.77	0.77	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.79
A car to W	0.07	0.21	0.27	0.29	0.30	0.31	0.31	0.32	0.32	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
C car to A	0.19	0.33	0.39	0.41	0.42	0.43	0.43	0.43	0.43	0.44	0.44	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43
C car to C	0.00	0.15	0.20	0.22	0.23	0.24	0.24	0.24	0.25	0.25	0.25	0.25	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24
C car to S	0.12	0.27	0.32	0.35	0.36	0.36	0.37	0.37	0.37	0.38	0.38	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37
C car to N	0.17	0.32	0.37	0.40	0.41	0.41	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42	0.42
C car to E	0.36	0.51	0.56	0.58	0.59	0.60	0.60	0.60	0.61	0.61	0.61	0.61	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60
C car to W	0.12	0.27	0.32	0.34	0.35	0.36	0.36	0.37	0.37	0.37	0.37	0.37	0.37	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36
S car to A	0.28	0.42	0.48	0.50	0.51	0.52	0.52	0.52	0.52	0.53	0.53	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52
S car to C	0.12	0.27	0.32	0.35	0.36	0.37	0.37	0.37	0.38	0.38	0.38	0.38	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37
S car to S	0.02	0.17	0.22	0.25	0.26	0.26	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27	0.27
S car to N	0.29	0.44	0.50	0.52	0.53	0.54	0.54	0.55	0.55	0.56	0.56	0.55	0.55	0.55	0.54	0.54	0.54	0.54	0.54	0.54	0.54
S car to E	0.25	0.40	0.45	0.48	0.49	0.49	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
S car to W	0.24	0.39	0.44	0.47	0.48	0.49	0.49	0.49	0.50	0.50	0.51	0.50	0.50	0.49	0.49	0.49	0.49	0.49	0.49	0.49	0.49
N car to A	0.36	0.52	0.58	0.61	0.63	0.64	0.65	0.66	0.68	0.69	0.71	0.72	0.73	0.73	0.74	0.74	0.74	0.74	0.74	0.74	0.74
N car to C	0.17	0.33	0.39	0.43	0.44	0.45	0.46	0.47	0.49	0.51	0.52	0.53	0.54	0.54	0.55	0.55	0.55	0.55	0.55	0.55	0.55
N car to S	0.29	0.45	0.52	0.55	0.57	0.58	0.59	0.60	0.62	0.63	0.65	0.66	0.67	0.67	0.67	0.68	0.68	0.68	0.68	0.68	0.68
N car to N	-0.03	0.13	0.19	0.22	0.23	0.24	0.24	0.24	0.25	0.25	0.25	0.25	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24	0.24
N car to E	0.19	0.33	0.39	0.41	0.42	0.43	0.43	0.43	0.43	0.44	0.44	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43
N car to W	0.29	0.45	0.51	0.55	0.56	0.58	0.58	0.59	0.61	0.63	0.64	0.65	0.66	0.67	0.67	0.67	0.67	0.67	0.67	0.67	0.68
E car to A	0.53	0.68	0.74	0.76	0.77	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.77	0.77	0.77	0.77	0.77	0.77	0.77	0.77
E car to C	0.36	0.52	0.58	0.61	0.63	0.64	0.65	0.66	0.68	0.69	0.71	0.72	0.73	0.73	0.74	0.74	0.74	0.74	0.74	0.74	0.74
E car to S	0.25	0.40	0.45	0.48	0.49	0.49	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
E car to N	0.19	0.33	0.39	0.41	0.42	0.43	0.43	0.43	0.44	0.44	0.44	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43	0.43
E car to E	0.00	0.16	0.22	0.24	0.25	0.26	0.26	0.27	0.27	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28	0.28
E car to W	0.46	0.61	0.66	0.69	0.70	0.70	0.71	0.71	0.71	0.71	0.71	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70
W car to A	0.07	0.21	0.27	0.29	0.30	0.31	0.31	0.31	0.31	0.32	0.32	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31
W car to C	0.12	0.43	0.56	0.60	0.62	0.61	0.60	0.60	0.62	0.63	0.64	0.64	0.64	0.64	0.63	0.63	0.62	0.61	0.61	0.61	0.61
W car to S	0.24	0.55	0.68	0.73	0.74	0.74	0.73	0.73	0.74	0.76	0.77	0.77	0.77	0.77	0.76	0.75	0.75	0.74	0.74	0.74	0.74
W car to N	0.29	0.60	0.73	0.78	0.79	0.79	0.78	0.78	0.79	0.80	0.81	0.81	0.81	0.81	0.81	0.80	0.79	0.79	0.79	0.78	0.78
W car to E	0.46	0.61	0.66	0.68	0.69	0.69	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.70
W car to W	-0.04	0.15	0.22	0.26	0.28	0.29	0.30	0.30	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.30	0.30	0.30	0.30	0.30

Travel time	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
A PT to A	0.59	0.52	0.49	0.47	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
A PT to C	0.94	0.88	0.84	0.82	0.80	0.80	0.79	0.79	0.81	0.82	0.83	0.83	0.83	0.83	0.82	0.81	0.80	0.79	0.79	0.78	0.78
A PT to S	0.82	0.76	0.73	0.71	0.70	0.69	0.69	0.69	0.70	0.70	0.71	0.71	0.71	0.71	0.72	0.71	0.71	0.71	0.71	0.71	0.72
A PT to N	1.15	1.06	1.01	0.98	0.96	0.95	0.94	0.94	0.96	0.97	0.98	0.98	0.98	0.98	0.97	0.96	0.95	0.94	0.93	0.93	0.92
A PT to E	1.12	1.03	0.97	0.94	0.92	0.91	0.91	0.91	0.91	0.92	0.92	0.92	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93	0.93
A PT to W	0.53	0.53	0.53	0.52	0.52	0.52	0.52	0.53	0.53	0.54	0.54	0.54	0.55	0.55	0.55	0.55	0.55	0.55	0.55	0.55	0.55
C PT to A	0.68	0.66	0.64	0.64	0.63	0.63	0.63	0.63	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62
C PT to C	0.51	0.49	0.47	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
C PT to S	0.60	0.58	0.58	0.57	0.57	0.57	0.57	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56
C PT to N	0.67	0.64	0.62	0.61	0.61	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60	0.60
C PT to E	0.90	0.86	0.83	0.82	0.81	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80	0.80
C PT to W	0.60	0.58	0.57	0.57	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56	0.56
S PT to A	0.79	0.75	0.72	0.71	0.70	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.69
S PT to C	0.60	0.59	0.58	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57	0.57
S PT to S	0.50	0.46	0.44	0.43	0.42	0.42	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41	0.41
S PT to N	0.82	0.78	0.75	0.73	0.73	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71
S PT to E	0.77	0.73	0.70	0.68	0.68	0.67	0.67	0.67	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
S PT to W	0.42	0.39	0.37	0.36	0.35	0.35	0.34	0.34	0.34	0.35	0.35	0.35	0.35	0.36	0.36	0.36	0.36	0.36	0.36	0.36	0.36
N PT to A	0.92	0.87	0.85	0.83	0.83	0.82	0.82	0.83	0.83	0.84	0.85	0.86	0.87	0.87	0.87	0.87	0.88	0.88	0.88	0.88	0.88
N PT to C	0.69	0.67	0.65	0.65	0.65	0.65	0.65	0.65	0.66	0.67	0.68	0.69	0.69	0.70	0.70	0.70	0.70	0.70	0.70	0.70	0.71
N PT to S	0.84	0.80	0.78	0.77	0.76	0.76	0.76	0.77	0.77	0.78	0.79	0.80	0.81	0.81	0.81	0.81	0.81	0.82	0.82	0.82	0.82
N PT to N	0.53	0.50	0.48	0.47	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
N PT to E	0.68	0.67	0.66	0.66	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65
N PT to W	0.84	0.80	0.77	0.76	0.76	0.76	0.76	0.76	0.77	0.78	0.79	0.79	0.80	0.80	0.81	0.81	0.81	0.81	0.81	0.81	0.81
E PT to A	1.10	1.02	0.97	0.94	0.92	0.91	0.90	0.90	0.90	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89
E PT to C	0.92	0.88	0.86	0.86	0.85	0.85	0.85	0.85	0.86	0.87	0.88	0.89	0.89	0.90	0.90	0.90	0.90	0.90	0.91	0.91	0.91
E PT to S	0.77	0.73	0.70	0.68	0.68	0.67	0.67	0.67	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66	0.66
E PT to N	0.68	0.67	0.66	0.66	0.66	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65	0.65
E PT to E	0.57	0.52	0.49	0.47	0.46	0.46	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45	0.45
E PT to W	1.02	0.94	0.90	0.87	0.85	0.84	0.84	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.82	0.82	0.82	0.82	0.82	0.82
W PT to A	0.53	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52	0.52
W PT to C	0.85	0.80	0.76	0.74	0.73	0.72	0.71	0.72	0.73	0.74	0.74	0.74	0.74	0.73	0.72	0.71	0.70	0.70	0.69	0.68	0.68
W PT to S	0.51	0.41	0.35	0.32	0.30	0.29	0.28	0.28	0.28	0.27	0.27	0.27	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26	0.26
W PT to N	1.07	0.99	0.93	0.90	0.88	0.87	0.86	0.86	0.87	0.88	0.89	0.89	0.89	0.88	0.87	0.86	0.85	0.84	0.84	0.83	0.83
W PT to E	1.04	0.95	0.90	0.87	0.85	0.84	0.84	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83
W PT to W	0.55	0.48	0.45	0.43	0.42	0.41	0.41	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40	0.40

Appendix 3. Equations for random utility theory
Residential location choice

The equation is shown below (see Equation 1).

$$P_{i*} = \frac{e^{U_{i*}}}{\sum_i e^{U_i}}$$

Equation 1

i: index for area

i*: a particular area

P_{i*}: Probability that urban district i* is chosen

U_i: Utility value in urban district i

The utility value in Equation 1 describes the attributes of urban district. The equation is shown below

$$U_i = \alpha_1 X_{1i} + \alpha_2 X_{2i} + \cdots + \alpha_n X_{ni}$$

Equation 2

n: index for attribute of urban district

X_{ni}: value of attribute n of urban district i

α_n: Parameter for attribute n

Pagliara found that dwelling type, air quality, traffic noise, street in front of dwelling, taxes or rent, time for travel to work, time for travel to shopping, time for travel to school, Crossing on walking trip to school are the utilities for household to choose dwellings. In this model, rather than consider dwellings we only consider which urban district is chosen. So the utilities that vary between different dwellings but have no different between urban districts are not considered in this model. Such utilities are dwelling type, air quality, traffic noise, street in front of dwelling, taxes or rent, time for travel to shopping, time for travel to school, Crossing on walking trip to school. So, time for travel to work is the only utility for household choosing urban district.

And Hjorthol (2003) found that the distance to city center also influence the location choice for cities in Norway . That might because in Bergen area, most pubs, threats and cinemas locate in city center (Bergenhus). So, travel time to city center is also a utility in this model. The equation for residential location choice in this model can be seen below.

$$P_{i*,j*} = \frac{e^{\alpha_W T_{i*,j*} + \alpha_C T_{i*,C}}}{\sum_i e^{\alpha_W T_{i,j*} + \alpha_C T_{i,C}}}$$

Equation 3

$P_{i*,j*}$: Probability that people who work in j* chose to live in area i*

$T_{i,j}$: Travel time from i to j in morning peak.

$T_{i,C}$: Travel time from i to city center in normal hour.

α_w: Parameter for travel time to job

α_C: Parameter for travel time to city center

Job location choice

The equation can be seen below.

$$P_{i*,j*} = \frac{e^{\alpha_w T_{i*,j*}}}{\sum_j e^{\alpha_w T_{i*,j}}}$$

Equation 4

$P_{i*,j*}$: Probability that people who live in i* chose to work in area j*

$T_{i,j}$: Travel time from i to j in morning peak.

α_w: Parameter for travel time to job

Appendix 4. Equations for model

adult__family_relocaters[Resident,Working](t) =
adult__family_relocaters[Resident,Working](t - dt) +
(new_adult__family_relocater[Resident,Working] -
AFR__stop_searching[Resident,Working] -
AFR__leaving_old_dwelling[Resident,Working]) * dt
INIT adult__family_relocaters[Resident,Working] = 1
INFLOWS:
new_adult__family_relocater[Resident,Working] =
(AHFS_decision_to_move[Resident,Working]+AHLS_decision_to_move[Resident,W
orking])
OUTFLOWS:
AFR__stop_searching[Resident,Working] =
MAX(adult__family_relocaters[Resident,Working]/time_to_stop_seeking-AFR__leav
ing_old_dwelling[Resident,Working],0)
AFR__leaving_old_dwelling[Resident,Working] =
AFP_get_vacancy_sort_by_WP[Working]/AFR_sorted_by_working_place[Working]
*adult__family_relocaters[Resident,Working]
Adult__married_or_cohibited[Resident,Working](t) =
Adult__married_or_cohibited[Resident,Working](t - dt) +
(Adult_moving_together[Resident,Working] +
net_migration_AMC[Resident,Working] + AMC_job_change[Resident,Working] -
Retirement_of_AMC[Resident,Working] - Adult_becoming_single[Resident,Working]
- Death_of_AMC[Resident,Working]) * dt
INIT Adult__married_or_cohibited[Resident,Working] =
MC_commuter_real[Resident,Working]
INFLOWS:
Adult_moving_together[Resident,Working] =
Adult_single[Resident,Working]*marriage_rate*(1+Frac_cohibit_to_marriage)*2
net_migration_AMC[Resident,Working] =
imigration_AMC[Resident,Working]-emigration_AMC[Resident,Working]
AMC_job_change[Resident,Working] =
NOJS_get_job[Resident,Working]-NOJS_Job_quiting[Resident,Working]
OUTFLOWS:
Retirement_of_AMC[Resident,Working] =
Adult__married_or_cohibited[Resident,Working]/47
Adult_becoming_single[Resident,Working] =
Adult__married_or_cohibited[Resident,Working]*Divorce_rate*(1+Frac_cohibit_to_
marriage)*2+Death_of_AMC[Resident,Working]
Death_of_AMC[Resident,Working] =
Adult__married_or_cohibited[Resident,Working]*Death_rate_adult
Adult_Household_with_free_space[Resident,Working](t) =
Adult_Household_with_free_space[Resident,Working](t - dt) +
(Adult_household_lossing_young_member[Resident,Working] -
AHFS_stay_in_position[Resident,Working] -
AHFS_decision_to_move[Resident,Working]) * dt
INIT Adult_Household_with_free_space[Resident,Working] =
Adult_household_lossing_young_member[Resident,Working]
INFLOWS:

Adult_household_lossing_young_member[Resident,Working] =
household_wiith_child_leaving[Resident,Working]
OUTFLOWS:
AHFS_stay_in_position[Resident,Working] =
Adult_Household_with_free_space[Resident,Working]*(1-frac_move_by_deserted)*
decision_time_for_adjust_living_condition
AHFS_decision_to_move[Resident,Working] =
Adult_Household_with_free_space[Resident,Working]*frac_move_by_deserted*decision_time_for_adjust_living_condition
Adult_Household_with_less_space[Resident,Working](t) =
Adult_Household_with_less_space[Resident,Working](t - dt) +
(Household_with_new_member[Resident,Working] -
AHLS_stay_in_postion[Resident,Working] -
AHLS_decision_to_move[Resident,Working]) * dt
INIT Adult_Household_with_less_space[Resident,Working] =
Household_with_new_member[Resident,Working]
INFLOWS:
Household_with_new_member[Resident,Working] =
household_with_new_adult[Resident,Working]+household_with_new_babies[Resident,Working]
OUTFLOWS:
AHLS_stay_in_postion[Resident,Working] =
Adult_Household_with_less_space[Resident,Working]*(1-frac_move_by_crowd)/decision_time_for_adjust_living_condition
AHLS_decision_to_move[Resident,Working] =
Adult_Household_with_less_space[Resident,Working]*frac_move_by_crowd/decision_time_for_adjust_living_condition
Adult_single[Resident,Working](t) = Adult_single[Resident,Working](t - dt) +
(Youth_starting__living_alone[Resident,Working] +
Adult_becoming_single[Resident,Working] + net_migration_AS[Resident,Working]
+ AS_job_change[Resident,Working] - Retiremen_of_AS[Resident,Working] -
Adult_moving_together[Resident,Working] - Death_of_AS[Resident,Working]) * dt
INIT Adult_single[Resident,Working] = singel_commuter_real[Resident,Working]
INFLOWS:
Youth_starting__living_alone[Resident,Working] =
YS_get_vacancy[Resident,Working]
Adult_becoming_single[Resident,Working] =
Adult__married_or_cohibited[Resident,Working]*Divorce_rate*(1+Frac_cohibit_to_marriage)*2+Death_of_AMC[Resident,Working]
net_migration_AS[Resident,Working] =
imigration_AS[Resident,Working]-emigration_AS[Resident,Working]
AS_job_change[Resident,Working] =
SOJS_get_job[Resident,Working]-SOJSJob_quiting[Resident,Working]
OUTFLOWS:
Retiremen_of_AS[Resident,Working] = Adult_single[Resident,Working]/47
Adult_moving_together[Resident,Working] =
Adult_single[Resident,Working]*marriage_rate*(1+Frac_cohibit_to_marriage)*2
Death_of_AS[Resident,Working] =
Adult_single[Resident,Working]*Death_rate_adult

Divorced__adult_relocater[Resident,Working](t) =
Divorced__adult_relocater[Resident,Working](t - dt) +
(new_Divorced__adult_relocater[Resident,Working] -
DAR__leaving_old_dwelling[Resident,Working]) * dt
INIT Divorced__adult_relocater[Resident,Working] =
new_Divorced__adult_relocater[Resident,Working]*dt
INFLOWS:
new_Divorced__adult_relocater[Resident,Working] =
New_divorced_dwelling_seeker[Resident,Working]
OUTFLOWS:
DAR__leaving_old_dwelling[Resident,Working] =
DAR_get_vacancy_sort_by_WP[Working]/DAR_sorted_by_working_place[Working
]*Divorced__adult_relocater[Resident,Working]
Dwelling_Vacancy[Askoy](t) = Dwelling_Vacancy[Askoy](t - dt) +
(New_dwelling_vacancy[Askoy] - Vacancy_Taken[Askoy]) * dt
INIT Dwelling_Vacancy[Askoy] = 1108
Dwelling_Vacancy[Center_Bergen](t) = Dwelling_Vacancy[Center_Bergen](t - dt) +
(New_dwelling_vacancy[Center_Bergen] - Vacancy_Taken[Center_Bergen]) * dt
INIT Dwelling_Vacancy[Center_Bergen] = 726
Dwelling_Vacancy[South_Bergen](t) = Dwelling_Vacancy[South_Bergen](t - dt) +
(New_dwelling_vacancy[South_Bergen] - Vacancy_Taken[South_Bergen]) * dt
INIT Dwelling_Vacancy[South_Bergen] = 2156
Dwelling_Vacancy[North_Bergen](t) = Dwelling_Vacancy[North_Bergen](t - dt) +
(New_dwelling_vacancy[North_Bergen] - Vacancy_Taken[North_Bergen]) * dt
INIT Dwelling_Vacancy[North_Bergen] = 1013
Dwelling_Vacancy[East_Bergen](t) = Dwelling_Vacancy[East_Bergen](t - dt) +
(New_dwelling_vacancy[East_Bergen] - Vacancy_Taken[East_Bergen]) * dt
INIT Dwelling_Vacancy[East_Bergen] = 621
Dwelling_Vacancy[West_Bergen](t) = Dwelling_Vacancy[West_Bergen](t - dt) +
(New_dwelling_vacancy[West_Bergen] - Vacancy_Taken[West_Bergen]) * dt
INIT Dwelling_Vacancy[West_Bergen] = 2427
INFLOWS:
New_dwelling_vacancy[Resident] =
emigration_leaving_dwellings[Resident]+household_diminish[Resident]+Household_
leaving_dwellings[Resident]+moving_together[Resident]+new_dwelling_array[Resid
ent]
OUTFLOWS:
Vacancy_Taken[Resident] =
MIN(TOTAL_seeker[Resident],Dwelling_Vacancy[Resident]/Average_transit_time_f
or_Dwelling)
Family_first_time_buyer[Working](t) = Family_first_time_buyer[Working](t - dt) +
(new_FFTB[Working] - FFB_moving_in[Working] -
FFTB_become_long_distance_commuter[Working]) * dt
INIT Family_first_time_buyer[Working] = new_FFTB[Working]*dt
INFLOWS:
new_FFTB[Working] = family_immigration_household[Working]
OUTFLOWS:
FFB_moving_in[Working] = FFB_get_vacancy_sort_by_WP[Working]

FFTB_become_long_distance_commuter[Working] =
MAX(Family_first_time_buyer[Working]/time_to_become_long_distance_commuter
-FFB_moving_in[Working],0)
First_time_job_seeker(t) = First_time_job_seeker(t - dt) + (new_FTJS -
FTJS_getting_job) * dt
INIT First_time_job_seeker = new_FTJS*dt
INFLOWS:
new_FTJS = Stating_job_seeking
OUTFLOWS:
FTJS_getting_job = FTJS_get_vacancy
imigration_job_seeker(t) = imigration_job_seeker(t - dt) + (new_IJS - IJS_getting_job)
* dt
INIT imigration_job_seeker = new_IJS*dt
INFLOWS:
new_IJS = needed_immigration
OUTFLOWS:
IJS_getting_job = ARRAYSUM(IJS_get_job[*])
Job_vacancy[Askoy_W](t) = Job_vacancy[Askoy_W](t - dt) +
(New_job_acancy[Askoy_W] - job_offers[Askoy_W]) * dt
INIT Job_vacancy[Askoy_W] = 104
Job_vacancy[Center_Bergen_W](t) = Job_vacancy[Center_Bergen_W](t - dt) +
(New_job_acancy[Center_Bergen_W] - job_offers[Center_Bergen_W]) * dt
INIT Job_vacancy[Center_Bergen_W] = 4289
Job_vacancy[South_Bergen_W](t) = Job_vacancy[South_Bergen_W](t - dt) +
(New_job_acancy[South_Bergen_W] - job_offers[South_Bergen_W]) * dt
INIT Job_vacancy[South_Bergen_W] = 1080
Job_vacancy[North_Bergen_W](t) = Job_vacancy[North_Bergen_W](t - dt) +
(New_job_acancy[North_Bergen_W] - job_offers[North_Bergen_W]) * dt
INIT Job_vacancy[North_Bergen_W] = 188
Job_vacancy[East_Bergen_W](t) = Job_vacancy[East_Bergen_W](t - dt) +
(New_job_acancy[East_Bergen_W] - job_offers[East_Bergen_W]) * dt
INIT Job_vacancy[East_Bergen_W] = 56
Job_vacancy[West_Bergen_W](t) = Job_vacancy[West_Bergen_W](t - dt) +
(New_job_acancy[West_Bergen_W] - job_offers[West_Bergen_W]) * dt
INIT Job_vacancy[West_Bergen_W] = 298
INFLOWS:
New_job_acancy[Working] =
Emigration_leaving_job[Working]+job_quitting[Working]+net_job_creation_sorted_
by_working_place[Working]+Retirement[Working]+death_leaving_job[Working]
OUTFLOWS:
job_offers[Working] =
MIN(Job_vacancy[Working]/Average_transit_time_for_job,Total_search_effect[Work
ing])
Juveniles(t) = Juveniles(t - dt) + (Birth + net_miration_J - Stating_job_seeking -
Death_of_J) * dt
INIT Juveniles = 65036
INFLOWS:
Birth = live_birth_Bergen_Askoy
net_miration_J = EXO_imigration_juveniles-EXO_emigration_juveniles
OUTFLOWS:

Stating_job_seeking =
Juveniles/19*(1-STEP(1,2012))+(total_Juveniles_proj/19)*(0+STEP(1,2012))
Death_of_J = Juveniles*Death_rate_juveniles
long_distance_commuter(t) = long_distance_commuter(t - dt) +
(new_long_distance_commuter) * dt
INIT long_distance_commuter = 0
INFLOWS:
new_long_distance_commuter =
ARRAYSUM(FFTB_become_long_distance_commuter[*])+ARRAYSUM(SFTB_become_long_distance_commuter[*])
NONsingle_on_job_seeker[Resident,Working](t) =
NONsingle_on_job_seeker[Resident,Working](t - dt) +
(new_NOJS[Resident,Working] - NOJS_Job_quiting[Resident,Working]) * dt
INIT NONsingle_on_job_seeker[Resident,Working] =
new_NOJS[Resident,Working]*dt
INFLOWS:
new_NOJS[Resident,Working] =
Adult__married_or_cohibited[Resident,Working]*turnover_rate
OUTFLOWS:
NOJS_Job_quiting[Resident,Working] =
NOJS_get_vacancy_sort_by_Resident[Resident]/NOJS_sorted_by_resident_place[Resident]*NONsingle_on_job_seeker[Resident,Working]
Pensioner__married_or_cohibited[Resident,Working](t) =
Pensioner__married_or_cohibited[Resident,Working](t - dt) +
(Retirement_of_AMC[Resident,Working] + Net_migration_PMC[Resident,Working] - Death_of_PMC[Resident,Working] - Becoming_widow[Resident,Working]) * dt
INIT Pensioner__married_or_cohibited[Resident,Working] =
ini_pensioner_MC[Resident,Working]
INFLOWS:
Retirement_of_AMC[Resident,Working] =
Adult__married_or_cohibited[Resident,Working]/47
Net_migration_PMC[Resident,Working] = -emigration_FMC[Resident,Working]
OUTFLOWS:
Death_of_PMC[Resident,Working] =
Pensioner__married_or_cohibited[Resident,Working]*Death_rate_pensioner
Becoming_widow[Resident,Working] = Death_of_PMC[Resident,Working]
Pensioner__single[Resident,Working](t) = Pensioner__single[Resident,Working](t - dt) + (Retiremen_of_AS[Resident,Working] + Becoming_widow[Resident,Working] + Net_migration_PS[Resident,Working] - Death_of_PS[Resident,Working]) * dt
INIT Pensioner__single[Resident,Working] =
ini_pensioner_single[Resident,Working]
INFLOWS:
Retiremen_of_AS[Resident,Working] = Adult_single[Resident,Working]/47
Becoming_widow[Resident,Working] = Death_of_PMC[Resident,Working]
Net_migration_PS[Resident,Working] =
imigration_PS[Resident,Working]-emigration_PS[Resident,Working]
OUTFLOWS:
Death_of_PS[Resident,Working] =
Pensioner__single[Resident,Working]*Death_rate_pensioner

single__adult_dwelling_seeker[Resident,Working](t) =
single__adult_dwelling_seeker[Resident,Working](t - dt) +
(new_single__adult_dwelling_seeker[Resident,Working] -
SAR__leaving_old_dwelling[Resident,Working] -
SAR__stop_searching[Resident,Working]) * dt

