The Use of Diffusion Algorithms for the Estimation of Construction Year Within Urban Models

https://github.com/emunozh/ConstructionYearTechRep

M. Esteban Muñoz H.[1]*, Irene Peters[1].
[1]Technical Urban Infrastructure Systems Group,
HafenCity University, Hamburg, Germany

October 30, 2015

Abstract

We present a mechanism developed for the estimation of missing construction years within a rich digital cadastre of the city of Hamburg. For the estimation of the missing construction years we implement a diffusion algorithm. This algorithm defines a neighbourhood for each individual building within a PostgreSQL database and subsequently ranks each building in the neighborhood according to: (1) the minimal euclidean distance between spatial objects and (2) distance between attributes of the buildings.

The model is constructed upon a set of python functions defining the different simulation steps and a script controlling the iterative procedure used within the estimation.

Results form the model present a method for the estimation of construction years for buildings of the city digital cadastre. The performance of the algorithm is computed as the difference between estimated and observed construction years of a random 1% sample of the initial building stock. The performance of the algorithm shows unsatisfactory results.

We end the paper by discussing alternative implementations for this type of models in the domain of urban modeling, further improvements of the mechanism and alternative mechanism for the estimation of missing parameters within digital building databases.

Keywords: Digital Cadastre, Diffusion, Urban Model.

*Corresponding author

Contents

1 The Role of Construction Year of Buildings in Urban Energy Models

The past years have seen a proliferation of urban models developed for the estimation of urban energy demand (Calderón *et al.* 2015; Mata *et al.* 2014; Muñoz Hidalgo 2014; Muñoz Hidalgo & Peters 2014; Chingcuanco & Miller 2012; Fracastoro & Serraino 2011). Space heating has a large share (in many cities of higher latitudes: the largest share) in urban energy demand. Space heating used to be estimated based on urban area typologies, which offer different settlement types associated with typical heat demands (Roth & Häubi 1980, 1981; Blesl 2002; Everding 2004; Blesl *et al.* 2007; Genske *et al.* 2009; Erhorn-Kluttig 2011). These heat demands were based on empirical findings in case studies.

The dominant methods for estimating space heating demand today focus on the individual building. They either use building typologies with typical energy demand coefficients for each building type (requiring the analyst to classify individual buildings into types) or compute heat demand at the individual building level.

For both methods, the construction year of buildings is a variable of paramount importance. It is the construction year which exerts the greatest influence on the physical properties of the building envelope, and building envelope, in turn, is the single most important determinant of heat demand in many countries of higher latitude, as it is responsible for heat transmission losses. Construction techniques and legal requirements on the energy efficiency of buildings vary between construction epochs. In Germany,

for example, there have been legal requirements on the energy efficiency of newly constructed buildings since the 1970s. They have been updated, on average, every twelve years, setting increasingly stringent standards on the heat transmission properties of the building shell.

Construction years are not always available though. For example, in the electronic cadastre of the city of Hamburg (a city of 1,8 million inhabitants and 300.00 buildings) around half of all residential buildings do not carry a construction year. In this paper, we offer several algorithms for estimating construction years based on existing data and illustrate the effect of these methods. The algorithms follow a diffusion logic, aiming to replicate the real-world processes of the past that gave rise to new construction in different parts of the city over the years.

Diffusion models have been applied in urban simulation to the topic of technology diffusion. Linder (2013) uses a diffusion algorithm for the simulation of the installation of photovoltaic panels and its contribution to an increasing distributed energy supply system. Guseo & Guidolin (2008) implement a cellular automate framework for the simulation of innovation diffusion, the authors compare their model with an aggregate Generalized Bass Model (GBM), the author argue in favor of a disaggregated, simulation based approach. Schmid & Madlener (2008) develop an agent-based model to simulate the diffusion of biogas plants within Switzerland. Peters *et al.* (2002) develop an agent-based model which incorporates the diffusion of a novel sanitation technology.

The aim of our paper is to provide city and energy planners with a method to gain important information for the estimation of current and future heat demand, needed for dimensioning and managing a city's heat supply infrastructure.

There is a tendency towards the development of disaggregated simulation based models for the estimation and forecast of urban energy demand models (Balaras *et al.* 2007; Kavgic *et al.* 2010; Dascalaki *et al.* 2010, 2011; Dall'O' *et al.* 2012; Caputo *et al.* 2013; Hrabovszky-Horváth *et al.* 2013; Kragh & Wittchen 2013; Singh *et al.* 2013) For a review of energy demand models see Swan & Ugursal (2009) and Keirstead *et al.* (2012), on both papers the authors argue in favor of disaggregated simulation models for the estimation of energy demand.

2 Diffusion Mechanisms

The use of diffusion models has been traditionally used on models developed to represent occurring natural phenomena. The use of diffusion models has also been used on models developed for the simulation of technology diffusion over time. A constant among diffusion models has been the concept of time. The evolution or spatial development of a technology, idea or population. In this paper we make use of a diffusion model to

represent a static phenomena. The idea behind the use of a diffusion model within a static framework is to take advantage of an interaction between elements of the model. This interaction, define as the diffusion process, is an iterative process. In a classical definition each iteration would represent a simulation step in time. In the presented paper the iterative process does not represent time and is only present as a tool to achieve a state of convergence within the model.

The focus on the diffusion terminology has the aim to facilitate the understanding of the developed algorithm for the estimation of missing years. The goal of this algorithm is the estimation of missing construction years by selecting a similar building within a given radius, the "know" construction years propagate in space until every building has a "known" construction year. This propagation of years gives our algorithm the characteristics of a diffusion model.

In the energy planning community the use of diffusion models are not very common. A good example is the use of technology diffusion algorithm for the simulation of the diffusion of PV panels in a large region (Linder 2013). The use of a diffusion model is also used by Bagchi *et al.* (2013) for the simulation of a fire spread of an electricity grid. Wittmann & Bruckner (2007) present an agent-based model developed for the simulation of energy supply systems under liberalized markets. Within this model the authors define a technology diffusion model based on the individual decisions of the model agents. Many cellular automata models could be described as a diffusion algorithm, the use of a cellular automata for the forecasting of urban growth could be classified as diffusion models (Han *et al.* 2009; Batty *et al.* 1999). Guseo & Guidolin (2008) makes use of a cellular automata model for the development of a technology diffusion model.

