



Raising Open and User-friendly Transparency- Enabling Technologies for Public Administrations



Project number 645860
H2020-INSO-2014

D4.6 Beta version of SIM

(Final, version 1.0, 30/01/2017)



WISE & MUNRO



Document produced by

Organisation: Warsaw School of Economics

Authors in alphabetic order: Marcin Czapryna (MC), Bogumił Kamiński (BK), Przemysław Szufel (PS), Michał Wasiluk (MW), Anna Wiertelowska (AW)

Corresponding author / mail: Przemysław Szufel / pszufe@sgh.waw.pl

Subject: Beta version of SIM

Due date: 31 January 2017

Dissemination level: [Select among Public PU, Confidential CO, Classified CI]

Reviewed and approved by

Date	Name	Organization
23-01-2017	Jonathan Groff	CNRS Paris
19-01-2017	Ilias Trochidis	Ortelio

Revision History

Version	Date	Authors	Status	Description of Changes
0.1	14-10-2016	MC/BK/PS	Draft	Organization – verification and updates of SIM alpha version
0.2	17-11-2016	MC	Draft	Dublin synthetic population
0.3	14-12-2016	MC	Draft	Exponential random graph models and problem description added
0.4	18-12-2016	MC/BK	Draft	Model description updated
0.4.1	20-12-2016	BK/PS/AW	Draft	Initial ODGM stochastic decision modelling (SilverDecisions) documentation
0.5	07-01-2017	BK	Draft	Simulation results updated
0.5.2	07-01-2017	AW/PS	Draft	SilverDecisions Documentation provided
0.6	07-01-2017	MC	Draft	Extension of simulation results description
0.7	14-01-2017	PS/AW/BK	Draft	Decision tree modelling – conclusions from Pilot meetings
0.8	20-01-2017	PS	Draft	Final draft for internal review
0.81	23-01-2017	PS/MC/BK	Draft	Rewording of selected paragraphs
0.92	30-01-2017	PS	Final	Final version

TABLE OF CONTENTS

1	Introduction.....	8
2	Simulating opinion dynamics in social networks.....	9
2.1	introduction	9
2.2	Requirements.....	10
2.2.1	Opinion representativeness issues	10
2.2.2	Population simulation and SPOD	11
2.3	Motivation.....	11
2.3.1	Elicitation of preferences	13
2.3.2	Synthetic population modelling.....	13
2.3.3	Statistical properties of social networks	15
2.3.4	Opinion dynamics	16
2.3.5	ExpOnenTial Random Graph models	17
3	Architectural issues for simulation design.....	21
3.1	Agent based modelling and simulation.....	21
3.2	Algorithm for reconstruction of synthetic populationS	24
3.3	Meta-modelling.....	26
4	Pilot data and simulation experiments result.....	33
4.1	Pilot's data	33
4.1.1	Prato	33
4.1.2	Dublin	37
4.2	Simulation Results.....	43
4.2.1	Simulation experiments.....	43
4.2.2	Opinion dynamics in social networks	45
4.2.3	Simulation results - conclusions	57
5	PA's decision process visualisation and analysis for the ODGM	58
5.1	SilverDecisions' Development process.....	58
5.1.1	Pilot's requirements	58
5.1.2	Development methodology	58
5.1.3	User Activity Monitoring.....	62
5.2	User interface layout.....	64
5.3	Modelling decision trees with SilverDecisions	65
5.3.1	Decision tree model.....	66
5.3.2	Create your first decision tree with SilverDecisions	68
5.3.3	Actions supported by application	72
5.3.4	Application settings	75
5.3.5	Usage tricks and known issues	78
6	Conclusions.....	81
7	Bibliography.....	82
8	Appendix A – Feedback received From Pilots.....	86
8.1	ODGM decision modelling platform visualization feature proposal	86
8.1.1	Conclusions from discussion with pilots	86
8.1.2	Use cases scenarios discussed with pilots	87
8.2	Scenarios for supporting PA in presenting decision making process to citizen within the ODGM model	89
8.3	Pilot's reactions.....	92
9	Appendix B – Simulation algorithm details	93
10	Appendix C – Tools for ODGM data visualisation for SPOD	98
11	Appendix D – SilverDecisions' development guide	99

11.1	Dependencies	99
11.2	Project structure	99
11.2.1	Source files directory structure	100
11.2.2	Main GIT repository branches:	100
11.3	Quick start.....	100
11.4	Building and working with project	100
11.4.1	Gulp tasks	100
11.4.2	Sample usage	100
11.4.3	Configuration	101
11.4.4	Custom javascript events	103
11.5	JSON file format	103
12	Appendix E – Responsible Research and Innovation criteria in ICT (RRI-ICT)	105

Acronyms

The following are definitions and acronyms used within the document:

Term	Definition
OD	open data
SPOD	<u>S</u> ocial <u>P</u> latform of <u>O</u> pen <u>D</u> ata
ICT	Information Communication Technology
HTML	HyperText Markup Language
CSS	Cascading Style Sheets,
IE	Internet Explore
API	Application Program Interface
DOM	Document Object Model
WP	Work Package
PA	Public Administration
ODGM	Open Data Governance Model

Executive Summary

In this report, we consider a scenario where a Public Administration (PA) uses an online social platform to give citizens access to open data and to collect information regarding citizens' preferences. Apart from the core functionalities provided in SPOD, namely: giving access to the data, visualizing the data and allowing for discussions and collaboration between citizens and the PA about the data, there is also a need on the PA side (and possibly also on behalf of third party NGOs) for open data governance. The SIM deliverable provides implementation of the open data governance model developed to support SPOD deliverables.

In order to develop the best possible Open Data Governance Model and implement it in the SIM deliverable, on the basis of the outline of the assumptions and objectives formulated in the ROUTE-TO-PA proposal, we have used the following methods to define the functional and non-functional requirements for SIM:

- detailed discussions of the requirements of PAs participating in the ROUTE-TO-PA project (the summary of the collected requirements is given in Appendix A)
- a review of the literature to identify best practices in modelling such problems;
- Internal discussions with the SPOD platform delivery team in order to coordinate SIM and SPOD functionalities and data models.

This procedure allowed us to formulate detailed requirements for SIM. Following the ROUTE-TO-PA delivery plan, SIM will be integrated with SPOD. Therefore, it is feasible that the requirements given in this document might be augmented in the subsequent implementation stages of ROUTE-TO-PA, based on feedback from the pilot implementation. Thus, the objective of the Beta version of SIM, is to extend the baseline formulation presented in the alpha pilot PA participants, to review and augment them so as to best meet the requirement to produce a flexible and usable Open Data Governance Model. We provide a revised functional and non-functional requirements along with a baseline implementation of SIM.

It should be noted, however, that conceptually SIM is a complex solution, especially in the area providing the ability to analyse citizens' opinions expressed on the SPOD platform. Consequently, in the Beta version of SIM we already provide an extended, initial implementation of this module (including data visualisation module and ODGM decision process modelling module).

The main functionalities of the SIM module can be grouped into 2 main categories:

- Build network models for understanding opinion diffusion in social networks and provide tools for generalization of the opinions expressed and collected in the SPOD platform across the entire population
- Providing ODGM tools for PA to model stochastic decision process (also collaboratively with citizens and NGOs) and present the conclusions (decision scenarios with uncertainty) to citizens in a readable manner.

The beta version of the SIM report extends and updates the alpha version in many areas. In particular, we have extended the general problem definition after thorough discussions with pilots. Prato synthetic population description has been extended and a new population model of Prato has been added.

In this report, after several discussions with pilots and analysing their needs, we also have developed a completely new functionality – SilverDecisions. The need for this tool was a results of a discussion with Pilots (Prato, Dublin) that took place during the project meeting in Prato – September 13th-15th, 2016.

SilverDecisions is a tool for PA to model and present decision process to citizens along the ODGM framework. The software has been tightly integrated with SPOD and is available in several SPOD modules. However, in order to increase project dissemination factor it also can also be run in the standalone mode.

1 INTRODUCTION

The report presents the Beta version of the SIM module for the SPOD Platform. In this report, we also present a Beta version of the open data governance model (ODGM). The goal of the ODGM is to help the Public Administration (PA) provide information about citizens' activities on SPOD and in particular to design an efficient system for elicitation of social preferences within heterogeneous communities. The social platform for open data (SPOD) allows citizens to monitor the allocation and spending of financial resources, controlling PA and hence increasing its efficiency. The SPOD platform is described in DL 4.1. For the purposes of this model we assume that the PA shares the information with citizens and thus PA actions are controlled by a participatory society. This improved knowledge of societal needs and preferences leads to more efficient decisions and regulatory actions taken by the PA. It should be noted, that the open data increases transparency and thus not only the PA control by the citizens but also the citizens involvement in local affairs. Moreover, the discussions among citizens and between PA and citizens can be based on the empirical evidence and fact.

The main functionalities of the SIM module can be grouped into 2 main categories (1) build network models for understanding opinion diffusion in social networks and provide tools for generalization of the opinions expressed and collected in the SPOD platform across the entire population and (2) providing ODGM tools for PA to model, present and explain decision process with uncertainty (also allowing to collaborate on the decision making process with citizens and NGOs). Additionally, within the SIM module we have developed a set of open source tools for Open Data visualisation available as SPOD datalet plugin (see Appendix C).

In order to understand the key challenge faced when citizens' opinions are analysed, it is important to note that the opinions of the subpopulation that uses SPOD may be not representative of the entire population. In the deliverable we develop a method for analysing the dynamics of preferences in the population according to the limited online social network data.

The available data for analysis of preferences include information collected by the PA from the online platform (we assume that it is to be run and administered by the PA) and Census data regarding the population. Hence, the PA has access to basic personal data of platform users, their position in the online social network and the opinions revealed on the platform. The online user data can be analyzed along with the aggregated census data of the whole population.

We have implemented a multi-agent simulation model that takes into account the distribution of personal attributes, social network data, the influence of social links and opinion diffusion dynamics. We analyze how different algorithms can allow the PA to generalize preferences collected by the online platform to the citizenry as a whole.

2 SIMULATING OPINION DYNAMICS IN SOCIAL NETWORKS

2.1 INTRODUCTION

In order to define the main SIM functionalities in area of opinion dynamics modeling, discussions have been held with pilot participants of the ROUTE-TO-PA project as well as with other project participants.

The main functionalities can be grouped into 2 categories:

1. SPOD users' statistics. It is important to collect and present data about usage of the SPOD platform: the number of active users, the level of interest in different data sets, statistics about discussions and opinions on the platform. This will allow the PA to understand and analyze how the SPOD is used, the main topics of interest for citizens, the most important data as well as the most active and involved users/citizens.
2. SPOD results generalization. This functionality will support the PA with information on how representative the opinions expressed by the citizens (sample) observed on the SPOD platform are. It will also provide the PA with information on how the opinion distribution might appear across the whole population (at a certain confidence level). The main socio-demographic features that determine the opinions shared by citizens are to be identified and presented.

One of the important objectives in this respect is to achieve representativeness of societal opinions in the analysis of preferences revealed on the SPOD platform. This creates a need for optimal design of preference elicitation and aggregation systems with heterogeneity in citizens' geographical location and demographic structure. Public resources are allocated to initiatives by democratic representatives of the citizens', who often are aided by the judgments of field experts. Similar types of decisions, yet with a different degree of detail, are taken on respective levels of public decision making (local governments and central government). Different initiatives influence the economic and social well-being of citizens with a different scope and magnitude (e.g. effects can be global, local only, with local overlaps, with externalities, or with a specific mechanism of propagation through the system). The problem of preference elicitation, their aggregation and translation (representation) into operational budgets, is a well-known issue in the field of economic theory – e.g. see Gajdos et al. (2008) and for a literature review see Fischhoff & Manski (2000).

A feature of a model economy, which draws a distinct line between the results derived from theoretical research, is the degree of heterogeneity of agents in that economy – e.g. see Kirman (1992). The economics of heterogeneous agents is a relatively new branch of economics, and it is true both for state of the art mainstream economic paradigms - neoclassical economics, as well as more interaction-oriented approaches - agent-based economics.

As an example application, consider the case where financial resources assigned by the public administration to possible initiatives represent citizens' preferences as closely as possible, especially taking into account the fact that discriminative outcomes/equilibria, in which underrepresented initiatives are never financed, are not allowed over a longer time horizon. The main question that we address is "How to choose socially optimal

regulatory actions that will properly take into account both heterogeneous agents and underrepresented social groups?” We approach this problem by modelling preference dynamics in heterogeneous communities.

Analysis of sharing information on SPOD necessitates the use of modelling tools that take into consideration the heterogeneity of economic agents, their geographic location, virtual and real-world social networks and information flow (including data comprehension) within those networks. The tool that allows modelling and finding the optimal design of such complex economic systems is agent-based simulation and modelling – e.g. see Farmer & Foley (2009) and Tesfatsion (2002).

The agent-based simulation model will be built within the MASON simulation framework (see Luke et al. 2005) and implemented in Java. The implemented model will be provided together with tools allowing for rigorous statistical analysis of simulation experiments. The statistical analysis and visualization of the simulation will be implemented in Gnu R and Python. The simulation model will provide the Open Data implementation guidelines for government decision makers.

2.2 REQUIREMENTS

We consider a scenario where public administration (PA) uses an online social platform to collect information regarding citizens' preferences. However, opinions of the subpopulation that uses the online platform may be unrepresentative.

2.2.1 OPINION REPRESENTATIVENESS ISSUES

Let us assume that a PA considers alternative investment decisions and should select the one preferred by the majority of the population. However, the number of citizens who reveal their opinions on an online platform is limited and may commonly be unrepresentative of the population as a whole. Bias is introduced when a group of non-representative citizens uses the online platform (selection bias), such as when the percentage of younger people using the platform is higher than their proportion in the general population. Classical statistical methods can only mitigate this selection bias. Let us, however, also consider the case where the discussion is significantly influenced by a few of the most active users (persuasiveness bias).

We assume that the opinion diffusion process takes place across the entire population. However, a PA can only observe a sample subpopulation. In this approach, unobserved population members influence the observed information diffusion. Moreover, we assume that opinion diffusion has the same dynamics on the subsample as on the entire population in this version of the model.

Using these assumptions, we develop a method to generalize the dynamics of preferences observed on the social platform to the entire population. The available data includes information collected by the PA from the online platform (we assume that it is run and administered by the PA) and census data regarding the population. Hence, the PA has access to basic personal data of platform users (e.g. gender, age), position in the online social network and opinions revealed on the platform.

The online user data can be analyzed along with the aggregated census data for the entire population.

We have implemented a multi-agent simulation model that takes into account the distribution of personal attributes, social network data and opinion diffusion dynamics. Such algorithms enable generalization of preferences collected by the online platform to the entire population and correction for not only representativeness but for persuasiveness biases as well.

The main advantage of using the SIM functionality for PA is twofold. Firstly, the PA can learn and thus better understand the unbiased opinion of the entire population based on the observed discussion on the SPOD platform of only a part (sometimes relatively small) of this population. Moreover, the SIM (the trinomial model implemented in SIM) enables a more detailed analysis of how particular attributes influence the opinions of citizens and how opinions spread throughout the population in general. Secondly, the PA can potentially perform simulation of the population opinion dynamics and thus better analyse and understand how to better inform the population about new initiatives, for example. These SIM features contribute to a better understanding of the entire population's opinions and thus they allow more optimal (from the entire population perspective) allocation of available resources.

2.2.2 POPULATION SIMULATION AND SPOD

We describe the basic steps for the PA to use SIM functionality in conjunction with the SPOD platform.