INIT single__adult_dwelling_seeker[Resident,Working] =
new_single__adult_dwelling_seeker[Resident,Working]*dt
INFLOWS:
new_single__adult_dwelling_seeker[Resident,Working] =
SHFS_decision_to_move[Resident,Working]
OUTFLOWS:
SAR__leaving_old_dwelling[Resident,Working] =
SAR_get_vacancy_sort_by_WP[Working]/SAR_sorted_by_working_place[Working]
*single__adult_dwelling_seeker[Resident,Working]
SAR__stop_searching[Resident,Working] =
MAX(single__adult_dwelling_seeker[Resident,Working]/time_to_stop_seeking-SAR
__leaving_old_dwelling[Resident,Working],0)
Single__pensioner_relocater[Resident,Working](t) =
Single__pensioner_relocater[Resident,Working](t - dt) +
(new_single__pensioner_relocater[Resident,Working] -
SPR_moving_out[Resident,Working] - SPR_stop_seeking[Resident,Working]) * dt
INIT Single__pensioner_relocater[Resident,Working] =
new_single__pensioner_relocater[Resident,Working]*dt
INFLOWS:
new_single__pensioner_relocater[Resident,Working] =
SPFS_decision_to_move[Resident,Working]
OUTFLOWS:
SPR_moving_out[Resident,Working] =
ARRAYSUM(SPR_get_vacancy[Resident,*])/ARRAYSUM(Single__pensioner_reloc
ater[Resident,*])*Single__pensioner_relocater[Resident,Working]
SPR_stop_seeking[Resident,Working] =
MAX(Single__pensioner_relocater[Resident,Working]/time_to_stop_seeking-SPR_m
oving_out[Resident,Working],0)
Single_first_time_buyer[Working](t) = Single_first_time_buyer[Working](t - dt) +
(new_SFTB[Working] - SFTB_moving_in[Working] -
SFTB_become_long_distance_commuter[Working]) * dt
INIT Single_first_time_buyer[Working] = new_SFTB[Working]*dt
INFLOWS:
new_SFTB[Working] = single_immigration[Working]
OUTFLOWS:
SFTB_moving_in[Working] = SFTB_get_vacancy_sort_by_WP[Working]
SFTB_become_long_distance_commuter[Working] =
MAX(Single_first_time_buyer[Working]/time_to_become_long_distance_commuter-
SFTB_moving_in[Working],0)
Single_household_with_free_space[Resident,Working](t) =
Single_household_with_free_space[Resident,Working](t - dt) +
(Adult_household_lossing_adult_member[Resident,Working] -
SHFS_stay_in_position[Resident,Working] -
SHFS_decision_to_move[Resident,Working]) * dt

INIT Single_household_with_free_space[Resident,Working] =
Adult_household_lossing_adult_member[Resident,Working]
INFLOWS:
Adult_household_lossing_adult_member[Resident,Working] =
Household_with_widow[Resident,Working]+houshold_with_adult_leaving[Resident,
Working]
OUTFLOWS:
SHFS_stay_in_position[Resident,Working] =
Single_household_with_free_space[Resident,Working]*(1-frac_move_by_deserted)*
decision_time_for_adjust_living_condition
SHFS_decision_to_move[Resident,Working] =
Single_household_with_free_space[Resident,Working]*frac_move_by_deserted*deci
sion_time_for_adjust_living_condition
single_on_job_seeker[Resident,Working](t) =
single_on_job_seeker[Resident,Working](t - dt) + (new_SOJS[Resident,Working] -
SOJSJob_quiting[Resident,Working]) * dt
INIT single_on_job_seeker[Resident,Working] = new_SOJS[Resident,Working]*dt
INFLOWS:
new_SOJS[Resident,Working] = Adult_single[Resident,Working]*turnover_rate
OUTFLOWS:
SOJSJob_quiting[Resident,Working] =
SOJS_get_vacancy_sort_by_Resident[Resident]/SOJS_sorted_by_resident_place[Res
ident]*single_on_job_seeker[Resident,Working]
single_pensioner_with_free_space[Resident,Working](t) =
single_pensioner_with_free_space[Resident,Working](t - dt) +
(pension_household_lossing_member[Resident,Working] -
SPFS_stay_in_position[Resident,Working] -
SPFS_decision_to_move[Resident,Working]) * dt
INIT single_pensioner_with_free_space[Resident,Working] =
pension_household_lossing_member[Resident,Working]
INFLOWS:
pension_household_lossing_member[Resident,Working] =
pensioner_household_with_widow[Resident,Working]
OUTFLOWS:
SPFS_stay_in_position[Resident,Working] =
single_pensioner_with_free_space[Resident,Working]*(1-frac_move_by_deserted)*d
ecision_time_for_adjust_living_condition
SPFS_decision_to_move[Resident,Working] =
single_pensioner_with_free_space[Resident,Working]*frac_move_by_deserted*decis
ion_time_for_adjust_living_condition
Total_job_opportunity(t) = Total_job_opportunity(t - dt) + (Net_change_of_job) * dt
INIT Total_job_opportunity = 155073+Total_working_age_population*0
INFLOWS:
Net_change_of_job =
net_job_creation*(1-STEP(1,2012))+net_job_change_projection*(0+STEP(1,2012))
Youth_seeker[Working](t) = Youth_seeker[Working](t - dt) + (New_YS[Working] -
YS_moving_in[Working]) * dt
INIT Youth_seeker[Working] = New_YS[Working]*dt
INFLOWS:
New_YS[Working] = newly_matured_dwelling_seeker[Working]

OUTFLOWS:
YS_moving_in[Working] = YS_get_vacancy_sort_by_WP[Working]
adult_with_resident_and_work[Resident,Working] =
(Adult__married_or_cohibited[Resident,Working]+Adult_single[Resident,Working])
AFP_get_vacancy_sort_by_WP[Working] =
ARRAYSUM(AFR_get_vacancy[*,Working])
AFR_get_vacancy[Resident,Working] =
Search_effect_AFR[Resident,Working]*Frac_seek_get_vacancy[Resident]
AFR_seeker_car_WP[Working] =
AFR_sorted_by_working_place[Working]*frac_owing_car
AFR_seeker_PT_WP[Working] =
AFR_sorted_by_working_place[Working]*(1-frac_owing_car)
AFR_sorted_by_working_place[Working] =
ARRAYSUM(adult__family_relocaters[*,Working])
age_2 = GRAPH(TIME)
(2001, 39349), (2002, 39578), (2003, 39770), (2004, 39857), (2005, 39798), (2006, 39883), (2007, 40069), (2008, 40263), (2009, 40703), (2010, 41555), (2011, 42143), (2012, 42379)
area_household[Resident] =
ARRAYSUM(Adult__married_or_cohibited[Resident,*])+ARRAYSUM(Adult_single[Resident,*])+ARRAYSUM(Pensioner__married_or_cohibited[Resident,*])+ARRAYSUM(Pensioner__single[Resident,*])
average_family_commuting_time =
ARRAYSUM(total_family_commuting_time[*,*])/ARRAYSUM(Adult__married_or_cohibited[*,*])
Average_number_of_child = GRAPH(TIME)
(2001, 2.50), (2011, 2.34)
average_sigle_commuting_time =
ARRAYSUM(total_single_commuting_time[*,*])/ARRAYSUM(Adult_single[*,*])
Average_transit_time_for_Dwelling = 1/12
Average_transit_time_for_job = 1/12
A_car_to_A = GRAPH(TIME)
(2001, -0.0683), (2002, 0.123), (2002, 0.203), (2003, 0.243), (2003, 0.263), (2004, 0.274), (2004, 0.28), (2005, 0.285), (2005, 0.291), (2006, 0.295), (2006, 0.296), (2007, 0.294), (2007, 0.293), (2008, 0.291), (2008, 0.288), (2009, 0.286), (2009, 0.284), (2010, 0.283), (2010, 0.282), (2011, 0.281), (2011, 0.28)
A_car_to_A_2 = GRAPH(TIME)
(2001, -0.456), (2002, -0.134), (2002, 0.0292), (2003, 0.119), (2003, 0.17), (2004, 0.2), (2004, 0.217), (2005, 0.229), (2005, 0.238), (2006, 0.243), (2006, 0.246), (2007, 0.244), (2007, 0.243), (2008, 0.242), (2008, 0.242), (2009, 0.242), (2009, 0.242), (2010, 0.242), (2010, 0.242), (2011, 0.243), (2011, 0.243)
A_car_to_C = GRAPH(TIME)
(2001, 0.187), (2002, 0.497), (2002, 0.624), (2003, 0.673), (2003, 0.687), (2004, 0.684), (2004, 0.673), (2005, 0.673), (2005, 0.69), (2006, 0.707), (2006, 0.719), (2007, 0.724), (2007, 0.728), (2008, 0.725), (2008, 0.719), (2009, 0.713), (2009, 0.707), (2010, 0.703), (2010, 0.7), (2011, 0.698), (2011, 0.698)
A_car_to_C_2 = GRAPH(TIME)
(2001, 0.00), (2002, 0.34), (2002, 0.515), (2003, 0.612), (2003, 0.673), (2004, 0.713), (2004, 0.74), (2005, 0.785), (2005, 0.863), (2006, 0.941), (2006, 1.01), (2007, 1.06),

(2007, 1.10), (2008, 1.12), (2008, 1.14), (2009, 1.15), (2009, 1.16), (2010, 1.18), (2010, 1.19), (2011, 1.21), (2011, 1.23)
A_car_to_E = GRAPH(TIME)
(2001, 0.534), (2002, 0.681), (2002, 0.725), (2003, 0.747), (2003, 0.758), (2004, 0.763), (2004, 0.766), (2005, 0.77), (2005, 0.776), (2006, 0.781), (2006, 0.784), (2007, 0.783), (2007, 0.784), (2008, 0.785), (2008, 0.784), (2009, 0.784), (2009, 0.784), (2010, 0.784), (2010, 0.784), (2011, 0.784), (2011, 0.785)
A_car_to_E_2 = GRAPH(TIME)
(2001, 0.00), (2002, 0.423), (2002, 0.628), (2003, 0.74), (2003, 0.804), (2004, 0.841), (2004, 0.863), (2005, 0.878), (2005, 0.89), (2006, 0.898), (2006, 0.902), (2007, 0.901), (2007, 0.902), (2008, 0.901), (2008, 0.901), (2009, 0.901), (2009, 0.901), (2010, 0.901), (2010, 0.901), (2011, 0.902), (2011, 0.902)
A_car_to_N = GRAPH(TIME)
(2001, 0.36), (2002, 0.67), (2002, 0.797), (2003, 0.846), (2003, 0.86), (2004, 0.857), (2004, 0.847), (2005, 0.847), (2005, 0.863), (2006, 0.88), (2006, 0.892), (2007, 0.898), (2007, 0.901), (2008, 0.899), (2008, 0.893), (2009, 0.887), (2009, 0.881), (2010, 0.877), (2010, 0.874), (2011, 0.872), (2011, 0.872)
A_car_to_N_2 = GRAPH(TIME)
(2001, 0.00), (2002, 0.429), (2002, 0.656), (2003, 0.784), (2003, 0.863), (2004, 0.914), (2004, 0.948), (2005, 0.996), (2005, 1.08), (2006, 1.16), (2006, 1.23), (2007, 1.28), (2007, 1.32), (2008, 1.34), (2008, 1.35), (2009, 1.37), (2009, 1.38), (2010, 1.39), (2010, 1.41), (2011, 1.43), (2011, 1.45)
A_car_to_S = GRAPH(TIME)
(2001, 0.276), (2002, 0.419), (2002, 0.469), (2003, 0.492), (2003, 0.504), (2004, 0.509), (2004, 0.512), (2005, 0.516), (2005, 0.522), (2006, 0.527), (2006, 0.53), (2007, 0.53), (2007, 0.531), (2008, 0.531), (2008, 0.531), (2009, 0.531), (2009, 0.53), (2010, 0.53), (2010, 0.531), (2011, 0.531), (2011, 0.532)
A_car_to_S_2 = GRAPH(TIME)
(2001, 0.00), (2002, 0.284), (2002, 0.417), (2003, 0.486), (2003, 0.525), (2004, 0.547), (2004, 0.56), (2005, 0.569), (2005, 0.578), (2006, 0.584), (2006, 0.587), (2007, 0.586), (2007, 0.585), (2008, 0.585), (2008, 0.584), (2009, 0.584), (2009, 0.584), (2010, 0.584), (2010, 0.585), (2011, 0.585), (2011, 0.586)
A_car_to_W = GRAPH(TIME)
(2001, 0.0667), (2002, 0.213), (2002, 0.267), (2003, 0.291), (2003, 0.303), (2004, 0.309), (2004, 0.312), (2005, 0.316), (2005, 0.322), (2006, 0.327), (2006, 0.33), (2007, 0.329), (2007, 0.331), (2008, 0.331), (2008, 0.331), (2009, 0.33), (2009, 0.33), (2010, 0.33), (2010, 0.33), (2011, 0.331), (2011, 0.332)
A_car_to_W_2 = GRAPH(TIME)
(2001, 0.00), (2002, 0.176), (2002, 0.25), (2003, 0.286), (2003, 0.305), (2004, 0.315), (2004, 0.32), (2005, 0.325), (2005, 0.332), (2006, 0.336), (2006, 0.338), (2007, 0.336), (2007, 0.336), (2008, 0.335), (2008, 0.334), (2009, 0.334), (2009, 0.334), (2010, 0.334), (2010, 0.335), (2011, 0.335), (2011, 0.335)
A_PT_to_A = GRAPH(TIME)
(2001, 0.586), (2002, 0.524), (2002, 0.491), (2003, 0.473), (2003, 0.463), (2004, 0.458), (2004, 0.454), (2005, 0.452), (2005, 0.451), (2006, 0.451), (2006, 0.45), (2007, 0.45), (2007, 0.45), (2008, 0.45), (2008, 0.45), (2009, 0.45), (2009, 0.45), (2010, 0.45), (2010, 0.45), (2011, 0.45), (2011, 0.45)
A_PT_to_A_2 = GRAPH(TIME)
(2001, -0.456), (2002, -0.134), (2002, 0.0292), (2003, 0.119), (2003, 0.17), (2004, 0.2), (2004, 0.217), (2005, 0.229), (2005, 0.238), (2006, 0.243), (2006, 0.246), (2007,

0.244), (2007, 0.243), (2008, 0.242), (2008, 0.242), (2009, 0.242), (2009, 0.242), (2010, 0.242), (2010, 0.242), (2011, 0.243), (2011, 0.243)
A_PT_to_C = GRAPH(TIME)
(2001, 0.94), (2002, 0.878), (2002, 0.841), (2003, 0.818), (2003, 0.805), (2004, 0.796), (2004, 0.789), (2005, 0.792), (2005, 0.808), (2006, 0.822), (2006, 0.831), (2007, 0.834), (2007, 0.835), (2008, 0.829), (2008, 0.82), (2009, 0.811), (2009, 0.801), (2010, 0.793), (2010, 0.786), (2011, 0.781), (2011, 0.777)
A_PT_to_C_2 = GRAPH(TIME)
(2001, 1.03), (2002, 0.954), (2002, 0.952), (2003, 0.983), (2003, 1.03), (2004, 1.07), (2004, 1.11), (2005, 1.15), (2005, 1.20), (2006, 1.23), (2006, 1.26), (2007, 1.30), (2007, 1.33), (2008, 1.36), (2008, 1.38), (2009, 1.39), (2009, 1.40), (2010, 1.41), (2010, 1.42), (2011, 1.42), (2011, 1.43)
A_PT_to_E = GRAPH(TIME)
(2001, 1.12), (2002, 1.03), (2002, 0.972), (2003, 0.941), (2003, 0.924), (2004, 0.915), (2004, 0.91), (2005, 0.909), (2005, 0.912), (2006, 0.917), (2006, 0.921), (2007, 0.923), (2007, 0.927), (2008, 0.928), (2008, 0.928), (2009, 0.928), (2009, 0.928), (2010, 0.927), (2010, 0.927), (2011, 0.928), (2011, 0.929)
A_PT_to_E_2 = GRAPH(TIME)
(2001, 1.22), (2002, 1.61), (2002, 1.57), (2003, 1.55), (2003, 1.55), (2004, 1.54), (2004, 1.54), (2005, 1.54), (2005, 1.54), (2006, 1.55), (2006, 1.55), (2007, 1.55), (2007, 1.55), (2008, 1.55), (2008, 1.55), (2009, 1.55), (2009, 1.55), (2010, 1.55), (2010, 1.55), (2011, 1.55), (2011, 1.55)
A_PT_to_N = GRAPH(TIME)
(2001, 1.15), (2002, 1.06), (2002, 1.01), (2003, 0.977), (2003, 0.959), (2004, 0.947), (2004, 0.939), (2005, 0.941), (2005, 0.955), (2006, 0.97), (2006, 0.978), (2007, 0.981), (2007, 0.982), (2008, 0.977), (2008, 0.968), (2009, 0.958), (2009, 0.949), (2010, 0.94), (2010, 0.934), (2011, 0.929), (2011, 0.925)
A_PT_to_N_2 = GRAPH(TIME)
(2001, 1.19), (2002, 1.58), (2002, 1.54), (2003, 1.52), (2003, 1.51), (2004, 1.51), (2004, 1.52), (2005, 1.55), (2005, 1.61), (2006, 1.69), (2006, 1.75), (2007, 1.81), (2007, 1.85), (2008, 1.87), (2008, 1.88), (2009, 1.89), (2009, 1.90), (2010, 1.92), (2010, 1.94), (2011, 1.96), (2011, 1.98)
A_PT_to_S = GRAPH(TIME)
(2001, 0.824), (2002, 0.758), (2002, 0.725), (2003, 0.708), (2003, 0.699), (2004, 0.695), (2004, 0.693), (2005, 0.694), (2005, 0.698), (2006, 0.703), (2006, 0.707), (2007, 0.71), (2007, 0.714), (2008, 0.715), (2008, 0.715), (2009, 0.715), (2009, 0.714), (2010, 0.714), (2010, 0.714), (2011, 0.715), (2011, 0.715)
A_PT_to_S_2 = GRAPH(TIME)
(2001, 0.978), (2002, 1.08), (2002, 1.06), (2003, 1.05), (2003, 1.05), (2004, 1.05), (2004, 1.04), (2005, 1.05), (2005, 1.05), (2006, 1.05), (2006, 1.05), (2007, 1.06), (2007, 1.06), (2008, 1.06), (2008, 1.06), (2009, 1.06), (2009, 1.06), (2010, 1.06), (2010, 1.06), (2011, 1.06), (2011, 1.06)
A_PT_to_W = GRAPH(TIME)
(2001, 0.53), (2002, 0.527), (2002, 0.525), (2003, 0.524), (2003, 0.524), (2004, 0.524), (2004, 0.525), (2005, 0.527), (2005, 0.532), (2006, 0.536), (2006, 0.54), (2007, 0.543), (2007, 0.546), (2008, 0.547), (2008, 0.548), (2009, 0.547), (2009, 0.547), (2010, 0.547), (2010, 0.547), (2011, 0.547), (2011, 0.548)
A_PT_to_W_2 = GRAPH(TIME)
(2001, 0.71), (2002, 0.642), (2002, 0.636), (2003, 0.632), (2003, 0.629), (2004, 0.627), (2004, 0.626), (2005, 0.627), (2005, 0.63), (2006, 0.633), (2006, 0.635), (2007, 0.637),

(2007, 0.639), (2008, 0.64), (2008, 0.64), (2009, 0.64), (2009, 0.64), (2010, 0.64), (2010, 0.641), (2011, 0.641), (2011, 0.642)
bergen_household = total_household-ARRAYSUM(Adult__married_or_cohibited[Askoy,*])-ARRAYSUM(Adult_single[Askoy,*])-ARRAYSUM(Pensioner__married_or_cohibited[Askoy,*])-ARRAYSUM(Pensioner__single[Askoy,*])
b_for_family_choosing_High_density = 10
b_for_single_choosing_Low_density = 10
car_travel_time_congestion[Askoy,Askoy_W] = A_car_to_A
car_travel_time_congestion[Askoy,Center_Bergen_W] = A_car_to_C
car_travel_time_congestion[Askoy,South_Bergen_W] = A_car_to_S
car_travel_time_congestion[Askoy,North_Bergen_W] = A_car_to_N
car_travel_time_congestion[Askoy,East_Bergen_W] = A_car_to_E
car_travel_time_congestion[Askoy,West_Bergen_W] = A_car_to_W
car_travel_time_congestion[Center_Bergen,Askoy_W] = C_car_to_A
car_travel_time_congestion[Center_Bergen,Center_Bergen_W] = C_car_to_C
car_travel_time_congestion[Center_Bergen,South_Bergen_W] = C_car_to_S
car_travel_time_congestion[Center_Bergen,North_Bergen_W] = C_car_to_N
car_travel_time_congestion[Center_Bergen,East_Bergen_W] = C_car_to_E
car_travel_time_congestion[Center_Bergen,West_Bergen_W] = C_car_to_W
car_travel_time_congestion[South_Bergen,Askoy_W] = S_car_to_A
car_travel_time_congestion[South_Bergen,Center_Bergen_W] = S_car_to_C
car_travel_time_congestion[South_Bergen,South_Bergen_W] = S_car_to_S
car_travel_time_congestion[South_Bergen,North_Bergen_W] = S_car_to_N
car_travel_time_congestion[South_Bergen,East_Bergen_W] = S_car_to_E
car_travel_time_congestion[South_Bergen,West_Bergen_W] = S_car_to_W
car_travel_time_congestion[North_Bergen,Askoy_W] = N_car_to_A
car_travel_time_congestion[North_Bergen,Center_Bergen_W] = N_car_to_C
car_travel_time_congestion[North_Bergen,South_Bergen_W] = N_car_to_S
car_travel_time_congestion[North_Bergen,North_Bergen_W] = N_car_to_N
car_travel_time_congestion[North_Bergen,East_Bergen_W] = N_car_to_E
car_travel_time_congestion[North_Bergen,West_Bergen_W] = N_car_to_W
car_travel_time_congestion[East_Bergen,Askoy_W] = E_car_to_A
car_travel_time_congestion[East_Bergen,Center_Bergen_W] = E_car_to_C
car_travel_time_congestion[East_Bergen,South_Bergen_W] = E_car_to_S
car_travel_time_congestion[East_Bergen,North_Bergen_W] = E_car_to_N
car_travel_time_congestion[East_Bergen,East_Bergen_W] = E_car_to_E
car_travel_time_congestion[East_Bergen,West_Bergen_W] = E_car_to_W
car_travel_time_congestion[West_Bergen,Askoy_W] = W_car_to_A
car_travel_time_congestion[West_Bergen,Center_Bergen_W] = W_car_to_C
car_travel_time_congestion[West_Bergen,South_Bergen_W] = W_car_to_S
car_travel_time_congestion[West_Bergen,North_Bergen_W] = W_car_to_N
car_travel_time_congestion[West_Bergen,East_Bergen_W] = W_car_to_E
car_travel_time_congestion[West_Bergen,West_Bergen_W] = W_car_to_W
car_travel_time_free[Askoy,Askoy_W] = A_car_to_A_2
car_travel_time_free[Askoy,Center_Bergen_W] = A_car_to_C_2
car_travel_time_free[Askoy,South_Bergen_W] = A_car_to_S_2
car_travel_time_free[Askoy,North_Bergen_W] = A_car_to_N_2
car_travel_time_free[Askoy,East_Bergen_W] = A_car_to_E_2
car_travel_time_free[Askoy,West_Bergen_W] = A_car_to_W_2

car_travel_time_free[Center_Bergen,Askoy_W] = C_car_to_A_2
car_travel_time_free[Center_Bergen,Center_Bergen_W] = C_car_to_C_2
car_travel_time_free[Center_Bergen,South_Bergen_W] = C_car_to_S_2
car_travel_time_free[Center_Bergen,North_Bergen_W] = C_car_to_N_2
car_travel_time_free[Center_Bergen,East_Bergen_W] = C_car_to_E_2
car_travel_time_free[Center_Bergen,West_Bergen_W] = C_car_to_W_2
car_travel_time_free[South_Bergen,Askoy_W] = S_car_to_A_2
car_travel_time_free[South_Bergen,Center_Bergen_W] = S_car_to_C_2
car_travel_time_free[South_Bergen,South_Bergen_W] = S_car_to_S_2
car_travel_time_free[South_Bergen,North_Bergen_W] = S_car_to_N_2
car_travel_time_free[South_Bergen,East_Bergen_W] = S_car_to_E_2
car_travel_time_free[South_Bergen,West_Bergen_W] = S_car_to_W_2
car_travel_time_free[North_Bergen,Askoy_W] = N_car_to_A_2
car_travel_time_free[North_Bergen,Center_Bergen_W] = N_car_to_C_2
car_travel_time_free[North_Bergen,South_Bergen_W] = N_car_to_S_2
car_travel_time_free[North_Bergen,North_Bergen_W] = N_car_to_N_2
car_travel_time_free[North_Bergen,East_Bergen_W] = N_car_to_E_2
car_travel_time_free[North_Bergen,West_Bergen_W] = N_car_to_W_2
car_travel_time_free[East_Bergen,Askoy_W] = E_car_to_A_2
car_travel_time_free[East_Bergen,Center_Bergen_W] = E_car_to_C_2
car_travel_time_free[East_Bergen,South_Bergen_W] = E_car_to_S_2
car_travel_time_free[East_Bergen,North_Bergen_W] = E_car_to_N_2
car_travel_time_free[East_Bergen,East_Bergen_W] = E_car_to_E_2
car_travel_time_free[East_Bergen,West_Bergen_W] = E_car_to_W_2
car_travel_time_free[West_Bergen,Askoy_W] = W_car_to_A_2
car_travel_time_free[West_Bergen,Center_Bergen_W] = W_car_to_C_2
car_travel_time_free[West_Bergen,South_Bergen_W] = W_car_to_S_2
car_travel_time_free[West_Bergen,North_Bergen_W] = W_car_to_N_2
car_travel_time_free[West_Bergen,East_Bergen_W] = W_car_to_E_2
car_travel_time_free[West_Bergen,West_Bergen_W] = W_car_to_W_2
commuter[Resident,Working] =
(Adult_single[Resident,Working]+Adult__married_or_cohibited[Resident,Working])*
employment_rate
C_car_to_A = GRAPH(TIME)
(2001, 0.187), (2002, 0.332), (2002, 0.386), (2003, 0.41), (2003, 0.421), (2004, 0.427), (2004, 0.429), (2005, 0.432), (2005, 0.435), (2006, 0.436), (2006, 0.436), (2007, 0.433), (2007, 0.432), (2008, 0.431), (2008, 0.43), (2009, 0.43), (2009, 0.43), (2010, 0.43), (2010, 0.43), (2011, 0.43), (2011, 0.431)
C_car_to_A_2 = GRAPH(TIME)
(2001, 0.00), (2002, 0.236), (2002, 0.347), (2003, 0.404), (2003, 0.436), (2004, 0.453), (2004, 0.463), (2005, 0.471), (2005, 0.477), (2006, 0.481), (2006, 0.483), (2007, 0.481), (2007, 0.48), (2008, 0.479), (2008, 0.478), (2009, 0.478), (2009, 0.478), (2010, 0.479), (2010, 0.479), (2011, 0.479), (2011, 0.48)
C_car_to_C = GRAPH(TIME)
(2001, -0.000276), (2002, 0.147), (2002, 0.2), (2003, 0.223), (2003, 0.234), (2004, 0.239), (2004, 0.242), (2005, 0.244), (2005, 0.246), (2006, 0.247), (2006, 0.247), (2007, 0.245), (2007, 0.244), (2008, 0.243), (2008, 0.243), (2009, 0.242), (2009, 0.242), (2010, 0.242), (2010, 0.242), (2011, 0.243), (2011, 0.243)
C_car_to_C_2 = GRAPH(TIME)