In this paper we present a small diffusion model for the estimation of missing construction years, the single most important parameter for the estimation of heat consumption of individual buildings. Nonetheless, the presented model can be used for the simulation of all kind of diffusion phenomena. An explain before the aim of this model is not to accurately represent simulation time but to achieve a predefined model convergence, the estimation of all building construction years. In order to use the presented model as a dynamic model the user would need to explicit define the handling of time steps within the framework.

For the development of the model we make use of the python language, we see python as the de-facto language for any simulation dealing with spatial referenced objects. De model development in the python language allows us to use powerful tools like the PostgreSQL database without much difficulty. At the same time the language offers powerful libraries for data manipulation (pandas) and for spatial objects (shapely). The use of the developed functions are presented in an Ipython notebook, making the reproduction of results extremely easy.

3 The Data We Work With

In this paper we apply the developed algorithm to a preprocessed data sample retrieved from the official digital cadastre of the city of Hamburg ALKIS (AdV 2008). The selected data is from year 2010 because many of our models use the year 2010 as a basis year. The entire digital cadastre is now open and can be freely downloaded from daten-hamburg.de[1]. The digital cadastre provides many information about the urban environment of the city. For this analysis we only make use of the data set describing the building stock. The data set describing the building stock contains information about every single building in the city. Each building is represented in space with an accurate geometry of its perimeter ans some attributes like: building use, construction type of construction year (see Table 3). The latter is an attribute that is not available for every building in the database. Table 2 list the share of buildings with an unknown construction year in the 2010 digital cadastre. 30% of all buildings and 51% of all residential buildings have a known construction year. We use this buildings as a starting point for the estimation of construction years for the rest of the building of the digital cadastre.

Table 1: Some of the attributes of the individual buildings on the digital cadastre

Attribute Description	**Name**	**Data type**
Unique Identifier	UUID	String
Construction type	BAW	Categorical
Building function	GFK	Categorical
Construction year	BAJ	Integer
District	Stadtteil	String
Living area	sqm	Float
Shell area of wall	shell_wall	Float
Shell area of building	shell	Float
ID of k nearest neighbours	neighbours	Array
Statistical area	statgeb	String
Geometry of the building	geometry	Shapely Polygon
Simplified geometry of the building	simple_geometry	Shapely Polygon

The data uploaded with the model scripts is a preprocessed data from the digital cadastre. With the available data we populate a PostgreSQL database and index the database with two columns containing geographical information. In order to simplify and increase the simulation speed of our models we abstract the detailed geometrical information of the individual buildings (Muñoz Hidalgo 2015 –in Press–). The used data contains only the simplified geometrical information. The geometrical information of the buildings is used in the model to compute the minimal distance between objects (through the shapely package) and in the PostgreSQL database for the definition of

[1] `http://daten-hamburg.de/geographie_geologie_geobasisdaten/ALKIS_Liegenschaftskarte/`

Table 2: Available records from the digital cadastre

Used filter **residential**	**# of records**	**Share**	**. . . from from total**
Records on the digital cadastre.	369 416	100%	/
➽ . . . with a known construction year	110 845	**30%**	/
➽ . . . residential buildings	194 996	53%	100%
➽ . . . with a known construction year	98 504	27%	**51%**

neighbourhoods.

In the presented paper we interpret the data set as a pandas data frame. This is possible because the building sample is relative small with 1757 buildings. In order to process larger amounts of data we make use of the PostgreSQL database from which we read the needed input data and write the estimated new data.

In order to allow the full reproducibility of the model we provide the used data set as a `hdf` file. This file can be directly imported as a pandas data frame into any python working environment. Other software tools should be able to read and interpret most of the data saved on this file, the geometrical data of the individual buildings is stores as Well Known Text (WKT) objects, a standard define by the Open Geospatial Consortium (OGC). We also provide the used script to communicate to our PostgreSQL database to generate the hdf file for information purposes.

4 Structure, Rationale, Results and Suggestion for Refining the Mechanism

On this section we present the main components of the developed model. First we describes the definition of neighbourhoods within the selected building sample. We define this neighborhoods within the PostgreSQL database in order to take advantage of the advance spatial indexing of the database. With the neighbourhood define we present the main algorithm used for the definition of the diffusion model, On a next step we rank each neighbour according to its characteristics for the estimation of construction years. We end the main section of the paper by discussing the results and proposing further improvements for the model.

4.1 Definition of k-nearest Neighbours

An essential component of the diffusion model is the definition of a neighborhood. There are many ways to define a neighbourhood. In this example we have a simple static definition of neighbourhood. We allocate neighbours to each element within our PostgreSQL database. The definition of neighbourhood is created as a function of two variables. The first variable is the radius r to select possible neighbours and the second the amount of neighbours k to select. Because the definition of this neighbourhood is rather simple we can define it within the spatial database, we make this via an SQL script. This script is performed for each building in the database.

$$N_{[1-k],i} = sort(B_{n,i}) \tag{1}$$

$$B_{n,i} = B_j \forall j : \text{within buffer}(B_i, r) \tag{2}$$

Where:

B	Buildings sample
N_i	k number of Neighbours within radius r of centroid of building i

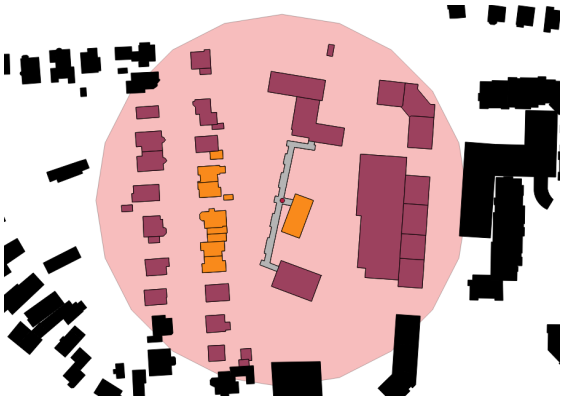


Figure 1: Example of the k-nearest algorithm with $k = 10$ and a radius $r = 100$

The developed scripts for the definition of neighbourhoods is described on scrips `spatialQNeighbours.sql`. On this scripts the radius is set to $r = 500$ meters (see code line 10) and the number of buildings, define as the nearest k neighbours is set to $k = 30$ on the last code line. The SQL script selects the first k elements because the elements are sorted by euclidean distance to the selected building for which we are defining the neighbourhood. The variable within this SQL script is the selected building `UUID`, in this case the `UUID` is set to `DEHHALKA10000w1Y`.