Phase 1 Building the network representation of the entire population

Synthetic population generation → Sample selection from SPOD → Network representation of the synthetic population

- 1) Synthetic population generation – depending on the available census data (the attributes of the citizens, the multi-way distributions available, the so-called seed-sample) the synthetic population can and should be generated. The synthetic population consists of all the citizens that belong (are of interest) to the particular PA that uses SPOD. The citizens are represented by a set of socio-demographic features (attributes), e.g. age, gender, number of children. etc.
- 2) Sample selection from SPOD – the subpopulation that uses SPOD is characterised by the socio-demographic attributes (in the ideal case the same as in the case of the synthetic population) and the links between them (citizen A is a friend of a citizen B). This sample should be used for statistical estimation in the consecutive step.
- 3) Network representation of the synthetic population – the missing links between synthetic population citizens are estimated in a statistically efficient way. In such a way, the synthetic population acquires a mathematical representation in the form of a network (graph). The graph consists of nodes (citizens and attributes) and links (edges) between citizens representing the fact that both citizens communicate and discuss relevant issues.

Phase 2 Simulating the opinion dynamics

Sample (discussion) selection from SPOD → Opinion dynamics simulation on the synthetic population (opinion generalisation) → Scenario analysis

- 4) Sample (discussion) selection from SPOD: collecting opinions from the social platform. The opinion can be expressed in many forms although the relevant form should enable comparability among citizens. One can use emoticons for this purpose.
- 5) The simulation of opinion dynamics upon the synthetic population (opinion generalisation) – the opinion dynamics is generated and the most probable opinion dynamics over the entire population is estimated and chosen. This scenario enables the analysis of the distribution of opinion in the entire population and the dynamics of opinion spreading within the population.
- 6) Scenario analysis – the simulation mechanism from the previous step can be used for the scenario analysis (a simulation is run with the selected and freely chosen parameter values)

2.3 MOTIVATION

Local governments are increasingly interested in improving communication with citizens and want to understand their preferences in order to put policies in place in an informed way. Therefore, they are implementing social

collaboration platforms, which allow members of local communities to discuss local governance issues. Such platforms allow for both C2C (citizen to citizen) and C2G (citizen to government) communication. Bertot et al. (2010) show that such platforms promote a culture of transparency, information openness and lead to a decrease in levels of corruption. It should be emphasized that such platforms provide a means of two-way communication, in which information flows not only from public administration to citizens, but also the other way around. Moreover, an information exchange between citizens themselves can be observed, i.e. citizens can discuss issues without explicitly directing their remarks to public administrators but among themselves. For the purpose of the present paper we will call such platforms 'social platforms', 'online platforms' or simply 'platforms', although we keep in mind the particular context in which they are used. Any kind of information retained from user activity on the platform will be referred to as online data.

A public administration usually has only limited information about its individual citizens, but has plenty of aggregated, census-type data, that can be disaggregated using synthetic population simulation, and used to gain additional insight about the preferences of the community on various issues. Such information is essential to the policy maker. It can be used by the public administration to support decision-making processes and to advise the PA in an informed way, i.e. based on the revealed preferences of the entire population, not only of the subjective opinions formed by the authority. Such a concept stands in line with the recently widely accepted paradigms of open governments and, in particular, of the open data trend. Such initiatives not only enable public administration to better understand the preferences of the population, which serves both ends – the community and the administration – but also acts to improve communication with citizens and allows community members to discuss local governance issues. Among other benefits, the issues discussed on the platform reveal priorities for the administration, e.g. by indicating matters that are most important for the citizens and therefore should be granted most concern on the part of administration and be included in PA strategies with the highest priority.

Public administrations, using the data collected from social platforms, would like to draw conclusions regarding the distribution of preferences on various matters and issues among the entire local community. However, the idea that all community members are keen social platform users, informed and advised using online data is generally misleading. Since users of social platforms do not have to be representative of the whole local population with respect to many characteristics, such as age, sex, income level, location, education etc., a risk emerges of being biased, perhaps significantly so, if only online data is used to inform and advise local authorities. This concerns both issues such as majority opinion as well as the distribution or heterogeneity of opinions within the community, which itself can be very informative for the public administration. Such bias decreases with increasing number of community members who participate in the platform, i.e. with the number of users, but one can expect that, especially in the early stages of platform implementations, biases could potentially be severe. Therefore, in order to generalize information obtained from online data, for example, from a survey or from voting (like/dislike/neutral), the public administration has to understand the qualitative and quantitative nature of heterogeneity of the respondents (users) with respect to such aspects as their geographic or demographic structure, both among platform users and among the rest of community. Such information is obtained from census data, when the entire population is concerned, and from user profile data, as far as platform users are concerned. Such data is used in classical approaches.

Classically, when one wants to infer community opinion on a certain issue, one conducts a survey, in which the sample of respondents is carefully selected, so that it is representative of the general community, i.e., its structure resembles the structure of the whole community along as many important dimensions as possible (such as location, age, sex, education, income etc.). In such situations, the survey sample is formed exogenously – experimenters construct it according to their wills and means. In the case of a social platform, to which users willingly subscribe, the situation is very different. One could also say that the sample population (population of

platform users) is formed endogenously – no one assumes that its structure will be of a certain form or character, it emerges with respect to citizens' propensity to participate. This propensity is not uniform over the entire population, but it is reasonable to assume that it correlates with census data along at least some dimensions, like age or income, or psychological characteristics, like extraversion or openness. Regardless of the underlying reasons, the opinions formed by users of the online platform cannot simply be extrapolated to the whole community.

Apart from census-type characteristics, however, data retained on the social platform logs provides the public administration with a new dimension of information, which is contained in the links or connections that users engage in when using the platform. This information is a rich resource and these links/connections are revealed when citizens discuss on-line certain issues/posts. It is worth mentioning that this is not necessarily a census-type link, like family membership, but can be formed between people that do not even know one another. On-line discussions can be direct, as when two citizens interact with each other in a direct discussion on a certain issue/post that was hosted on a forum, but they also can be indirect, as when two or more citizens discuss the same topic not directly with each other but with some other community members who discuss a given post, or even in an open way, i.e. posting their opinions publically, so that it becomes available for all other platform users involved in a certain conversation. For the purpose of this paper, in terms of a connection or a link between citizens established via an online platform, we are referring to a situation in which two citizens are involved in discussion of a certain post/issue, regardless of the fact as to whether they are conducting a direct discussion with each other – it will suffice that they just are involved in the discussion of the same post.

2.3.1 ELICITATION OF PREFERENCES

The elicitation of preferences problem is the classical problem considered in statistical and economical literature. We are interested in learning the preferences of the whole population; however, only the preferences of a small subsample are known. In an ideal case (when we are able to design the survey and draw the citizens randomly or the citizens are drawn randomly as a consequence of the selection process) we can directly generalize for the population (e.g. the sample mean is an unbiased estimator of the population mean) and calculate the estimation error. When the sample is biased no direct reasoning is generally available and the sample results must be rescaled using statistical methods (the corrections are applied) to deduce about the sample. Such techniques are applied to election surveys and polling data.

In the context of social networks the situation is complicated as the bias is not only due to differences in attribute values distribution between a sample and the population but also due to social processes (opinion changes as the consequence of social interactions among citizens) that may have a different form for the sample and population. This may lead to even more bias than that due merely to differences in attribute values distribution in the worst-case scenario. In such situations, traditional statistical measures for bias correction would underperform the method proposed.

2.3.2 SYNTHETIC POPULATION MODELLING

Synthetic population generation means creating the dataset that contains the micro data comprising all the citizens (for the public administration level considered e.g. the whole municipality) with all the relevant various attributes. These attributes are normally grouped into categories.

Due to privacy and other reasons such data is typically unavailable. The typical situation is that an anonymised¹ sample of citizens comprising a limited number of citizens is available together with marginal univariate distributions (histograms) with selected cross-tabulated multivariate distributions. Such data is available from different municipal, administrative levels, (sub) regional and national level. However, the data content (marginal distributions available) may differ depending on the administrative level. It should also be mentioned that the level of data available, the attributes presented and their aggregation level depend very much on the national level, see. e.g. Huang and Williamson (2001).

Depending on the availability of the sample of individual citizens we may distinguish synthetic population generations with and without samples, see. e.g. Lenormand and Deffuant (2013). Another characteristic of the data available is whether individual citizen data is available (a single citizen with typical attributes such as: age category, gender, place of residence, income category, employment status or marital status) or household data or both (household data typically comprises information on role (head of household), number of children, relation between heads of the household and spouse attributes, e.g. age difference).

Two main synthetic population reconstruction methods are considered in the literature: the synthetic reconstruction approach using the Monte-Carlo method and the combinatorial approach. These methods are compared by Huang and Williamson (2001).

Synthetic reconstruction with Monte-Carlo is done in two steps: generating the multivariate distribution of all relevant attributes and sampling citizens from this distribution. The iterative proportional fitting (IPF) technique is mainly applied for the generation of multivariate distribution (such distribution is normally not available as only two-or three way tables are available in a standard case). Using the sample data, the cross-table (comprising the number of citizens in the sample according to two or more attributes) is presented in a multi-dimensional matrix form. Such a matrix presents information on the correlation structure, although due to stochastic reasons the marginal distributions may differ. Using the IPF technique, matrix data is transformed in such a way that the marginal distributions will fit the known marginal distributions of the region and the correlation structure will be preserved to the extent possible.

The relevant formulas for one step in the two-dimensional case updating are presented below:

$$x^{k+1}(i, j) = \frac{x^k(i, j)}{x^k(i, .)} \times \tilde{x}(i) \quad (2.1)$$

$$x^{k+1}(i, j) = \frac{x^k(i, j)}{x^k(., j)} \times \tilde{y}(j) \quad (2.2)$$

Where $x^{k+1}(i, j)$ represents the number of citizens in the (i,j) cell (the citizens with the first attribute belonging to the i-th class and the second attribute to j-th class) in the k+1 step.

$x^k(i, .)$ and $x^k(., j)$ represents appropriate marginal distributions calculated based on the two-dimensional matrix values.

$\tilde{x}(i)$ and $\tilde{y}(j)$ are known marginal distributions of attributes one and two respectively.

The formulas can easily be generalized to the multi-dimensional case. The updating process ends when the matrix values changes cease to exceed the given threshold value. To generate the multivariate distribution of attributes considered, the hierarchical process is applied. The general idea behind this process is presented below, see Frick and Axhausen (2004). We start with univariate marginal distributions and step-by-step generate two-dimensional, then three-dimensional and so forth distributions, which are summarised below:

0 step:

¹ No data that would enable identification is being presented e.g. there is no detailed residence information, other information may be blurred by adding the random numbers.

(1,0,0), (0,1,0), (0,0,1)

1 step:

(1,0,0) and (0,1,0) \rightarrow (1,1,0)

(1,0,0) and (0,0,1) \rightarrow (1,0,1)

(0,1,0) and (0,0,1) \rightarrow (0,1,1)

2 step:

(1,1,0) and (1,0,1) and (0,1,1) \rightarrow (1,1,1)

In practice, some two- and three-way tables are already available for the subpopulation and there is no need to generate them. Sometimes data on the upper administrative level may be applied, as the starting point of the IPF process. Having generated the multivariate distribution, individual citizens are generated using the Monte-Carlo technique.

The combinatorial approach follows another approach. The available data sample is weighted in such a way that its composition fits the observed marginal distributions. The typical measures of fit are: total absolute error, standardised absolute error, phi and psi statistics and z-score.

Having both individual citizen and household data presents additional challenges. Barthelemy and Toint (2012) propose a method for generating both individual citizens and households at the same time. Their procedure consists of three consecutive steps: generating individual citizens, generating households' distributions and generating households (based on the generated individual citizens).

When no sample is available, one could use more advanced statistical techniques such as the maximum entropy generator, as proposed by Barthelemy and Toint (2012).

Guo and Bhat (2007) present and compare the software dedicated to the problem of synthetic population generation.

2.3.3 STATISCAL PROPERTIES OF SOCIAL NETWORKS

As only a sample of citizens will be active on the SPOD platform in the course of the self-selection process, this selection process should be modelled. As we represent the social structure of citizens by the graph $G=(V,E)$, where V represents nodes (citizens) and E edges (social links among citizens), graph sampling methods are suitable for this purpose.

Many methods of graph sampling are described in the literature. Depending on the object sampled we can distinguish between node and edge sampling. In the literature we consider the following methods of graph sampling, see Frank (1974):

- Random sampling without replacement
- Random sampling with replacement
- Bernoulli sampling
- Random walk
- Snowball sampling
- Homogenous Sampling

Random sampling without replacement allows drawing nodes uniformly at random without replacement, whereas random sampling with replacement method allows for nodes replacement (individual nodes can be chosen more than one time). Both methods have a simple statistical design (no information on the node connecting edges is necessary) and the sample size is defined beforehand.

In the Bernoulli sampling method, each node can be selected with the given (heterogeneous or homogeneous) probability. This method is characterised by simple statistical design (no information on edges is necessary). However, the sample size is undefined beforehand and may differ depending on the result of a single drawing.

Walk Sampling, see also Lovasz (1993).

The random walk is the iterative sampling procedure whereby, starting from the initial nodes, consecutive nodes are selected among nodes linked (by edges) to the last node selected.

Snowball sampling is the iterative sampling procedure that, starting with the initial sample of nodes, extends the sample by so-called waves (nodes sampled for the nodes that had not already been sampled that are adjacent to the nodes sampled in the previous wave).

Walk sampling allows us to sample without knowing the entire network. However, the statistical properties of the sample are more complex than in the case of homogeneous sampling procedure. Analysis of the sample statistics and generalization requires usage of Markov chain methods and bias correction induced by the unequal number of edges degrees for each node. Extensions of these methods can be found in the literature e.g. Ribeiro and Towsley (2010) propose multi-dimensional random walk sampling methods.

Depending on the sampling methods, sample statistics such as nodes distributions, dyads and triads distributions may differ, see Frank (1981) so that different methods must be used for statistical inference. The application of logit models and logistic regressions for the estimation of networks characteristics is proposed by Wasserman and Pattison (1996).

In case of the sparse data, the Bayesian approach could also be applied, see e.g. Butts (2003) or Farine and Strandburg-Peshkin (2015).

2.3.4 OPINION DYNAMICS

Generally, the main groups of learning models considered in the literature are: Bayesian updating and non-Bayesian updating, see Acemoglu and Ozdaglar (2011).

The Bayesian updating model is seen as a model of learning and rational opinions and beliefs updating process. However, the requirements according to the agent knowledge of priors (the prior beliefs distribution over all possible alternatives) and the high requirements for computational processing (classical Bayesian probability updating formula) required of citizens, make the practical application of this kind of updating in real life virtually impossible.

Therefore, simpler methods have been proposed for the purpose of opinion dynamics modelling, see deGroot (1977). This class of models permits the updating of opinions with the weighted linear mean of own opinion and the opinions the agent has relations with (represented by edges in the graph representations). The limitation of such a model is that, at the limit, agreement is reached and no permanent disagreements are possible. Many extensions and variations of the model have been proposed time varying weights, belief depending weights, Krause (2000). These models allow for permanent disagreements under mild conditions, see Lorenz (2005). Such an effect can also be achieved by introducing heterogeneity among agents, implementing so called stubborn agents (agents that do not change their opinions due to the influence of others). We may also allow for different levels of influence/persuasiveness (Zhou et al., 2015) or (Diao et al. 2014). The reaching of consensus is studied by e.g. Shang (2014).

2.3.5 EXPONENTIAL RANDOM GRAPH MODELS

Thomson and Frank (2000) consider a sampling-design approach to network modelling. In this approach, it is assumed that the population is fixed (not random) and that sampling is the only source of randomness. This approach is straightforward, although it requires knowledge of the sampling mechanism and at least part of the whole network structure to allow for parameter estimation and so cannot be applied in every case. Alternatively, a model-design approach is considered. One assumes that there exists a general random model of the network structure (governed by its respective functional form and parameters that describe the probability distribution of potential networks) and that the observed entire network is just a realization of this random variable. Such an approach allows for an estimation process even if the sampling process is not known or not fully known. Exponential random graph models are an example of the model based design applied to network modelling.