(2001, -0.241), (2002, -0.00267), (2002, 0.11), (2003, 0.168), (2003, 0.2), (2004, 0.218), (2004, 0.228), (2005, 0.236), (2005, 0.242), (2006, 0.245), (2006, 0.247), (2007, 0.245), (2007, 0.243), (2008, 0.242), (2008, 0.242), (2009, 0.242), (2009, 0.242), (2010, 0.242), (2010, 0.242), (2011, 0.243), (2011, 0.243)
C_car_to_E = GRAPH(TIME)
(2001, 0.36), (2002, 0.505), (2002, 0.559), (2003, 0.583), (2003, 0.594), (2004, 0.6), (2004, 0.602), (2005, 0.605), (2005, 0.608), (2006, 0.609), (2006, 0.609), (2007, 0.606), (2007, 0.605), (2008, 0.604), (2008, 0.603), (2009, 0.603), (2009, 0.603), (2010, 0.603), (2010, 0.603), (2011, 0.603), (2011, 0.603)
C_car_to_E_2 = GRAPH(TIME)
(2001, 0.00), (2002, 0.324), (2002, 0.487), (2003, 0.575), (2003, 0.625), (2004, 0.653), (2004, 0.669), (2005, 0.68), (2005, 0.689), (2006, 0.694), (2006, 0.696), (2007, 0.694), (2007, 0.693), (2008, 0.693), (2008, 0.692), (2009, 0.692), (2009, 0.692), (2010, 0.693), (2010, 0.693), (2011, 0.693), (2011, 0.693)
C_car_to_N = GRAPH(TIME)
(2001, 0.173), (2002, 0.318), (2002, 0.372), (2003, 0.396), (2003, 0.407), (2004, 0.413), (2004, 0.416), (2005, 0.418), (2005, 0.421), (2006, 0.422), (2006, 0.422), (2007, 0.419), (2007, 0.418), (2008, 0.417), (2008, 0.417), (2009, 0.416), (2009, 0.416), (2010, 0.416), (2010, 0.416), (2011, 0.416), (2011, 0.417)
C_car_to_N_2 = GRAPH(TIME)
(2001, 0.00), (2002, 0.229), (2002, 0.335), (2003, 0.39), (2003, 0.42), (2004, 0.436), (2004, 0.446), (2005, 0.453), (2005, 0.459), (2006, 0.463), (2006, 0.464), (2007, 0.462), (2007, 0.46), (2008, 0.459), (2008, 0.459), (2009, 0.459), (2009, 0.459), (2010, 0.459), (2010, 0.46), (2011, 0.46), (2011, 0.46)
C_car_to_S = GRAPH(TIME)
(2001, 0.12), (2002, 0.266), (2002, 0.321), (2003, 0.346), (2003, 0.357), (2004, 0.363), (2004, 0.367), (2005, 0.37), (2005, 0.374), (2006, 0.377), (2006, 0.378), (2007, 0.374), (2007, 0.372), (2008, 0.371), (2008, 0.37), (2009, 0.369), (2009, 0.369), (2010, 0.369), (2010, 0.369), (2011, 0.369), (2011, 0.37)
C_car_to_S_2 = GRAPH(TIME)
(2001, 0.00), (2002, 0.202), (2002, 0.293), (2003, 0.339), (2003, 0.363), (2004, 0.377), (2004, 0.384), (2005, 0.391), (2005, 0.397), (2006, 0.401), (2006, 0.403), (2007, 0.4), (2007, 0.398), (2008, 0.397), (2008, 0.396), (2009, 0.396), (2009, 0.396), (2010, 0.396), (2010, 0.396), (2011, 0.396), (2011, 0.397)
C_car_to_W = GRAPH(TIME)
(2001, 0.12), (2002, 0.266), (2002, 0.32), (2003, 0.343), (2003, 0.355), (2004, 0.36), (2004, 0.363), (2005, 0.365), (2005, 0.368), (2006, 0.37), (2006, 0.37), (2007, 0.367), (2007, 0.365), (2008, 0.364), (2008, 0.364), (2009, 0.363), (2009, 0.363), (2010, 0.363), (2010, 0.364), (2011, 0.364), (2011, 0.364)
C_car_to_W_2 = GRAPH(TIME)
(2001, 0.00), (2002, 0.202), (2002, 0.293), (2003, 0.338), (2003, 0.362), (2004, 0.376), (2004, 0.383), (2005, 0.389), (2005, 0.395), (2006, 0.398), (2006, 0.4), (2007, 0.398), (2007, 0.397), (2008, 0.396), (2008, 0.395), (2009, 0.395), (2009, 0.395), (2010, 0.395), (2010, 0.396), (2011, 0.396), (2011, 0.396)
C_PT_to_A = GRAPH(TIME)
(2001, 0.68), (2002, 0.657), (2002, 0.643), (2003, 0.635), (2003, 0.63), (2004, 0.628), (2004, 0.626), (2005, 0.625), (2005, 0.625), (2006, 0.624), (2006, 0.624), (2007, 0.624), (2007, 0.624), (2008, 0.624), (2008, 0.624), (2009, 0.624), (2009, 0.624), (2010, 0.624), (2010, 0.624), (2011, 0.624), (2011, 0.624)
C_PT_to_A_2 = GRAPH(TIME)

(2001, 0.852), (2002, 0.775), (2002, 0.724), (2003, 0.692), (2003, 0.674), (2004, 0.663), (2004, 0.657), (2005, 0.654), (2005, 0.652), (2006, 0.65), (2006, 0.65), (2007, 0.65), (2007, 0.65), (2008, 0.65), (2008, 0.651), (2009, 0.651), (2009, 0.652), (2010, 0.652), (2010, 0.652), (2011, 0.652), (2011, 0.652)
C_PT_to_C = GRAPH(TIME)
(2001, 0.512), (2002, 0.487), (2002, 0.471), (2003, 0.463), (2003, 0.457), (2004, 0.454), (2004, 0.453), (2005, 0.451), (2005, 0.451), (2006, 0.451), (2006, 0.45), (2007, 0.45), (2007, 0.45), (2008, 0.45), (2008, 0.45), (2009, 0.45), (2009, 0.45), (2010, 0.45), (2010, 0.45), (2011, 0.45), (2011, 0.45)
C_PT_to_C_2 = GRAPH(TIME)
(2001, -0.241), (2002, -0.00267), (2002, 0.11), (2003, 0.168), (2003, 0.2), (2004, 0.218), (2004, 0.228), (2005, 0.236), (2005, 0.242), (2006, 0.245), (2006, 0.247), (2007, 0.245), (2007, 0.243), (2008, 0.242), (2008, 0.242), (2009, 0.242), (2009, 0.242), (2010, 0.242), (2010, 0.242), (2011, 0.243), (2011, 0.243)
C_PT_to_E = GRAPH(TIME)
(2001, 0.9), (2002, 0.857), (2002, 0.832), (2003, 0.818), (2003, 0.809), (2004, 0.804), (2004, 0.801), (2005, 0.799), (2005, 0.798), (2006, 0.798), (2006, 0.798), (2007, 0.797), (2007, 0.797), (2008, 0.797), (2008, 0.797), (2009, 0.797), (2009, 0.797), (2010, 0.797), (2010, 0.797), (2011, 0.797), (2011, 0.797)
C_PT_to_E_2 = GRAPH(TIME)
(2001, 1.05), (2002, 1.27), (2002, 1.24), (2003, 1.23), (2003, 1.22), (2004, 1.22), (2004, 1.22), (2005, 1.22), (2005, 1.21), (2006, 1.21), (2006, 1.21), (2007, 1.21), (2007, 1.21), (2008, 1.21), (2008, 1.21), (2009, 1.21), (2009, 1.21), (2010, 1.21), (2010, 1.22), (2011, 1.22), (2011, 1.22)
C_PT_to_N = GRAPH(TIME)
(2001, 0.67), (2002, 0.64), (2002, 0.622), (2003, 0.612), (2003, 0.605), (2004, 0.602), (2004, 0.6), (2005, 0.599), (2005, 0.598), (2006, 0.598), (2006, 0.597), (2007, 0.597), (2007, 0.597), (2008, 0.597), (2008, 0.597), (2009, 0.597), (2009, 0.597), (2010, 0.597), (2010, 0.597), (2011, 0.597), (2011, 0.597)
C_PT_to_N_2 = GRAPH(TIME)
(2001, 0.885), (2002, 0.911), (2002, 0.886), (2003, 0.872), (2003, 0.863), (2004, 0.858), (2004, 0.856), (2005, 0.854), (2005, 0.854), (2006, 0.854), (2006, 0.855), (2007, 0.855), (2007, 0.856), (2008, 0.856), (2008, 0.857), (2009, 0.857), (2009, 0.857), (2010, 0.858), (2010, 0.858), (2011, 0.858), (2011, 0.859)
C_PT_to_S = GRAPH(TIME)
(2001, 0.6), (2002, 0.585), (2002, 0.576), (2003, 0.571), (2003, 0.568), (2004, 0.566), (2004, 0.565), (2005, 0.565), (2005, 0.564), (2006, 0.564), (2006, 0.564), (2007, 0.564), (2007, 0.564), (2008, 0.564), (2008, 0.564), (2009, 0.564), (2009, 0.564), (2010, 0.564), (2010, 0.564), (2011, 0.564), (2011, 0.564)
C_PT_to_S_2 = GRAPH(TIME)
(2001, 0.782), (2002, 0.765), (2002, 0.762), (2003, 0.76), (2003, 0.76), (2004, 0.759), (2004, 0.759), (2005, 0.76), (2005, 0.762), (2006, 0.764), (2006, 0.707), (2007, 0.708), (2007, 0.709), (2008, 0.709), (2008, 0.709), (2009, 0.709), (2009, 0.71), (2010, 0.71), (2010, 0.71), (2011, 0.71), (2011, 0.711)
C_PT_to_W = GRAPH(TIME)
(2001, 0.6), (2002, 0.582), (2002, 0.572), (2003, 0.566), (2003, 0.562), (2004, 0.56), (2004, 0.559), (2005, 0.558), (2005, 0.558), (2006, 0.557), (2006, 0.557), (2007, 0.557), (2007, 0.557), (2008, 0.557), (2008, 0.557), (2009, 0.557), (2009, 0.557), (2010, 0.557), (2010, 0.557), (2011, 0.557), (2011, 0.557)
C_PT_to_W_2 = GRAPH(TIME)

(2001, 0.782), (2002, 0.763), (2002, 0.76), (2003, 0.758), (2003, 0.756), (2004, 0.755), (2004, 0.755), (2005, 0.755), (2005, 0.756), (2006, 0.758), (2006, 0.76), (2007, 0.761), (2007, 0.762), (2008, 0.763), (2008, 0.763), (2009, 0.764), (2009, 0.764), (2010, 0.764), (2010, 0.765), (2011, 0.766), (2011, 0.766)
DAR_get_vacancy[Resident,Working] = Search_effect_DAR[Resident,Working]*Frac_seek_get_vacancy[Resident]
DAR_get_vacancy_sort_by_WP[Working] = ARRAYSUM(DAR_get_vacancy[*,Working])
DAR_seeker_car_WP[Working] = DAR_sorted_by_working_place[Working]*frac_owing_car
DAR_seeker_PT_WP[Working] = DAR_sorted_by_working_place[Working]*(1-frac_owing_car)
DAR_sorted_by_working_place[Working] = ARRAYSUM(Divorced__adult_relocater[*,Working])
death_leaving_job[Working] = ARRAYSUM(Death_of_AMC[*,Working])+ARRAYSUM(Death_of_AS[*,Working])
Death_rate_adult = GRAPH(TIME)
(2001, 0.00292), (2002, 0.00287), (2003, 0.00287), (2004, 0.00265), (2005, 0.00268), (2006, 0.00255), (2007, 0.00278), (2008, 0.00272), (2009, 0.00261), (2010, 0.00268), (2011, 0.00261)
Death_rate_juveniles = GRAPH(TIME)
(2001, 0.000399), (2002, 0.000388), (2003, 0.000361), (2004, 0.000288), (2005, 0.000441), (2006, 0.000327), (2007, 0.000279), (2008, 0.000256), (2009, 0.000229), (2010, 0.000228), (2011, 0.000334)
Death_rate_pensioner = GRAPH(TIME)
(2001, 0.0554), (2002, 0.0586), (2003, 0.0542), (2004, 0.0526), (2005, 0.049), (2006, 0.0524), (2007, 0.0518), (2008, 0.0521), (2009, 0.0494), (2010, 0.0509), (2011, 0.0524)
decision_time_for_adjust_living_condition = 1
demand_for_high_density[Resident] = ARRAYSUM(family_seeking_high_density[Resident,*])+ARRAYSUM(single_seeking_high_density[Resident,*])
demand_for_low_density[Resident] = ARRAYSUM(family_seeking_low_density[Resident,*])+ARRAYSUM(Single_seeking_low_density[Resident,*])
demand_supply_ratio_work[Working] = 1/Frac_get_job[Working]
Destinaion_choice_NOJS_car[Resident,Working] = frac_choice_sort_by_resident_placee_car_f[Resident,Working]*NOJS_car[Resident]
Destinaion_choice_SOJS_car[Resident,Working] = frac_choice_sort_by_resident_placee_car_s[Resident,Working]*SOJS_car[Resident]
Destinasion_choice_AFR_car[Resident,Working] = AFR_seeker_car_WP[Working]*frac_choice_sort_by_working_place_car_f[Resident,Working]
Destinasion_choice_AFR_PT[Resident,Working] = AFR_seeker_PT_WP[Working]*frac_choice_sort_by_working_place_PT_f[Resident,Working]
Destinasion_choice_DAR_car[Resident,Working] = frac_choice_sort_by_working_placee_car_s[Resident,Working]*DAR_seeker_car_WP[Working]

Destinasion_choice_DAR_PT[Resident,Working] =
frac_choice_sort_by_working_place_PT_s[Resident,Working]*DAR_seeker_PT_WP[Working]
Destinasion_choice_FFB_car[Resident,Working] =
frac_choice_sort_by_working_place_car_f[Resident,Working]*FFTB_seeker_car[Working]
Destinasion_choice_FFTB_PT[Resident,Working] =
frac_choice_sort_by_working_place_PT_f[Resident,Working]*FFTB_seeker_PT[Working]
Destinasion_choice_SAR_car[Resident,Working] =
frac_choice_sort_by_working_placee_car_s[Resident,Working]*SAR_seeker_car_WP[Working]
Destinasion_choice_SAR_PT[Resident,Working] =
frac_choice_sort_by_working_place_PT_s[Resident,Working]*SAR_seeker_PT_WP[Working]
Destinasion_choice_SFTB_car[Resident,Working] =
frac_choice_sort_by_working_placee_car_s[Resident,Working]*SFTB_seeker_car[Working]
Destinasion_choice_SFTB_PT[Resident,Working] =
frac_choice_sort_by_working_place_PT_s[Resident,Working]*SFTB_seeker_PT[Working]
Destinasion_choice_YS_car[Resident,Working] =
frac_choice_sort_by_working_placee_car_s[Resident,Working]*YS_seeker_car[Working]
Destinasion_choice_YS_PT[Resident,Working] =
frac_choice_sort_by_working_place_PT_s[Resident,Working]*YS_seeker_PT[Working]
Destinasion_SPR[Resident,Working] =
(Single__pensioner_relocater[Resident,Working])/dt
DEstination_choice_NOJS_PT[Resident,Working] =
frac_choice_sort_by_resident_place_PT_f[Resident,Working]*NOJS_PT[Resident]
DEstination_choice_SOJS_PT[Resident,Working] =
frac_choice_sort_by_resident_place_PT_s[Resident,Working]*SOJS_PT[Resident]
Divorce_rate = GRAPH(TIME)
(2001, 0.00611), (2002, 0.00541), (2003, 0.00566), (2004, 0.00629), (2005, 0.00537), (2006, 0.00569), (2007, 0.00522), (2008, 0.00479), (2009, 0.00489), (2010, 0.00516), (2011, 0.00481)
Emigration_adult[Resident,Working] =
EXO_emigration_adult/Total_adult_population*adult_with_resident_and_work[Resident,Working]*(1-STEP(1,2012))+emmigration_adult_proj/Total_adult_population*adult_with_resident_and_work[Resident,Working]*(0+STEP(1,2012))
emigration_AMC[Resident,Working] =
AFR__leaving_old_dwelling[Resident,Working]*2+Emigration_adult[Resident,Working]*(1-frac_single_in_migration_adult)
emigration_AS[Resident,Working] =
intra_urban_emigration[Resident,Working]+inter_urban_emigraion_AS[Resident,Working]
emigration_FMC[Resident,Working] =
Emigration_pensioner[Resident,Working]*(1-frac_single_in_migration_pensioner)

emigration_household[Resident,Working] = Emigration_adult[Resident,Working]*frac_single_in_migration_adult+Emigration_adult[Resident,Working]*(1-frac_single_in_migration_adult)/2+Emigration_pensioner[Resident,Working]*frac_single_in_migration_pensioner+Emigration_pensioner[Resident,Working]*(1-frac_single_in_migration_pensioner)/2
emigration_leaving_dwellings[Resident] = ARRAYSUM(emigration_household[Resident,*])
Emigration_leaving_job[Working] = ARRAYSUM(Emigration_adult[*,Working])
Emigration_pensioner[Resident,Working] = EXO_emigration_pensioner/total_pensioner_population*pensioner_with_work_and_resident[Resident,Working]*(1-STEP(1,2012))+emmigration_pensioner_proj/total_pensioner_population*pensioner_with_work_and_resident[Resident,Working]*(0+STEP(1,2012))
emigration_PS[Resident,Working] = SPR_moving_out[Resident,Working]+inter_urabn_emigraion_PS[Resident,Working]
emmigration_adult_proj = GRAPH(TIME)
(2012, 8481), (2013, 8661), (2013, 8828), (2014, 9017), (2015, 9219), (2015, 9433), (2016, 9630), (2017, 9826), (2017, 10010), (2018, 10193), (2019, 10356), (2019, 10515), (2020, 10668), (2021, 10820), (2021, 10967), (2022, 11102), (2023, 11241), (2023, 11377), (2024, 11508), (2025, 11641), (2025, 11769), (2026, 11908), (2027, 12042), (2027, 12178), (2028, 12313), (2029, 12450), (2029, 12597), (2030, 12744)
emmigration_pensioner_proj = GRAPH(TIME)
(2012, 839), (2013, 361), (2013, 507), (2014, 659), (2015, 692), (2015, 675), (2016, 853), (2017, 605), (2017, 627), (2018, 575), (2019, 488), (2019, 399), (2020, 437), (2021, 319), (2021, 150), (2022, 253), (2023, 130), (2023, 23.9), (2024, 10.2), (2025, -38.0), (2025, -136), (2026, -273), (2027, -346), (2027, -445), (2028, -498), (2029, -439), (2029, -478), (2030, -499)
employment_rate = 0.7
EXO_emigration_adult = GRAPH(time)
(2001, 5459), (2002, 5553), (2003, 5732), (2004, 6147), (2005, 6061), (2006, 6940), (2007, 7195), (2008, 6509), (2009, 6975), (2010, 7706), (2011, 8835)
EXO_emigration_juveniles = GRAPH(time)
(2001, 1841), (2002, 1873), (2003, 1934), (2004, 2073), (2005, 2165), (2006, 2218), (2007, 2270), (2008, 1975), (2009, 2075), (2010, 2257), (2011, 2429)
EXO_emigration_pensioner = GRAPH(time)
(2001, 1337), (2002, 1360), (2003, 1404), (2004, 1506), (2005, 1540), (2006, 1679), (2007, 1740), (2008, 1559), (2009, 1616), (2010, 1806), (2011, 2060)
EXO_imigration_adult = GRAPH(time)
(2001, 7943), (2002, 7853), (2003, 7951), (2004, 8094), (2005, 9087), (2006, 9654), (2007, 10638), (2008, 10584), (2009, 11041), (2010, 11282), (2011, 12548)
EXO_imigration_juveniles = GRAPH(time)
(2001, 2278), (2002, 2253), (2003, 2280), (2004, 2322), (2005, 2380), (2006, 2538), (2007, 2736), (2008, 2724), (2009, 2836), (2010, 3012), (2011, 2965)
E_car_to_A = GRAPH(TIME)
(2001, 0.534), (2002, 0.682), (2002, 0.737), (2003, 0.761), (2003, 0.772), (2004, 0.777), (2004, 0.779), (2005, 0.782), (2005, 0.784), (2006, 0.784), (2006, 0.783), (2007, 0.779), (2007, 0.776), (2008, 0.775), (2008, 0.774), (2009, 0.773), (2009, 0.773), (2010, 0.773), (2010, 0.773), (2011, 0.773), (2011, 0.773)
E_car_to_A_2 = GRAPH(TIME)

(2001, 0.00), (2002, 0.415), (2002, 0.63), (2003, 0.749), (2003, 0.816), (2004, 0.854), (2004, 0.877), (2005, 0.891), (2005, 0.902), (2006, 0.908), (2006, 0.91), (2007, 0.908), (2007, 0.907), (2008, 0.906), (2008, 0.905), (2009, 0.905), (2009, 0.905), (2010, 0.905), (2010, 0.905), (2011, 0.905), (2011, 0.905)
E_car_to_C = GRAPH(TIME)
(2001, 0.36), (2002, 0.518), (2002, 0.582), (2003, 0.613), (2003, 0.631), (2004, 0.643), (2004, 0.65), (2005, 0.661), (2005, 0.678), (2006, 0.695), (2006, 0.709), (2007, 0.719), (2007, 0.728), (2008, 0.733), (2008, 0.736), (2009, 0.738), (2009, 0.739), (2010, 0.74), (2010, 0.741), (2011, 0.742), (2011, 0.743)
E_car_to_C_2 = GRAPH(TIME)
(2001, 0.00), (2002, 0.338), (2002, 0.509), (2003, 0.601), (2003, 0.653), (2004, 0.683), (2004, 0.7), (2005, 0.714), (2005, 0.729), (2006, 0.742), (2006, 0.752), (2007, 0.757), (2007, 0.761), (2008, 0.763), (2008, 0.764), (2009, 0.765), (2009, 0.767), (2010, 0.768), (2010, 0.771), (2011, 0.773), (2011, 0.776)
E_car_to_E = GRAPH(TIME)
(2001, 0.00486), (2002, 0.158), (2002, 0.215), (2003, 0.241), (2003, 0.254), (2004, 0.261), (2004, 0.265), (2005, 0.269), (2005, 0.273), (2006, 0.277), (2006, 0.279), (2007, 0.278), (2007, 0.279), (2008, 0.279), (2008, 0.279), (2009, 0.279), (2009, 0.279), (2010, 0.279), (2010, 0.28), (2011, 0.28), (2011, 0.281)
E_car_to_E_2 = GRAPH(TIME)
(2001, -0.456), (2002, -0.13), (2002, 0.0343), (2003, 0.123), (2003, 0.173), (2004, 0.202), (2004, 0.218), (2005, 0.23), (2005, 0.238), (2006, 0.243), (2006, 0.245), (2007, 0.243), (2007, 0.242), (2008, 0.241), (2008, 0.241), (2009, 0.241), (2009, 0.241), (2010, 0.242), (2010, 0.242), (2011, 0.243), (2011, 0.243)
E_car_to_N = GRAPH(TIME)
(2001, 0.187), (2002, 0.332), (2002, 0.386), (2003, 0.41), (2003, 0.422), (2004, 0.427), (2004, 0.43), (2005, 0.432), (2005, 0.435), (2006, 0.437), (2006, 0.437), (2007, 0.434), (2007, 0.433), (2008, 0.432), (2008, 0.431), (2009, 0.431), (2009, 0.431), (2010, 0.431), (2010, 0.431), (2011, 0.432), (2011, 0.432)
E_car_to_N_2 = GRAPH(TIME)
(2001, 0.00), (2002, 0.236), (2002, 0.347), (2003, 0.404), (2003, 0.435), (2004, 0.453), (2004, 0.463), (2005, 0.47), (2005, 0.476), (2006, 0.48), (2006, 0.481), (2007, 0.479), (2007, 0.478), (2008, 0.477), (2008, 0.476), (2009, 0.476), (2009, 0.477), (2010, 0.477), (2010, 0.477), (2011, 0.477), (2011, 0.478)
E_car_to_S = GRAPH(TIME)
(2001, 0.253), (2002, 0.398), (2002, 0.452), (2003, 0.476), (2003, 0.487), (2004, 0.493), (2004, 0.495), (2005, 0.498), (2005, 0.501), (2006, 0.502), (2006, 0.502), (2007, 0.499), (2007, 0.497), (2008, 0.497), (2008, 0.496), (2009, 0.496), (2009, 0.496), (2010, 0.496), (2010, 0.496), (2011, 0.496), (2011, 0.496)
E_car_to_S_2 = GRAPH(TIME)
(2001, 0.00), (2002, 0.269), (2002, 0.4), (2003, 0.469), (2003, 0.507), (2004, 0.529), (2004, 0.541), (2005, 0.55), (2005, 0.557), (2006, 0.562), (2006, 0.563), (2007, 0.561), (2007, 0.56), (2008, 0.559), (2008, 0.559), (2009, 0.558), (2009, 0.559), (2010, 0.559), (2010, 0.559), (2011, 0.559), (2011, 0.56)
E_car_to_W = GRAPH(TIME)
(2001, 0.462), (2002, 0.609), (2002, 0.664), (2003, 0.688), (2003, 0.699), (2004, 0.704), (2004, 0.706), (2005, 0.708), (2005, 0.71), (2006, 0.71), (2006, 0.708), (2007, 0.704), (2007, 0.701), (2008, 0.7), (2008, 0.699), (2009, 0.698), (2009, 0.698), (2010, 0.697), (2010, 0.697), (2011, 0.697), (2011, 0.698)
E_car_to_W_2 = GRAPH(TIME)