A graphical representation of the neighbourhood definition is depicted on Figure 1. In this case we define the radius $r = 100$ and selected the nearest 10 buildings $k = 10$. This neighbourhood definition is static because it is equal for every single building in the database, in the final section of this paper we argue for a more elaborated definition of neighborhood. In order to develop more elaborated neighbourhood definitions we need a systematic mechanism to asses the performance of the different definitions. For the presented paper the definition of a neighborhood is not central as we aim to focus on the diffusion algorithm itself, a more elaborated neighborhood definition can be implemented at any time within the presented diffusion model.

4.2 Diffusion Model in Python

In the following section we present the developed model and discuss the individual model components used in the model. The individual components used in the model have been arrange as python function and each function is define within an individual python file. This arrangement has been chosen in order to easy the reader search for a specific model component, a disadvantage of this arrangement is that it created some small redundancies within the model. An implementation example of the individual components of the model is presented as an **ipython-notebook** file. The file can be visualize within the github repository under: `https://github.com/emunozh/ConstructionYearTechRep/blob/master/Main.ipynb` but the reader wont be able to run the file. In order to run the file and the individual scripts the reader will have to download the repository and configure a python environment with the required libraries listed on the home page of the repository. We also provide the small data-set used in this paper as a hdf (high density format) file. The file can be directly imported as a pandas dataframe.

The algorithm will estimate the construction year for all buildings in a database without a known construction year. The algorithm iterates through the data set until: (1) all buildings have a know construction year; or (2) the number of performed iterations reaches the maximum predefined number of iterations (variable `MAX_ITERATIONS`) defined in the global scope of the script; or (3) there is no diffusion in the iteration.

The python function `_getByear` defines the main iteration loop through the selected building sample for the estimation of all missing construction years, a simplified version of this function is presented on file `_getByear.py`. This function takes a *Pandas* data frame as input. For each building with an unknown construction year the loop calls a second python function: `_estimateByear`. The python function `getNeighbours` selects the predefined neighbours from the pandas data frame. The simplest implementation of this function is a subsample of the data frame by index, the function can pass SQL commands to the spatial database in the case of missing neighbours of the pandas data frame. The implementation of this function is not further discuss on this paper.

In this paper we present two functions than can be used as the `_estimateByear`

function depending on how the libraries are imported into the `_getByear` script. The two functions use: (1) a simple median computation of all known construction years of the building neighbours and (2) a ranking algorithm based on characteristics of the neighbours.

The first python function, called `_estimateByearMedian` is presented on the python file with the same name. The function takes two parameters as input: (1) a row of the original building sample (a row of the pandas data frame) i.e. an individual building; and (2) the entire data sample, needed to fetch the predefined neighbours. This script will simply take the median from all neighbours with a know construction year as the estimated construction year. A possible extension to this function would be to estimate the construction year as a weighted median based on distance and characteristics of the neighbours. We expand the algorithm into a ranking algorithm that will take into account distance and characteristics of the buildings but does not use a madian for the estimation of construction year. The biggest problem with this method is that the algorithm assumes a dominance of the neighbourhood for the estimation of the unknown construction year. In the presented graphical representation of the neighbourhood definition (Figure **??**) it is clear to see to which building ensemble the selected building corresponds, an estimation of its construction year would clearly be wrong. We aim to estimate the construction year of all buildings by selecting the closest and most similar building to the selected one.

4.3 A Ranking Algorithm

In order to define which building the most similar building is, we use the available building attributes from the digital cadastre to rank them based on the absolute difference to the selected building. The available and used characteristics from the digital cadastre are listed on Table 3. Below we make a small description of the used attributes and comment on how the data is interpret in order to rank the neighbours. The implemented code is presented on File `_getNeighbours.py`.

Table 3: Attributes of the individual buildings used for the clustering of buildings

Attribute Description	**Name**	**Data type**	**Used**
Construction type	BAW	Categorical	* as dummy variable
Building function	GFK	Categorical	* as dummy variable
ID of k nearest neighbours	neighbours	Array	* euclidean distance
Geometry of the building	geometry	Shapely Polygon	* euclidean distance

BAW cl: 11–18 The building construction type is used to: (1) distinguish between residential and non residential buildings, as non residential buildings do not have a

define BAW; and (2) differentiate between different construction types of residential buildings. If the selected building has a BAW the distance value can take three possible values: (1) = 1 if the neighbour building does not have a BAW value (is not residential); (2) = 0.5 if the BAW is different to the selected buildings; and (3) = 0 if the BAW values are the same. If the selected building does not have a BAW value the distance is equal to 1 if the neighbour has a BAW value and 0 otherwise. See Equation 3.

$$N_{i,j,baw} = \begin{cases} \text{if } B_{i,baw} = \emptyset & \begin{cases} 0 & \text{if } N_{i,j,baw} = \emptyset \\ 1 & \text{else} \end{cases} \\ \text{else} & \begin{cases} 0 & \text{if } B_{i,baw} = N_{i,j,baw} \\ 1 & \text{else if } N_{i,j,baw} = \emptyset \\ 0.5 & \text{else} \end{cases} \end{cases} \tag{3}$$

GKF cl: 19–21 The building function is interpreted as a boolean variable. If the building function is the same the distance value is set to 0 otherwise the value is set to 1. See Equation 4.

$$N_{i,j,gfk} = \begin{cases} 0 & \text{if } B_{i,gfk} = N_{i,j,gfk}\, 1 \\ \text{else} \end{cases} \tag{4}$$

SQM cl: 22–24 The floor space distance is expressed as the percentage absolute distance between the selected and the neighbouring buildings. See Equation 5.