Holland and Leinhardt (1981) first proposed use of the exponential family of distributions for modelling the graph structure. The presented model was applied to directed graphs (digraphs). Such graphs are characterized by directed edges connecting the nodes of the graphs. The model of Holland and Leinhardt (1981) allows modelling of the individual parameters (node characteristics, called “attractiveness” and “expansiveness”) that govern the probability of outgoing and incoming edges respectively. Moreover, the dependence between edges in each dyad (a dyad is a pair of directed edges connecting two particular nodes) is assumed, though explicit independence between different dyads is assumed in their model (dyads of the same node are dependent due to individual parameter in an implicit way). The original model was extended in several ways. For example, Wang and Wong (1987) model the graphs that may comprise different subgraphs based on external information (such as sex and age) with different graph characteristics as, for instance, edge density, in the form of so-called stochastic block models for graphs. Frank and Strauss (1986) consider the so-called Markov random graph models. This class of models assumes independence of the edges between nodes if no common nodes are present. Lazega and Duijn (1997) consider so called p_2 models (as an extension to the original p_1 model of Holland and Leinhardt) which also take the covariates (nodes characteristics) into account in the hierarchical estimation process. The more general form of exponential random graph models is that of the so-called p^* models, implemented by Frank (1991) and Wasserman and Pattison (1996), namely:

$$P_{\theta}(Y = y) = \exp(\theta u(y) - \varphi(\theta))$$

Where $P_{\theta}(Y = y)$ is the probability of observing a particular population (represented by the adjacency matrix y), θ denotes the vector of respective parameters, $u(y)$ – vector of sufficient network statistics and $\varphi(\theta)$ the normalizing constant such that the sum of probabilities $P_{\theta}(Y = y)$ equals 1.

The network statistics could be, for instance, the number of edges $u_1 = \sum_{i < j} y_{ij}$ or number of triangles $u_2 = \sum_{i < j < k} y_{ij} y_{ik} y_{jk}$. In general, based on the Hammersley-Clifford Theorem, see Besag (1974), the statistics are based on the number of cliques in the edge dependence graph (the edge dependence graph consists of edges of the original graph as nodes, links in this graphs represent mutual dependence of the edges). These models allow for mathematical representation of the complex social dependency structures observed in reality.

Different methods for estimation of the required parameters have been proposed in the literature. The original approach of Holland and Leinhardt (1981) was the iterative fitting procedure. Strauss and Ikeda (1990) proposed the use of a pseudolikelihood estimation for the model parameters estimation. In this approach, the conditional likelihood is considered (the likelihood of the edge conditional on the edges structure of the remaining network) and the model loglikelihood function equals the the sum of the conditional loglikelihood function of each potential edge. This approach simplified the parameters estimation process. However, it turned out that the method can only be applied if the dependency among different edges is limited. The parameters can also be estimated using simulation methods, as proposed by Dahmström and Dahmström (1993) and Corander et al. (1998) following the idea of Geyer and Thompson (1992). The simulation approach was further extended towards the MCMCMLE (Monte-Carlo Markov Chain Maximum Likelihood Estimation) method, see Snijders (2002). The general idea of this algorithm is to use Gibbs sampling, see Geman and Geman (1983) or the Metropolis-Hastings algorithm, see e.g. Hastings (1970) for sampling from the network and using the statistical parameters of the selected network statistics (sufficient statistics) for updating of parameters in consecutive steps. Details of the algorithm are available in Snijders (2002), who implements the Robbins-Monro (1951) algorithm in a network context. IT implementation of these algorithms is available in the R program, package ergm, see Handcock et al (2016) and Hunter et al. (2008).

Parameters selection for exponential random graph models

Homophily, the tendency of similar citizens to be joined with a higher probability than dissimilar citizens, has been cited in many empirical studies. For instance, Smith et al. (2014) find in their study of sex, race, religion, age and education homophily in the United States, covering the years 1985-2004, that age and race homophily is stable but also find some degree of increasing religion and educational homophily. Homophily was also observed by e.g. Marsden (1988) or Goodreau (2009) .

Geometrically weighted edgewise shared partner (GWESP) distribution is defined as $e^{\lambda} \sum_{k=1}^{n-2} (1 - (1 - e^{-\lambda})^k) \times EP_k$, see e.g. Snijders et al. (2006). This statistic enables modelling of the influence of mutual friends (people we know) on the probability of a relation (link) between two selected citizens. However, the marginal influence decreases with an increasing number of mutual friends.

Proposed model and discussion

Based on the literature discussion presented above we propose to use the following model for modelling the probability of the link between two citizens:

$$(1) \text{ logit}(p_{ij}) = \mu_i + \mu_j + \beta \times |x_i - x_j| + \gamma \times e^{\lambda} (1 - (1 - e^{-\lambda})^k)$$

Where k is the number of edgewise shared partners which is $k_{ij} = \sum_k y_{ik} y_{jk}$ with decay parameter λ as introduced by Snijders et al. (2006). Such a model was also used by e.g. Handcock and Gile (2010).

The model takes into consideration the intrinsic psychological factors (tendency to acquaintance with other people, so called “attractiveness” or “expansiveness”) of citizens i and j by parameters μ_i and μ_j , degree of homophily with parameters β as well as the influence of mutual friends on the potential links between citizens, whereas the marginal influence of the next mutual friend decreases (changing of common friends number from 0 to one has higher impact on the probability of existence of the common link between citizens i and j than increase e.g. from 20 to 21)

Such formulations lead to the following network statistics: degree distribution, sum of $|x_i - x_j|$ for all edges in the network and geometrically weighted edgewise shared partner distribution which is defined as $e^\lambda \sum_{k=1}^{n-2} (1 - (1 - e^{-\lambda})^k) \times EP_k$, where EP_k is the number of edges having exactly k shared partners.

The model can be estimated using the MCMCMLE method in the case of the constant decay parameter λ or using the approach of Hunter and Handcock (2006) for the estimation of the curved exponential family models if λ is to be estimated.

In the case of large networks, simplifications of the model may be necessary:

- $\text{logit}(p_{ij}) = \mu_i + \mu_j + \beta \times |x_i - x_j| + \gamma \times e^\lambda (1 - (1 - e^{-\lambda})^{\bar{k}})$, (2) where \bar{k} is the population average number of edgewise shared partners
- $\text{logit}(p_{ij}) = \alpha + \beta \times |x_i - x_j|$ (3)

In both cases, we can apply the MLE approach using the logistic regression approach with a different number of parameters for the simplified models (2) and (3)

Exponential random graph models in the case of the network sample

Let us assume that we can represent the population as an indirect graph. Such a graph can be represented by the adjacency matrix Y with n rows and n columns (where n is the number of citizens). Each element of the matrix Y - y_{ij} is a binary variable (which assumes the value 1 if there exists a relation between citizens i and j and 0 otherwise). In our case we assume that the graph is undirected, i.e. $y_{ij} = y_{ji}$.

Following Handcock and Gile (2010) we divide the adjacency matrix Y into an observable and unobservable part - Y_{obs} and Y_{mis} . Let us introduce a random binary matrix D that corresponds to the adjacency matrix Y and indicates whether its respective element is observable or not. By the above definition of Y , D is symmetric.

$$Y_{obs} = Y_{ij} \circ D_{ij}$$

$$Y_{mis} = Y_{ij} \circ (1 - D_{ij})$$

We also assume that the conditional probability $P(D = d|Y, \varphi)$ is known, where $Y = Y_{obs} + Y_{mis}$ represents the whole network (including its unobservable part) and a set φ of unknown sampling parameters

For each observable citizen i we observe the set of socio-demographic variables x_i . We also know the multivariate distribution of the variables x for the whole population.

Moreover, for each citizen we define the opinion function $o(\cdot, t) \rightarrow [-1, 1]$, $t \in (0, 1, 2, 3, \dots, m)$. For each discrete time and each observable citizen i such that $\exists j: D_{ij} = 1$ we can observe the opinion expressed by this citizen. We assume that the function value set is $[-1, 1]$ where the value -1 represents an extremely negative opinion and the value 1 an extremely positive opinion. Although we assume that citizens can progressively express the strength of opinion we also consider the scenario where the value set is discrete and the function can only assume 3 values $\{-1, 0, 1\}$ where -1 represents negative, 0 neutral and 1 positive opinion. Let us also define the vector $o(t) = [o(1, t), o(2, t), \dots, o(n, t)]$. We can also divide $o(t) = o_{obs}(t) + o_{mis}(t)$ into observable and unobservable parts. We also consider the opinion dynamics function that transforms $o(\cdot, t) \rightarrow o(\cdot, t + 1)$, $t \in (0, 1, 2, 3, \dots, m - 1)$ and depends on the vector $o(t)$, adjacency matrix Y and unknown parameter set β that characterizes the citizen i .

The problem is to find the vector $o(m)$ for both observable and unobservable citizens.

We solve the problem in 2 steps. First, we find the distribution $Y|Y_{obs}$ and than for each $Y|Y_{obs}$ we simulate different opinion dynamics and thus get $o^{sim}(t, \delta) = o_{obs}^{sim}(t, \delta) + o_{mis}^{sim}(t, \delta)$.

We then choose such a parameter set δ_{opt} that the mean distance between $d(o_{obs}^{sim}(m, \delta), o_{obs}(m))$ is minimal.

The solution to our problem is then the vector $o^{sol}(t, \delta) = o_{obs}(m) + o_{mis}^{sim}(t, \delta_{opt})$

Direct application of the algorithm would require knowledge of the entire network. In the case where only part of the network can be observed modifications will be necessary.

Following Handcock and Gile (2010) we write the likelihood function as:

$$L(\theta, \varphi | Y_{obs} = y_{obs}, D = d) \propto \sum_{v \in Y_{mis} | y_{obs}} P(D = d | Y = y_{obs} + v, \varphi) \times P_{\theta}(Y = y_{obs} + v)$$

Where the second factor of the product is the probability of the whole network (including observable and missing part), whereas the first factor of the product gives the probability of observing the particular sample conditioned on the whole network and sampling parameters.

Handcock and Gile (2010) define the sampling mechanism as amenable if $P(D = d | Y = y_{obs} + v, \varphi) = P(D = d | Y = y_{obs}, \varphi)$, the sample probability depends on the observable part of the network structure.

We assume snowball sampling as the sampling method that represents the way citizens are attracted to the social platform. This sampling is k-wave sampling, where for each wave W_i the wave W_{i+1} consists of citizens linked with those citizens in wave W_i independently sampled with probability γ , which has not been sampled yet $\notin UW_i$, $i \in 0, \dots, k$. Parameters γ, k form the parameters set ϕ . We can easily see that in step j the probability of being sampled for the citizen i , provided she/he had not been sampled before, is $1 - (1 - \gamma)^l$ where l is the number of links of citizen i with citizens in wave W_{j-1} .

As in general (except for $\gamma = 0, 1$) the probability of the sample depends on the missing part of the network this sampling design is not amenable.

For parameter estimation we propose extending the MCMCMLE method by two network statistics for the purpose of sampling parameter estimations. Namely, the total number of edges (for the parameter γ) and one of the distance based statistics - average distance, average eccentricity (which is the maximal distance from the node) or the radius, which is the minimal eccentricity (for parameter k). Also, the MCMC has two phases: 1 – sampling of the whole network (as specified in the model-based approach), 2 – sampling from the whole network (we always start with the same citizens of the wave W_0 and sample using the current iteration values of the sampling the parameters set ϕ).

To facilitate the process we can apply the fixed decay parameter λ for the single MCMCMLE run and do the grid search for λ .

It is known that that the maximum likelihood (ML) estimate for an exponential family is also the solution of the moment equation. This can be extended to the current design (the likelihood function includes a snowball sampling component).

3 ARCHITECTURAL ISSUES FOR SIMULATION DESIGN

3.1 AGENT BASED MODELLING AND SIMULATION

Socio-economic systems are classified as complex systems, which means that the system as a whole exhibits qualitatively different aggregate macro characteristics than behaviours that can be inferred from simple aggregation of micro level individual actions of individuals, households, enterprises or institutions which constitute its parts.

The emergent differences between macro- and micro-behaviour stems from the effects of mutual interactions between individuals. Therefore, in order to effectively model complex socio-economic systems, it is not enough to capture the behaviour of individual elements and aggregate them, but it is essential to understand and represent the overall dynamics of the system, see Axtell (2007) and Tesfatsion (2002).

This principle is a founding element of agent based modelling, which is a methodology that allows researchers to quantitatively explain complex social and economic phenomena. In this way, it is possible to explain emergent behaviour observable on a macro scale that is present due to micro scale interactions, e.g. network effects.

Agent based modelling overcomes a major shortcoming of standard economic modelling that assumes that it is enough to model a single homogenous type of agent for each class of agents, the so-called representative agent. In this approach, every individual, household, company etc. is identical and fully rational. By full rationality it is meant that it possesses full knowledge, makes optimal decisions and incurs zero costs in the decision-making process. This approach is clearly not valid empirically. In some situations, it was found to be sufficient and to provide satisfactory predictive power. However, when we wish to explain the effects of interactions between agents the fact that they are different and not entirely rational is crucial.

The key feature of the agent based model is that it contains multiple heterogeneous entities: individuals, households, families, companies etc. adapting their actions to a dynamically evolving environment. Usually agents form hierarchies, e.g. a group of individuals constitutes a household and connections, e.g. social networks. Those three elements, i.e. heterogeneity, adaptive behaviour and complex relationships between agents imply that, although in theory it is possible to write down a full mathematical specification of such a model, in practice it is not feasible and computer code is a widely applied and accepted method for detailed specification of such models. Additionally, it is not only the specification of the model that is complex. Also, when solving them it is usually not feasible to use standard mathematical tools used for proving of theorems but computer simulation is used as an alternative. Therefore, because of the complexity of the model, its specification is not explicit (mathematical model) but implicit (computer code) and the method of analysis is not deductive (theorem proving) but inductive (statistical analysis of computer simulation output), see Kamiński (2012).

The foundation of traditional economic modelling was originally built upon what is called now Cowles Commission approach. This approach, in short, consisted of the estimation of potentially large systems of quantitative relationships (i.e. equations) specified between aggregate economic variables such as output, investment, unemployment, inflation, money supply, etc. These are ad hoc relationships which, abstracted from individual choice, bottom-up aggregation of economic dynamics and often abstracted from the way in which economic agents form expectations. Rapid development in computing power has allowed for the creation and usage of agent-based simulation techniques in the 1990s. Agent based models are built according to the bottom-up method. It means that agent based models are designed on the micro level, where interactions and behaviours of the individual agents are specified, and then the macro dynamics is observed as an emergent outcome of the model's simulation, see e.g. Oeffner (2009).

The fact that the agent based model is represented and analysed using computer simulation introduces some practical constraints. The most important one of these is the size of the population of agents in the model. Modelling of populations consisting of millions of agents is usually not feasible (though it should be noted that such big models are met in practice, yet they require enormous amounts of computing power) and, as an alternative, synthetic populations of agents are constructed. They usually feature much lower numbers of agents (in the order of thousands). In such cases the characteristics of agents are chosen so as to accurately represent the target population. A typical approach is to collect aggregated data about the distribution of attributes of entities in real life (e.g. gender, age, income, location) along with their interdependences and then create a synthetic population in an agent based model that features similar distributions.

One important benefit of the synthetic population approach in agent based modelling is that it allows the researcher or practitioner to analyse counterfactual scenarios. This means that we are able not only to consider and model the behaviour of the actual population (as e.g., in econometric modelling). We can also consider what-if scenarios that assume alternative narrations or future events. A good example of such an application is a model of the new NASDAQ Stock Exchange (Darley and Outkin, 2007), which the model's designers used in predicting the consequences of new trading regulations on this market. The constructed agent based model was able to predict most of the emergent behaviour that later occurred in practice (like the evolution of predatory trading). Such conclusions were not possible using traditional methods that focused on analysis of historical data.