(2001, 0.00), (2002, 0.378), (2002, 0.572), (2003, 0.678), (2003, 0.737), (2004, 0.771), (2004, 0.791), (2005, 0.804), (2005, 0.813), (2006, 0.818), (2006, 0.82), (2007, 0.818), (2007, 0.816), (2008, 0.815), (2008, 0.814), (2009, 0.814), (2009, 0.814), (2010, 0.814), (2010, 0.814), (2011, 0.814), (2011, 0.814)
E_PT_to_A = GRAPH(TIME)
(2001, 1.10), (2002, 1.02), (2002, 0.968), (2003, 0.938), (2003, 0.92), (2004, 0.909), (2004, 0.903), (2005, 0.899), (2005, 0.896), (2006, 0.895), (2006, 0.894), (2007, 0.893), (2007, 0.892), (2008, 0.892), (2008, 0.892), (2009, 0.892), (2009, 0.891), (2010, 0.891), (2010, 0.891), (2011, 0.891), (2011, 0.891)
E_PT_to_A_2 = GRAPH(TIME)
(2001, 1.17), (2002, 1.59), (2002, 1.58), (2003, 1.57), (2003, 1.57), (2004, 1.56), (2004, 1.56), (2005, 1.56), (2005, 1.56), (2006, 1.56), (2006, 1.56), (2007, 1.56), (2007, 1.56), (2008, 1.56), (2008, 1.56), (2009, 1.56), (2009, 1.56), (2010, 1.56), (2010, 1.56), (2011, 1.56), (2011, 1.56)
E_PT_to_C = GRAPH(TIME)
(2001, 0.92), (2002, 0.884), (2002, 0.865), (2003, 0.855), (2003, 0.851), (2004, 0.85), (2004, 0.851), (2005, 0.854), (2005, 0.863), (2006, 0.872), (2006, 0.881), (2007, 0.889), (2007, 0.895), (2008, 0.899), (2008, 0.901), (2009, 0.903), (2009, 0.904), (2010, 0.905), (2010, 0.906), (2011, 0.907), (2011, 0.909)
E_PT_to_C_2 = GRAPH(TIME)
(2001, 1.07), (2002, 1.31), (2002, 1.29), (2003, 1.27), (2003, 1.26), (2004, 1.26), (2004, 1.26), (2005, 1.26), (2005, 1.26), (2006, 1.27), (2006, 1.28), (2007, 1.29), (2007, 1.29), (2008, 1.29), (2008, 1.29), (2009, 1.30), (2009, 1.30), (2010, 1.30), (2010, 1.30), (2011, 1.30), (2011, 1.31)
E_PT_to_E = GRAPH(TIME)
(2001, 0.574), (2002, 0.522), (2002, 0.492), (2003, 0.474), (2003, 0.464), (2004, 0.458), (2004, 0.454), (2005, 0.452), (2005, 0.451), (2006, 0.45), (2006, 0.45), (2007, 0.45), (2007, 0.45), (2008, 0.45), (2008, 0.45), (2009, 0.45), (2009, 0.45), (2010, 0.45), (2010, 0.45), (2011, 0.45), (2011, 0.45)
E_PT_to_E_2 = GRAPH(TIME)
(2001, -0.456), (2002, -0.13), (2002, 0.0343), (2003, 0.123), (2003, 0.173), (2004, 0.202), (2004, 0.218), (2005, 0.23), (2005, 0.238), (2006, 0.243), (2006, 0.245), (2007, 0.243), (2007, 0.242), (2008, 0.241), (2008, 0.241), (2009, 0.241), (2009, 0.241), (2010, 0.242), (2010, 0.242), (2011, 0.243), (2011, 0.243)
E_PT_to_N = GRAPH(TIME)
(2001, 0.68), (2002, 0.668), (2002, 0.662), (2003, 0.658), (2003, 0.655), (2004, 0.654), (2004, 0.653), (2005, 0.653), (2005, 0.652), (2006, 0.652), (2006, 0.652), (2007, 0.652), (2007, 0.652), (2008, 0.652), (2008, 0.652), (2009, 0.652), (2009, 0.653), (2010, 0.653), (2010, 0.653), (2011, 0.653), (2011, 0.653)
E_PT_to_N_2 = GRAPH(TIME)
(2001, 0.904), (2002, 0.947), (2002, 0.921), (2003, 0.906), (2003, 0.896), (2004, 0.891), (2004, 0.887), (2005, 0.885), (2005, 0.885), (2006, 0.885), (2006, 0.885), (2007, 0.885), (2007, 0.885), (2008, 0.886), (2008, 0.886), (2009, 0.886), (2009, 0.887), (2010, 0.887), (2010, 0.888), (2011, 0.888), (2011, 0.889)
E_PT_to_S = GRAPH(TIME)
(2001, 0.77), (2002, 0.726), (2002, 0.7), (2003, 0.685), (2003, 0.676), (2004, 0.671), (2004, 0.668), (2005, 0.666), (2005, 0.665), (2006, 0.664), (2006, 0.664), (2007, 0.664), (2007, 0.664), (2008, 0.663), (2008, 0.663), (2009, 0.663), (2009, 0.663), (2010, 0.663), (2010, 0.663), (2011, 0.663), (2011, 0.663)
E_PT_to_S_2 = GRAPH(TIME)

(2001, 0.912), (2002, 1.00), (2002, 1.00), (2003, 1.00), (2003, 1.00), (2004, 1.00),
(2004, 1.00), (2005, 1.00), (2005, 1.00), (2006, 1.00), (2006, 1.00), (2007, 1.00),
(2007, 1.00), (2008, 1.00), (2008, 1.00), (2009, 1.00), (2009, 1.00), (2010, 1.00),
(2010, 1.00), (2011, 1.00), (2011, 1.01)
E_PT_to_W = GRAPH(TIME)
(2001, 1.02), (2002, 0.944), (2002, 0.896), (2003, 0.868), (2003, 0.851), (2004, 0.841),
(2004, 0.835), (2005, 0.832), (2005, 0.829), (2006, 0.828), (2006, 0.827), (2007,
0.826), (2007, 0.826), (2008, 0.825), (2008, 0.825), (2009, 0.825), (2009, 0.825),
(2010, 0.825), (2010, 0.825), (2011, 0.825), (2011, 0.824)
E_PT_to_W_2 = GRAPH(TIME)
(2001, 1.11), (2002, 1.43), (2002, 1.43), (2003, 1.42), (2003, 1.42), (2004, 1.42),
(2004, 1.42), (2005, 1.42), (2005, 1.41), (2006, 1.41), (2006, 1.41), (2007, 1.41),
(2007, 1.41), (2008, 1.41), (2008, 1.41), (2009, 1.41), (2009, 1.41), (2010, 1.41),
(2010, 1.41), (2011, 1.41), (2011, 1.41)
family_household[Resident,Working] =
Adult__married_or_cohibited[Resident,Working]/2
family_immigration_household[Working] =
IJS_get_job[Working]*(1-frac_single_in_migration_adult)/2
family_seeking_high_density[Resident,Working] =
Frac_family_chooing_High_density[Resident,Working]*total_family_search_effect[Resident,Working]
family_seeking_low_density[Resident,Working] =
(1-Frac_family_chooing_High_density[Resident,Working])*total_family_search_effect[Resident,Working]
FFB_get_vacancy_sort_by_WP[Working] =
ARRAYSUM(FFTB_get_vacancy[*,Working])
FFTB_get_vacancy[Resident,Working] =
Search_effect_FFTB[Resident,Working]*Frac_seek_get_vacancy[Resident]
FFTB_seeker_car[Working] = Family_first_time_buyer[Working]*frac_owing_car
FFTB_seeker_PT[Working] =
Family_first_time_buyer[Working]*(1-frac_owing_car)
frac_choice_sort_by_resident_placee_car_f[Resident,Working] =
workplace_utility_car_f[Resident,Working]/total_utility_car_worker_f[Resident]*(1-switch_logit)+switch_logit*frac_job_seeker_f[Resident,Working]
frac_choice_sort_by_resident_placee_car_s[Resident,Working] =
workplace_utility_car_s[Resident,Working]/total_utility_car_worker_s[Resident]*(1-switch_logit)+switch_logit*frac_job_seeker_s[Resident,Working]
frac_choice_sort_by_resident_place_PT_f[Resident,Working] =
workplace_utility_PT_f[Resident,Working]/total_utility_PT_worker_f[Resident]*(1-switch_logit)+switch_logit*frac_job_seeker_f[Resident,Working]
frac_choice_sort_by_resident_place_PT_s[Resident,Working] =
workplace_utility_PT_s[Resident,Working]/total_utility_PT_worker_s[Resident]*(1-switch_logit)+switch_logit*frac_job_seeker_s[Resident,Working]
frac_choice_sort_by_working_placee_car_s[Resident,Working] =
resident_utility_car_s[Resident,Working]/total_utility_car_s[Working]*(1-switch_logit)+frac_s[Resident,Working]*switch_logit
frac_choice_sort_by_working_place_car_f[Resident,Working] =
resident_utility_car_f[Resident,Working]/total_utility_car_f[Working]*(1-switch_logit)+frac_f[Resident,Working]*switch_logit

frac_choice_sort_by_working_place_PT_f[Resident,Working] =
resident_utility_PT_f[Resident,Working]/total_utility_PT_f[Working]*(1-switch_logit)+frac_f[Resident,Working]*switch_logit
frac_choice_sort_by_working_place_PT_s[Resident,Working] =
resident_utility_PT_S[Resident,Working]/total_utility_PT_S[Working]*(1-switch_logit)+frac_s[Resident,Working]*switch_logit
Frac_cohibit_to_marriage = GRAPH(TIME)
(2001, 0.6), (2011, 0.7)
FRAC_commuters[Resident,Working] =
adult_with_resident_and_work[Resident,Working]/ARRAYSUM(adult_with_resident_and_work[*,Working])
frac_demand_for_high_density[Resident] =
demand_for_high_density[Resident]/(demand_for_high_density[Resident]+demand_for_low_density[Resident])
frac_f[Resident,Working] =
[0.878790481321,0.0671367476708,0.043420637753,0.0222686246833,0.0134465915751,0.121534220282,
0.0170482044773,0.227103534288,0.141355492808,0.117806170825,0.0659775229366,0.123556832847,
0.0213554749484,0.254638630595,0.542313736146,0.0686675984988,0.106339170155,0.154773768234,
0.0111946996208,0.137234802746,0.0630867507232,0.636291995711,0.132169792721,0.0811335663917,
0.00239908688244,0.0367894457294,0.0306112240292,0.0630552076778,0.626568136185,0.0173873780851,
0.0692120527505,0.27709683897,0.17921215854,0.0919104026041,0.055498786427,0.50161423416]
Frac_family_chooing_High_density[Resident,Working] =
MAX_frac_family_chooing_High_density/(1+EXP(-relative_negative_family_commuting_time[Resident,Working]*b_for_family_choosing_High_density))
Frac_get_job[Working] = job_offers[Working]/Total_search_effect[Working]
frac_job_seeker_f[Resident,Working] =
[0.49479838363,0.238280427515,0.0864933147733,0.0177733322703,0.00353325280072,0.15912128901,
0.00700476288458,0.588198387838,0.205480634151,0.0686144156783,0.0126511606124,0.118050638836,
0.00527036496722,0.396132878662,0.473505988632,0.0240223699812,0.0122474110192,0.0888209867385,
0.00497152784774,0.384172998849,0.0991194780549,0.400558938765,0.0273923448399,0.0837847116434,
0.00313678517218,0.303212561177,0.14159988847,0.11686710795,0.382319698869,0.0528639583621,
0.0183462365939,0.46300104916,0.168064561168,0.0345352388949,0.00686543905741,0.309187475126]
frac_job_seeker_s[Resident,Working] =
[0.49479838363,0.238280427515,0.0864933147733,0.0177733322703,0.00353325280072,0.15912128901,
0.00700476288458,0.588198387838,0.205480634151,0.0686144156783,0.0126511606124,0.118050638836,

0.00527036496722,0.396132878662,0.473505988632,0.0240223699812,0.01224741
10192,0.0888209867385,
0.00497152784774,0.384172998849,0.0991194780549,0.400558938765,0.02739234
48399,0.0837847116434,
0.00313678517218,0.303212561177,0.14159988847,0.11686710795,0.38231969886
9,0.0528639583621,
0.0183462365939,0.46300104916,0.168064561168,0.0345352388949,0.0068654390
5741,0.309187475126]
Frac_Job_vacancy[Working] =
Job_vacancy[Working]/ARRAYSUM(Job_vacancy[*])
frac_move_by_crowd = (1-0.35)
frac_move_by_deserted = 1-0.811
frac_new_dwellings[Askoy] = GRAPH(TIME)
(2001, 0.12), (2002, 0.146), (2003, 0.104), (2004, 0.103), (2005, 0.102), (2006, 0.104), (2007, 0.159), (2008, 0.192), (2009, 0.124), (2010, 0.179), (2011, 0.136)
frac_new_dwellings[Center_Bergen] = GRAPH(TIME)
(2001, 0.203), (2002, 0.14), (2003, 0.257), (2004, 0.3), (2005, 0.224), (2006, 0.267), (2007, 0.198), (2008, 0.0691), (2009, 0.217), (2010, 0.205), (2011, 0.213)
frac_new_dwellings[South_Bergen] = GRAPH(TIME)
(2001, 0.393), (2002, 0.436), (2003, 0.379), (2004, 0.323), (2005, 0.439), (2006, 0.308), (2007, 0.319), (2008, 0.353), (2009, 0.222), (2010, 0.236), (2011, 0.415)
frac_new_dwellings[North_Bergen] = GRAPH(TIME)
(2001, 0.0639), (2002, 0.113), (2003, 0.135), (2004, 0.128), (2005, 0.0899), (2006, 0.0626), (2007, 0.108), (2008, 0.0891), (2009, 0.103), (2010, 0.118), (2011, 0.0926)
frac_new_dwellings[East_Bergen] = GRAPH(TIME)
(2001, 0.0293), (2002, 0.017), (2003, 0.0141), (2004, 0.00675), (2005, 0.0143), (2006, 0.0349), (2007, 0.0308), (2008, 0.0256), (2009, 0.112), (2010, 0.0636), (2011, 0.0282)
frac_new_dwellings[West_Bergen] = GRAPH(TIME)
(2001, 0.191), (2002, 0.148), (2003, 0.11), (2004, 0.14), (2005, 0.131), (2006, 0.223), (2007, 0.186), (2008, 0.272), (2009, 0.222), (2010, 0.199), (2011, 0.114)
frac_new_dwelling_proj[Askoy] = GRAPH(TIME)
(2001, 0.162), (2001, 0.158), (2002, 0.153), (2002, 0.149), (2002, 0.144), (2003, 0.138), (2003, 0.133), (2004, 0.128), (2004, 0.124), (2004, 0.124), (2005, 0.128), (2005, 0.135), (2005, 0.143), (2006, 0.154), (2006, 0.17), (2006, 0.192), (2007, 0.22), (2007, 0.251), (2007, 0.284), (2008, 0.315), (2008, 0.343), (2009, 0.368), (2009, 0.389), (2009, 0.405), (2010, 0.416), (2010, 0.422), (2010, 0.423), (2011, 0.42), (2011, 0.415)
frac_new_dwelling_proj[Center_Bergen] = GRAPH(TIME)
(2001, 0.251), (2001, 0.246), (2002, 0.242), (2002, 0.239), (2002, 0.234), (2003, 0.229), (2003, 0.224), (2004, 0.219), (2004, 0.216), (2004, 0.213), (2005, 0.211), (2005, 0.207), (2005, 0.203), (2006, 0.196), (2006, 0.188), (2006, 0.177), (2007, 0.164), (2007, 0.149), (2007, 0.135), (2008, 0.121), (2008, 0.109), (2009, 0.0971), (2009, 0.0863), (2009, 0.0761), (2010, 0.0665), (2010, 0.0573), (2010, 0.0488), (2011, 0.0411), (2011, 0.0343)
frac_new_dwelling_proj[South_Bergen] = GRAPH(TIME)
(2001, 0.26), (2001, 0.269), (2002, 0.278), (2002, 0.285), (2002, 0.29), (2003, 0.294), (2003, 0.294), (2004, 0.289), (2004, 0.278), (2004, 0.264), (2005, 0.254), (2005, 0.252), (2005, 0.263), (2006, 0.284), (2006, 0.313), (2006, 0.343), (2007, 0.369), (2007, 0.388), (2007, 0.399), (2008, 0.403), (2008, 0.4), (2009, 0.393), (2009, 0.383), (2009, 0.371), (2010, 0.36), (2010, 0.351), (2010, 0.343), (2011, 0.336), (2011, 0.33)

frac_new_dwelling_proj[North_Bergen] = GRAPH(TIME)
(2001, 0.12), (2001, 0.121), (2002, 0.122), (2002, 0.122), (2002, 0.123), (2003, 0.125), (2003, 0.128), (2004, 0.132), (2004, 0.137), (2004, 0.141), (2005, 0.142), (2005, 0.139), (2005, 0.132), (2006, 0.121), (2006, 0.108), (2006, 0.0936), (2007, 0.0795), (2007, 0.067), (2007, 0.057), (2008, 0.0497), (2008, 0.0453), (2009, 0.0435), (2009, 0.0441), (2009, 0.0466), (2010, 0.0508), (2010, 0.0563), (2010, 0.0628), (2011, 0.0701), (2011, 0.0782)
frac_new_dwelling_proj[East_Bergen] = GRAPH(TIME)
(2001, 0.0288), (2001, 0.0273), (2002, 0.0261), (2002, 0.0258), (2002, 0.0268), (2003, 0.0291), (2003, 0.0321), (2004, 0.0356), (2004, 0.0394), (2004, 0.0428), (2005, 0.045), (2005, 0.0451), (2005, 0.0428), (2006, 0.0385), (2006, 0.0331), (2006, 0.0279), (2007, 0.0235), (2007, 0.0202), (2007, 0.0181), (2008, 0.0171), (2008, 0.0169), (2009, 0.0175), (2009, 0.0186), (2009, 0.0202), (2010, 0.022), (2010, 0.024), (2010, 0.026), (2011, 0.0279), (2011, 0.0298)
frac_new_dwelling_proj[West_Bergen] = GRAPH(TIME)
(2012, 0.178), (2013, 0.179), (2013, 0.179), (2014, 0.18), (2015, 0.181), (2015, 0.185), (2016, 0.19), (2016, 0.197), (2017, 0.206), (2018, 0.215), (2018, 0.221), (2019, 0.222), (2020, 0.217), (2020, 0.205), (2021, 0.187), (2022, 0.166), (2022, 0.145), (2023, 0.125), (2024, 0.108), (2024, 0.0945), (2025, 0.0856), (2025, 0.0806), (2026, 0.0791), (2027, 0.0805), (2027, 0.0842), (2028, 0.0896), (2029, 0.0964), (2029, 0.104), (2030, 0.113)
frac_new_job[Askoy_W] = GRAPH(TIME)
(2001, 0.0747), (2002, 0.0747), (2003, 0.0712), (2004, 0.0677), (2005, 0.0642), (2006, 0.0607), (2007, 0.0572), (2008, 0.0538), (2009, 0.0551), (2010, 0.0564), (2011, 0.0578)
frac_new_job[Center_Bergen_W] = GRAPH(TIME)
(2001, -0.00159), (2002, -0.00159), (2003, 0.0753), (2004, 0.152), (2005, 0.229), (2006, 0.306), (2007, 0.383), (2008, 0.46), (2009, 0.392), (2010, 0.324), (2011, 0.256)
frac_new_job[South_Bergen_W] = GRAPH(TIME)
(2001, 0.536), (2002, 0.536), (2003, 0.488), (2004, 0.44), (2005, 0.391), (2006, 0.343), (2007, 0.295), (2008, 0.247), (2009, 0.327), (2010, 0.407), (2011, 0.487)
frac_new_job[North_Bergen_W] = GRAPH(TIME)
(2001, 0.195), (2002, 0.195), (2003, 0.182), (2004, 0.17), (2005, 0.157), (2006, 0.144), (2007, 0.131), (2008, 0.118), (2009, 0.101), (2010, 0.0847), (2011, 0.0679)
frac_new_job[East_Bergen_W] = GRAPH(TIME)
(2001, 0.0348), (2002, 0.0348), (2003, 0.0342), (2004, 0.0335), (2005, 0.0329), (2006, 0.0323), (2007, 0.0316), (2008, 0.031), (2009, 0.0102), (2010, -0.0106), (2011, -0.0314)
frac_new_job[West_Bergen_W] = GRAPH(TIME)
(2001, 0.161), (2002, 0.161), (2003, 0.149), (2004, 0.137), (2005, 0.126), (2006, 0.114), (2007, 0.102), (2008, 0.0907), (2009, 0.115), (2010, 0.139), (2011, 0.163)
frac_new_job_proj[Askoy_W] = GRAPH(TIME)
(2012, 0.0374), (2013, 0.038), (2013, 0.0386), (2014, 0.0392), (2015, 0.0397), (2015, 0.0403), (2016, 0.041), (2016, 0.0418), (2017, 0.0423), (2018, 0.0431), (2018, 0.0438), (2019, 0.0446), (2020, 0.0456), (2020, 0.0468), (2021, 0.0482), (2022, 0.0493), (2022, 0.0504), (2023, 0.0515), (2024, 0.0526), (2024, 0.0536), (2025, 0.0546), (2025, 0.0556), (2026, 0.0566), (2027, 0.0587), (2027, 0.0599), (2028, 0.0611), (2029, 0.0623), (2029, 0.0634), (2030, 0.0645)
frac_new_job_proj[Center_Bergen_W] = GRAPH(TIME)

(2012, 0.541), (2013, 0.539), (2013, 0.536), (2014, 0.531), (2015, 0.525), (2015, 0.518), (2016, 0.511), (2016, 0.501), (2017, 0.492), (2018, 0.482), (2018, 0.473), (2019, 0.465), (2020, 0.46), (2020, 0.458), (2021, 0.453), (2022, 0.452), (2022, 0.453), (2023, 0.457), (2024, 0.461), (2024, 0.466), (2025, 0.471), (2025, 0.475), (2026, 0.479), (2027, 0.486), (2027, 0.489), (2028, 0.491), (2029, 0.491), (2029, 0.491), (2030, 0.491)
frac_new_job_proj[South_Bergen_W] = GRAPH(TIME)
(2012, 0.19), (2013, 0.193), (2013, 0.198), (2014, 0.202), (2015, 0.207), (2015, 0.212), (2016, 0.217), (2016, 0.221), (2017, 0.226), (2018, 0.229), (2018, 0.233), (2019, 0.235), (2020, 0.239), (2020, 0.242), (2021, 0.244), (2022, 0.244), (2022, 0.243), (2023, 0.241), (2024, 0.24), (2024, 0.239), (2025, 0.238), (2025, 0.239), (2026, 0.239), (2027, 0.238), (2027, 0.238), (2028, 0.238), (2029, 0.237), (2029, 0.236), (2030, 0.236)
frac_new_job_proj[North_Bergen_W] = GRAPH(TIME)
(2012, 0.0781), (2013, 0.08), (2013, 0.082), (2014, 0.0838), (2015, 0.0855), (2015, 0.0875), (2016, 0.0898), (2016, 0.0927), (2017, 0.0954), (2018, 0.098), (2018, 0.1), (2019, 0.102), (2020, 0.103), (2020, 0.104), (2021, 0.104), (2022, 0.105), (2022, 0.105), (2023, 0.105), (2024, 0.104), (2024, 0.103), (2025, 0.101), (2025, 0.1), (2026, 0.0991), (2027, 0.0981), (2027, 0.0973), (2028, 0.0966), (2029, 0.0961), (2029, 0.0957), (2030, 0.0953)
frac_new_job_proj[East_Bergen_W] = GRAPH(TIME)
(2012, 0.0394), (2013, 0.0358), (2013, 0.0324), (2014, 0.0302), (2015, 0.0296), (2015, 0.0281), (2016, 0.0267), (2016, 0.0255), (2017, 0.0251), (2018, 0.0247), (2018, 0.0237), (2019, 0.0234), (2020, 0.0235), (2020, 0.0238), (2021, 0.0243), (2022, 0.025), (2022, 0.0258), (2023, 0.0234), (2024, 0.022), (2024, 0.0219), (2025, 0.0222), (2025, 0.0225), (2026, 0.0226), (2027, 0.0227), (2027, 0.0227), (2028, 0.0227), (2029, 0.0227), (2029, 0.0227), (2030, 0.0227)
frac_new_job_proj[West_Bergen_W] = GRAPH(TIME)
(2012, 0.114), (2013, 0.114), (2013, 0.113), (2014, 0.113), (2015, 0.113), (2015, 0.114), (2016, 0.115), (2016, 0.117), (2017, 0.12), (2018, 0.123), (2018, 0.126), (2019, 0.13), (2020, 0.129), (2020, 0.127), (2021, 0.126), (2022, 0.125), (2022, 0.124), (2023, 0.122), (2024, 0.12), (2024, 0.117), (2025, 0.113), (2025, 0.108), (2026, 0.103), (2027, 0.0965), (2027, 0.0926), (2028, 0.0912), (2029, 0.0908), (2029, 0.0908), (2030, 0.0909)
frac_owing_car = GRAPH(TIME)
(2001, 0.385), (2002, 0.389), (2003, 0.393), (2004, 0.403), (2005, 0.415), (2006, 0.428), (2007, 0.441), (2008, 0.441), (2009, 0.442), (2010, 0.442), (2011, 0.442)
frac_s[Resident,Working] =
[0.80519040406,0.0376615052593,0.0281780251819,0.0140168380235,0.00966663457319,0.0731045032275,
0.0500746129884,0.408401170019,0.294071375794,0.237711618138,0.152049432384,0.23825280812,
0.0224286504221,0.163735302427,0.403409034962,0.049543784641,0.0876268127383,0.106714533104,
0.0141836771198,0.106454685855,0.0566129335592,0.553829728482,0.131388828814,0.0674853124486,
0.00269840193162,0.0253342491099,0.0243861065652,0.0487219919906,0.552941019998,0.0128388778121,
0.105424253479,0.25841308733,0.193342523938,0.0961760387251,0.0663272714926,0.501603965288]