$$N_{i,j,sqm} = |B_{i,sqm} - N_{i,j,sqm}| \tag{5}$$

Shell Area cl: 25–28 Analogue to the floor space the building shell area distance is computed as the percentage absolute distance between buildings. See Equation **??**.

$$N_{i,j,shell} = |B_{i,shell} - N_{i,j,shell}| \tag{6}$$

Euclidian Distance cl: 29–32 The euclidean distance between building is computed internally by the shapely library and is computed as the minimal distance between geometries. See Equation 7.

$$N_{i,j,dis} = distance(B_{i,dis} - N_{i,j,dis}) \tag{7}$$

With the computed individual distance measures from the building attributes we compute a rank for each neighbour j of building i. The rank is computes as the percent-

age sum of all the computed distances, the mathematical expression of the computed rank is presented on Equation 8.

$$rank_{i,j} = \sum_k N_{i,j,k} \div \max\left(\sum_k N_{i,j,k}\right) \tag{8}$$

An example of the ranking algorithm is depicted on Figure 2. The figure shows the ranking of each predefine neighbour for a random selected building of the sample. Each building is ranked on a scale from 1 to 0. The lower the value the higher the similitude to the selected building. The corresponding values for this neighborhood are listed on Table 4. The ranking algorithm does not differentiate between attributes, each attribute has the same weight to the final ranking of the neighbour. On a more elaborated algorithm the weighting of attributes could be modified globally of dynamically depending of the characteristics of context of the selected building or based of available training data. Similar to the definition of the neighbourhood such a elaborated algorithm can only be assessed within a systematic comparison of different implementations. The implementation of the ranking computation occurs within the neighbourhood definition (see File `_getNeighbours.py`), the function using the computed rank called `_estimateByearRank` is presented on File `lst:_estimateByearRank.py`. The algorithm will simple take the minimum value and define its construction year from the selected neighbour. Within this algorithm we could define a maximum allowed ranking. This value could be estimated by statistical means based on a robust sample with known construction years or computed through a calibration method with a predefined training data set. Notice that the building geometry depicted on Figure 2 is a simplified building geometry. This simplification process is described in detail by Muñoz Hidalgo (2015 –in Press–).

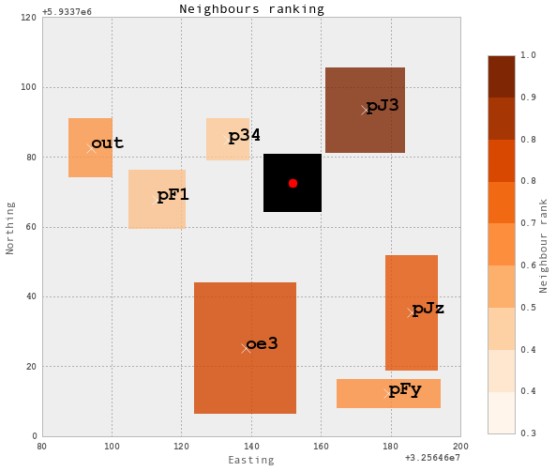


Figure 2: Ranking of neighbours based on building characteristics and euclidean distance

Table 4: Neighbours rank based on building characteristics and euclidean distance

	baw	gfk	bja	sqm	shell_wall	distance	rank
oe3	0.5	0	2002	1.00	0.50	0.42	0.77
pF1	0.5	0	2001	0.00	0.00	0.47	0.31
pJ3	0.5	1	2008	0.63	1.00	0.02	1.00
p34	0.5	0	2001	0.08	0.21	0.09	0.28
pJz	0.5	0	1961	0.42	0.84	0.46	0.70
pFy	0.5	0	1950	0.03	0.05	1.00	0.50

This process does not find a solution for all buildings, depending on the parameters defining the neighborhood a building might not have any suitable building within this neighborhood in order to estimate it's construction year. In order to deal with this issue the algorithm has to run a couple of times so that all buildings have a known construction year. On every iteration the original data-set is updated and all estimated construction years are interpreted as known construction years. This iterative process emulates a diffusion process, isolated buildings will eventually get a neighbor with similar characteristics in order to estimate it's construction year.

The computed results are presented in this section. The main results are presented with the help of two figures: (1) a figure showing the diffusion process by differentiating the known and unknown construction years of the building sample (see Figure 3) and (2) showing the estimated construction years for of each iteration (see Figure 4).

On this example the algorithm can't estimate a construction year for each building. This is because the neighborhood definition is performed for the entire city, leaving the building at the edge of the sample without a neighbourhood. The implementation of this algorithm only looks for neighbouring buildings within the preselected sample, the implemented algorithm within an under developing agent based model will fetch the missing buildings from the PostgreSQL database. There might also be problems with the neighborhood definition. The two variables, r and k, are set globally. This is clearly not ideal, a proper radius should be set based on the urban context, for more compact urban setting a small radius could be good but won't be enough to define a good neighborhood on a less dense urban area. Even within homogeneous urban areas we find large building for which the algorithm might not find any neighbour because they are larger than the predefined radius.

Figure 3 shows clearly the diffusion process within the building sample. On iteration $iter = 0$ (initial state) only a handful of buildings have a known construction year (black). After the first iteration most of the buildings have a know construction year and after 6 iterations the algorithm can't find any more construction years for the remaining buildings. The algorithm is not able to estimate the construction year for every