The agent based modelling approach is very flexible in the sense that using it, one can replicate a wide range of dynamic phenomena. This flexibility can sometimes, however, be challenging. Scientific theories are based on abstraction and agent based models that try to replicate or describe reality, which can be thought of as an overload with respect to scientific theory. According to general opinion, models should not be as complex as reality, Leijonhufvud (2006). But this argument seems to be valid, if simplicity is needed, i.e., when the purpose of modelling is generalization (for the sake of an example). Generalization, however, does not have to constitute the sole purpose of economic modelling. Apart from abstraction, one can be, and in fact often is, interested in predictions of how a given system behaves under given circumstances or what are the consequences of changes in the structure of the system. It does not mean that every other system, even a very similar one, must, or is expected to, reproduce the same behaviour as the system in question. From such a perspective, close correspondence between the model and the reality seems desirable, even if not essential.

In fact, as mentioned earlier, within an agent based framework, a one-to-one correspondence between the model and the real-world economy could in principle be possible. One can, however, argue whether such a correspondence makes any practical sense, since a large number of arbitrary choices would have to be made, regarding the decision-making processes of all the groups of agents within the model. These could, in principle, be assumed on the basis of micro studies, experimental investigations or ad hoc assumptions, but clearly, even despite this, it would be extremely difficult to preserve the stability of the model on an aggregated level. Moreover, once stability has been achieved, it seems to be very fragile with respect to changes in agents' decision making functions and the structure of the model.

On a high level of abstraction, summarizing the observations of Fagiolo et al. (2007) and Oeffner (2009), the following features of agent based models can be emphasised:

1. Bottom-up perspective. Macro-level dynamics appears as a result of behaviour and explicit interactions of individuals on the micro level Tesfatsion (2002), Pyka and Fagiolo (2005).
2. Heterogeneity. Agents are heterogeneous in their behaviour, competencies, (bounded) rationality, computational skills etc.
3. Evolving complex system approach modelled by a network of direct interactions. All agents live in a network which is a complex dynamically evolving system (Kirman, 1997), aggregated properties emerge after repeated interaction between agents take place, agents' decisions are based on

present and past experience, trading of goods and services are modelled explicitly and as a result, the general equilibrium does not hold.

4. Non-linearity. Interactions between agents are highly non-linear, agent based models can contain feedback loops between micro and macro levels (small scale interactions create macro level dynamics, which in turn influences activity on the micro level).
5. Direct interactions. Agents interact with each other directly, their decisions depend on past and present choices made by other agents (Fagiolo. 1998), subgroups of agents (local networks) can emerge and their structure can change endogenously over time, agents can decide with whom to interact according to expected payoffs (in a bounded-rational way).
6. Bounded rationality. Agents live in a world which is too complex for exact (hyper) rationality, only local or partial rationality can be imposed, agents behave as rational individuals with adaptive expectations.
7. Learning. In numerous agent based models learning algorithms are introduced Windrum and Moneta (2007), agents engage in an open-ended search within a dynamically changing environment, observed patterns constitute a relevant ingredient for learning and adaptation, initial conditions often place agents as units without knowledge about the environment in which they live.
8. Dynamics. Agent based models, due to adaptive expectations, are characterized by dynamics which are irreversible.
9. Endogenous and persistent novelty. Economic systems are non-stationary with constantly introduced novelty, which leads to emergence of new behaviour patterns, which in turn drives adaptation and learning, on top of which agents find it difficult to adapt and learn in such a turbulent and changing environment, e.g. firms introducing new products into the market in order to increase payoffs while the results of research and development cannot be known ex ante (Dosi et al., 2006).
10. Selection mechanisms on the market. Goods and services produced by companies are filtered and selected by consumers, selection criteria are complex and involve numerous dimensions (e.g. product features), additional turbulence can be created firms entering or dropping out of the market (Windrum, 2005).

In a more explicit, implementation-oriented manner, a minimalistic ABM consists of the following ingredients:

1. Agents. They are specified as objects of predefined types (e.g. households, firms, banks, and the government) and implemented within the simulated economic environment as autonomous and interactive entities. Agents are characterized by micro-parameters according to which they can differ (e.g. education type, age or productivity). Micro-parameters can be fixed or variable over simulation iterations. Each agent has a set of decision micro-variables attached, which are updated according to ex ante assumed decision rules (e.g. consumption, labour demand, wage offered).
2. Interaction structure. Agents interact with each other exchanging resources they have at disposal (e.g. trading consumption goods, hiring labour supply, borrowing money holdings) and information contained in their information sets (e.g. wages, prices, labour market status). Interaction structure defines who interacts with whom and how.
3. Time. Models are simulated in discrete time steps, e.g. days in Legnick (2013), weeks in Ashraf et al. (2011), months in Giovanni (2010), and quarters in Gaffeo (2008). Different kinds of decisions can be made in different timeframes.
4. Macro variables. Result as an explicit aggregation of micro variables. Some can be defined exogenously on the macro level (e.g. a rate of interest).

The agent based model is usually so complex that it is impossible to parameterize it exactly using empirical data. Usually we have to calibrate it and test its behaviour under different values of its parameters. Therefore, the workflow of working with the agent based model is multi step, as shown in Figure 1 below.



Figure 1 Steps of development and analysis of agent based model.

The above process can be iterative if the obtained results do not reflect the modelled phenomenon accurately. In the following sections, we describe the components used for implementation of the agent based model developed in this work package.

3.2 ALGORITHM FOR RECONSTRUCTION OF SYNTHETIC POPULATIONS

We assume that we can only observe a sample of the population (users of the SPOD platform). In particular, we observe the opinions expressed by these group of citizens (e.g., in forms of emoticons) over certain time intervals (users are assumed not only to express their opinions but also to change their opinions due to discussion with

the citizens they have links to, e.g. as friends) and the links to other citizens that represent relations between citizens such as friendship. Additionally, we assume that we can observe personal attributes (e.g. sex, age, income, social status, employment, family status) of the sample citizens. We also know the marginal distributions of the same attributes. We model this kind of situation as a network where nodes represent the citizens and the edges represent the existing links between them.

In the following section the method which describes the preferences of agents, who are not users of the online platform, are inferred using census data and data from the platform are explained in more detail. The method consists of three major steps:

- I. synthetic population generation
- II. opinion propagation dynamics simulation within the synthetic population
- III. opinion dynamics selection and results evaluation

In the first step of the procedure the synthetic population is generated. The synthetic population resembles the true population regarding the marginal distributions of the attributes considered. The distribution of primary opinions (the opinions of citizens expressed for the first time with no or little influence by others) and their correlation with socio-demographic data as well as the links between citizens and the correlation with socio-demographic data (in particular the degree of being homophile, where the similarity between citizens increases the probability of the link existence between them) are estimated based on the observed sample and then reconstructed for the synthetic population.

Wasserman and K. Faust (1994) point out that a social network has the following four distinguishing characteristics:

- (1) Actors and their actions are viewed as interdependent rather than independent, autonomous units*
- (2) Relational ties (linkages) between actors are channels for transfer or "flow" of resources (either material or nonmaterial)*
- (3) Network models focusing on individuals view the network structural environment as providing opportunities for or constraints on individual action*
- (4) Network models conceptualize structure (social, economic, political, and so forth) as lasting patterns of relations among actors."*

These are network characteristics that should be considered with particular care when designing a multi-agent model of a social network.

Below we outline a 6-step procedure which produces a synthetic population of agents and implements the opinion diffusion simulation, using the network graph representation. The citizens are represented by nodes and the relations among them by edges. We also use the term agent for a citizen. We use the following notation for the citizens. Namely V^P denotes all the citizens in the synthetic population, V^S denotes only these citizens of V^P whose preferences and links are observable (SPOD users) and $V^{NS} = V^P \setminus V^S$ all those citizens that are not observable. Similar superscripts are used for the edges.

1. For a given network structure $G^S = (V^S, E^S)$, using data available for agents in V^S , i.e., $d(v)$ for $v \in V$, estimate a model M_E which predicts a probability that two given agents $v, u \in V^S$ are connected by an edge in G , i.e. a probability, that $(v, u) \in E^S$. This probability will be denoted by $p_{v,u}$.
2. Using model M_E , for all agents $v \in V^{NS} = V^P \setminus V^S$ reconstruct edges of the form (v, u) , such that $u \in V^{NS}$ and $u \neq v$.
3. For a given discussion/post $p \in P$, primary opinions $o(v, 0)$ and data $d(v)$ for all agents $v \in V^S$, estimate a model M_o , which uses $d(v)$ to predict $o(v, 0)$.

4. Using model M_o , for all agents $v \in V^{NS} = V^P \setminus V^S$ reconstruct their primary opinions $o(v, p)$.
5. Using an opinion diffusion algorithm A_o , simulate how opinions of agents $v \in V^S$ change, when interaction of all agents is taken into account, i.e. when opinions of agents $v \in V^S$ are influenced by opinions of agents $u \in V^{NS}$, for which an edge (v, u) is predicted by a model M_E . Different diffusion algorithms are allowed. They differ in how agents value her/his own opinion comparing to the opinion of the citizens she/he has a relation with and how the agents adopt the opinions of these citizens.
6. Compare the generated opinions dynamics of the agents $v \in V^S$ with the dynamics observed on the platform and choose the opinions dynamics process that generates the minimal difference. For all agents $v \in V^{NS} = V^P \setminus V^S$ the final opinions $o(v, n)$ generated by these opinions dynamics are considered.

Details of the procedure are presented in Appendix B of this document.

3.3 META-MODELLING

One of the significant challenges of agent based modelling is the difficulty, often impossibility, of derivation of the analytical characteristics of their properties using deductive methods. In such cases, statistical induction methods are required. In order to precisely describe the above concept, we use the following formalism.

Assume that there is a given mathematical model M and its property W that is of interest for the researcher or practitioner. In general, there exist the following scenarios concerning their relationship:

- S1) it is possible to verify property W using simulation and analytically;
- S2) it is possible to verify property W only analytically;
- S3) it is possible to verify property W only using simulation;
- S4) it is not possible to verify property W using simulation or analytically;

It is worth noting that the impossibility of analytical verification of the property can be objective (the analytical method of verification is not known) or operational (in theory the problem could be solved but the cost involved is so significant that in practice it is not achievable).

In the case of agent-based models their specification is usually so complex that it is impossible to solve them analytically and one must apply simulation methods to analyse them. Therefore, we are left with S3 and S4 scenarios only.

Let us emphasize the difference between those two scenarios. Consider an agent based model of open data governance (such as is prepared in this work package), that is model M in the above notation. Assume that we analyse the social network between the agents in this model and we are interested in the clustering coefficient of the network of connections (this parameter is actually measured in our analysis). We might be interested in two properties – $W1$: the clustering coefficient is equal to exactly 0.5 and $W2$: the clustering coefficient is greater or equal than 0.5.

It is impossible to verify property $W1$ using simulation, even if it were in fact true. On the other hand, property $W2$ can be verified. For an extended discussion of this distinction, see Kaminski (2015).

In practice, the analyst is simply restricted to consideration of properties that can be verified using simulation. The basic workflow of the simulation experiment with the agent based model is given below:

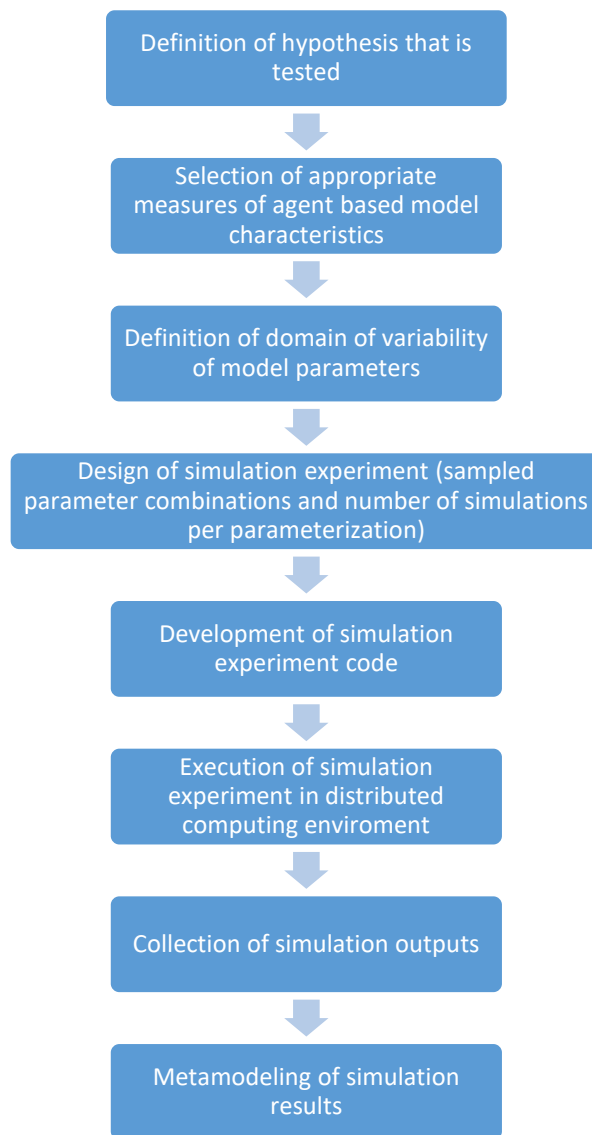


Figure 2 Process of analysis of agent based model properties.

In the above process the two critical elements are design of the simulation experiment and meta-modelling, and these will be explained in subsequent paragraphs.

When it comes to design of computer experiments, the starting point is identification of the model parameters and their domains. Assume that the considered model has parameters P_1, P_2, \dots, P_i , with their domains equal to D_1, D_2, \dots, D_i . That means that, for example, the value of parameter P_1 must be in the set D_1 . Therefore, the entire space of the parameters is a Cartesian product $D_1 \times D_2 \times \dots \times D_i$.

Depending on the number of parameters in the product, $D_1 \times D_2 \times \dots \times D_i$, four major methods of sampling points for the experiment design are used in practice:

- 1) Full Cartesian product;
- 2) Random sampling;
- 3) Latin hypercube design;
- 4) Low discrepancy sequences.

In order to avoid technical details below we illustrate all four techniques with the assumption that we have only two parameters P_1 and P_2 .

Full Cartesian product (see figure below) assumes that we select a subset of values of each domain and simulate all combinations of all parameters.

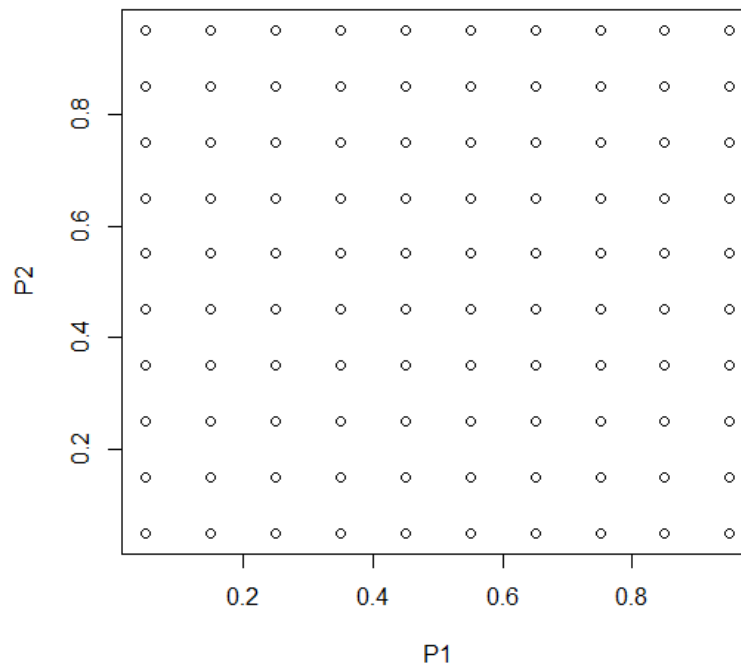


Figure 3 Full Cartesian product.