Frac_seek_get_vacancy[Resident] =
Vacancy_Taken[Resident]/TOTAL_seeker[Resident]
Frac_single_adult[Resident] =
ARRAYSUM(Adult_single[Resident,*])/(ARRAYSUM(Adult_single[Resident,*])+ARRAYSUM(Adult__married_or_cohibited[Resident,*]))
frac_single_adult_real[Askoy] = GRAPH(TIME)
(2001, 0.245), (2011, 0.253)
frac_single_adult_real[Center_Bergen] = GRAPH(TIME)
(2001, 0.486), (2011, 0.52)
frac_single_adult_real[South_Bergen] = GRAPH(TIME)
(2001, 0.243), (2011, 0.279)
frac_single_adult_real[North_Bergen] = GRAPH(TIME)
(2001, 0.3), (2011, 0.319)
frac_single_adult_real[East_Bergen] = GRAPH(TIME)
(2001, 0.289), (2011, 0.293)
frac_single_adult_real[West_Bergen] = GRAPH(TIME)
(2001, 0.338), (2011, 0.36)
frac_single_chooing_Low_density[Resident,Working] =
MAX_frac_single_chooing_Low_density/(1+EXP(-relative_single_commuting_time[Resident,Working]*b_for_single_choosing_Low_density))
frac_single_in_migration_adult = 0.45
frac_single_in_migration_pensioner = 0.69
FTJS_get_job[Working] = Frac_get_job[Working]*search_effect_FTJS[Working]
FTJS_get_vacancy = ARRAYSUM(FTJS_get_job[*])
Gap_between_jobs_and_workers =
Total_job_opportunity-Total_working_age_population
household_diminish[Resident] =
ARRAYSUM(Death_of_AS[Resident,*])+ARRAYSUM(Death_of_PS[Resident,*])
Household_leaving_dwellings[Resident] =
ARRAYSUM(AFR__leaving_old_dwelling[Resident,*])+ARRAYSUM(SAR__leaving_old_dwelling[Resident,*])+ARRAYSUM(SPR_moving_out[Resident,*])
household_wiith_child_leaving[Resident,Working] =
ARRAYSUM(Youth_starting__living_alone[*,*])/Average_number_of_child/total_family_household*family_household[Resident,Working]
household_with_new_adult[Resident,Working] =
Adult_moving_together[Resident,Working]/2
household_with_new_babies[Resident,Working] =
Birth/Average_number_of_child/total_family_household*family_household[Resident,Working]
Household_with_widow[Resident,Working] = Death_of_AMC[Resident,Working]
houshold_with_adult_leaving[Resident,Working] =
(Adult_becoming_single[Resident,Working]-Household_with_widow[Resident,Working])/2
IJS_get_job[Working] = Frac_get_job[Working]*Search_effect_IJS[Working]
imigration_AMC[Resident,Working] =
AFR_get_vacancy[Resident,Working]*2+FFTB_get_vacancy[Resident,Working]*2
imigration_AS[Resident,Working] =
SAR_get_vacancy[Resident,Working]+SFTB_get_vacancy[Resident,Working]+DAR_get_vacancy[Resident,Working]
imigration_PS[Resident,Working] = SPR_get_vacancy[Resident,Working]

ini_pensioner_MC[Resident,Working] =
[463.736263736264,185.494505494505,46.3736263736264,46.3736263736264,9.27472527472528,92.7472527472527,
33.510424213153,2446.0934154391,285.341262174998,854.515817435402,52.6113660146502,490.927714722692,
9.0168504522364,1120.4338371949,1168.22314459175,289.080225498699,79.8892950068145,244.356647255607,
2.97399803070639,648.331570693994,39.3162539659385,774.964406841472,20.0447467269611,145.369023740928,
1.42722117202268,229.211720226843,88.3449905482042,107.041587901701,289.012287334594,39.9621928166352,
9.8095807961705,1309.87332371265,97.7034247298582,475.470381190384,19.4229699764176,874.720319594523]
ini_pensioner_single[Resident,Working] =
[657.142857142857,262.857142857143,65.7142857142857,65.7142857142857,13.1428571428571,131.428571428571,
65.2660387990018,4764.09450213314,555.740320373501,1664.28398937455,102.467680914433,956.147468405377,
10.2651468219551,1275.54714409615,1329.95242225251,329.100607111882,90.9492008425226,278.185478874985,
3.77447941358813,822.836512162211,49.898617847635,983.553845592794,25.439991247584,184.496553736188,
1.5992438563327,256.838563327032,98.9931947069943,119.943289224953,323.846880907372,44.7788279773157,
15.0927457674844,2015.3343423322,150.323747844145,731.54538734997,29.8836366196192,1345.82014008659]
inter_urabn_emigraion_PS[Resident,Working] =
Emigration_pensioner[Resident,Working]*frac_single_in_migration_pensioner
inter_urban_emigraion_AS[Resident,Working] =
Emigration_adult[Resident,Working]*frac_single_in_migration_adult
intra_urban_emigration[Resident,Working] =
DAR__leaving_old_dwelling[Resident,Working]+SAR__leaving_old_dwelling[Resident,Working]
in_saane =
ARRAYSUM(commuter[*,North_Bergen_W])-commuter[North_Bergen,North_Bergen_W]
job_quitting[Working] =
ARRAYSUM(NOJS_Job_quiting[*,Working])+ARRAYSUM(SOJSJob_quiting[*,Working])
live_birth_Bergen_Askoy = GRAPH(TIME)
(2001, 3420), (2002, 3497), (2003, 3487), (2004, 3623), (2005, 3715), (2006, 3692), (2007, 3745), (2008, 3844), (2009, 3935), (2010, 3936), (2011, 3898)
logit_for_askoy = 20
Logit_for_time_to_center_f = -0.4
Logit_for_time_to_center_s = -0.7
Logit_for_time_to_work_f = -0.5
Logit_for_time_to_work_s = -0.55
long_commuter = employment_rate*total_commuter
market_demand[Resident] =
TOTAL_seeker[Resident]/ARRAYSUM(TOTAL_seeker[*])

marriage_rate = GRAPH(time)
(2001, 0.0225), (2002, 0.0216), (2003, 0.0207), (2004, 0.0219), (2005, 0.021), (2006, 0.0211), (2007, 0.0212), (2008, 0.0214), (2009, 0.0213), (2010, 0.0193), (2011, 0.0189)
MAX_frac_family_chooing_High_density = 0.8
MAX_frac_single_chooing_Low_density = 0.7
MC_commuter_real[Resident,Working] = (real_commuter[Resident,Working]-singel_commuter_real[Resident,Working]*employment_rate)/employment_rate
moving_together[Resident] = ARRAYSUM(Adult_moving_together[Resident,*])/2
nature_labor_force_change = ARRAYSUM(Retirement[*])-Stating_job_seeking+ARRAYSUM(death_leaving_job[*])
needed_immigration = Gap_between_jobs_and_workers+DELAY(EXO_emigration_adult,1/12)+DELAY(nature_labor_force_change,1/12)
net_job_change_projection = GRAPH(TIME)
(2012, 2172), (2013, 1863), (2013, 2278), (2014, 2494), (2015, 2689), (2015, 2343), (2016, 2305), (2017, 2036), (2017, 1974), (2018, 1587), (2019, 1452), (2019, 1310), (2020, 1243), (2021, 1112), (2021, 883), (2022, 898), (2023, 801), (2023, 708), (2024, 708), (2025, 602), (2025, 750), (2026, 665), (2027, 657), (2027, 644), (2028, 672), (2029, 806), (2029, 812), (2030, 854)
net_job_creation = GRAPH(TIME)
(2001, 2158), (2002, 1764), (2003, 1517), (2004, 2510), (2005, 2496), (2006, 3076), (2007, 4201), (2008, 3717), (2009, 3189), (2010, 3082), (2011, 3083)
net_job_creation_sorted_by_working_place[Working] = new_job_array[Working]*(1-STEP(1,2012))+new_job_array_proj[Working]*(0+STEP(1,2012))
newly_matured_dwelling_seeker[Working] = FTJS_get_job[Working]
new_dewlling_proj = GRAPH(TIME)
(2012, 2671), (2013, 2786), (2014, 2439), (2015, 2332), (2016, 2289), (2017, 2270), (2018, 2262), (2019, 2263), (2020, 2260), (2021, 2268), (2022, 2262), (2023, 2268), (2024, 2139), (2025, 1914), (2026, 1830), (2027, 1829), (2028, 1847), (2029, 1864), (2030, 0.00)
New_divorced_dwelling_seeker[Resident,Working] = houshold_with_adult_leaving[Resident,Working]
new_dwellings[Resident] = frac_new_dwellings[Resident]*total_new_dwellings
new_dwelling_array[Resident] = new_dwellings[Resident]*(1-STEP(1,2012))+new_dwelling_arrray_proj[Resident]*(0+STEP(1,2012))
new_dwelling_arrray_proj[Resident] = IF ARRAYIDX()=2
THEN new_dewlling_proj*frac_new_dwelling_proj[Resident]
ELSE IF policy_switch=0
    THEN new_dewlling_proj*frac_new_dwelling_proj[Resident]
    ELSE
new_dewlling_proj*(1-frac_new_dwelling_proj[Center_Bergen])*(policy_frac[Resident]/(1-policy_frac[Center_Bergen]))
new_job_array[Working] = frac_new_job[Working]*net_job_creation

new_job_array_proj[Working] =
frac_new_job_proj[Working]*net_job_change_projection+Pulse_job_for_Policy_test[Working]
NOJS_car[Resident] = frac_owing_car*NOJS_sorted_by_resident_place[Resident]
NOJS_get_job[Resident,Working] =
Frac_get_job[Working]*Search_effect_NOJS[Resident,Working]
NOJS_get_vacancy_sort_by_Resident[Resident] =
ARRAYSUM(NOJS_get_job[Resident,*])
NOJS_PT[Resident] =
NOJS_sorted_by_resident_place[Resident]*(1-frac_owing_car)
NOJS_sorted_by_resident_place[Resident] =
ARRAYSUM(NONsingle_on_job_seeker[Resident,*])
N_car_to_A = GRAPH(TIME)
(2001, 0.36), (2002, 0.518), (2002, 0.582), (2003, 0.613), (2003, 0.631), (2004, 0.642), (2004, 0.65), (2005, 0.661), (2005, 0.677), (2006, 0.694), (2006, 0.709), (2007, 0.719), (2007, 0.727), (2008, 0.733), (2008, 0.735), (2009, 0.737), (2009, 0.738), (2010, 0.738), (2010, 0.739), (2011, 0.741), (2011, 0.742)
N_car_to_A_2 = GRAPH(TIME)
(2001, 0.00), (2002, 0.338), (2002, 0.509), (2003, 0.601), (2003, 0.653), (2004, 0.683), (2004, 0.7), (2005, 0.714), (2005, 0.73), (2006, 0.743), (2006, 0.753), (2007, 0.759), (2007, 0.763), (2008, 0.765), (2008, 0.766), (2009, 0.767), (2009, 0.768), (2010, 0.77), (2010, 0.773), (2011, 0.775), (2011, 0.778)
N_car_to_C = GRAPH(TIME)
(2001, 0.173), (2002, 0.331), (2002, 0.394), (2003, 0.425), (2003, 0.443), (2004, 0.455), (2004, 0.463), (2005, 0.473), (2005, 0.49), (2006, 0.506), (2006, 0.521), (2007, 0.531), (2007, 0.539), (2008, 0.545), (2008, 0.547), (2009, 0.549), (2009, 0.55), (2010, 0.551), (2010, 0.552), (2011, 0.553), (2011, 0.554)
N_car_to_C_2 = GRAPH(TIME)
(2001, 0.00), (2002, 0.242), (2002, 0.356), (2003, 0.415), (2003, 0.447), (2004, 0.465), (2004, 0.475), (2005, 0.485), (2005, 0.498), (2006, 0.51), (2006, 0.519), (2007, 0.523), (2007, 0.527), (2008, 0.528), (2008, 0.529), (2009, 0.531), (2009, 0.532), (2010, 0.534), (2010, 0.536), (2011, 0.538), (2011, 0.541)
N_car_to_E = GRAPH(TIME)
(2001, 0.187), (2002, 0.332), (2002, 0.386), (2003, 0.409), (2003, 0.421), (2004, 0.426), (2004, 0.429), (2005, 0.431), (2005, 0.434), (2006, 0.435), (2006, 0.435), (2007, 0.432), (2007, 0.431), (2008, 0.43), (2008, 0.429), (2009, 0.429), (2009, 0.429), (2010, 0.429), (2010, 0.429), (2011, 0.429), (2011, 0.429)
N_car_to_E_2 = GRAPH(TIME)
(2001, 0.00), (2002, 0.235), (2002, 0.346), (2003, 0.403), (2003, 0.434), (2004, 0.452), (2004, 0.462), (2005, 0.469), (2005, 0.475), (2006, 0.479), (2006, 0.48), (2007, 0.478), (2007, 0.477), (2008, 0.476), (2008, 0.475), (2009, 0.475), (2009, 0.475), (2010, 0.476), (2010, 0.476), (2011, 0.476), (2011, 0.476)
N_car_to_N = GRAPH(TIME)
(2001, -0.0262), (2002, 0.131), (2002, 0.191), (2003, 0.218), (2003, 0.231), (2004, 0.237), (2004, 0.24), (2005, 0.243), (2005, 0.246), (2006, 0.247), (2006, 0.247), (2007, 0.245), (2007, 0.244), (2008, 0.243), (2008, 0.243), (2009, 0.242), (2009, 0.242), (2010, 0.242), (2010, 0.242), (2011, 0.243), (2011, 0.243)
N_car_to_N_2 = GRAPH(TIME)
(2001, -0.347), (2002, -0.0651), (2002, 0.0729), (2003, 0.146), (2003, 0.187), (2004, 0.21), (2004, 0.223), (2005, 0.233), (2005, 0.24), (2006, 0.245), (2006, 0.246), (2007,

0.244), (2007, 0.243), (2008, 0.242), (2008, 0.242), (2009, 0.242), (2009, 0.242), (2010, 0.242), (2010, 0.242), (2011, 0.243), (2011, 0.243)
N_car_to_S = GRAPH(TIME)
(2001, 0.293), (2002, 0.452), (2002, 0.516), (2003, 0.548), (2003, 0.567), (2004, 0.579), (2004, 0.587), (2005, 0.598), (2005, 0.617), (2006, 0.634), (2006, 0.65), (2007, 0.659), (2007, 0.668), (2008, 0.672), (2008, 0.675), (2009, 0.676), (2009, 0.677), (2010, 0.678), (2010, 0.678), (2011, 0.68), (2011, 0.681)
N_car_to_S_2 = GRAPH(TIME)
(2001, 0.00), (2002, 0.304), (2002, 0.456), (2003, 0.536), (2003, 0.581), (2004, 0.607), (2004, 0.621), (2005, 0.634), (2005, 0.65), (2006, 0.663), (2006, 0.673), (2007, 0.677), (2007, 0.681), (2008, 0.682), (2008, 0.683), (2009, 0.684), (2009, 0.686), (2010, 0.687), (2010, 0.69), (2011, 0.692), (2011, 0.695)
N_car_to_W = GRAPH(TIME)
(2001, 0.293), (2002, 0.452), (2002, 0.515), (2003, 0.546), (2003, 0.564), (2004, 0.575), (2004, 0.583), (2005, 0.594), (2005, 0.611), (2006, 0.627), (2006, 0.642), (2007, 0.652), (2007, 0.661), (2008, 0.666), (2008, 0.669), (2009, 0.67), (2009, 0.671), (2010, 0.672), (2010, 0.673), (2011, 0.674), (2011, 0.675)
N_car_to_W_2 = GRAPH(TIME)
(2001, 0.00), (2002, 0.304), (2002, 0.455), (2003, 0.535), (2003, 0.58), (2004, 0.606), (2004, 0.62), (2005, 0.633), (2005, 0.648), (2006, 0.661), (2006, 0.67), (2007, 0.676), (2007, 0.68), (2008, 0.682), (2008, 0.683), (2009, 0.684), (2009, 0.685), (2010, 0.687), (2010, 0.689), (2011, 0.692), (2011, 0.694)
N_PT_to_A = GRAPH(TIME)
(2001, 0.92), (2002, 0.872), (2002, 0.846), (2003, 0.832), (2003, 0.826), (2004, 0.824), (2004, 0.824), (2005, 0.827), (2005, 0.835), (2006, 0.844), (2006, 0.852), (2007, 0.86), (2007, 0.867), (2008, 0.87), (2008, 0.873), (2009, 0.874), (2009, 0.875), (2010, 0.876), (2010, 0.877), (2011, 0.878), (2011, 0.88)
N_PT_to_A_2 = GRAPH(TIME)
(2001, 1.03), (2002, 1.25), (2002, 1.25), (2003, 1.25), (2003, 1.25), (2004, 1.25), (2004, 1.25), (2005, 1.25), (2005, 1.26), (2006, 1.27), (2006, 1.28), (2007, 1.28), (2007, 1.29), (2008, 1.29), (2008, 1.29), (2009, 1.29), (2009, 1.30), (2010, 1.30), (2010, 1.30), (2011, 1.30), (2011, 1.31)
N_PT_to_C = GRAPH(TIME)
(2001, 0.69), (2002, 0.665), (2002, 0.653), (2003, 0.647), (2003, 0.646), (2004, 0.646), (2004, 0.648), (2005, 0.652), (2005, 0.66), (2006, 0.669), (2006, 0.678), (2007, 0.686), (2007, 0.693), (2008, 0.696), (2008, 0.699), (2009, 0.7), (2009, 0.701), (2010, 0.702), (2010, 0.703), (2011, 0.705), (2011, 0.706)
N_PT_to_C_2 = GRAPH(TIME)
(2001, 0.853), (2002, 0.901), (2002, 0.904), (2003, 0.906), (2003, 0.907), (2004, 0.908), (2004, 0.909), (2005, 0.912), (2005, 0.922), (2006, 0.932), (2006, 0.942), (2007, 0.951), (2007, 0.957), (2008, 0.961), (2008, 0.963), (2009, 0.965), (2009, 0.968), (2010, 0.97), (2010, 0.973), (2011, 0.976), (2011, 0.979)
N_PT_to_E = GRAPH(TIME)
(2001, 0.68), (2002, 0.668), (2002, 0.66), (2003, 0.656), (2003, 0.654), (2004, 0.652), (2004, 0.651), (2005, 0.651), (2005, 0.65), (2006, 0.65), (2006, 0.65), (2007, 0.65), (2007, 0.65), (2008, 0.65), (2008, 0.65), (2009, 0.65), (2009, 0.65), (2010, 0.65), (2010, 0.65), (2011, 0.65), (2011, 0.65)
N_PT_to_E_2 = GRAPH(TIME)
(2001, 0.852), (2002, 0.884), (2002, 0.879), (2003, 0.876), (2003, 0.875), (2004, 0.874), (2004, 0.873), (2005, 0.873), (2005, 0.873), (2006, 0.874), (2006, 0.874),

(2007, 0.874), (2007, 0.875), (2008, 0.875), (2008, 0.875), (2009, 0.875), (2009, 0.876), (2010, 0.876), (2010, 0.876), (2011, 0.876), (2011, 0.877)
N_PT_to_N = GRAPH(TIME)
(2001, 0.532), (2002, 0.498), (2002, 0.478), (2003, 0.466), (2003, 0.46), (2004, 0.456), (2004, 0.453), (2005, 0.452), (2005, 0.451), (2006, 0.451), (2006, 0.45), (2007, 0.45), (2007, 0.45), (2008, 0.45), (2008, 0.45), (2009, 0.45), (2009, 0.45), (2010, 0.45), (2010, 0.45), (2011, 0.45), (2011, 0.45)
N_PT_to_N_2 = GRAPH(TIME)
(2001, -0.347), (2002, -0.0651), (2002, 0.0729), (2003, 0.146), (2003, 0.187), (2004, 0.21), (2004, 0.223), (2005, 0.233), (2005, 0.24), (2006, 0.245), (2006, 0.246), (2007, 0.244), (2007, 0.243), (2008, 0.242), (2008, 0.242), (2009, 0.242), (2009, 0.242), (2010, 0.242), (2010, 0.242), (2011, 0.243), (2011, 0.243)
N_PT_to_S = GRAPH(TIME)
(2001, 0.84), (2002, 0.8), (2002, 0.779), (2003, 0.768), (2003, 0.764), (2004, 0.763), (2004, 0.763), (2005, 0.767), (2005, 0.775), (2006, 0.784), (2006, 0.792), (2007, 0.8), (2007, 0.806), (2008, 0.81), (2008, 0.812), (2009, 0.814), (2009, 0.815), (2010, 0.816), (2010, 0.817), (2011, 0.818), (2011, 0.82)
N_PT_to_S_2 = GRAPH(TIME)
(2001, 0.973), (2002, 1.13), (2002, 1.13), (2003, 1.13), (2003, 1.13), (2004, 1.13), (2004, 1.13), (2005, 1.13), (2005, 1.14), (2006, 1.15), (2006, 1.16), (2007, 1.16), (2007, 1.17), (2008, 1.17), (2008, 1.17), (2009, 1.17), (2009, 1.18), (2010, 1.18), (2010, 1.18), (2011, 1.18), (2011, 1.18)
N_PT_to_W = GRAPH(TIME)
(2001, 0.84), (2002, 0.798), (2002, 0.775), (2003, 0.763), (2003, 0.758), (2004, 0.756), (2004, 0.756), (2005, 0.76), (2005, 0.768), (2006, 0.777), (2006, 0.786), (2007, 0.794), (2007, 0.8), (2008, 0.804), (2008, 0.806), (2009, 0.807), (2009, 0.808), (2010, 0.809), (2010, 0.81), (2011, 0.812), (2011, 0.813)
N_PT_to_W_2 = GRAPH(TIME)
(2001, 0.973), (2002, 1.13), (2002, 1.13), (2003, 1.13), (2003, 1.12), (2004, 1.12), (2004, 1.12), (2005, 1.13), (2005, 1.13), (2006, 1.14), (2006, 1.15), (2007, 1.16), (2007, 1.17), (2008, 1.17), (2008, 1.17), (2009, 1.17), (2009, 1.17), (2010, 1.18), (2010, 1.18), (2011, 1.18), (2011, 1.18)
out_asane = ARRAYSUM(commuter[North_Bergen,*])-commuter[North_Bergen,North_Bergen_W]
pensioner_household_with_widow[Resident,Working] = Becoming_widow[Resident,Working]
pensioner_with_work_and_resident[Resident,Working] = Pensioner__married_or_cohibited[Resident,Working]+Pensioner__single[Resident,Working]
policy_frac[Resident] = IF policy_switch=1
THEN market_demand[Resident]
ELSE IF policy_switch=2
    THEN ARRAYVALUE(frac_new_job[*],ARRAYIDX())
    ELSE IF policy_switch=3
        THEN ARRAYVALUE(Frac_Job_vacancy[*],ARRAYIDX())
        ELSE 1
policy_switch = 0
PT_travel_time_congestion[Askoy,Askoy_W] = A_PT_to_A
PT_travel_time_congestion[Askoy,Center_Bergen_W] = A_PT_to_C