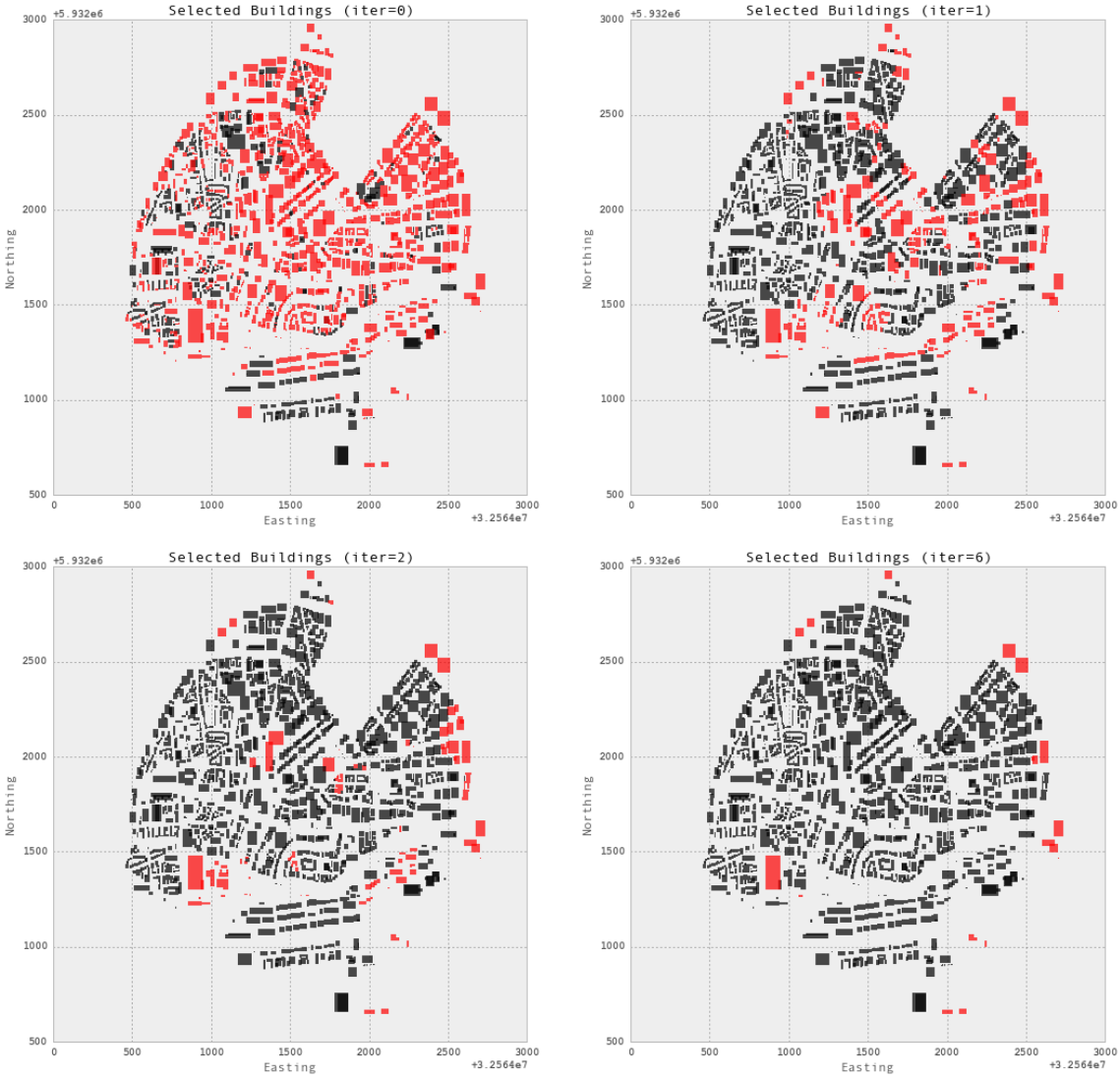


Figure 3: Diffusion process of estimated years for iteration 0, 1, 2 and 6. Unknown construction years in red

single building because of the initial definition of neighborhood. It is clear that a better definition of neighbours needs to be develop in order to improve the performance of the algorithm.

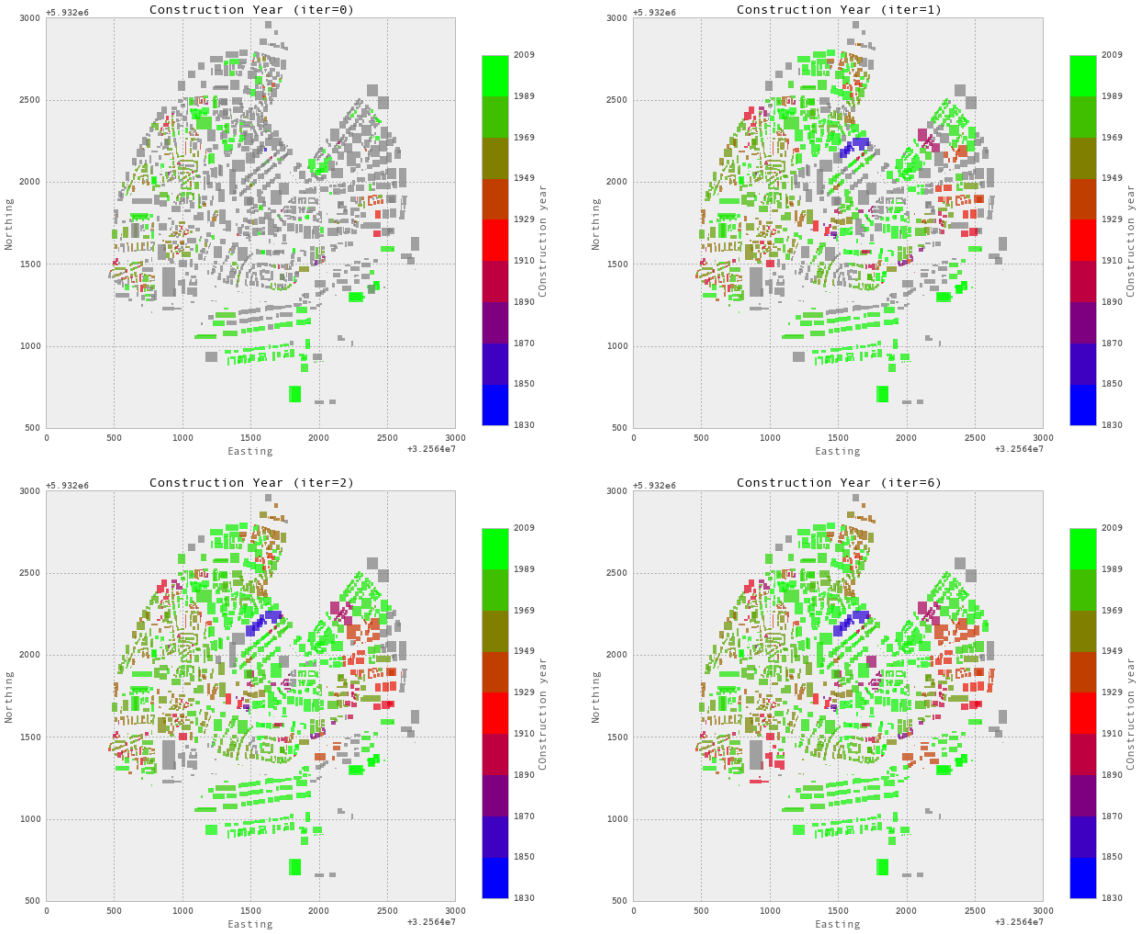


Figure 4: Estimated construction years on iteration 0 (initial state), 1, 2, and 6 (final state)