The advantage of the full Cartesian design is that it evenly fills the parameter space. Unfortunately, as the number of parameters grows the number of simulation parameterization in the full Cartesian design grows exponentially. For example, with 10 parameters and each measured at 10 values we need to sample 10^{10} points. Another disadvantage is that in each dimension we get only 10 distinct values and are unable to precisely capture the behaviour of the model between them. Finally, one needs to know how many simulations one wants to run prior to the simulation experiment.

The simplest approach that endeavours to overcome these shortcomings is random design, depicted in Figure 4 below. In this approach we sequentially randomly pick a point from the design space.

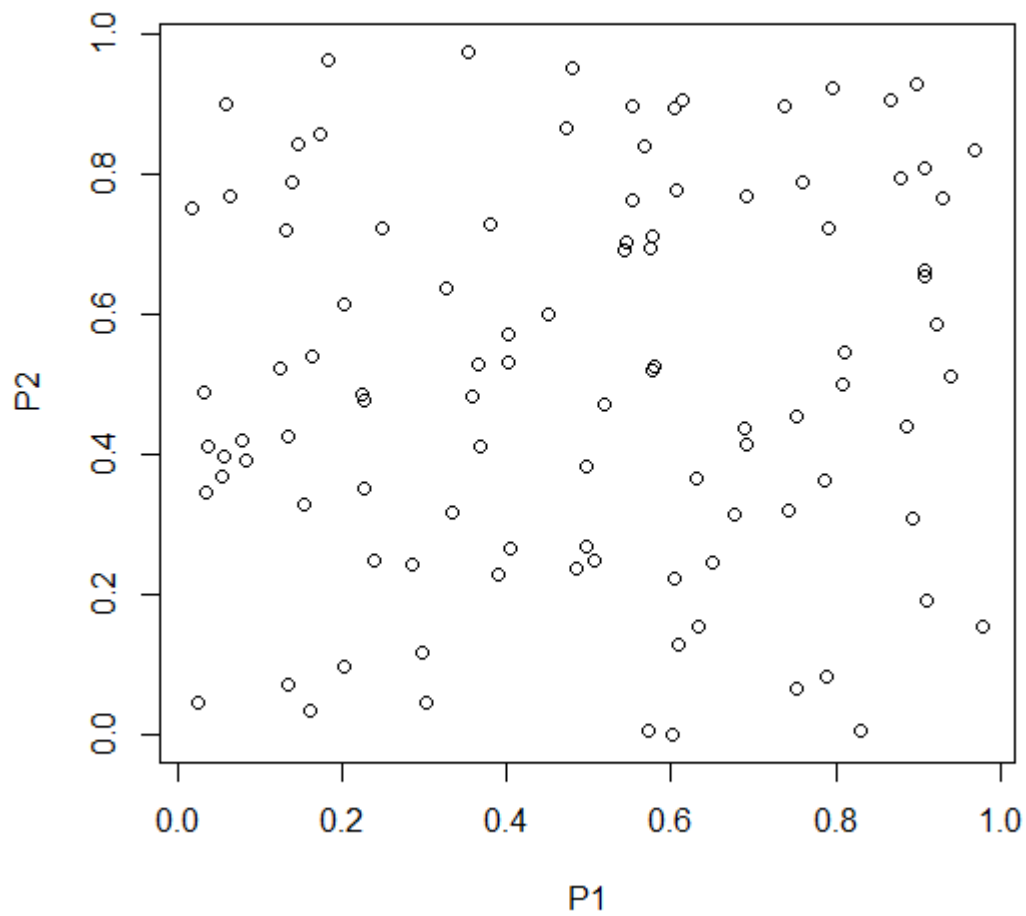


Figure 4 Random design.

The advantages of this approach are that one does not have to plan upfront how many points it wants to sample and that in each dimension all values of parameters are equally likely. However, its shortcoming is that it does not guarantee an even distribution of points in the design space, which can be seen in Figure 4 above, where in some areas there are clusters of points whereas other areas are blank.

A popular method that lies somewhere between full Cartesian product and random design is the Latin hypercube design, see Figure 5 below.

In this approach each dimension is selected uniformly (exactly as in the Cartesian product approach) but instead of calculating their full product the dimensions are randomly matched.

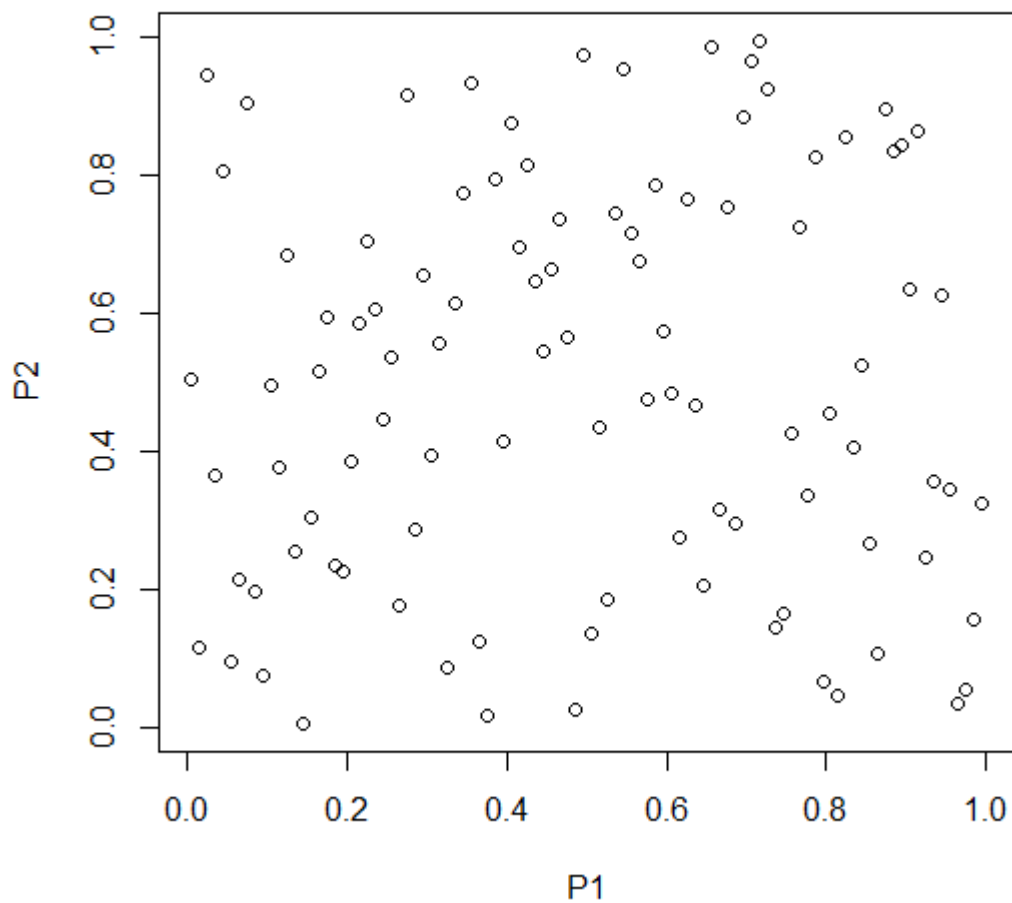


Figure 5 Latin hypercube design.

This approach guarantees (as opposed to random design) that marginal distributions of each parameter are exactly uniform while, at the same time, the number of sampled points is elastic even for highly-dimensional data.

The drawback of the Latin hypercube design is that one has to select the number of sampled points before the experiment (this is because – as mentioned earlier – marginal distributions of parameters are not random but predefined, as in the full Cartesian product).

In cases where one does not know the number of simulations that would be run a good method of choice is “low discrepancy sequences”, also called quasi-random sequences, see Figure 6 below.

In this approach, intuitively, a new point in the design space is iteratively placed in the biggest “hole” in the parameter space.

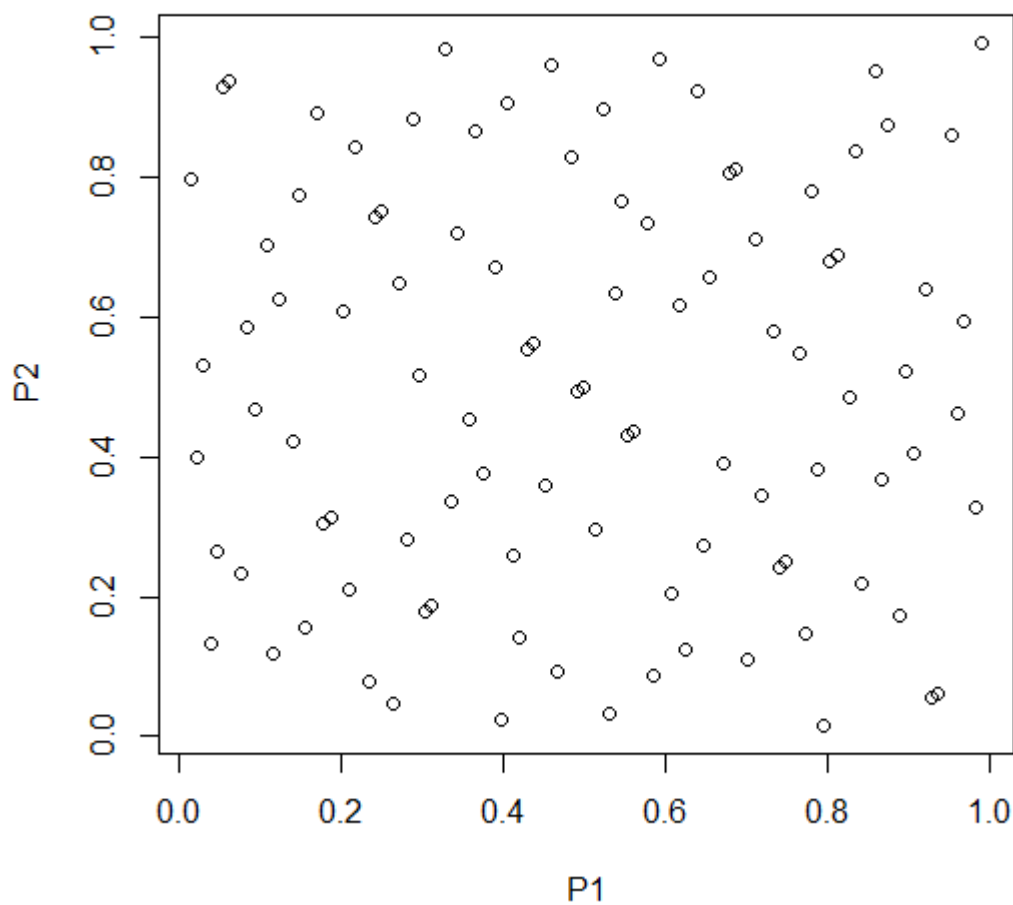


Figure 6 Low discrepancy sequence.

The most popular method for generating low discrepancy sequences are the so-called Sobol sequences, see Bratley and Fox (1988). They can be recommended when the number of simulations run is not known when the simulation experiment is started. In practice, such a situation is quite common because usually only the computational budget is known (e.g. how many hours are given for running of the simulation) and the duration of a single run of simulation is not fixed but is changing randomly from one run to another.

In the Alpha stage of the SIM deliverable we have used a reduced dimensionality of design space and thus the full Cartesian product for experiment design was suitable. However, in the beta stage we plan to extend the simulation results analysis and then more advanced methods for analysis of simulation results will be applied.

The second key step in analysis of the simulation model is so-called meta-modelling. As discussed by Kamiński (2015), complex stochastic simulations are often viewed as black boxes in the way they transform their input parameters into output characteristics of modelled systems. Therefore, it is often proposed to use their simpler approximations, referred to as simulation meta-models, cf. Barton (1992). The relevant literature identifies various reasons why meta-models can be useful for a researcher (Kleijnen, 2000; Santos, 2007). They can be summarized in three major groups: (i) understanding the shape of the relationship between the inputs and outputs of a model, (ii) prediction and (iii) optimization. These different usage scenarios imply different

approaches to the selection of the functional specification of a meta-model, simulation experiment design and parameter estimation. A review and comparison of meta-model types applied in practice is presented in Wang (2006). In the analysis of the agent based model prepared in the current work package, the meta-modelling has two major objectives: understanding and prediction. Hence, the meta-models are expected to have two features: simple interpretation of their structure and good predictive power.

Formally, if we denote by $S(x)$ a simulation model that given parameters x produces a random value $S(x)$ the objective of meta-modelling in the context of the work package is to find a deterministic function $f(x)$ such that it approximates the expected value of simulation $E(S(x))$ as closely as possible. If we assume that the domain of the parameter space is D (i.e., $x \in D$) then we wish to find the function $f(x)$ that minimizes the following objective function:

$$Q(f, S, D, d) = \int_{x \in D} d(f(x), E(S(x))) dx,$$

where $d(\cdot, \cdot)$ is a measure of distance between $f(x)$ and $E(S(x))$. The most common distance measures d are the absolute value $d_a(x, y) = |x - y|$ and the squared distance $d_s(x, y) = d_a^2(x, y)$.

In practice, it is impossible to evaluate $Q(f, S, D, d)$ and minimize it. As a result, its approximation based on the sampled data is used. Assuming that we have run the simulation in points x_1, x_2, \dots, x_n respectively n_1, n_2, \dots, n_k times and collected observations $s_{i,j}$ the simplest approximation of $Q(f, S, D, d)$ is the following:

$$q(f, S, D, d) = \sum_{i=1}^n d\left(f(x_i) - \frac{\sum_{j=1}^{n_i} s_{i,j}}{n_i}\right).$$

This estimator is unbiased i.e., $E(q(f, S, D, d)) = Q(f, S, D, d)$.

The key challenge in this process is the choice of an appropriate space F of approximation functions f . As mentioned previously this space should have two desirable properties: good explanatory power and ease of implementation.

In the current work package, we have selected the following classes of models. They all have the desired properties noted above. Here they are ordered by increasing explanatory power and decreasing simplicity of interpretation:

- Linear regression;
- Generalized additive models;
- Random forest.

4 PILOT DATA AND SIMULATION EXPERIMENTS RESULT

The starting point for the simulations is the synthetic population generated using the local census data. The census data available is heterogeneous with respect to the attributes available, the aggregation of attributes into separate categories, the level of availability (individual citizen versus household) and the cross-tables (mostly two- or three-way tables showing the joint distribution for 2 or 3 different attributes at the same time). Within this phase of the project we have used the census data from the Prato municipality. Such data comprises marginal distributions for single citizens and is described below in more detail. However, the synthetic population for Dublin based on the respective census data (available in other form than Prato census data) is also generated.

4.1 PILOT'S DATA

4.1.1 PRATO

Data for Prato (an Italian city in Tuscany region) is available on an individual citizen level. We have received the census demographic data from the Prato municipality and income data based on tax information. In particular, the following tables were available and were the basis for the Prato synthetic population generation:

- Population table that contains the distribution of Prato population among 6 regions of residence (00, East, West, North, South and Central regions), 14 age categories (0-2,3-5,6-10,11-13,14-17,18-24,25-34,35-44,44-54,55-59,60-64,65-74,75-84,84-) and sex (woman, man)
- Marital status table containing contains the distribution of PRATO population among 6 regions of residence (00, East, West, North, South and Central regions), 4 marital status categories (single, married, divorced and widowed) and gender (woman, man)
- Employment table containing the distribution of PRATO population among different employment categories and gender
- Income table containing the distribution of PRATO population among 7 income categories (0-10000,10000-15000,15000-26000,26000-55000,55000-75000,75000-120000,>120000)

The goal was to generate a synthetic population of heads of households, each characterized by the following features: region of residence, age, sex, marital status, income category. As we did not have all the cross tables we made some assumptions about the correlation structure among attributes considered within the synthetic population generation process. Such assumptions are not crucial as the citizen data sample (of citizens that registered on the SPOD platform and provided the socio-demographic information) will be available after the SPOD platform is launched and thus the IPF procedure as described in Chapter 2 may be used for generating the correlation structure among all considered attributes.