PT_travel_time_congestion[Askoy,South_Bergen_W] = A_PT_to_S
PT_travel_time_congestion[Askoy,North_Bergen_W] = A_PT_to_N
PT_travel_time_congestion[Askoy,East_Bergen_W] = A_PT_to_E
PT_travel_time_congestion[Askoy,West_Bergen_W] = A_PT_to_W
PT_travel_time_congestion[Center_Bergen,Askoy_W] = C_PT_to_A
PT_travel_time_congestion[Center_Bergen,Center_Bergen_W] = C_PT_to_C
PT_travel_time_congestion[Center_Bergen,South_Bergen_W] = C_PT_to_S
PT_travel_time_congestion[Center_Bergen,North_Bergen_W] = C_PT_to_N
PT_travel_time_congestion[Center_Bergen,East_Bergen_W] = C_PT_to_E
PT_travel_time_congestion[Center_Bergen,West_Bergen_W] = C_PT_to_W
PT_travel_time_congestion[South_Bergen,Askoy_W] = S_PT_to_A
PT_travel_time_congestion[South_Bergen,Center_Bergen_W] = S_PT_to_C
PT_travel_time_congestion[South_Bergen,South_Bergen_W] = S_PT_to_S
PT_travel_time_congestion[South_Bergen,North_Bergen_W] = S_PT_to_N
PT_travel_time_congestion[South_Bergen,East_Bergen_W] = S_PT_to_E
PT_travel_time_congestion[South_Bergen,West_Bergen_W] = S_PT_to_W
PT_travel_time_congestion[North_Bergen,Askoy_W] = N_PT_to_A
PT_travel_time_congestion[North_Bergen,Center_Bergen_W] = N_PT_to_C
PT_travel_time_congestion[North_Bergen,South_Bergen_W] = N_PT_to_S
PT_travel_time_congestion[North_Bergen,North_Bergen_W] = N_PT_to_N
PT_travel_time_congestion[North_Bergen,East_Bergen_W] = N_PT_to_E
PT_travel_time_congestion[North_Bergen,West_Bergen_W] = N_PT_to_W
PT_travel_time_congestion[East_Bergen,Askoy_W] = E_PT_to_A
PT_travel_time_congestion[East_Bergen,Center_Bergen_W] = E_PT_to_C
PT_travel_time_congestion[East_Bergen,South_Bergen_W] = E_PT_to_S
PT_travel_time_congestion[East_Bergen,North_Bergen_W] = E_PT_to_N
PT_travel_time_congestion[East_Bergen,East_Bergen_W] = E_PT_to_E
PT_travel_time_congestion[East_Bergen,West_Bergen_W] = E_PT_to_W
PT_travel_time_congestion[West_Bergen,Askoy_W] = W_PT_to_A
PT_travel_time_congestion[West_Bergen,Center_Bergen_W] = W_PT_to_C
PT_travel_time_congestion[West_Bergen,South_Bergen_W] = W_PT_to_S
PT_travel_time_congestion[West_Bergen,North_Bergen_W] = W_PT_to_N
PT_travel_time_congestion[West_Bergen,East_Bergen_W] = W_PT_to_E
PT_travel_time_congestion[West_Bergen,West_Bergen_W] = W_PT_to_W
PT_travel_time_free[Askoy,Askoy_W] = A_PT_to_A_2
PT_travel_time_free[Askoy,Center_Bergen_W] = A_PT_to_C_2
PT_travel_time_free[Askoy,South_Bergen_W] = A_PT_to_S_2
PT_travel_time_free[Askoy,North_Bergen_W] = A_PT_to_N_2
PT_travel_time_free[Askoy,East_Bergen_W] = A_PT_to_E_2
PT_travel_time_free[Askoy,West_Bergen_W] = A_PT_to_W_2
PT_travel_time_free[Center_Bergen,Askoy_W] = C_PT_to_A_2
PT_travel_time_free[Center_Bergen,Center_Bergen_W] = C_PT_to_C_2
PT_travel_time_free[Center_Bergen,South_Bergen_W] = C_PT_to_S_2
PT_travel_time_free[Center_Bergen,North_Bergen_W] = C_PT_to_N_2
PT_travel_time_free[Center_Bergen,East_Bergen_W] = C_PT_to_E_2
PT_travel_time_free[Center_Bergen,West_Bergen_W] = C_PT_to_W_2
PT_travel_time_free[South_Bergen,Askoy_W] = S_PT_to_A_2
PT_travel_time_free[South_Bergen,Center_Bergen_W] = S_PT_to_C_2
PT_travel_time_free[South_Bergen,South_Bergen_W] = S_PT_to_S_2
PT_travel_time_free[South_Bergen,North_Bergen_W] = S_PT_to_N_2

PT_travel_time_free[South_Bergen,East_Bergen_W] = S_PT_to_E_2
PT_travel_time_free[South_Bergen,West_Bergen_W] = S_PT_to_W_2
PT_travel_time_free[North_Bergen,Askoy_W] = N_PT_to_A_2
PT_travel_time_free[North_Bergen,Center_Bergen_W] = N_PT_to_C_2
PT_travel_time_free[North_Bergen,South_Bergen_W] = N_PT_to_S_2
PT_travel_time_free[North_Bergen,North_Bergen_W] = N_PT_to_N_2
PT_travel_time_free[North_Bergen,East_Bergen_W] = N_PT_to_E_2
PT_travel_time_free[North_Bergen,West_Bergen_W] = N_PT_to_W_2
PT_travel_time_free[East_Bergen,Askoy_W] = E_PT_to_A_2
PT_travel_time_free[East_Bergen,Center_Bergen_W] = E_PT_to_C_2
PT_travel_time_free[East_Bergen,South_Bergen_W] = E_PT_to_S_2
PT_travel_time_free[East_Bergen,North_Bergen_W] = E_PT_to_N_2
PT_travel_time_free[East_Bergen,East_Bergen_W] = E_PT_to_E_2
PT_travel_time_free[East_Bergen,West_Bergen_W] = E_PT_to_W_2
PT_travel_time_free[West_Bergen,Askoy_W] = W_PT_to_A_2
PT_travel_time_free[West_Bergen,Center_Bergen_W] = W_PT_to_C_2
PT_travel_time_free[West_Bergen,South_Bergen_W] = W_PT_to_S_2
PT_travel_time_free[West_Bergen,North_Bergen_W] = W_PT_to_N_2
PT_travel_time_free[West_Bergen,East_Bergen_W] = W_PT_to_E_2
PT_travel_time_free[West_Bergen,West_Bergen_W] = W_PT_to_W_2
Pulse_job_for_Policy_test[Askoy_W] = 0
Pulse_job_for_Policy_test[Center_Bergen_W] = 0
Pulse_job_for_Policy_test[South_Bergen_W] = 0
Pulse_job_for_Policy_test[North_Bergen_W] = PULSE(3000,2015,0)*0
Pulse_job_for_Policy_test[East_Bergen_W] = 0
Pulse_job_for_Policy_test[West_Bergen_W] = 0
real_commuter[Askoy,Askoy_W] = GRAPH(TIME)
(2000, 4606), (2011, 5758)
real_commuter[Askoy,Center_Bergen_W] = GRAPH(TIME)
(2000, 2297), (2011, 2847)
real_commuter[Askoy,South_Bergen_W] = GRAPH(TIME)
(2000, 768), (2011, 1063)
real_commuter[Askoy,North_Bergen_W] = GRAPH(TIME)
(2000, 115), (2011, 237)
real_commuter[Askoy,East_Bergen_W] = GRAPH(TIME)
(2000, 20.4), (2011, 48.3)
real_commuter[Askoy,West_Bergen_W] = GRAPH(TIME)
(2000, 1337), (2011, 1989)
real_commuter[Center_Bergen,Askoy_W] = GRAPH(TIME)
(2000, 72.0), (2011, 177)
real_commuter[Center_Bergen,Center_Bergen_W] = GRAPH(TIME)
(2000, 20125), (2011, 25386)
real_commuter[Center_Bergen,South_Bergen_W] = GRAPH(TIME)
(2000, 4435), (2011, 5349)
real_commuter[Center_Bergen,North_Bergen_W] = GRAPH(TIME)
(2000, 1424), (2011, 1808)
real_commuter[Center_Bergen,East_Bergen_W] = GRAPH(TIME)
(2000, 401), (2011, 281)
real_commuter[Center_Bergen,West_Bergen_W] = GRAPH(TIME)
(2000, 2985), (2011, 2909)

real_commuter[South_Bergen,Askoy_W] = GRAPH(TIME)
(2000, 51.5), (2011, 156)
real_commuter[South_Bergen,Center_Bergen_W] = GRAPH(TIME)
(2000, 9769), (2011, 11324)
real_commuter[South_Bergen,South_Bergen_W] = GRAPH(TIME)
(2000, 9354), (2011, 14407)
real_commuter[South_Bergen,North_Bergen_W] = GRAPH(TIME)
(2000, 440), (2011, 744)
real_commuter[South_Bergen,East_Bergen_W] = GRAPH(TIME)
(2000, 363), (2011, 327)
real_commuter[South_Bergen,West_Bergen_W] = GRAPH(TIME)
(2000, 2136), (2011, 2560)
real_commuter[North_Bergen,Askoy_W] = GRAPH(TIME)
(2000, 24.2), (2011, 90.4)
real_commuter[North_Bergen,Center_Bergen_W] = GRAPH(TIME)
(2000, 6714), (2011, 7963)
real_commuter[North_Bergen,South_Bergen_W] = GRAPH(TIME)
(2000, 1391), (2011, 1682)
real_commuter[North_Bergen,North_Bergen_W] = GRAPH(TIME)
(2000, 5099), (2011, 6995)
real_commuter[North_Bergen,East_Bergen_W] = GRAPH(TIME)
(2000, 357), (2011, 475)
real_commuter[North_Bergen,West_Bergen_W] = GRAPH(TIME)
(2000, 1002), (2011, 1487)
real_commuter[East_Bergen,Askoy_W] = GRAPH(TIME)
(2000, 3.56), (2011, 20.0)
real_commuter[East_Bergen,Center_Bergen_W] = GRAPH(TIME)
(2000, 1939), (2011, 1481)
real_commuter[East_Bergen,South_Bergen_W] = GRAPH(TIME)
(2000, 676), (2011, 778)
real_commuter[East_Bergen,North_Bergen_W] = GRAPH(TIME)
(2000, 470), (2011, 675)
real_commuter[East_Bergen,East_Bergen_W] = GRAPH(TIME)
(2000, 1556), (2011, 2201)
real_commuter[East_Bergen,West_Bergen_W] = GRAPH(TIME)
(2000, 148), (2011, 330)
real_commuter[West_Bergen,Askoy_W] = GRAPH(TIME)
(2000, 182), (2011, 573)
real_commuter[West_Bergen,Center_Bergen_W] = GRAPH(TIME)
(2000, 12949), (2011, 13504)
real_commuter[West_Bergen,South_Bergen_W] = GRAPH(TIME)
(2000, 4332), (2011, 5040)
real_commuter[West_Bergen,North_Bergen_W] = GRAPH(TIME)
(2000, 649), (2011, 1126)
real_commuter[West_Bergen,East_Bergen_W] = GRAPH(TIME)
(2000, 115), (2011, 229)
real_commuter[West_Bergen,West_Bergen_W] = GRAPH(TIME)
(2000, 7541), (2011, 9433)
real_long_distance_commuter =
ARRAYSUM(real_commuter[*,*])-real_commuter[Askoy,Askoy_W]-real_commuter

[Center_Bergen,Center_Bergen_W]-real_commuter[South_Bergen,South_Bergen_W]-real_commuter[North_Bergen,North_Bergen_W]-real_commuter[East_Bergen,East_Bergen_W]-real_commuter[West_Bergen,West_Bergen_W]
related_adult_a[Askoy] = GRAPH(TIME)
(2001, 9166), (2011, 11502)
related_adult_a[Center_Bergen] = GRAPH(TIME)
(2001, 21338), (2011, 25519)
related_adult_a[South_Bergen] = GRAPH(TIME)
(2001, 22423), (2011, 27174)
related_adult_a[North_Bergen] = GRAPH(TIME)
(2001, 16206), (2011, 16257)
related_adult_a[East_Bergen] = GRAPH(TIME)
(2001, 5069), (2011, 5256)
related_adult_a[West_Bergen] = GRAPH(TIME)
(2001, 26047), (2011, 26981)
relative_negative_family_commuting_time[Resident,Working] =
(average_family_commuting_time-Weighted_travel_time_effect[Resident,Working])/average_family_commuting_time
relative_single_commuting_time[Resident,Working] =
(Weighted_travel_time_effect[Resident,Working]-average_sigle_commuting_time)/average_sigle_commuting_time
Resident_adult_population[Resident] =
ARRAYSUM(Adult__married_or_cohibited[Resident,*])+ARRAYSUM(Adult_single[Resident,*])
resident_utility_car_f[Resident,Working] = IF ARRAYIDX(2)=1
THEN EXP(utility_car_to_work_f[Resident,Working]*logit_for_askoy)
ELSE
EXP(utility_car_to_center_f[Resident]+utility_car_to_work_f[Resident,Working])
resident_utility_car_s[Resident,Working] = IF ARRAYIDX(2)=1
THEN EXP(utility_car_to_work_s[Resident,Working]*logit_for_askoy)
ELSE
EXP(utility_car_to_center_s[Resident]+utility_car_to_work_s[Resident,Working])
resident_utility_PT_f[Resident,Working] = IF ARRAYIDX(2)=1
THEN EXP(utility_PT_to_work_f[Resident,Working]*logit_for_askoy)
ELSE
EXP(utility_PT_to_center_f[Resident]+utility_PT_to_work_f[Resident,Working])
resident_utility_PT_S[Resident,Working] = IF ARRAYIDX(2)=1
THEN EXP(utility_PT_to_work_s[Resident,Working]*logit_for_askoy)
ELSE
EXP(utility_PT_to_center_s[Resident]+utility_PT_to_work_s[Resident,Working])
Retirement[Working] =
ARRAYSUM(Retirement_of_AMC[*,Working])+ARRAYSUM(Retiremen_of_AS[*,Working])
SAR_get_vacancy[Resident,Working] =
Search_effect_SAR[Resident,Working]*Frac_seek_get_vacancy[Resident]
SAR_get_vacancy_sort_by_WP[Working] =
ARRAYSUM(SAR_get_vacancy[*,Working])
SAR_seeker_car_WP[Working] =
SAR_sorted_by_working_place[Working]*frac_owing_car

SAR_seeker_PT_WP[Working] =
SAR_sorted_by_working_place[Working]*(1-frac_owing_car)
SAR_sorted_by_working_place[Working] =
ARRAYSUM(single__adult_dwelling_seeker[*,Working])
Search_effect_AFR[Resident,Working] =
(Destinasion_choice_AFR_car[Resident,Working]+Destinasion_choice_AFR_PT[Resident,Working])/dt
search_effect_DAR[Resident,Working] =
(Destinasion_choice_DAR_car[Resident,Working]+Destinasion_choice_DAR_PT[Resident,Working])/dt
Search_effect_FFTB[Resident,Working] =
(Destinasion_choice_FFB_car[Resident,Working]+Destinasion_choice_FFTB_PT[Resident,Working])/dt
search_effect_FTJS[Working] =
First_time_job_seeker*Frac_Job_vacancy[Working]/dt
Search_effect_IJS[Working] = Frac_Job_vacancy[Working]*imigration_job_seeker/dt
Search_effect_NOJS[Resident,Working] =
(Destinaion_choice_NOJS_car[Resident,Working]+DEstination_choice_NOJS_PT[Resident,Working])/dt
Search_effect_SAR[Resident,Working] =
(Destinasion_choice_SAR_car[Resident,Working]+Destinasion_choice_SAR_PT[Resident,Working])/dt
Search_effect_SFTB[Resident,Working] =
(Destinasion_choice_SFTB_car[Resident,Working]+Destinasion_choice_SFTB_PT[Resident,Working])/dt
Search_effect_SOJS[Resident,Working] =
(Destinaion_choice_SOJS_car[Resident,Working]+DEstination_choice_SOJS_PT[Resident,Working])/dt
Search_effect_YS[Resident,Working] =
(Destinasion_choice_YS_car[Resident,Working]+Destinasion_choice_YS_PT[Resident,Working])/dt
SFTB_get_vacancy[Resident,Working] =
Search_effect_SFTB[Resident,Working]*Frac_seek_get_vacancy[Resident]
SFTB_get_vacancy_sort_by_WP[Working] =
ARRAYSUM(SFTB_get_vacancy[*,Working])
SFTB_seeker_car[Working] = Single_first_time_buyer[Working]*frac_owing_car
SFTB_seeker_PT[Working] = Single_first_time_buyer[Working]*(1-frac_owing_car)
singel_commuter_real[Resident,Working] =
real_commuter[Resident,Working]*frac_single_adult_real[Resident]/employment_rate
single_immigration[Working] =
IJS_get_job[Working]*frac_single_in_migration_adult
single_seeking_high_density[Resident,Working] =
(1-frac_single_chooing_Low_density[Resident,Working])*total_single_search_effect[Resident,Working]
Single_seeking_low_density[Resident,Working] =
frac_single_chooing_Low_density[Resident,Working]*total_single_search_effect[Resident,Working]
SOJS_car[Resident] = frac_owing_car*SOJS_sorted_by_resident_place[Resident]

SOJS_get_job[Resident,Working] =
Frac_get_job[Working]*Search_effect_SOJS[Resident,Working]
SOJS_get_vacancy_sort_by_Resident[Resident] =
ARRAYSUM(SOJS_get_job[Resident,*])
SOJS_PT[Resident] = SOJS_sorted_by_resident_place[Resident]*(1-frac_owing_car)
SOJS_sorted_by_resident_place[Resident] =
ARRAYSUM(single_on_job_seeker[Resident,*])
SPR_get_vacancy[Resident,Working] =
Destinasion_SPR[Resident,Working]*Frac_seek_get_vacancy[Resident]
switch_logit = 0
S_car_to_A = GRAPH(TIME)
(2001, 0.276), (2002, 0.421), (2002, 0.475), (2003, 0.499), (2003, 0.51), (2004, 0.516), (2004, 0.518), (2005, 0.521), (2005, 0.524), (2006, 0.525), (2006, 0.525), (2007, 0.522), (2007, 0.52), (2008, 0.519), (2008, 0.518), (2009, 0.518), (2009, 0.518), (2010, 0.518), (2010, 0.518), (2011, 0.518), (2011, 0.518)
S_car_to_A_2 = GRAPH(TIME)
(2001, 0.00), (2002, 0.28), (2002, 0.417), (2003, 0.49), (2003, 0.53), (2004, 0.553), (2004, 0.566), (2005, 0.575), (2005, 0.583), (2006, 0.587), (2006, 0.589), (2007, 0.587), (2007, 0.585), (2008, 0.584), (2008, 0.584), (2009, 0.584), (2009, 0.584), (2010, 0.584), (2010, 0.585), (2011, 0.585), (2011, 0.585)
S_car_to_C = GRAPH(TIME)
(2001, 0.12), (2002, 0.267), (2002, 0.322), (2003, 0.347), (2003, 0.359), (2004, 0.365), (2004, 0.368), (2005, 0.372), (2005, 0.377), (2006, 0.381), (2006, 0.384), (2007, 0.378), (2007, 0.374), (2008, 0.372), (2008, 0.37), (2009, 0.369), (2009, 0.369), (2010, 0.368), (2010, 0.368), (2011, 0.368), (2011, 0.369)
S_car_to_C_2 = GRAPH(TIME)
(2001, 0.00), (2002, 0.203), (2002, 0.295), (2003, 0.341), (2003, 0.365), (2004, 0.379), (2004, 0.386), (2005, 0.393), (2005, 0.399), (2006, 0.403), (2006, 0.406), (2007, 0.401), (2007, 0.399), (2008, 0.397), (2008, 0.395), (2009, 0.395), (2009, 0.395), (2010, 0.395), (2010, 0.395), (2011, 0.395), (2011, 0.395)
S_car_to_E = GRAPH(TIME)
(2001, 0.253), (2002, 0.398), (2002, 0.452), (2003, 0.476), (2003, 0.487), (2004, 0.493), (2004, 0.495), (2005, 0.498), (2005, 0.501), (2006, 0.502), (2006, 0.502), (2007, 0.499), (2007, 0.497), (2008, 0.496), (2008, 0.496), (2009, 0.496), (2009, 0.495), (2010, 0.496), (2010, 0.496), (2011, 0.496), (2011, 0.496)
S_car_to_E_2 = GRAPH(TIME)
(2001, 0.00), (2002, 0.269), (2002, 0.4), (2003, 0.469), (2003, 0.507), (2004, 0.529), (2004, 0.541), (2005, 0.55), (2005, 0.557), (2006, 0.561), (2006, 0.563), (2007, 0.561), (2007, 0.56), (2008, 0.559), (2008, 0.559), (2009, 0.558), (2009, 0.559), (2010, 0.559), (2010, 0.559), (2011, 0.559), (2011, 0.56)
S_car_to_N = GRAPH(TIME)
(2001, 0.293), (2002, 0.44), (2002, 0.495), (2003, 0.52), (2003, 0.532), (2004, 0.539), (2004, 0.542), (2005, 0.546), (2005, 0.551), (2006, 0.555), (2006, 0.558), (2007, 0.552), (2007, 0.548), (2008, 0.546), (2008, 0.544), (2009, 0.543), (2009, 0.543), (2010, 0.542), (2010, 0.542), (2011, 0.542), (2011, 0.543)
S_car_to_N_2 = GRAPH(TIME)
(2001, 0.00), (2002, 0.292), (2002, 0.436), (2003, 0.513), (2003, 0.556), (2004, 0.58), (2004, 0.594), (2005, 0.604), (2005, 0.613), (2006, 0.618), (2006, 0.622), (2007, 0.618), (2007, 0.615), (2008, 0.614), (2008, 0.612), (2009, 0.612), (2009, 0.612), (2010, 0.612), (2010, 0.612), (2011, 0.612), (2011, 0.612)

S_car_to_S = GRAPH(TIME)
(2001, 0.0242), (2002, 0.172), (2002, 0.225), (2003, 0.248), (2003, 0.259), (2004, 0.264), (2004, 0.266), (2005, 0.269), (2005, 0.271), (2006, 0.272), (2006, 0.272), (2007, 0.27), (2007, 0.268), (2008, 0.267), (2008, 0.267), (2009, 0.266), (2009, 0.266), (2010, 0.266), (2010, 0.266), (2011, 0.266), (2011, 0.266)
S_car_to_S_2 = GRAPH(TIME)
(2001, -0.298), (2002, -0.0365), (2002, 0.0898), (2003, 0.156), (2003, 0.193), (2004, 0.214), (2004, 0.226), (2005, 0.234), (2005, 0.241), (2006, 0.245), (2006, 0.246), (2007, 0.244), (2007, 0.243), (2008, 0.242), (2008, 0.242), (2009, 0.242), (2009, 0.242), (2010, 0.242), (2010, 0.242), (2011, 0.243), (2011, 0.243)
S_car_to_W = GRAPH(TIME)
(2001, 0.24), (2002, 0.387), (2002, 0.443), (2003, 0.467), (2003, 0.48), (2004, 0.486), (2004, 0.489), (2005, 0.493), (2005, 0.498), (2006, 0.502), (2006, 0.505), (2007, 0.499), (2007, 0.495), (2008, 0.493), (2008, 0.491), (2009, 0.49), (2009, 0.49), (2010, 0.49), (2010, 0.49), (2011, 0.49), (2011, 0.49)
S_car_to_W_2 = GRAPH(TIME)
(2001, 0.00), (2002, 0.266), (2002, 0.394), (2003, 0.461), (2003, 0.498), (2004, 0.519), (2004, 0.531), (2005, 0.54), (2005, 0.549), (2006, 0.554), (2006, 0.558), (2007, 0.554), (2007, 0.551), (2008, 0.55), (2008, 0.549), (2009, 0.548), (2009, 0.548), (2010, 0.548), (2010, 0.548), (2011, 0.548), (2011, 0.549)
S_PT_to_A = GRAPH(TIME)
(2001, 0.789), (2002, 0.746), (2002, 0.721), (2003, 0.706), (2003, 0.697), (2004, 0.692), (2004, 0.689), (2005, 0.688), (2005, 0.687), (2006, 0.686), (2006, 0.686), (2007, 0.685), (2007, 0.685), (2008, 0.685), (2008, 0.685), (2009, 0.685), (2009, 0.685), (2010, 0.685), (2010, 0.685), (2011, 0.685), (2011, 0.685)
S_PT_to_A_2 = GRAPH(TIME)
(2001, 0.946), (2002, 1.06), (2002, 1.05), (2003, 1.05), (2003, 1.05), (2004, 1.05), (2004, 1.05), (2005, 1.04), (2005, 1.04), (2006, 1.05), (2006, 1.05), (2007, 1.05), (2007, 1.05), (2008, 1.05), (2008, 1.05), (2009, 1.05), (2009, 1.05), (2010, 1.05), (2010, 1.05), (2011, 1.05), (2011, 1.05)
S_PT_to_C = GRAPH(TIME)
(2001, 0.6), (2002, 0.586), (2002, 0.578), (2003, 0.573), (2003, 0.571), (2004, 0.569), (2004, 0.568), (2005, 0.568), (2005, 0.569), (2006, 0.57), (2006, 0.57), (2007, 0.569), (2007, 0.568), (2008, 0.567), (2008, 0.567), (2009, 0.567), (2009, 0.566), (2010, 0.566), (2010, 0.566), (2011, 0.566), (2011, 0.566)
S_PT_to_C_2 = GRAPH(TIME)
(2001, 0.847), (2002, 0.834), (2002, 0.806), (2003, 0.789), (2003, 0.778), (2004, 0.772), (2004, 0.769), (2005, 0.768), (2005, 0.769), (2006, 0.772), (2006, 0.709), (2007, 0.709), (2007, 0.709), (2008, 0.709), (2008, 0.709), (2009, 0.709), (2009, 0.709), (2010, 0.709), (2010, 0.709), (2011, 0.71), (2011, 0.71)
S_PT_to_E = GRAPH(TIME)
(2001, 0.77), (2002, 0.726), (2002, 0.7), (2003, 0.685), (2003, 0.676), (2004, 0.671), (2004, 0.668), (2005, 0.666), (2005, 0.665), (2006, 0.664), (2006, 0.664), (2007, 0.664), (2007, 0.664), (2008, 0.663), (2008, 0.663), (2009, 0.663), (2009, 0.663), (2010, 0.663), (2010, 0.663), (2011, 0.663), (2011, 0.663)
S_PT_to_E_2 = GRAPH(TIME)
(2001, 0.912), (2002, 1.00), (2002, 1.00), (2003, 1.00), (2003, 1.00), (2004, 1.00), (2004, 1.00), (2005, 1.00), (2005, 1.00), (2006, 1.00), (2006, 1.00), (2007, 1.00), (2007, 1.00), (2008, 1.00), (2008, 1.00), (2009, 1.00), (2009, 1.00), (2010, 1.00), (2010, 1.00), (2011, 1.00), (2011, 1.00)