Figure 4 shows the estimated construction years for the same iterations as the previous figure. The diffusion process only takes distance between buildings and attributes as parameters to decide the diffusion direction. Available data of the urban context like streets should be included in the algorithm to decide the diffusion direction. On the east side of the building sample we can clearly identify a red section of buildings. The urban fabric of this section of the city is not constructed along a north-south axis but rather on a east-west axis. A island of buildings arranged from north to south is rather unlikely for this particular urban setting. The introduction of street data into the algorithm could help with this inconsistencies.

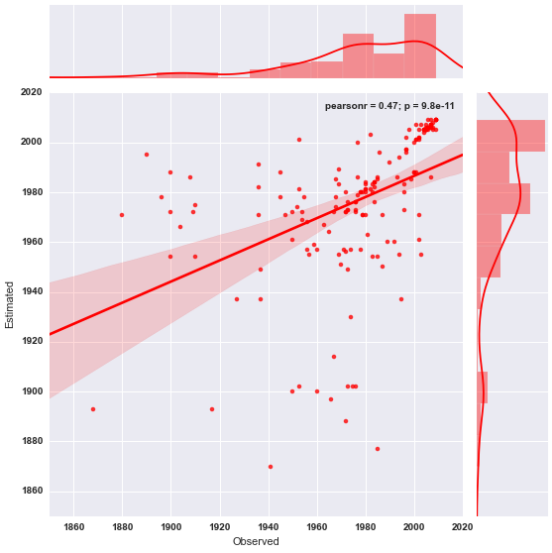
4.4 Performance of the Mechanism

In this sub section we discuss and present a method to asses the performance of the developed mechanism for the estimation of missing construction years within digital building stock databases. The code for this performance quantification method is also submitted as an Ipython Notebook as part of the supplementary material to this paper[2].

In order to quantify the performance of the algorithm we perform a second estimation of construction years. For this second estimation we select a random 1% of buildings with a known construction year from the original sample and change them to unknown values. After the estimation procedure we compare the estimated years for the random sub sample with the original known construction years. Figure 5a shows the relationship between observed and estimated construction years for the randomly selected 1% buildings and the pearson R coefficient.

We compare the performance of the mechanism with a simple linear regression model[3]. The results form the linear regression model are displayed on Figure 5b. This plot is analogue to the plot (a) showing the performance of the diffusion mechanism.

(a) diffusion algorithm



(b) linear regression

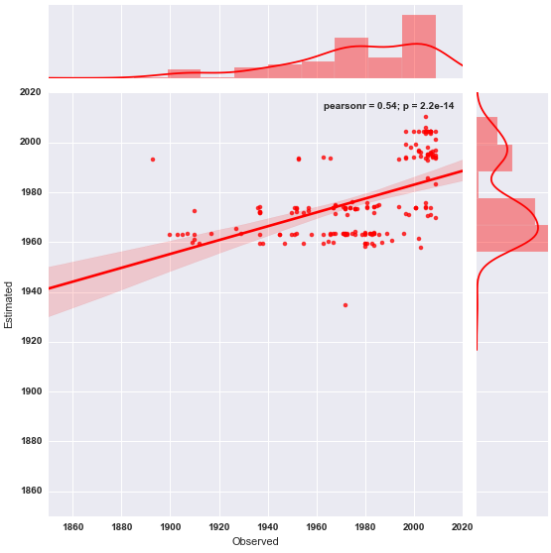


Figure 5: Performance of the algorithm showing some estimated construction years and the real "observed" construction years

[2] https://github.com/emunozh/ConstructionYearTechRep/blob/master/Performance.ipynb
[3] https://github.com/emunozh/ConstructionYearTechRep/blob/master/PerformanceOLS.ipynb

Table 5: Performance of the diffusion and linear regression mechanisms

Method	**Pearson R^2**	**Mean**	**SD**
Diffusion	0.47	18.09	26.89
Linear Regression	0.54	17.95	18.08

The performance results for the developed mechanism are unsatisfactory. There is a clear bias towards newer construction years but also a big spread in the estimation. The results from the linear model are better that the diffusion model but equally unsatisfactory.

Many heat estimation models use a neighborhood based mechanism for the estimation of heat demand (Hermelink *et al.* 2011; Munoz 2011; Muñoz Hidalgo & Peters 2014; Muñoz Hidalgo 2015 –in Press–). This type of models need to incorporate the uncertainties attached to this type of estimation mechanism into the overall assessment. Better yet, we need better methods for the estimation of missing data within urban stock datasets.

4.5 On How to Improve the Mechanism

The introduction of streets as barriers between neighbourhoods could be used for a more accurate definition of neighbourhood. In the same line as the previous conclusions we argue for a more elaborate definition of neighbourhoods. The architecture of the diffusion model proved to be straightforward and can be easily improved. In the other hand the definition of neighbourhoods requires more attention and the processing of more data and the construction of algorithms capable to deal with complex data interactions.

Another point to improve the model is the definition of a ranking limit. If the computed rank isn't below a globally (or locally) define threshold the building won't be taken into account. In this paper we did not define such a threshold because of the following two reasons: (1) we see the use of this type of parameters attached to some kind of model calibration, we need to define a robust method to calibrate this type of parameters; and (2) the definition of a ranking limit won't work with the current ranking definition because the ranking is made at a neighbourhood level rather than globally. In order to set a ranking limit the model needs to be able to benchmark the computed ranks to a predefined level.