Synthetic population generation (we have only considered heads of households) followed the following steps:

- 1) Selection of citizens with a minimum age of 18
- 2) Reading in, as a starting point, the population table with a marginal distribution according to: region of residence, age and gender
- 3) Assuming a theoretical age and marital status correlation structure (correlation structure among age and marital status is briefly characterised by a higher than proportional share of singles among younger citizens and a higher than proportional share of widows/widowers among older citizens) we obtain the correlation structure among these two attributes for Prato by fitting the theoretical

correlation structure to the real marginal distributions for age and marital status (applying Iterative Proportionate Fitting)

- 4) Heads of households were randomly (with equal probability for a woman or a man) chosen from the married citizens

The distributions received for Prato are presented in the following figures.

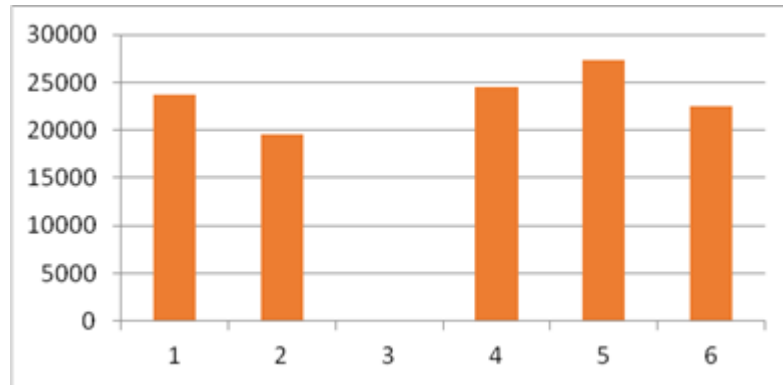


Figure 7 Prato synthetic population distribution according to district of residence.

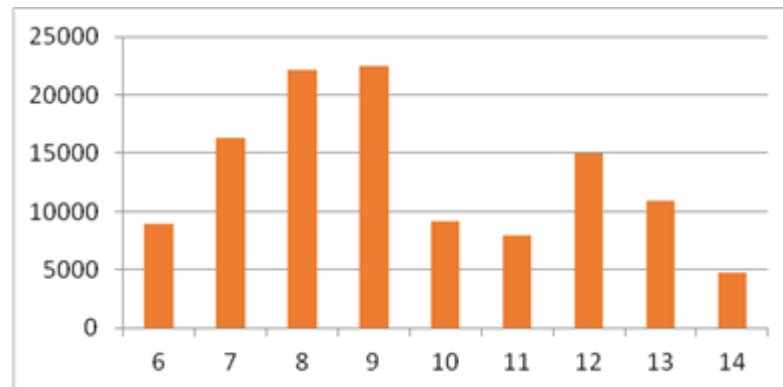


Figure 8 Prato synthetic population distribution according to age.

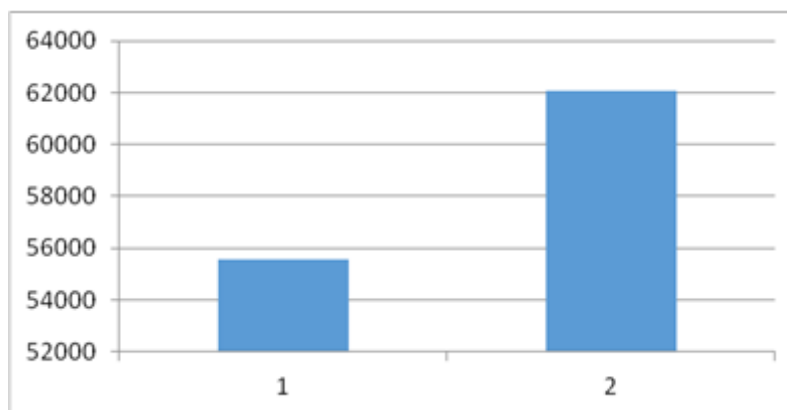


Figure 9 Prato synthetic population distribution according to gender.

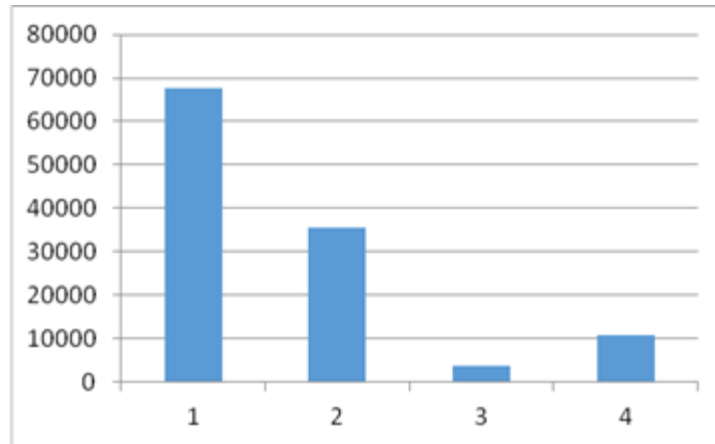


Figure 10 Prato synthetic population distribution according to marital status.

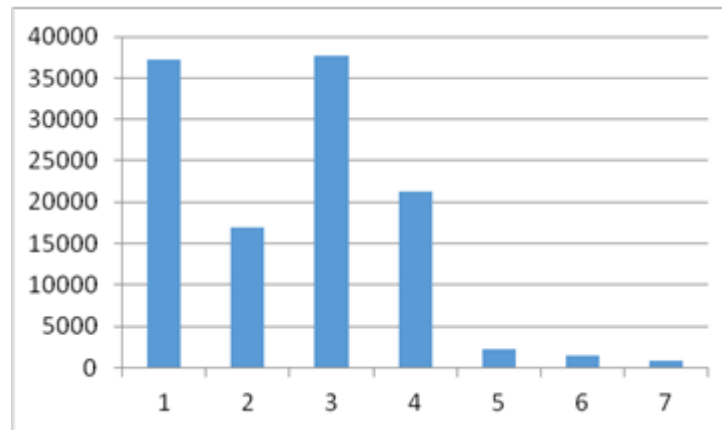


Figure 11 Prato synthetic population distribution according to income.

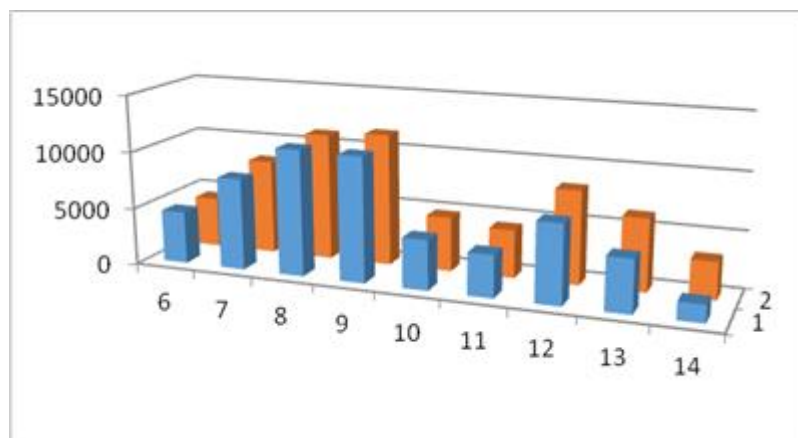


Figure 12 Prato synthetic population distribution according to age and gender.

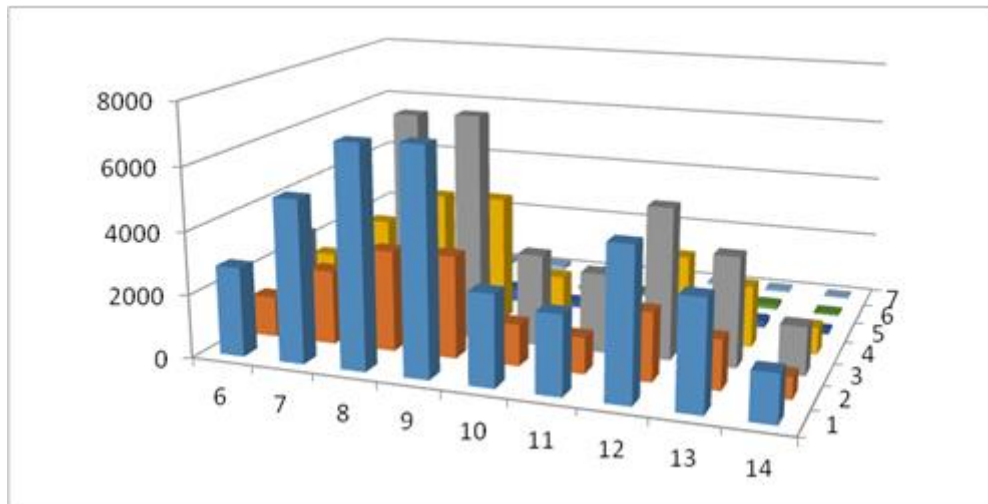


Figure 13 Prato synthetic population distribution according to age and income.

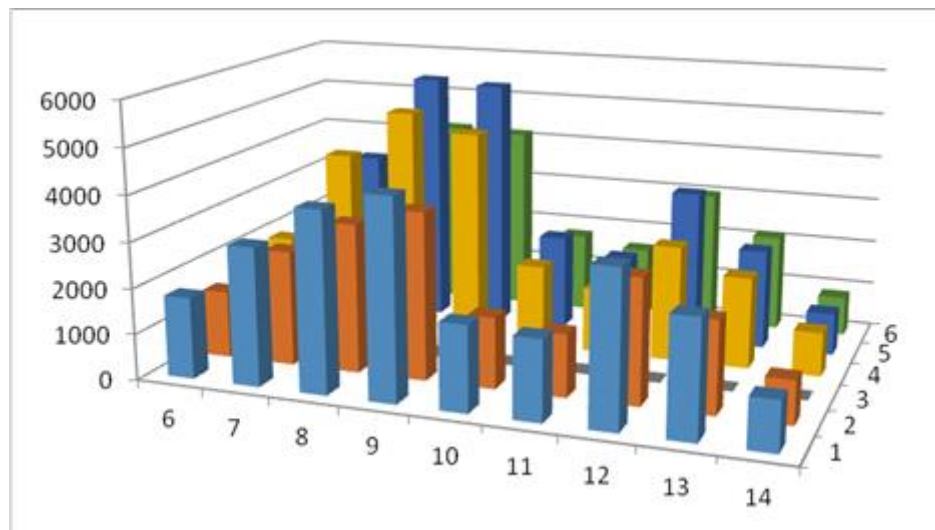


Figure 14 Prato synthetic population distribution according to age and district of residence.

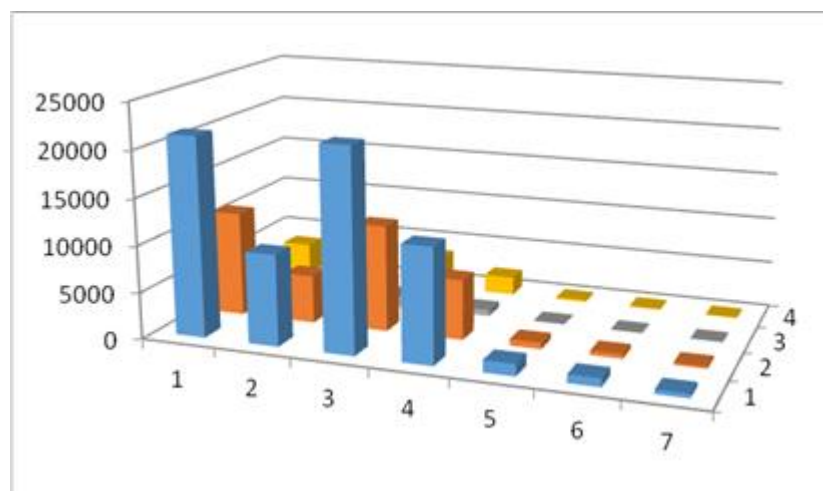


Figure 15 Prato synthetic population distribution according to income and marital status.

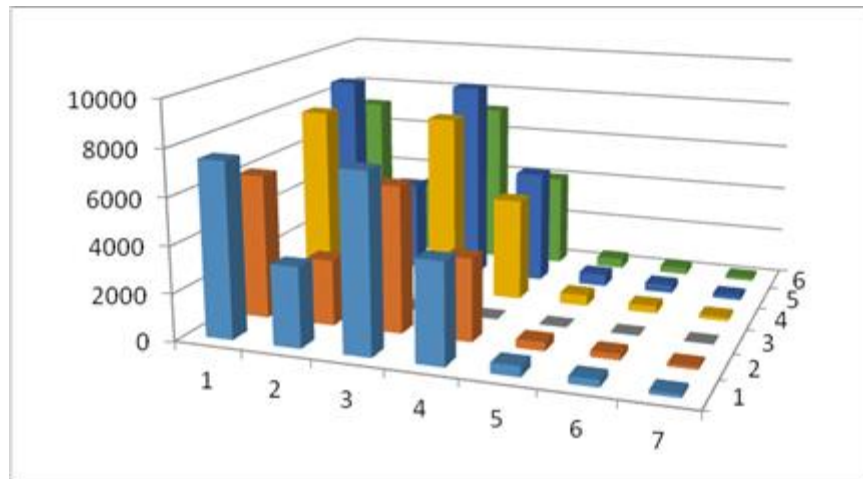


Figure 16 Prato synthetic population distribution according to income and district of residence.

4.1.2 DUBLIN

Data sent by the Dublin municipality, as well as the 2011 census data for Dublin (available at the web page <http://data.cso.ie/>), are the data sources used for the synthetic population generation. 2016 census data was available only as preliminary results at the time of report preparation. The synthetic data is generated at the regional level (Leinster) with the total number of citizens equal to 2 504 814 and a county level of: Dublin City (527 612 citizens), Dún Laoghaire-Rathdown (206 261 citizens), Fingal (273 991 citizens), and South Dublin (265 205 citizens).

Ultimately, the following attributes (selected based on socio-demographic importance and data availability) are considered:

- Gender
- Age category
- Marital status
- Basic economic status
- Number of children
- County (if Dublin city perspective considered)

The following categories are considered for each of the citizens attributes listed above. Gender (Male, Female), age category (0 – 4 years, 5 - 9, 10 - 14, 15 - 19, 20 - 24, 25 - 29, 30 - 34, 35 - 39, 40 - 44, 45 - 49, 50 - 54, 55 - 59, 60 - 64, 65 - 69, 70 - 74, 75 - 79, 80 - 84, 85+), marital status (Single, Married, Separated (including deserted), Divorced, Widowed), basic economic status (Employer or Self-employed, Employee, Assisting relative, Unemployed and looking for first regular job, Unemployed having lost or given up previous job, Student or pupil, Looking after home/family, Retired, Unable to work due to permanent sickness or disability, Other economic status), number of children (0,1,2,3,4 and more).

We use original category names in this paragraph. E.g. “All married” is the original name for the category that comprises subcategories as “Married”, “Divorced and Remarried” etc.

The synthetic population generation was done in the following steps:

Step 1

The table “Population (Number) by Province County or City, Sex, Age Group and Census Year” that consists of the distribution of citizens according to county, gender and age group was read in.

Step 2

The table “Population (Number) by Province County or City, Sex, Detailed Marital Status and Census Year” that consists of the distribution of citizens according to county, gender and marital status was read in.