S_PT_to_N = GRAPH(TIME)
(2001, 0.82), (2002, 0.776), (2002, 0.75), (2003, 0.735), (2003, 0.726), (2004, 0.721), (2004, 0.718), (2005, 0.717), (2005, 0.717), (2006, 0.717), (2006, 0.718), (2007, 0.716), (2007, 0.715), (2008, 0.715), (2008, 0.714), (2009, 0.714), (2009, 0.714), (2010, 0.714), (2010, 0.713), (2011, 0.713), (2011, 0.713)
S_PT_to_N_2 = GRAPH(TIME)
(2001, 1.05), (2002, 1.21), (2002, 1.16), (2003, 1.13), (2003, 1.12), (2004, 1.11), (2004, 1.10), (2005, 1.10), (2005, 1.10), (2006, 1.10), (2006, 1.10), (2007, 1.10), (2007, 1.09), (2008, 1.09), (2008, 1.09), (2009, 1.09), (2009, 1.09), (2010, 1.09), (2010, 1.09), (2011, 1.09), (2011, 1.09)
S_PT_to_S = GRAPH(TIME)
(2001, 0.498), (2002, 0.463), (2002, 0.441), (2003, 0.428), (2003, 0.42), (2004, 0.416), (2004, 0.413), (2005, 0.412), (2005, 0.411), (2006, 0.411), (2006, 0.411), (2007, 0.412), (2007, 0.412), (2008, 0.412), (2008, 0.412), (2009, 0.412), (2009, 0.412), (2010, 0.412), (2010, 0.412), (2011, 0.412), (2011, 0.412)
S_PT_to_S_2 = GRAPH(TIME)
(2001, -0.298), (2002, -0.0365), (2002, 0.0898), (2003, 0.156), (2003, 0.193), (2004, 0.214), (2004, 0.226), (2005, 0.234), (2005, 0.241), (2006, 0.245), (2006, 0.246), (2007, 0.244), (2007, 0.243), (2008, 0.242), (2008, 0.242), (2009, 0.242), (2009, 0.242), (2010, 0.242), (2010, 0.242), (2011, 0.243), (2011, 0.243)
S_PT_to_W = GRAPH(TIME)
(2001, 0.422), (2002, 0.393), (2002, 0.371), (2003, 0.358), (2003, 0.35), (2004, 0.346), (2004, 0.344), (2005, 0.343), (2005, 0.345), (2006, 0.347), (2006, 0.349), (2007, 0.351), (2007, 0.354), (2008, 0.355), (2008, 0.357), (2009, 0.358), (2009, 0.358), (2010, 0.359), (2010, 0.359), (2011, 0.359), (2011, 0.36)
S_PT_to_W_2 = GRAPH(TIME)
(2001, 0.973), (2002, 1.07), (2002, 1.04), (2003, 1.03), (2003, 1.02), (2004, 1.01), (2004, 1.01), (2005, 1.00), (2005, 1.00), (2006, 1.01), (2006, 1.01), (2007, 1.01), (2007, 1.01), (2008, 1.00), (2008, 1.00), (2009, 1.00), (2009, 1.00), (2010, 1.00), (2010, 1.00), (2011, 1.00), (2011, 1.00)
time_to_become_long_distance_commuter = 1
time_to_stop_seeking = 1
total_adult_and_pensioner =
ARRAYSUM(Adult__married_or_cohibited[*,*])+ARRAYSUM(Adult_single[*,*])+ARRAYSUM(Pensioner__married_or_cohibited[*,*])+ARRAYSUM(Pensioner__single[*,*])
Total_adult_population = ARRAYSUM(Resident_adult_population[*])
total_commuter =
Total_adult_population-adult_with_resident_and_work[Askoy,Askoy_W]-adult_with_resident_and_work[Center_Bergen,Center_Bergen_W]-adult_with_resident_and_work[South_Bergen,South_Bergen_W]-adult_with_resident_and_work[North_Bergen,North_Bergen_W]-adult_with_resident_and_work[East_Bergen,East_Bergen_W]-adult_with_resident_and_work[West_Bergen,West_Bergen_W]
total_commuter_asane = in_saane+out_asane
total_family_commuting_time[Resident,Working] =
Adult__married_or_cohibited[Resident,Working]*Weighted_travel_time_effect[Resident,Working]
total_family_household = ARRAYSUM(Adult__married_or_cohibited[*,*])/2
total_family_search_effect[Resident,Working] =
Search_effect_AFR[Resident,Working]+Search_effect_FFTB[Resident,Working]

total_household =
ARRAYSUM(Adult__married_or_cohibited[*,*])/2+ARRAYSUM(Adult_single[*,*])+ARRAYSUM(Pensioner__married_or_cohibited[*,*])/2+ARRAYSUM(Pensioner__single[*,*])
TOTAL_job[Working] =
ARRAYSUM(Adult_single[*,Working])+ARRAYSUM(Adult__married_or_cohibited[*,Working])
total_Juveniles_proj = GRAPH(TIME)
(2012, 71353), (2013, 72169), (2013, 72997), (2014, 73658), (2015, 74377), (2015, 75047), (2016, 75911), (2017, 76832), (2017, 77728), (2018, 78596), (2019, 79636), (2019, 80681), (2020, 81666), (2021, 82563), (2021, 83462), (2022, 84244), (2023, 85011), (2023, 85605), (2024, 86148), (2025, 86703), (2025, 87312), (2026, 87774), (2027, 88178), (2027, 88532), (2028, 88836), (2029, 89080), (2029, 89272), (2030, 89410)
total_new_dwellings = GRAPH(TIME)
(2001, 2183), (2002, 1879), (2003, 2504), (2004, 2492), (2005, 3180), (2006, 2792), (2007, 2345), (2008, 2434), (2009, 2176), (2010, 1718), (2011, 1906)
total_pensioner_population =
ARRAYSUM(Pensioner__married_or_cohibited[*,*])+ARRAYSUM(Pensioner__single[*,*])
Total_search_effect[Working] =
(search_effect_FTJS[Working]+Search_effect_IJS[Working]+arraysum(Search_effect_NOJS[*,Working])+ARRAYSUM(Search_effect_SOJS[*,Working]))
TOTAL_seeker[Resident] =
ARRAYSUM(Search_effect_AFR[Resident,*])+ARRAYSUM(Search_effect_FFTB[Resident,*])+ARRAYSUM(Search_effect_SAR[Resident,*])+ARRAYSUM(Search_effect_SFTB[Resident,*])+ARRAYSUM(Destinasion_SPR[Resident,*])+ARRAYSUM(Search_effect_YS[Resident,*])+ARRAYSUM(search_effect_DAR[Resident,*])
total_single_commuting_time[Resident,Working] =
Weighted_travel_time_effect[Resident,Working]*Adult_single[Resident,Working]
total_single_search_effect[Resident,Working] =
search_effect_DAR[Resident,Working]+Search_effect_SAR[Resident,Working]+Search_effect_SFTB[Resident,Working]+Search_effect_YS[Resident,Working]
total_utility_car_f[Working] = ARRAYSUM(resident_utility_car_f[*,Working])
total_utility_car_s[Working] = ARRAYSUM(resident_utility_car_s[*,Working])
total_utility_car_worker_f[Resident] =
ARRAYSUM(workplace_utility_car_f[Resident,*])
total_utility_car_worker_s[Resident] =
ARRAYSUM(workplace_utility_car_s[Resident,*])
total_utility_PT_f[Working] = ARRAYSUM(resident_utility_PT_f[*,Working])
total_utility_PT_S[Working] = ARRAYSUM(resident_utility_PT_S[*,Working])
total_utility_PT_worker_f[Resident] =
ARRAYSUM(workplace_utility_PT_f[Resident,*])
total_utility_PT_worker_s[Resident] =
ARRAYSUM(workplace_utility_PT_s[Resident,*])
Total_working_age_population =
ARRAYSUM(Adult__married_or_cohibited[*,*])+ARRAYSUM(Adult_single[*,*])
Travel_time_to_Center_for_HL_choice[Resident] =
car_travel_time_congestion[Resident,Center_Bergen_W]

travel_time_to_work_for_HL_choice[Resident,Working] =
car_travel_time_congestion[Resident,Working]
travel_time_unit = 6
turnover_rate = 0.1
utility_car_to_center_f[Resident] =
car_travel_time_congestion[Resident,Center_Bergen_W]*travel_time_unit*Logit_for_time_to_center_f
utility_car_to_center_s[Resident] =
car_travel_time_congestion[Resident,Center_Bergen_W]*travel_time_unit*Logit_for_time_to_center_s
utility_car_to_work_f[Resident,Working] =
car_travel_time_congestion[Resident,Working]*travel_time_unit*Logit_for_time_to_work_f
utility_car_to_work_s[Resident,Working] =
car_travel_time_congestion[Resident,Working]*travel_time_unit*Logit_for_time_to_work_s
utility_car_to_work_worker_f[Resident,Working] = IF ARRAYIDX(1)=1
THEN
EXP(car_travel_time_congestion[Resident,Working]*6*Logit_for_time_to_work_f*logit_for_askoy)
ELSE
EXP(car_travel_time_congestion[Resident,Working]*6*Logit_for_time_to_work_f)
utility_car_to_work_worker_s[Resident,Working] = IF ARRAYIDX(1)=1
THEN
EXP(car_travel_time_congestion[Resident,Working]*6*Logit_for_time_to_work_s*logit_for_askoy)
ELSE
EXP(car_travel_time_congestion[Resident,Working]*6*Logit_for_time_to_work_s)
utility_PT_to_center_f[Resident] =
PT_travel_time_congestion[Resident,Center_Bergen_W]*6*Logit_for_time_to_center_f
utility_PT_to_center_s[Resident] =
PT_travel_time_congestion[Resident,Center_Bergen_W]*travel_time_unit*Logit_for_time_to_center_s
utility_PT_to_work_f[Resident,Working] =
PT_travel_time_congestion[Resident,Working]*travel_time_unit*Logit_for_time_to_work_f
utility_PT_to_work_s[Resident,Working] =
PT_travel_time_congestion[Resident,Working]*travel_time_unit*Logit_for_time_to_work_s
utility_PT_to_work_worker_f[Resident,Working] = IF ARRAYIDX(1)=1
THEN
EXP(PT_travel_time_congestion[Resident,Working]*6*Logit_for_time_to_work_f*logit_for_askoy)
ELSE
EXP(PT_travel_time_congestion[Resident,Working]*6*Logit_for_time_to_work_f)
utility_PT_to_work_worker_s[Resident,Working] = IF ARRAYIDX(1)=1
THEN
EXP(PT_travel_time_congestion[Resident,Working]*6*Logit_for_time_to_work_s*logit_for_askoy)

ELSE
EXP(PT_travel_time_congestion[Resident,Working]*6*Logit_for_time_to_work_s)
weighted_center_to_work_for_HL_choice = 2.5
Weighted_travel_time_effect[Resident,Working] =
Travel_time_to_Center_for_HL_choice[Resident]*weighted_center_to_work_for_HL_choice+travel_time_to_work_for_HL_choice[Resident,Working]
workplace_utility_car_f[Resident,Working] =
utility_car_to_work_worker_f[Resident,Working]
workplace_utility_car_s[Resident,Working] =
utility_car_to_work_worker_s[Resident,Working]
workplace_utility_PT_f[Resident,Working] =
utility_PT_to_work_worker_f[Resident,Working]
workplace_utility_PT_s[Resident,Working] =
utility_PT_to_work_worker_s[Resident,Working]
W_car_to_A = GRAPH(TIME)
(2001, 0.0667), (2002, 0.212), (2002, 0.266), (2003, 0.289), (2003, 0.301), (2004, 0.306), (2004, 0.309), (2005, 0.311), (2005, 0.314), (2006, 0.315), (2006, 0.315), (2007, 0.312), (2007, 0.311), (2008, 0.31), (2008, 0.309), (2009, 0.309), (2009, 0.309), (2010, 0.309), (2010, 0.309), (2011, 0.309), (2011, 0.309)
W_car_to_A_2 = GRAPH(TIME)
(2001, 0.00), (2002, 0.174), (2002, 0.248), (2003, 0.284), (2003, 0.302), (2004, 0.312), (2004, 0.318), (2005, 0.323), (2005, 0.328), (2006, 0.33), (2006, 0.331), (2007, 0.328), (2007, 0.327), (2008, 0.326), (2008, 0.325), (2009, 0.325), (2009, 0.325), (2010, 0.326), (2010, 0.326), (2011, 0.326), (2011, 0.326)
W_car_to_C = GRAPH(TIME)
(2001, 0.12), (2002, 0.429), (2002, 0.555), (2003, 0.604), (2003, 0.617), (2004, 0.614), (2004, 0.603), (2005, 0.602), (2005, 0.615), (2006, 0.628), (2006, 0.637), (2007, 0.64), (2007, 0.641), (2008, 0.637), (2008, 0.631), (2009, 0.625), (2009, 0.619), (2010, 0.615), (2010, 0.612), (2011, 0.61), (2011, 0.609)
W_car_to_C_2 = GRAPH(TIME)
(2001, 0.00), (2002, 0.203), (2002, 0.295), (2003, 0.341), (2003, 0.365), (2004, 0.379), (2004, 0.386), (2005, 0.393), (2005, 0.399), (2006, 0.403), (2006, 0.406), (2007, 0.401), (2007, 0.399), (2008, 0.397), (2008, 0.395), (2009, 0.395), (2009, 0.395), (2010, 0.395), (2010, 0.395), (2011, 0.395), (2011, 0.395)
W_car_to_E = GRAPH(TIME)
(2001, 0.462), (2002, 0.607), (2002, 0.655), (2003, 0.677), (2003, 0.688), (2004, 0.693), (2004, 0.696), (2005, 0.698), (2005, 0.701), (2006, 0.702), (2006, 0.702), (2007, 0.699), (2007, 0.698), (2008, 0.697), (2008, 0.696), (2009, 0.696), (2009, 0.696), (2010, 0.696), (2010, 0.696), (2011, 0.696), (2011, 0.696)
W_car_to_E_2 = GRAPH(TIME)
(2001, 0.00), (2002, 0.382), (2002, 0.568), (2003, 0.67), (2003, 0.728), (2004, 0.762), (2004, 0.781), (2005, 0.794), (2005, 0.804), (2006, 0.809), (2006, 0.812), (2007, 0.81), (2007, 0.809), (2008, 0.809), (2008, 0.808), (2009, 0.808), (2009, 0.809), (2010, 0.809), (2010, 0.809), (2011, 0.81), (2011, 0.81)
W_car_to_N = GRAPH(TIME)
(2001, 0.293), (2002, 0.603), (2002, 0.729), (2003, 0.777), (2003, 0.791), (2004, 0.787), (2004, 0.777), (2005, 0.775), (2005, 0.789), (2006, 0.802), (2006, 0.811), (2007, 0.814), (2007, 0.815), (2008, 0.811), (2008, 0.805), (2009, 0.799), (2009, 0.793), (2010, 0.789), (2010, 0.786), (2011, 0.784), (2011, 0.783)
W_car_to_N_2 = GRAPH(TIME)

(2001, 0.00), (2002, 0.393), (2002, 0.6), (2003, 0.716), (2003, 0.788), (2004, 0.835), (2004, 0.866), (2005, 0.913), (2005, 0.99), (2006, 1.07), (2006, 1.14), (2007, 1.19), (2007, 1.23), (2008, 1.25), (2008, 1.26), (2009, 1.27), (2009, 1.29), (2010, 1.30), (2010, 1.32), (2011, 1.34), (2011, 1.36)
W_car_to_S = GRAPH(TIME)
(2001, 0.24), (2002, 0.551), (2002, 0.677), (2003, 0.727), (2003, 0.741), (2004, 0.738), (2004, 0.728), (2005, 0.727), (2005, 0.742), (2006, 0.756), (2006, 0.767), (2007, 0.768), (2007, 0.769), (2008, 0.765), (2008, 0.759), (2009, 0.752), (2009, 0.746), (2010, 0.742), (2010, 0.739), (2011, 0.737), (2011, 0.736)
W_car_to_S_2 = GRAPH(TIME)
(2001, 0.00), (2002, 0.367), (2002, 0.558), (2003, 0.665), (2003, 0.732), (2004, 0.775), (2004, 0.804), (2005, 0.323), (2005, 0.328), (2006, 0.33), (2006, 0.331), (2007, 0.328), (2007, 0.327), (2008, 0.326), (2008, 0.325), (2009, 0.325), (2009, 0.325), (2010, 0.326), (2010, 0.326), (2011, 0.326), (2011, 0.326)
W_car_to_W = GRAPH(TIME)
(2001, -0.0435), (2002, 0.145), (2002, 0.224), (2003, 0.262), (2003, 0.282), (2004, 0.293), (2004, 0.298), (2005, 0.303), (2005, 0.309), (2006, 0.313), (2006, 0.315), (2007, 0.313), (2007, 0.312), (2008, 0.31), (2008, 0.307), (2009, 0.305), (2009, 0.304), (2010, 0.302), (2010, 0.301), (2011, 0.3), (2011, 0.299)
W_car_to_W_2 = GRAPH(TIME)
(2001, -0.42), (2002, -0.11), (2002, 0.0449), (2003, 0.129), (2003, 0.177), (2004, 0.204), (2004, 0.22), (2005, 0.231), (2005, 0.239), (2006, 0.244), (2006, 0.246), (2007, 0.244), (2007, 0.243), (2008, 0.242), (2008, 0.242), (2009, 0.242), (2009, 0.242), (2010, 0.242), (2010, 0.242), (2011, 0.243), (2011, 0.243)
W_PT_to_A = GRAPH(TIME)
(2001, 0.53), (2002, 0.524), (2002, 0.521), (2003, 0.519), (2003, 0.518), (2004, 0.518), (2004, 0.517), (2005, 0.517), (2005, 0.517), (2006, 0.517), (2006, 0.517), (2007, 0.517), (2007, 0.517), (2008, 0.517), (2008, 0.517), (2009, 0.517), (2009, 0.517), (2010, 0.517), (2010, 0.517), (2011, 0.517), (2011, 0.517)
W_PT_to_A_2 = GRAPH(TIME)
(2001, 0.693), (2002, 0.619), (2002, 0.616), (2003, 0.614), (2003, 0.613), (2004, 0.613), (2004, 0.613), (2005, 0.613), (2005, 0.613), (2006, 0.613), (2006, 0.613), (2007, 0.614), (2007, 0.614), (2008, 0.614), (2008, 0.614), (2009, 0.615), (2009, 0.615), (2010, 0.615), (2010, 0.615), (2011, 0.615), (2011, 0.615)
W_PT_to_C = GRAPH(TIME)
(2001, 0.85), (2002, 0.795), (2002, 0.762), (2003, 0.742), (2003, 0.729), (2004, 0.721), (2004, 0.714), (2005, 0.715), (2005, 0.726), (2006, 0.736), (2006, 0.741), (2007, 0.741), (2007, 0.739), (2008, 0.732), (2008, 0.723), (2009, 0.713), (2009, 0.704), (2010, 0.697), (2010, 0.69), (2011, 0.684), (2011, 0.68)
W_PT_to_C_2 = GRAPH(TIME)
(2001, 0.847), (2002, 0.834), (2002, 0.806), (2003, 0.789), (2003, 0.778), (2004, 0.772), (2004, 0.769), (2005, 0.768), (2005, 0.769), (2006, 0.772), (2006, 0.709), (2007, 0.709), (2007, 0.709), (2008, 0.709), (2008, 0.709), (2009, 0.709), (2009, 0.709), (2010, 0.709), (2010, 0.709), (2011, 0.71), (2011, 0.71)
W_PT_to_E = GRAPH(TIME)
(2001, 1.04), (2002, 0.949), (2002, 0.897), (2003, 0.867), (2003, 0.85), (2004, 0.841), (2004, 0.835), (2005, 0.832), (2005, 0.831), (2006, 0.831), (2006, 0.831), (2007, 0.831), (2007, 0.831), (2008, 0.831), (2008, 0.831), (2009, 0.831), (2009, 0.831), (2010, 0.831), (2010, 0.831), (2011, 0.831), (2011, 0.831)
W_PT_to_E_2 = GRAPH(TIME)

(2001, 1.14), (2002, 1.43), (2002, 1.41), (2003, 1.40), (2003, 1.40), (2004, 1.40), (2004, 1.40), (2005, 1.40), (2005, 1.40), (2006, 1.40), (2006, 1.40), (2007, 1.40), (2007, 1.40), (2008, 1.40), (2008, 1.40), (2009, 1.40), (2009, 1.40), (2010, 1.40), (2010, 1.41), (2011, 1.41), (2011, 1.41)
W_PT_to_N = GRAPH(TIME)
(2001, 1.07), (2002, 0.985), (2002, 0.934), (2003, 0.903), (2003, 0.885), (2004, 0.873), (2004, 0.864), (2005, 0.864), (2005, 0.874), (2006, 0.883), (2006, 0.888), (2007, 0.889), (2007, 0.886), (2008, 0.879), (2008, 0.87), (2009, 0.861), (2009, 0.852), (2010, 0.844), (2010, 0.837), (2011, 0.831), (2011, 0.827)
W_PT_to_N_2 = GRAPH(TIME)
(2001, 1.14), (2002, 1.44), (2002, 1.40), (2003, 1.39), (2003, 1.38), (2004, 1.38), (2004, 1.39), (2005, 1.42), (2005, 1.48), (2006, 1.56), (2006, 1.62), (2007, 1.67), (2007, 1.71), (2008, 1.73), (2008, 1.75), (2009, 1.76), (2009, 1.77), (2010, 1.78), (2010, 1.80), (2011, 1.82), (2011, 1.84)
W_PT_to_S = GRAPH(TIME)
(2001, 0.511), (2002, 0.405), (2002, 0.35), (2003, 0.32), (2003, 0.302), (2004, 0.291), (2004, 0.285), (2005, 0.28), (2005, 0.276), (2006, 0.273), (2006, 0.27), (2007, 0.267), (2007, 0.265), (2008, 0.263), (2008, 0.262), (2009, 0.261), (2009, 0.26), (2010, 0.259), (2010, 0.259), (2011, 0.259), (2011, 0.258)
W_PT_to_S_2 = GRAPH(TIME)
(2001, 1.06), (2002, 1.29), (2002, 1.28), (2003, 1.28), (2003, 1.28), (2004, 1.29), (2004, 1.30), (2005, 0.613), (2005, 0.613), (2006, 0.613), (2006, 0.613), (2007, 0.614), (2007, 0.614), (2008, 0.614), (2008, 0.614), (2009, 0.615), (2009, 0.615), (2010, 0.615), (2010, 0.615), (2011, 0.615), (2011, 0.615)
W_PT_to_W = GRAPH(TIME)
(2001, 0.551), (2002, 0.484), (2002, 0.448), (2003, 0.428), (2003, 0.416), (2004, 0.409), (2004, 0.406), (2005, 0.403), (2005, 0.401), (2006, 0.4), (2006, 0.4), (2007, 0.399), (2007, 0.398), (2008, 0.398), (2008, 0.398), (2009, 0.398), (2009, 0.398), (2010, 0.398), (2010, 0.398), (2011, 0.398), (2011, 0.398)
W_PT_to_W_2 = GRAPH(TIME)
(2001, -0.42), (2002, -0.11), (2002, 0.0449), (2003, 0.129), (2003, 0.177), (2004, 0.204), (2004, 0.22), (2005, 0.231), (2005, 0.239), (2006, 0.244), (2006, 0.246), (2007, 0.244), (2007, 0.243), (2008, 0.242), (2008, 0.242), (2009, 0.242), (2009, 0.242), (2010, 0.242), (2010, 0.242), (2011, 0.243), (2011, 0.243)
w_r_ratio[Resident] = ARRAYVALUE(TOTAL_job[*],ARRAYIDX())/Resident_adult_population[Resident]
YS_get_vacancy[Resident,Working] = Search_effect_YS[Resident,Working]*Frac_seek_get_vacancy[Resident]
YS_get_vacancy_sort_by_WP[Working] = ARRAYSUM(YS_get_vacancy[*,Working])
YS_seeker_car[Working] = Youth_seeker[Working]*frac_owing_car
YS_seeker_PT[Working] = Youth_seeker[Working]*(1-frac_owing_car)

Appendix 5. Bergensim

Bergensim is the combination model of three models about Bergen area. They are transportation model made by Torgeir Brandsar, migration model described in this thesis and land-using model made by Carolin Schulze. The connections of models are shown below.

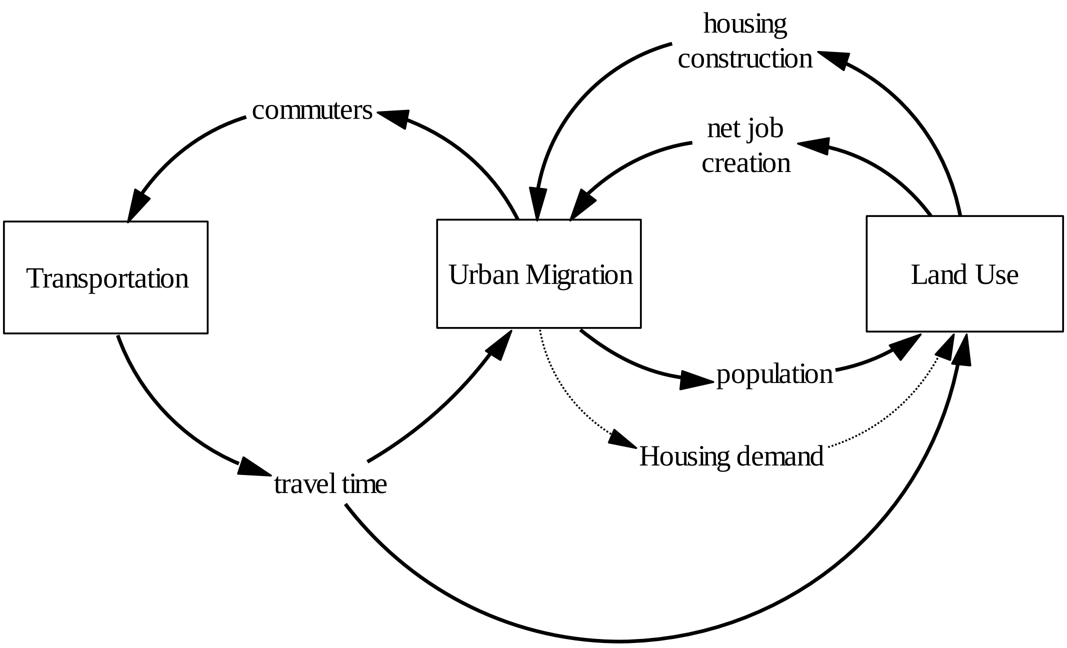
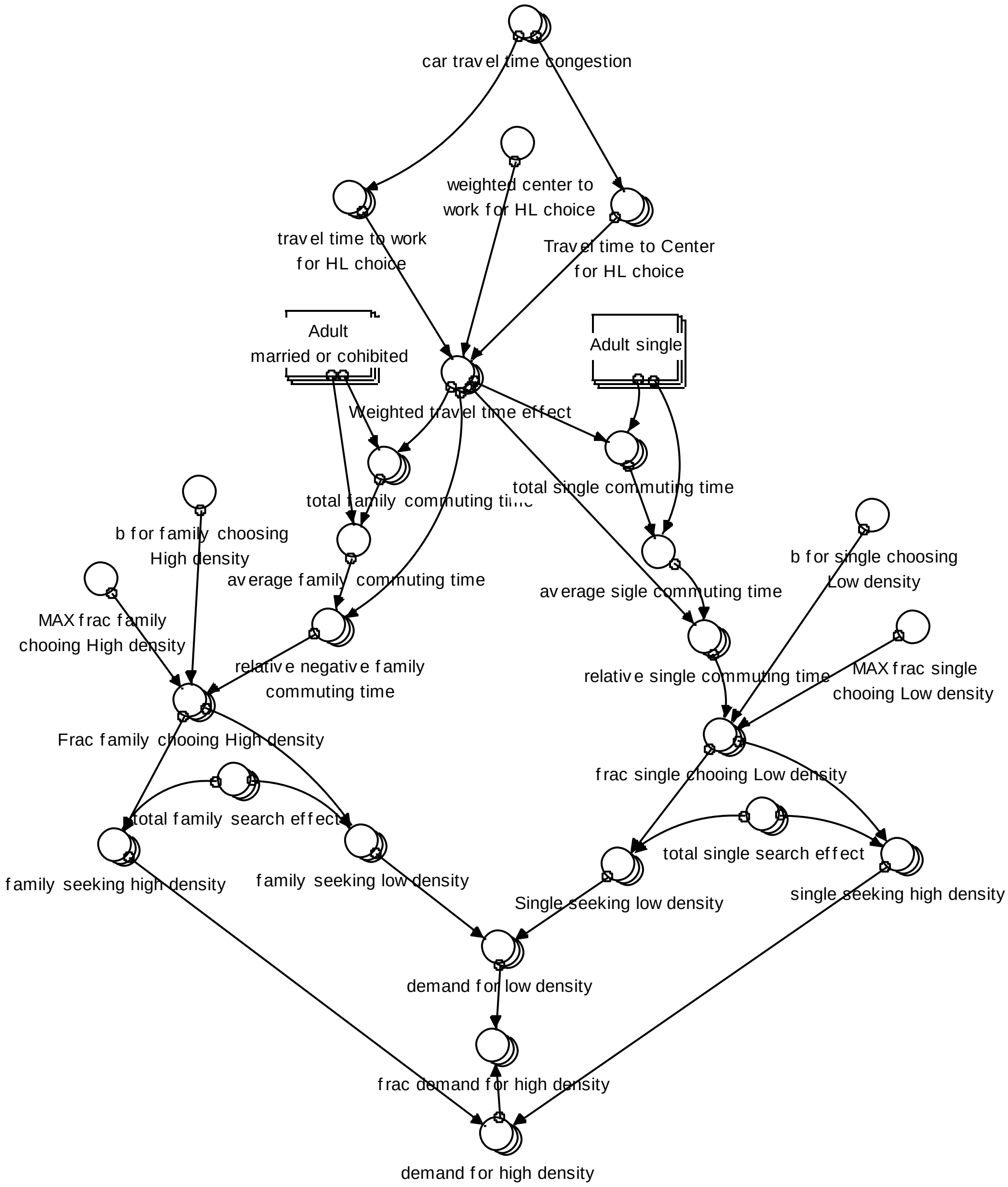


Figure by Bergensim group

With Bergensim, policies can be tested in a bigger system. It is also possible to compare policies in different sectors. For example, results for business tax abatement policy and toll fee policy can be compared.