We want to take advantage of the rich available spatial data from the Hamburg digital cadastre and start connecting the already available urban objects from the database. The use of streets, urban transport networks, green areas as well as water bodies can be integrated for a more accurate definition of neighborhoods. An accurate definition of neighborhood will allow us to simulate all kind of urban diffusion processes.

The presented results show a bias towards new constructed buildings. This is because new buildings are more likely to have a known construction year that older buildings are. We need to develop a mechanism to control for this bias. A possibility it to predefine a construction year distribution at an aggregate level and stochastically estimate the construction year of buildings with a monte-carlo method until the prior distribution is met varying a construction year weighting parameter.

5 Outlook: Further Refinements and Applications

In this paper we present a diffusion model developed for the estimation of unknown construction year of buildings described geometrically in space. We make use of well known python libraries for the definition of the functions used within the model. This model takes a predefine neighborhood (defines within a PostgreSQL database) and ranks each building in the neighbourhood based on the minimum euclidean distance to the building with an unknown construction year as well as the difference between buildings attributes.

The estimation process is repeated iteratively until each building in the database has a known construction year. This process gives the model a diffusion character. We see the development of this model as a first step towards more elaborate urban models able to: (1) appropriately represent neighbourhoods for the simulation of diffusion processes and (2) a model able to estimate missing data from spatial databases, needed for the estimation of energy demand at a low level of aggregation.

We aim to improve this method by developing an algorithm able to weight the individual characteristics of the buildings in a dynamic fashion based on contextual information of specific urban areas, both available at a micro level or aggregated at a statistical area, or available statistics of the national building stock.

We also aim to further develop the definition of neighborhoods. We aim to develop an algorithm for the construction of neighborhoods, able to define dynamic neighborhoods with: (a) a clustering algorithm or (b) rich contextual spatial elements. The use of a clustering algorithm would allow us to construct more homogenic neighborhoods.

The interaction between models working a different level of scale is also an interesting path to follow. As mentioned above, the use of predefined distributions available at an aggregate level as simulation benchmarks is an interesting path to follow not only for a static estimation of missing data but for the projection of simulations. Muñoz Hidalgo *et al.* (2015) project the retrofit of the building stock at an aggregate level and use these projections at benchmarks for the simulation of heat demand at a micro level. In a similar fashion simulation results from a diffusion model could be benchmarked to aggregated distributions.

6 Appendix:

See `https://github.com/emunozh/ConstructionYearTechRep` for the entire repository.

You can clone the repository to run in locally via:
`git clone https://github.com/emunozh/ConstructionYearTechRep.git`

The **mechanise** description can be read directly on github under:
`https://github.com/emunozh/ConstructionYearTechRep/blob/master/Main.ipynb`
or via nbviewer under:
`http://nbviewer.ipython.org/github/emunozh/ConstructionYearTechRep/blob/master/Main.ipynb`.

The **performance** of the mechanism can be read directly on github under:
`https://github.com/emunozh/ConstructionYearTechRep/blob/master/Performance.ipynb`
or via nbviewer under:
`http://nbviewer.ipython.org/github/emunozh/ConstructionYearTechRep/blob/master/Performance.ipynb`

The **OLS** performance can be read directly on github under:
`https://github.com/emunozh/ConstructionYearTechRep/blob/master/PerformanceOLS.ipynb`
or via nbviewer under:
`http://nbviewer.ipython.org/github/emunozh/ConstructionYearTechRep/blob/master/PerformanceOLS.ipynb`

Bibliography

AdV (2008). Dokumentation zur Modellierung der Geoinformationen des amtlichen Vermessungswesens: (GeoInfoDok): Erläuterungen zu ALKIS® Version 6.

Bagchi, A., Sprintson, A. & Singh, C. (2013). Modeling the impact of fire spread on an electrical distribution network. *Electric Power Systems Research* **100**, 15 – 24.

Balaras, C. A., Gaglia, A. G., Georgopoulou, E., Mirasgedis, S., Sarafidis, Y. & Lalas, D. P. (2007). European residential buildings and empirical assessment of the hellenic building stock, energy consumption, emissions and potential energy savings. *Building and Environment* **42**(3), 1298–1314.

Batty, M., Xie, Y. & Sun, Z. (1999). Modeling urban dynamics through gis-based cellular automata. *Computers, Environment and Urban Systems*

23, 205–233. URL http://www.bartlett.ucl.ac.uk/casa/latest/software/files-for-download/ceus-paper.pdf.

Blesl, M. (2002). *Räumlich hoch aufgelöste Modellierung leitungsgebundener Energieversorgungssysteme zur Deckung des Niedertemperaturwärmebedarfs*. Ph.D. thesis, Univ, Stuttgart and Stuttgart and Stuttgart. URL http://elib.uni-stuttgart.de/opus/volltexte/2002/1193/pdf/FB92.pdf.

Blesl, M., Kempe, S., Ohl, M. & Fahl, U. (2007). Wärmeatlas Baden-Württemberg: Erstellung eines Leitfadens und Umsetzung für Modellregionen: Institut für Energiewirtschaft und Rationelle Energieanwendung. Universität Stuttgart. URL http://elib.uni-stuttgart.de/opus/volltexte/2009/4840/.

Calderón, C., James, P., Urquizo, J. & McLoughlin, A. (2015). A GIS domestic building framework to estimate energy end-use demand in UK sub-city areas. *Energy and Buildings* **96**, 236 – 250.

Caputo, P., Costa, G. & Ferrari, S. (2013). A supporting method for defining energy strategies in the building sector at urban scale. *Energy Policy* **55**, 261–270.

Chingcuanco, F. & Miller, E. J. (2012). A microsimulation model of urban energy use: Modelling residential space heating demand in ilute. *Computers, Environment and Urban Systems* **36**(2), 186–194.

Dall'O', G., Galante, A. & Torri, M. (2012). A methodology for the energy performance classification of residential building stock on an urban scale. *Energy and Buildings* **48**, 211–219.

Dascalaki, E. G., Droutsa, K., Gaglia, A. G., Kontoyiannidis, S. & Balaras, C. A. (2010). Data collection and analysis of the building stock and its energy performance—an example for hellenic buildings. *Energy and Buildings* **42**(8), 1231–1237.