Step 3

The table “Population (Number) by Aggregate Town or Rural Area, Detailed Marital Status, Sex, Age Group and Census Year” was read in. This table was treated as a seed in a multi-dimensional Iterative Proportional Fitting Procedure (mipfp), see. Barthelemy and Suesse (2016). The distributions obtained in the previous steps for each county were treated as the marginal distributions. As a result we obtained the distribution of citizens according to gender, age, and detailed marital status for each county separately.

Step 4

The table “Population Aged 15 Years and Over (Number) by Regional Authority, Sex, Age Group, Principal Economic Status and Census Year” was read in. The table represents the distribution of citizens according to gender, approximate age group and principal economic status for the Dublin authority. Further approximate categories were used for age, namely (15 - 19 years, 20 - 24, 25 - 34, 35 - 44, 45 - 54, 55 - 64, 65+) . The distribution was further used as the conditional distribution of the principal economic status attributes, conditioned on approximate age category and gender.

Step 5

The table “Private Households (Number) by Type of Family Unit, Age of Parent, Children in Family Unit and Census Year” was read in. The table gives the number of children (in form of the following categories: No children, One child aged 0-4 years, One child aged 5-14 years, One child aged 15 years and over, Two children where the youngest child is 0-4 years, Two children where the youngest child is 5-14 years, Two children where the youngest child is aged 15 years and over, Three children where youngest child is aged 0-4 years, Three children where the youngest child is aged 5-14 years, Three children where the youngest child is aged 15 years and over, Four or more children where the youngest child is aged 0-4 years, Four or more children where the youngest child is aged 5-14 years, Four or more children where the youngest child is aged 15 years and over) for the following parents groups: (Females in husband and wife / cohabiting couple type family unit, Males in husband and wife / cohabiting couple type family unit). We treated this distribution as the conditional distribution of the number of children conditioned on age category and gender for citizens with “Married” marital status. The conditional distribution of the number of children for all other marital status groups was adjusted in such a way that the number of children in each county is correct.

Step 6

Using the marginal distribution (from Step 1 – Step 3) and 2 conditional distributions (from Step4 and Step 5 respectively) we received the final distribution in the form of the frequencies for each citizen category. The number of citizens in each category was generated in a standard way. Namely the final frequency in each category was multiplied by the total number of citizens in each county and the number of citizens in each category was set to the integer part of the multiplication result , possibly increased by 1 with a probability equal to the fractional part of the multiplication.

Selected results for Dublin City Council are shown below in graphical form.

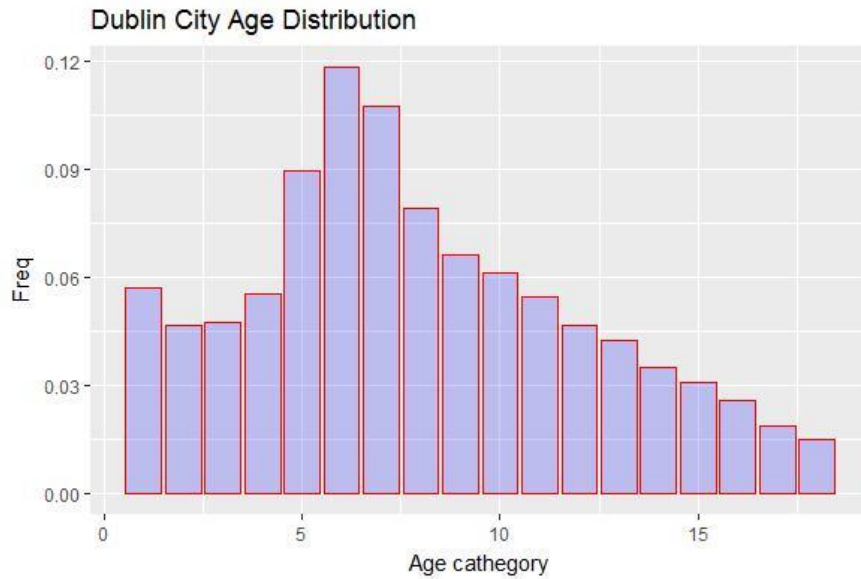


Figure 17 Dublin synthetic population distribution according to age category: 1 - 0 - 4 years; 2 - 5 - 9 years; 3 - 10 - 14 years; 4 - 15 - 19 years; 5 - 20 - 24 years; 6 - 25 - 29 years; 7 - 30 - 34 years; 8 - 35 - 39 years; 9 - 40 - 44 years; 10 - 45 - 49 years; 11 - 50 - 54 years; 12 - 55 - 59 years; 13 - 60 - 64 years; 14 - 65 - 69 years; 15 - 70 - 74 years; 16 - 75 - 79 years; 17 - 80 - 84 years; 18 - 85 years and over

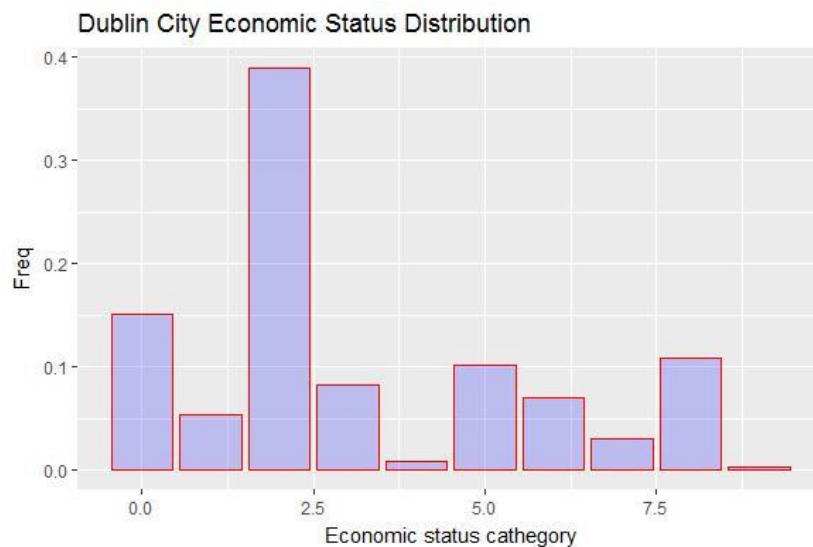


Figure 18 Dublin synthetic population distribution according to economic status category, 0 - Under age; 1 - Employer or Self-employed; 2 - Employee; 3 - Unemployed having lost or given up previous job; 4 - Unemployed and looking for first regular job; 5 - Student or pupil; 6 - Assisting relative, Looking after home/family; 7 - Unable to work due to permanent sickness or disability; 8 - Retired; 9 - Other economic status



Figure 19 Dublin synthetic population distribution according to gender category: 1 - Male; 2 - Female

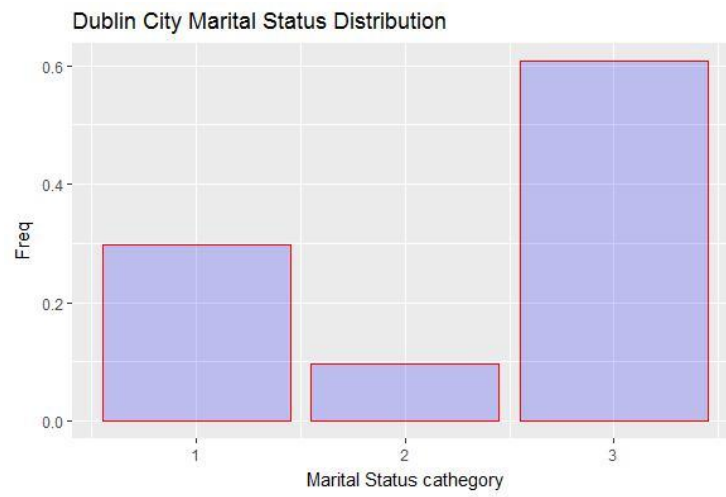


Figure 20 Dublin synthetic population distribution according to marital status category: 1 - Married; 2 - Separated, Divorced, Widowed; 3 - Single

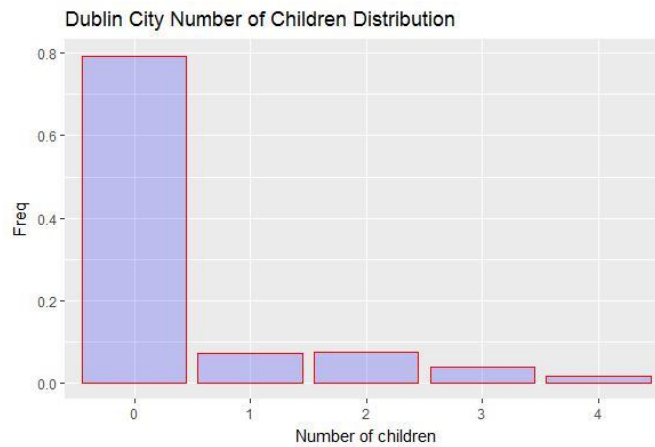


Figure 21 Dublin synthetic population distribution according to number of children

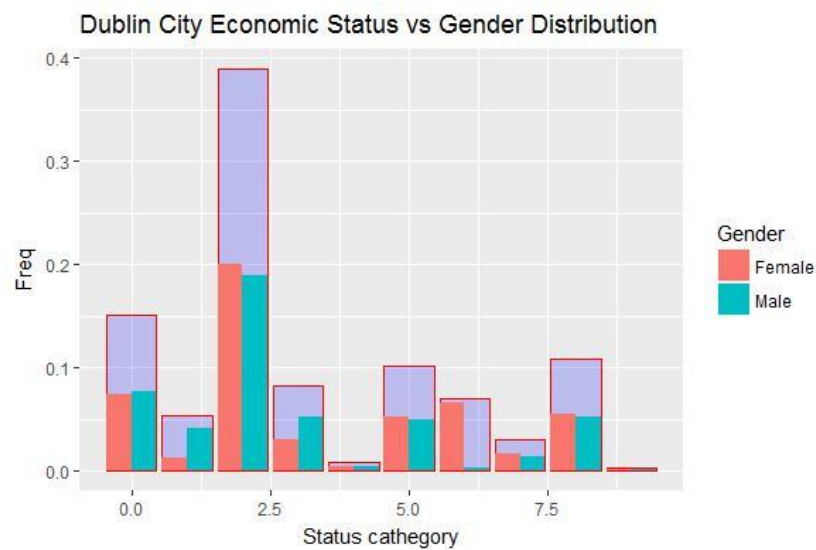


Figure 22 Dublin synthetic population distribution according to economic status: 0 - Under age; 1 - Employer or d; own account worker 2 - Employee; 3 - Unemployed having lost or given up previous job; 4 - Unemployed and looking for first regular job; 5 - Student or pupil; 6 - Assisting relative, Looking after home/family; 7 - Unable to work due to permanent sickness or disability; 8 - Retired; 9 - Other economic status and gender category

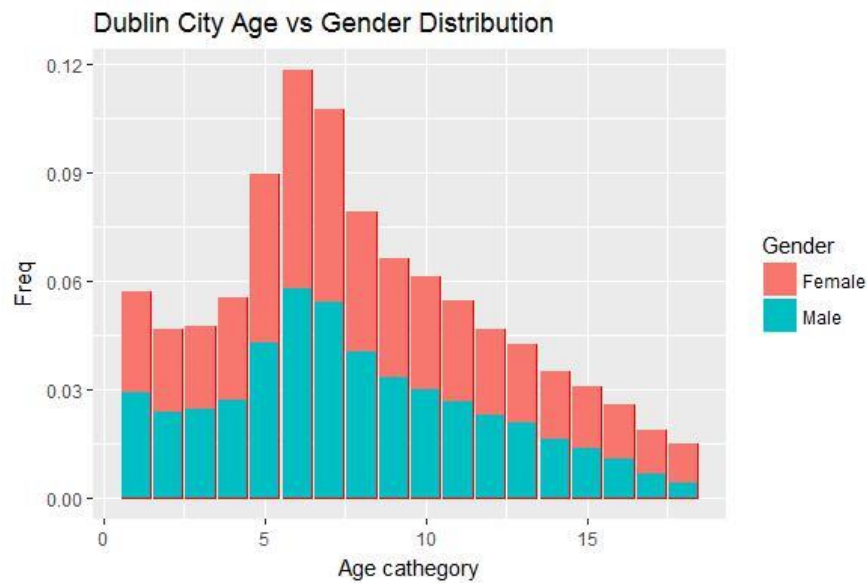


Figure 23 Dublin synthetic population distribution according to age category: 1 - 0 - 4 years; 2 - 5 - 9 years; 3 - 10 - 14 years; 4 - 15 - 19 years; 5 - 20 - 24 years; 6 - 25 - 29 years; 7 - 30 - 34 years; 8 - 35 - 39 years; 9 - 40 - 44 years; 10 - 45 - 49 years; 11 - 50 - 54 years; 12 - 55 - 59 years; 13 - 60 - 64 years; 14 - 65 - 69 years; 15 - 70 - 74 years; 16 - 75 - 79 years; 17 - 80 - 84 years; 18 - 85 years and over and gender category

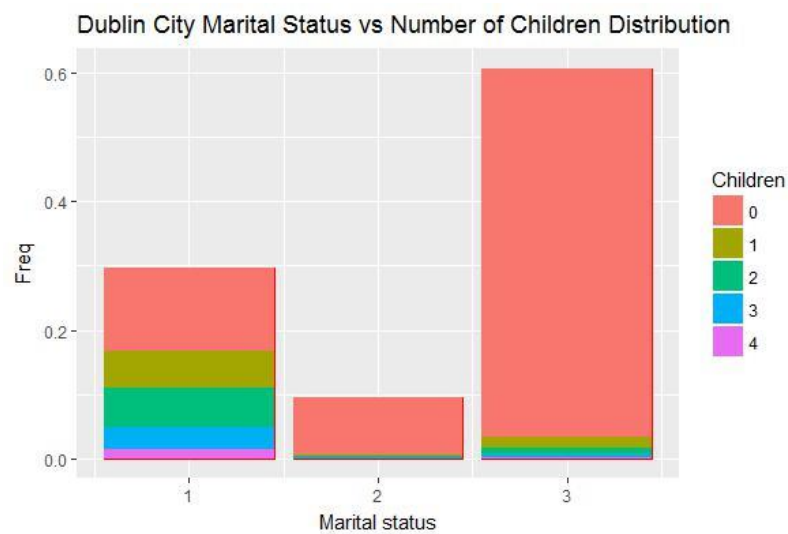


Figure 24 Dublin synthetic population distribution according to marital status category: 1 - All married; 2 – Separated (including deserted) , Divorced, Widowed; 3 - Single and number of children

4.2 SIMULATION RESULTS

This chapter consists of two sections. Firstly, the scenarios for simulation experiments are presented. Secondly, the results of multi-agent simulations are provided and discussed. The experiments have been run on an HPC large scale computing cluster running Linux Ubuntu nodes and Oracle Java. For analysis of the results, the meta-modelling approach has been used, as detailed earlier in section 3.3

4.2.1 SIMULATION EXPERIMENTS

The simulation experiments in the alpha SIM are based on data provided by the pilot (municipality of Prato – see previous Subsection). The population of Prato amounted to 191,000 citizens at the end of 2014. We have used the census data with the following socio-demographic features: city district of inhabitant, sex, age category, occupational status, marital status and yearly income category. We have all the marginal distributions and additionally selected cross marginal distributions. Based on this, a representative synthetic population of 1,420 citizens was generated to use in the simulation experiments. The primary opinions are based on the observed socio-economic features and the trinomial model. In order to introduce a representativeness bias in our model of preferences, older citizens tend to vote for “yes” and the wealthier tend to vote “no”. We have used the trinomial model with 3 different opinion values, although one can also allow for a continuous preferences distribution. We also consider such a simulation where the opinion of a citizen takes continuous rather than discrete values from the interval $[-1,1]$, in order to consider and model the case where a citizen can progressively express the strength of her/his opinion e.g. be completely in favour of the project, be almost completely in favour of the project, be in favour of the project, be moderately in favour of the project, etc.