The dotted lines shows a connection have not been done so far. The structure from migration to housing demand output has been done. The structure is shown below. The equations can also be found in appendix 3.



The housing demand is divided into high density demand and low density demand. The definition of low and high density can be found in Schulze (2013). And the demand is the sum of demand from family seekers and single seekers.

And whether a seeker might seek a high density or low density dwelling depends on the relative commuting time. If a seeker is searching a dwelling with longer commuting, it is more likely that he is searching a low density dwelling.

The relative commuting time is the ratio of seeker's commuting time by average commuting time. The commuting time here is the combination of commuting to work and travel time to city center.

The list below is the equations for connections in Bergensim. The calculation is for the sectore form net housing construction output to net housing construction input.

Askoy_housing_gap =
(Askøy.finishing_housing_construction[Low]-Askøy.housing_demolition[Low])/Askøy.housing_space_per_housing_unit[Low]+(Askøy.finishing_housing_construction[High]-Askøy.housing_demolition[High])/Askøy.housing_space_per_housing_unit[High]

Bergen_Center_housing_gap =
(Bergen_Center.finishing_housing_construction[Low]-Bergen_Center.housing_demolition[Low])/Bergen_Center.housing_space_per_housing_unit[Low]+(Bergen_Center.finishing_housing_construction[High]-Bergen_Center.housing_demolition[High])/Bergen_Center.housing_space_per_housing_unit[High]

Bergen_East_housing_gap =
(Bergen_East.finishing_housing_construction[Low]-Bergen_East.housing_demolition[Low])/Bergen_East.housing_space_per_housing_unit[Low]+(Bergen_East.finishing_housing_construction[High]-Bergen_East.housing_demolition[High])/Bergen_East.housing_space_per_housing_unit[High]

Bergen_North_housing_gap =
(Bergen_North.finishing_housing_construction[Low]-Bergen_North.housing_demolition[Low])/Bergen_North.housing_space_per_housing_unit[Low]+(Bergen_North.finishing_housing_construction[High]-Bergen_North.housing_demolition[High])/Bergen_North.housing_space_per_housing_unit[High]

Bergen_South_housing_gap =
(Bergen_South.finishing_housing_construction[Low]-Bergen_South.housing_demolition[Low])/Bergen_South.housing_space_per_housing_unit[Low]+(Bergen_South.finishing_housing_construction[High]-Bergen_South.housing_demolition[High])/Bergen_South.housing_space_per_housing_unit[High]

Bergen_West_housing_gap =
(Bergen_West.finishing_housing_construction[Low]-Bergen_West.Noname_1[Low])/Bergen_West.housing_space_per_housing_unit[Low]+(Bergen_West.finishing_housing_construction[High]-Bergen_West.Noname_1[High])/Bergen_West.housing_space_per_housing_unit[High]

Frac_new_housing[Askoy] = Askoy_housing_gap/Total_housing_gap

Frac_new_housing[Center_Bergen] =

Bergen_Center_housing_gap/Total_housing_gap

Frac_new_housing[South_Bergen] = Bergen_South_housing_gap/Total_housing_gap

Frac_new_housing[North_Bergen] = Bergen_North_housing_gap/Total_housing_gap

Frac_new_housing[East_Bergen] = Bergen_East_housing_gap/Total_housing_gap

Frac_new_housing[West_Bergen] = Bergen_West_housing_gap/Total_housing_gap

Frac_new_job[Askoy_W] =

ARRAYSUM(Askøy.positions_gap[*])/total_position_gap

Frac_new_job[Center_Bergen_W] =

ARRAYSUM(Bergen_Center.positions_gap[*])/total_position_gap

Frac_new_job[South_Bergen_W] =

ARRAYSUM(Bergen_South.positions_gap[*])/total_position_gap

Frac_new_job[North_Bergen_W] =

ARRAYSUM(Bergen_North.positions_gap[*])/total_position_gap

Frac_new_job[East_Bergen_W] =

ARRAYSUM(Bergen_East.positions_gap[*])/total_position_gap

Frac_new_job[West_Bergen_W] =

ARRAYSUM(Bergen_West.positions_gap[*])/total_position_gap

Total_housing_gap =

Askoy_housing_gap+Bergen_Center_housing_gap+Bergen_East_housing_gap+Bergen_North_housing_gap+Bergen_South_housing_gap+Bergen_West_housing_gap

total_position_gap = IF

(ARRAYSUM(Askøy.positions_gap[*])+ARRAYSUM(Bergen_Center.positions_gap[*])+ARRAYSUM(Bergen_East.positions_gap[*])+ARRAYSUM(Bergen_North.positions_gap[*])+ARRAYSUM(Bergen_South.positions_gap[*])+ARRAYSUM(Bergen_West.positions_gap[*]))=0

THEN 0.001

ELSE

(ARRAYSUM(Askøy.positions_gap[*])+ARRAYSUM(Bergen_Center.positions_gap[*])+ARRAYSUM(Bergen_East.positions_gap[*])+ARRAYSUM(Bergen_North.positions_gap[*])+ARRAYSUM(Bergen_South.positions_gap[*])+ARRAYSUM(Bergen_West.positions_gap[*]))

Appendix 6. Policies sensitively test

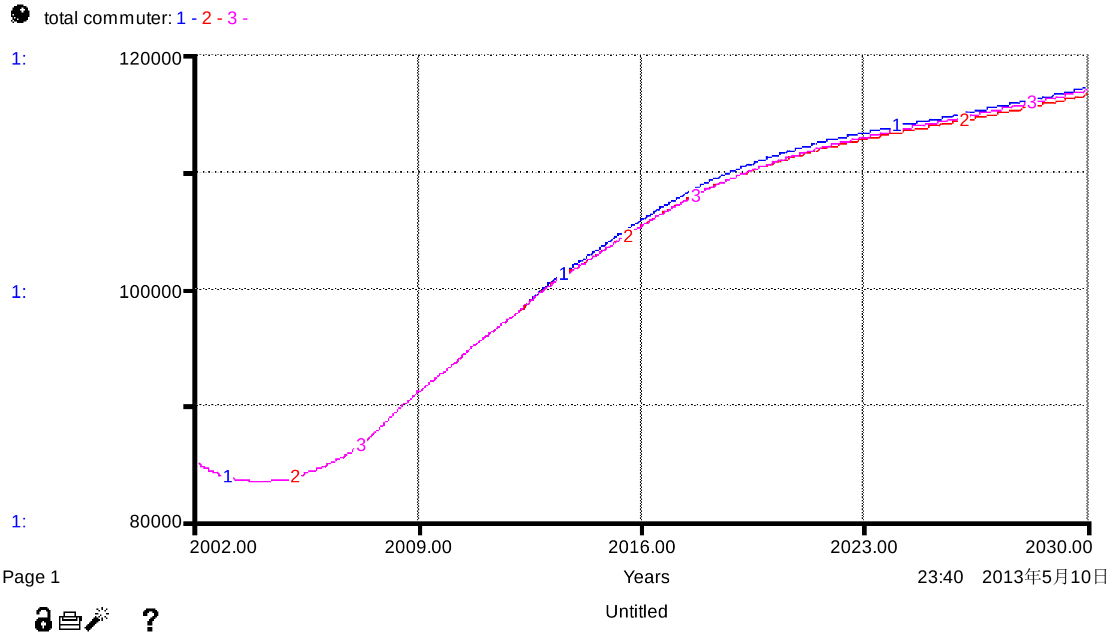


Figure: Policies sensitively test for high residential mobility.

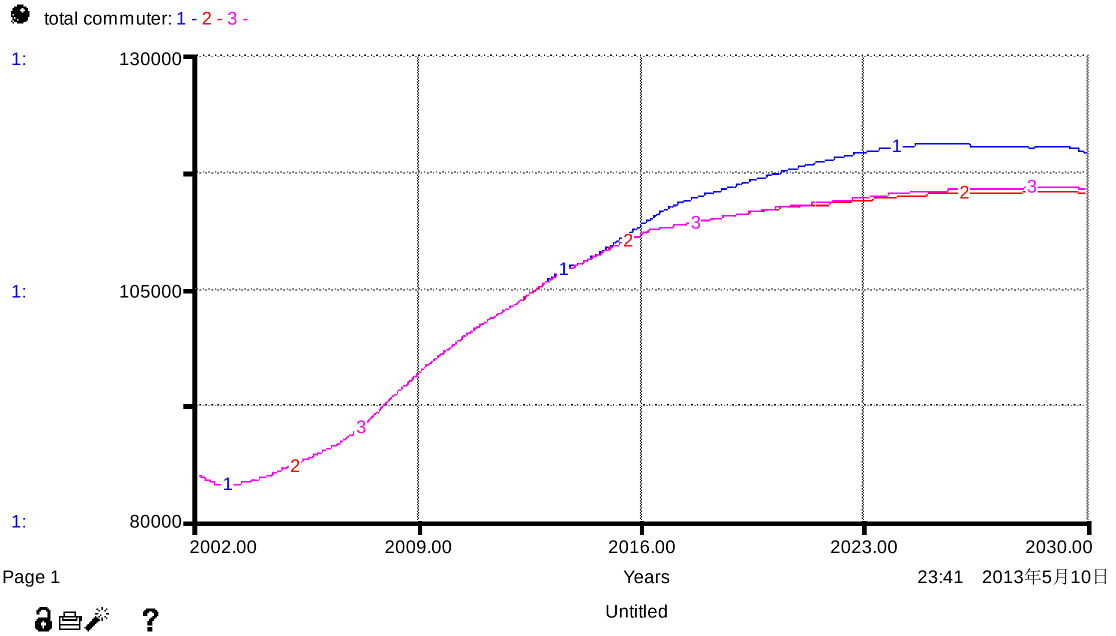


Figure: Policies sensitively test for low residential mobility.

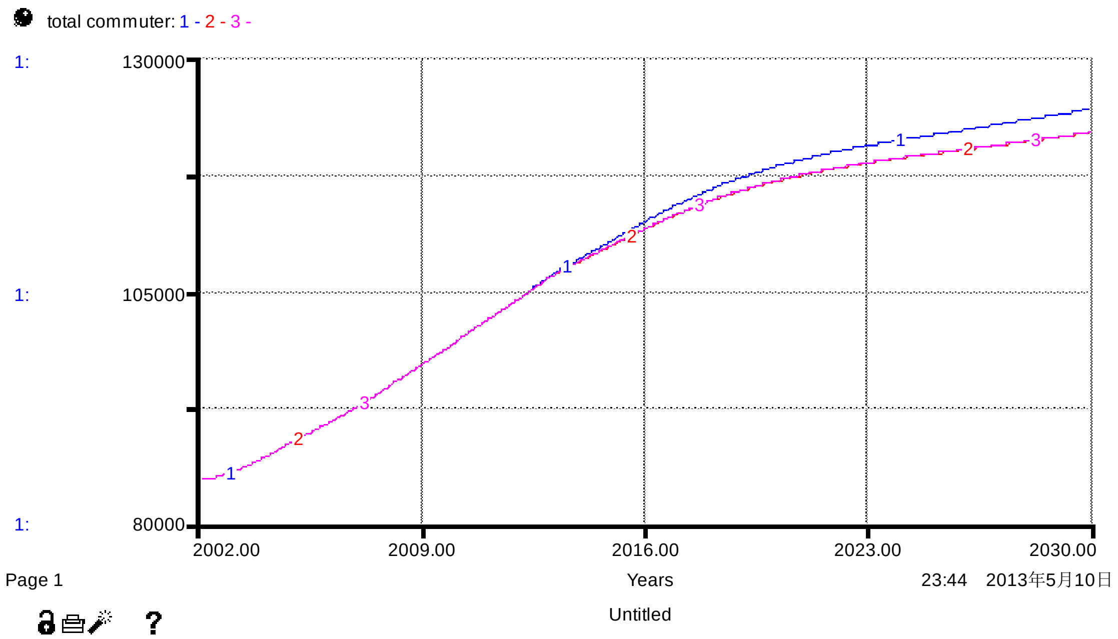


Figure: Policies sensitively test for high willingness to commute.

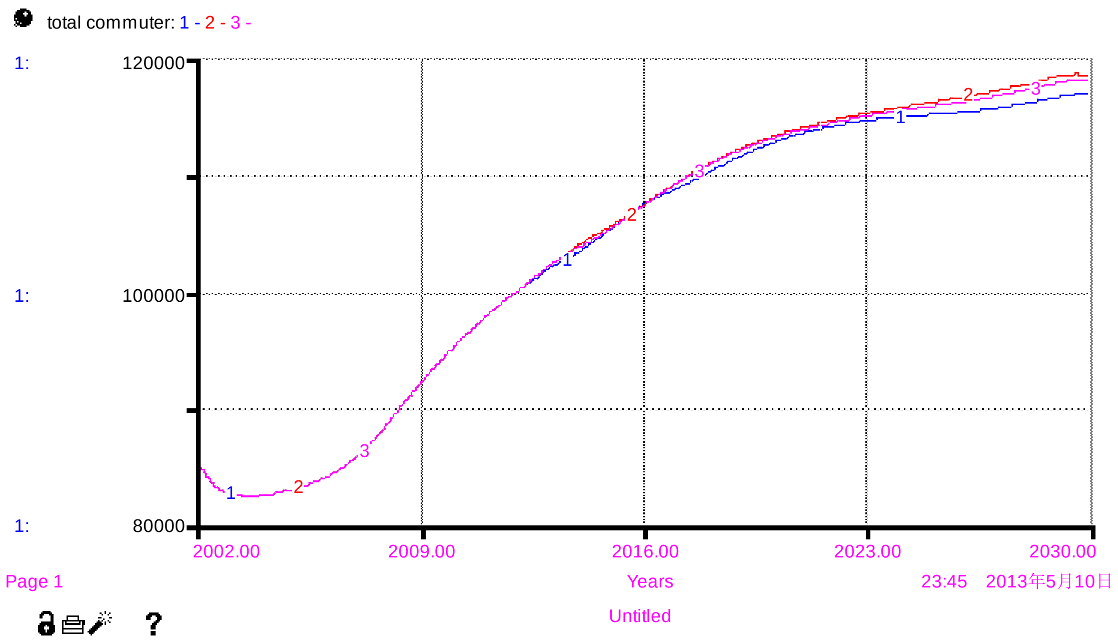


Figure: Policies sensitively test for low willingness to commute.

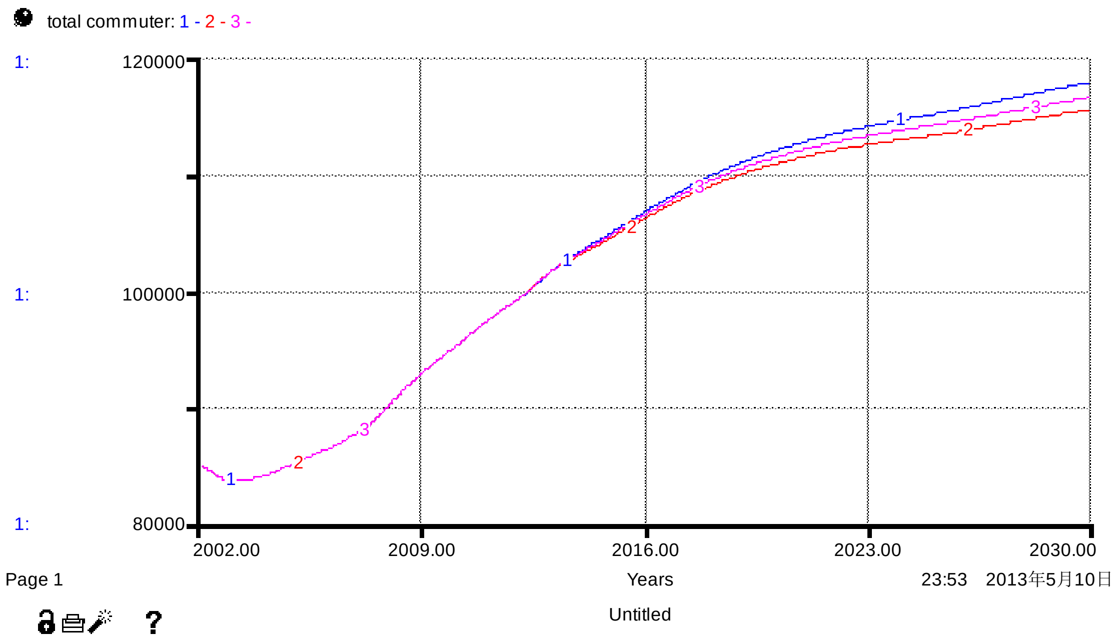


Figure: Policies sensitively test for high turnover rate.

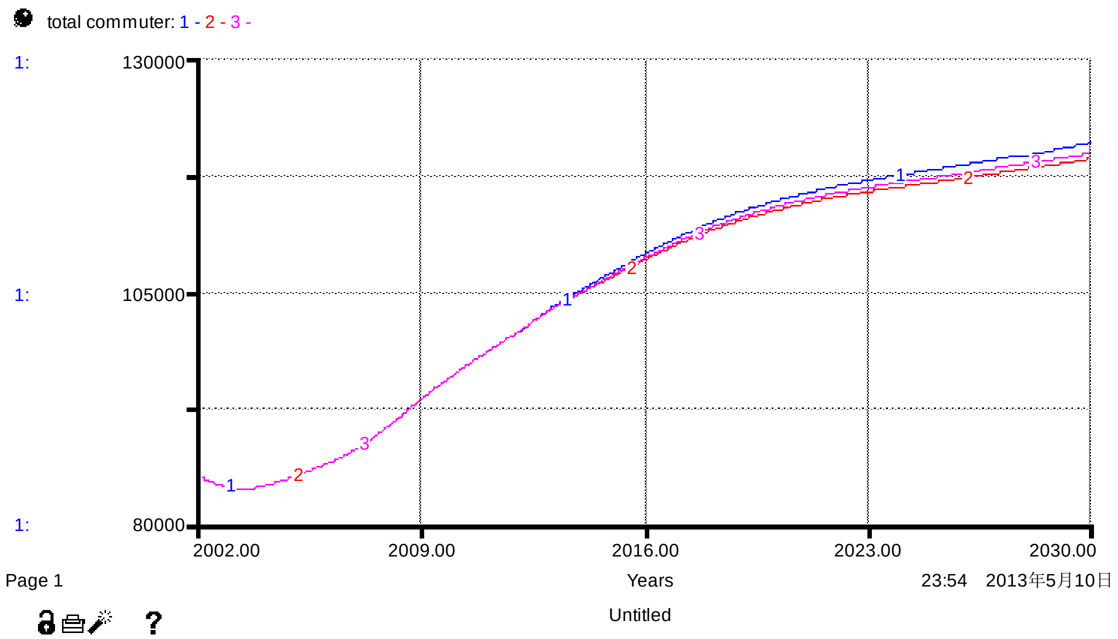


Figure: Policies sensitively test for low turnover rate.

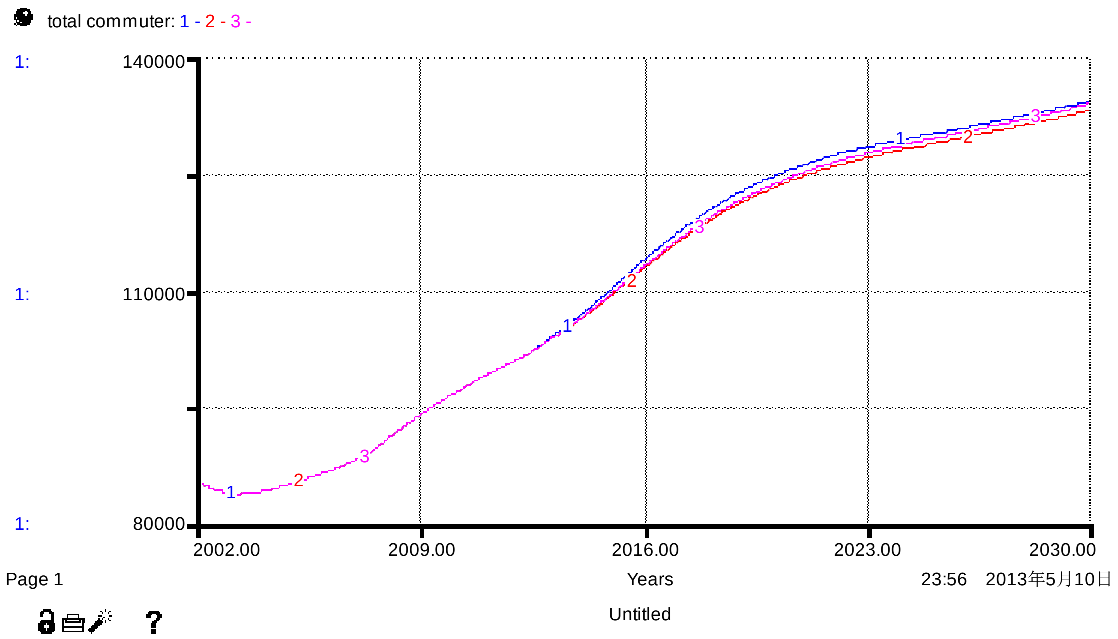


Figure: Policies sensitively test for high population increase.

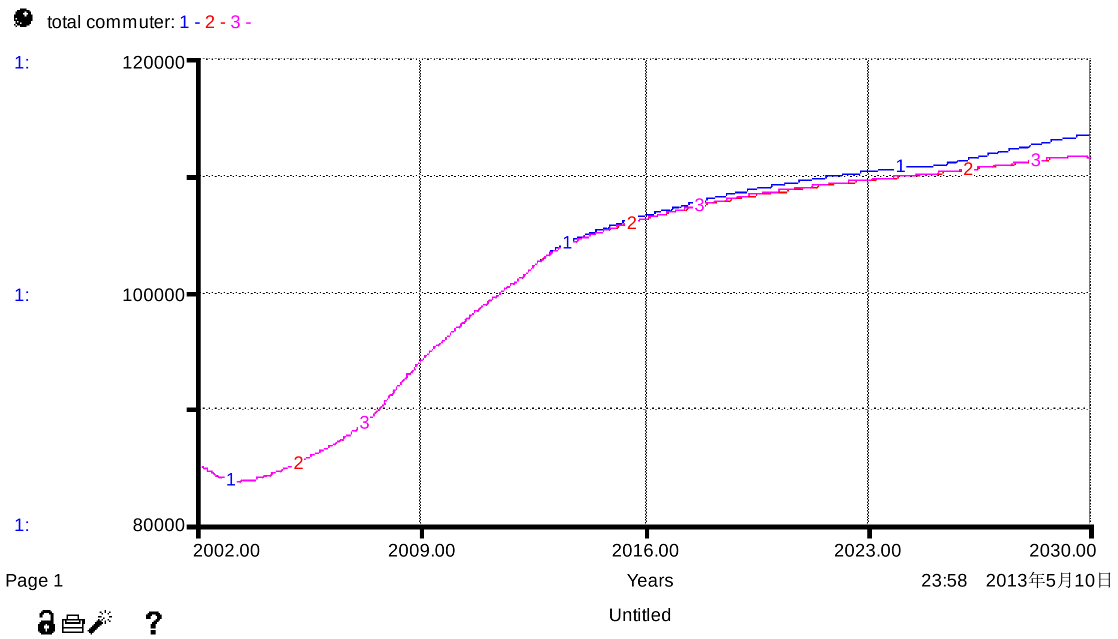


Figure: Policies sensitively test for low population increase.