Dascalaki, E. G., Droutsa, K. G., Balaras, C. A. & Kontoyiannidis, S. (2011). Building typologies as a tool for assessing the energy performance of residential buildings – a case study for the hellenic building stock. *Energy and Buildings* **43**(12), 3400–3409.

Erhorn-Kluttig, H. (2011). *Energetische Quartiersplanung: Methoden - Technologien - Praxisbeispiele*. Stuttgart: Fraunhofer-IRB-Verl.

Everding, D. D. (2004). Leitbilder und Potenziale eines solaren Städtebaus. URL http://www.ecofys.com/files/files/zusammenfassung_000.pdf.

Fracastoro, G. V. & Serraino, M. (2011). A methodology for assessing the energy performance of large scale building stocks and possible applications. *Energy and Buildings* **43**(4), 844–852.

Genske, D. D., Jödecke, T. & Ruff, A. (2009). Nutzung städtischer freiflächen für erneuerbare energien. URL http://www.bbsr.bund.de/BBSR/DE/Veroeffentlichungen/BMVBS/Sonderveroeffentlichungen/2009/DL_NutzungFreiflaechen.pdf.

Guseo, R. & Guidolin, M. (2008). Cellular automata and riccati equation models for diffusion of innovations. *Statistical Methods and Applications* **17**(3), 291–308.

Han, J., Hayashi, Y., Cao, X. & Imura, H. (2009). Application of an integrated system dynamics and cellular automata model for urban growth assessment: A case study of shanghai, china. *Landscape and Urban Planning* **91**(3), 133–141.

Hermelink, A., Manteuffel, B. v., Lindner, S. & John, A. (2011). Flächendeckende erhebung des energetischen zustandes des hamburger gebäudebestandes: Kurzdarstellung der arbeitsschritte.

Hrabovszky-Horváth, S., Pálvölgyi, T., Csoknyai, T. & Talamon, A. (2013). Generalized residential building typology for urban climate change mitigation and adaptation strategies: The case of hungary. *Energy and Buildings* **62**, 475–485.

Kavgic, M., Mavrogianni, A., Mumovic, D., Summerfield, A., Stevanovic, Z. & Djurovic-Petrovic, M. (2010). A review of bottom-up building stock models for energy consumption in the residential sector. *Building and Environment* **45**(7), 1683–1697.

Keirstead, J., Jennings, M. & Sivakumar, A. (2012). A review of urban energy system models: Approaches, challenges and opportunities. *Renewable and Sustainable Energy Reviews* **16**(6), 3847–3866.

Kragh, J. & Wittchen, K. (2013). Development of two danish building typologies for residential buildings. *Energy and Buildings* **68, Part A**, 79 – 86.

Linder, S. (2013). *Räumliche Diffusion von Photovoltaik-Anlagen in Baden-Württemberg*. Ph.D. thesis, Universität Würzburg, Würzburg. URL http://opus.bibliothek.uni-wuerzburg.de/volltexte/2013/7778/pdf/Linder_Susanne_Diss_WGA109.pdf.

Mata, É., Kalagasidis, A. S. & Johnsson, F. (2014). Building-stock aggregation through archetype buildings: France, germany, spain and the UK. *Building and Environment* **81**, 270 – 282.

Muñoz Hidalgo, M. E. (2014). A microsimulation approach to generate occupancy rates of small urban areas. In: *2nd Asia Conference on International Building Performance Simulation Associatio, IBPSA Japan*. Nagoya University, Japan. URL http://ibpsa.org/proceedings/asim2014/140_AsimC3-28-370.pdf.

Muñoz Hidalgo, M. E. (2015 –in Press–). Construction of building typologies from a regional material catalog: Assessment of urban heat demand and the environmental impact of retrofit policies. *Management of Environmental Quality - An international journal* **-**.

Muñoz Hidalgo, M. E. & Peters, I. (2014). Constructing an urban microsimulation model to assess the influence of demographics on heat consumption. *International Journal of Microsimulation* **7**(1), 127–157.

Muñoz Hidalgo, M. E., Vidyattama, Y. & Tanton, R. (2015). The influence of an ageing population and an efficient building stock on heat consumption patterns. In: *14th International Conference of the International Building Performance Simulation Association (IBPSA).*

Munoz, E. (2011). Determining energy characteristics of the building stock: Wedding micro- and macro approaches.

Peters, I., Brassel, K.-H. & Spörri, C. (2002). A microsimulation model for assessing urine flows in urban wastewater management. In: *Integrated Assessment and Decision Support. Proceedings of the First Biennial Meeting of the International Environmental Modelling and Software Society.* (Rizzoli, A. E. & Jakeman, A. J., eds.).

Roth, U. & Häubi, F. (1980). *Wechselwirkungen zwischen der Siedlungsstruktur und Wärmeversorgungssystemen : Forschungsprojekt BMBau RS II 4-704102-77.10(1980).* [Schriftenreihe des Bundesministers für Raumordnung, Bauwesen und Städtebau / 6] Schriftenreihe des Bundesministers für Raumordnung, Bauwesen und Städtebau / Deutschland, Bundesrepublik, Bundesminister für Raumordnung, Bauwesen und Städtebau. Bonn.

Roth, U. & Häubi, F. (1981). Wechselwirkungen zwischen der siedlungsstruktur und wärmeversorgungssystemen. *Schweizer Ingenieur und Architekt* **99**(44), 970–983.

Schmid, C. & Madlener, R. (2008). Diffusion der Biogastechnologie in der Schweiz: eine GIS-basierte Multiagenten-Simulation. *Zeitschrift für Energiewirtschaft* **32**(4), 271–279.

Singh, M. K., Mahapatra, S. & Teller, J. (2013). An analysis on energy efficiency initiatives in the building stock of liege, belgium. *Energy Policy* **62**, 729–741.

Swan, L. G. & Ugursal, V. I. (2009). Modeling of end-use energy consumption in the residential sector: A review of modeling techniques. *Renewable and Sustainable Energy Reviews* **13**, 1819–1835.

Wittmann, T. & Bruckner, T. (2007). Agentenbasierte modellierung urbaner energiesysteme. *WIRTSCHAFTSINFORMATIK* **49**(5), 352–360.