In each step (we allow for a limited number of steps in the simulation as it represents a more realistic assumption of user activity on the social platform) we may observe the preference dynamics of the subpopulation (sample) and population.

For simulation experiments we consider a 5-dimensional parameter space (citizenClassVersion, edgesVersion – limited to just 1 version in the presented results for greater readability, betaVersion, primaryOpinionVersion – only one in the presented results, selectionVersion). The explanation of each parameter along with its possible parameter values have been presented in Table 1. A full parameter sweep is considered. The full Cartesian product of all parameters consists of $2 \times 1 \times 16 \times 8$ points. For each parameter set we repeat the simulation 30 times with different random seed values that determine the order in which opinions are updated within the social network.

Within the parameter space we consider 8 distinct scenarios (0-15) for subpopulation selection. The parameterization for these scenarios is given in Table 2. “Selection probability” is the snowball sampling probability that a particular node will be selected. Network propagation depth defines how many neighbour levels are used for sampling. The number of citizens in a sub-network defines the initial selection size.

Parameter name	Parameter description /detailed model description can be found in section 3.2/	Values considered in the parameter sweep
citizenClass	Type of decision-making process for agents within the model. Two types are considered: {a, b}. For a detailed discussion see Appendix	(a) Discrete opinion (b) Continuous opinion
edgesVersion	Type of edge propagation in the social network	We consider parameters that influence: a) The average links b) Degree of homophily c) The mean and standard deviation of the citizens' individual features (individual attractiveness) distribution
Beta	β , such that $\beta \in (0,1)$ (homogenous for all the agents in the current implementation) represents the weight of the agent's own opinion relative to the opinions of the neighbouring agents that the agent is connected to.	a) 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.9 for the own opinion weight b) 0 or 0.1 for the portal/population average opinion weight c) Remaining weight gives the weight of friends (neighbours opinion)
primaryOpinionVersion	Type of primary opinion	0
selectionVersion	How is a subpopulation of agents selected?	0 - 7

Table 1 Parameter values. We perform a full Cartesian product parameter sweep. Hence the parameter space size is $2 \times 16 \times 1 \times 8 = 256$. For each data point 30 simulations are carried out that last over 6 periods on a population of 1420 agents.

Subpopulation selection type	Selection probability	Network propagation depth	Number of citizens in a sub network
0	0.3000	1	30
1	0.3000	1	40
2	0.3000	2	30
3	0.3000	2	40
4	0.4000	1	30
5	0.4000	1	40
6	0.4000	2	30
7	0.4000	2	40

Table 2 Parameterization for subpopulation selection procedure

4.2.2 OPINION DYNAMICS IN SOCIAL NETWORKS

The goal of this section is to present the initial simulation results from the SIM alpha module. We start with illustrative results from a selected single simulation. Following this, we move on to meta-modelling (see Section 3.3) across the entire parameter space.

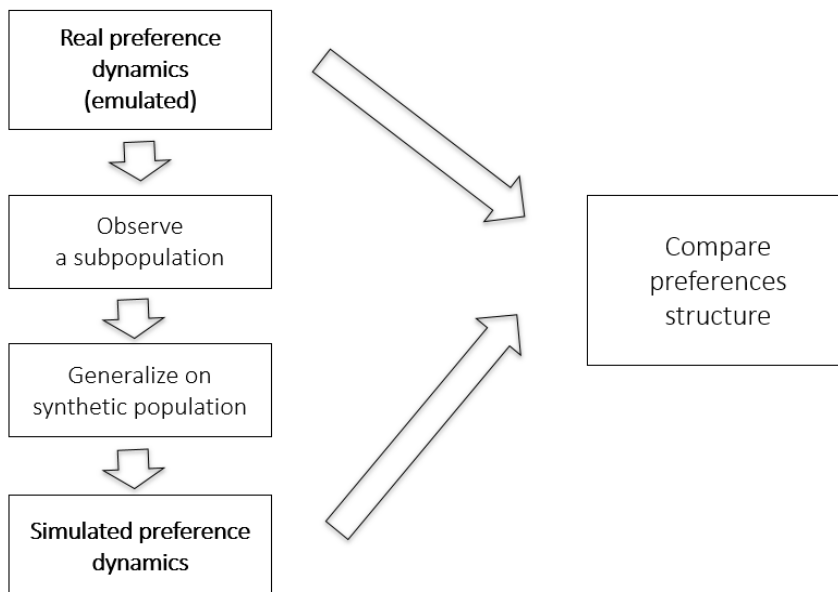


Figure 25 The goal of simulation experiments is to measure preference concordance between real preferences and simulated preferences. The concordance is a measure of the validity of the approach taken for preference solicitation in the Open Data Governance Model.

The simulations were divided into simulation experiments phases (Phase 1 and Phase 2) . The main difference between phases is the method we used for the synthetic population generation and the links between citizens in particular. In the (simulation experiment) Phase 1 we used the simplified model that describes the probability of the link between two citizens (see. Paragraph 2.3.5 for detailed explanations). The influence of mutual friends (mathematically modeled as geometrically weighted edgewise shared partners) is treated in a simplified way. Namely we the same (population average) number of mutual friends for each pair of citizens. In the (simulation experiment) Phase 2 the influence of mutual friends is modeled for each pair of citizens individually using the actual number of the mutual friends. Phase 2 requires the application of complex statistical procedure as Monte-Carlo Markov Chain (MCMC) method.

4.2.2.1 SIMULATION PHASE 1

Phase 1 includes building the network representation of the entire population and has the following sequence of steps:

- 1) Synthetic population generation – depending on the available census data (the attributes of citizens, the multi-way distributions available, the so-called seed-sample), the synthetic population is generated. The synthetic population consists of all the citizens that belong (are of interest) to the particular PA that uses SPOD. The citizens are mathematically represented by a set of socio-demographic features (attributes), e.g. age, gender, number of children, etc.
- 2) Links generation (to build the mathematical representation of the synthetic population as a graph) – the model of the probability of the link existence between a pair of the citizens in the synthetic population, that we use takes into consideration:
 - a. intrinsic, psychological features (“attractiveness”) that influence the propensity to build social links with other citizens, independent of socio-demographic features.
 - b. Homophily – the tendency of the citizens with the same or similar socio-demographic features to establish social relations (links) more often that citizens with different socio-demographic features. Mathematical details are given in Appendix 1.
- 3) Sample selection from synthetic population – the sample (which represents the future subpopulation that will use SPOD) is characterized by socio-demographic attributes (the same as the generated in previous steps synthetic population) and the links between them. We do the snowball sampling using the following parameters:
 - a. a chosen number of initial citizens (so-called first users)
 - b. the number of the consecutive waves (1st wave consist of all the direct friends of the initial citizens, that were encouraged by the initial citizens to join the SPOD platform. 2nd wave consist of all the direct friends of 1st wave citizens, that were encouraged by 1st wave citizens to join SPOD platform, etc. In the simulation we limit the number of waves.)
 - c. the probability that a citizen is encouraged by her/his friend to join SPOD platform.
- 4) Model estimation – based on the selected (in previous step) sample we estimate 2 models
 - a. The logistic regression model, that gives the probability of the link between selected pair of the citizens
 - b. The trinomial model, that gives the probability of the initial opinion for each citizen, based on her/his socio-demographic features
- 5) Network representation of the synthetic population – the missing links between synthetic population citizens are estimated in a statistically efficient way. In such a manner, the synthetic population acquires a mathematical representation in the form of a network (graph). However, contrary to the graph representation of the entire synthetic population (see. step 2), the links are reconstructed based on the selected sample, (see step 3) and the estimated logistic regression model (see. Step 4). The graph consists of nodes (citizens and attributes) and links (edges) between citizens representing the fact that both citizens communicate and discuss relevant issues in real life.

- 6) Primary opinion reconstruction – the missing primary opinions of the citizens that are not in a sample (see step 3) are reconstructed based on the socio-demographic features and the estimated trinomial model (see. step 4).

For the final usage of SIM only steps 1), 2), 4) 5) and 6) will be applied. Step 3) will not be necessary as we will treat the SPOD users as a selected sample. Moreover, we will not observe the social links between citizens that are not on the SPOD platform. The numerical experimental design we propose (that leverages on the assumption that we know the entire synthetic population including the links between citizens) gives us, however, the opportunity to assess whether it is possible to estimate the opinion dynamics for the whole population based on the observed sample, what the potential error is and which factors influence the results the most.

The results show that the most important factors that influence the reconstructed opinion dynamics are the citizen type (representing the way that observed opinions – in a discrete or continuous way - are expressed and processed by citizens) and the beta factors that represent both the way a citizen is attached to their opinion and the way the opinion of other citizens (social influence) lead to revision of one's own opinion. The second factor is the number and structure of the sample.

The results also show that for beta factors having extreme values (a citizen is attached to their opinion in an extreme way or changes their opinion very easily based on the opinions of other citizens), we can reconstruct the opinion dynamics for the whole population with very limited error. For the moderate beta values (which better correspond to real situations) the reconstruction error (MAE) is limited.

Explained variable: sample_yes_PCT on results_f0

* denotes terminal node

```

1) root 34530 1091.9120000 0.8741121
  2) citizenClass=Dominating_opin. 17280 720.1088000 0.7856621
    4) beta< 1.5 960 17.4951200 0.4881210 *
    5) beta>=1.5 16320 612.6248000 0.8031645
      10) beta< 13.5 11520 491.3273000 0.7726209
        20) beta>=2.5 10560 455.7070000 0.7561138
          40) beta< 3.5 960 14.0395800 0.5157529 *
          41) beta>=3.5 9600 380.6587000 0.7801499
            82) beta>=4.5 8640 345.7402000 0.7603964
              164) beta< 5.5 960 14.0592500 0.5062146 *
              165) beta>=5.5 7680 261.9039000 0.7921691
                330) beta>=6.5 6720 230.0527000 0.7682903
                  660) beta< 7.5 960 11.1472300 0.5872638 *
                  661) beta>=7.5 5760 182.2024000 0.7984614
                    1322) beta>=8.5 4800 150.8030000 0.7661015
                      2644) beta< 9.5 960 5.1121150 0.6102834 *
                      2645) beta>=9.5 3840 116.5558000 0.8050560
                        5290) beta>=10.5 2880 80.9230300 0.7498370
                          10580) beta< 11.5 960 4.3963990 0.6188206 *
                          10581) beta>=11.5 1920 51.8086000 0.8153452
                            21162) beta>=12.5 960 5.6016810 0.6613552 *
                            21163) beta< 12.5 960 0.6780876 0.9693352 *
                              5291) beta< 10.5 960 0.5067081 0.9707129 *
                                1323) beta< 8.5 960 1.2410550 0.9602610 *
                                  331) beta< 6.5 960 1.1974500 0.9593206 *
                                    83) beta< 4.5 960 1.2051410 0.9579315 *
                                      21) beta< 2.5 960 1.0909730 0.9541995 *
                                        11) beta>=13.5 4800 84.7572500 0.8764692
                                          22) beta< 17.5 3840 69.1782200 0.8481309
                                            44) beta>=14.5 2880 48.9669900 0.8066713
                                              88) beta< 15.5 960 3.0072640 0.7110340 *
                                              89) beta>=15.5 1920 32.7887600 0.8544900
                                                178) beta>=16.5 960 2.2270720 0.7289119 *
                                                179) beta< 16.5 960 0.2835529 0.9800681 *
                                                  45) beta< 14.5 960 0.4095194 0.9725098 *

```

```

23) beta>=17.5 960 0.1603808 0.9898220 *
3) citizenClass=Mean_neighbour 17250 101.1916000 0.9627159
6) beta< 3.5 2850 62.1092100 0.8763703 *
7) beta>=3.5 14400 13.6287100 0.9798051 *

```

And varImpPlot for this model

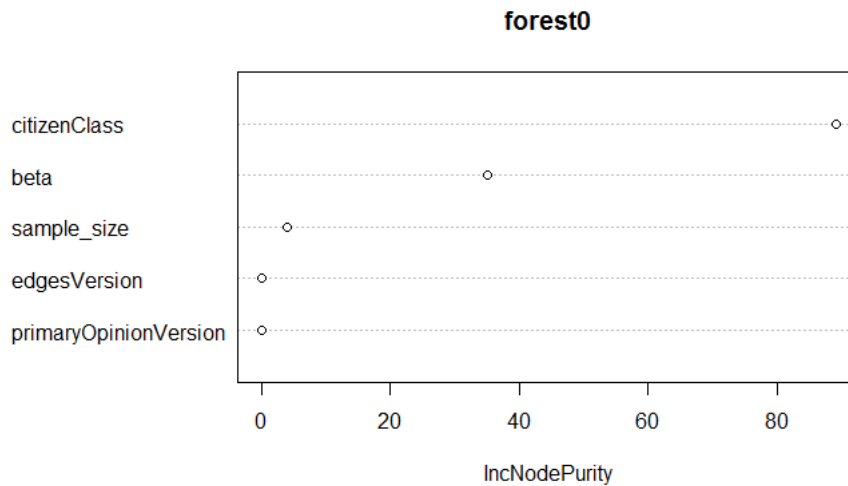


Figure 26 Random forest node purity at the beginning of simulation across the parameter sweep and simulation runs. It can be clearly seen that the preference concordance is determined by the type of opinion diffusion dynamics and the weight of an agent's own opinion.

The results from linear model for this variable are:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	8.335e-01	3.376e-03	246.89	<2e-16	***
citizenClassDominating_opin.	-1.769e-01	1.570e-03	-112.68	<2e-16	***
beta	9.406e-03	1.514e-04	62.11	<2e-16	***
sample_size	6.517e-04	4.679e-05	13.93	<2e-16	***

Explained variable: sample_yes_PCT on results_f5

* denotes terminal node

```

1) root 34530 6.189301e+03 7.252969e-01
2) citizenClass=Dominating_opin. 17280 4.087435e+03 4.862159e-01
4) beta< 1.5 960 3.560555e-03 6.218905e-05 *
5) beta>=1.5 16320 3.847193e+03 5.148132e-01
10) beta< 17.5 15360 3.634374e+03 4.863199e-01
20) beta>=16.5 960 0.000000e+00 0.000000e+00 *
21) beta< 16.5 14400 3.392191e+03 5.187412e-01
42) beta< 15.5 13440 3.176857e+03 4.860963e-01
84) beta>=14.5 960 0.000000e+00 0.000000e+00 *
85) beta< 14.5 12480 2.932570e+03 5.234883e-01
170) beta< 13.5 11520 2.713133e+03 4.852447e-01
340) beta>=12.5 960 0.000000e+00 0.000000e+00 *
341) beta< 12.5 10560 2.466540e+03 5.293579e-01
682) beta< 11.5 9600 2.254333e+03 4.846074e-01
1364) beta>=10.5 960 3.983333e+00 4.166667e-03 *
1365) beta< 10.5 8640 2.004138e+03 5.379896e-01
2730) beta< 9.5 7680 1.796583e+03 4.832491e-01
5460) beta>=8.5 960 0.000000e+00 0.000000e+00 *
5461) beta< 8.5 6720 1.540368e+03 5.522847e-01
10922) beta< 7.5 5760 1.345530e+03 4.829899e-01

```



```

21844) beta>=6.5 960 0.000000e+00 0.000000e+00 *
21845) beta< 6.5 4800 1.076792e+03 5.795879e-01
43690) beta< 5.5 3840 8.948104e+02 4.825644e-01
87380) beta>=4.5 960 1.995833e+00 2.083333e-03 *
87381) beta< 4.5 2880 5.973111e+02 6.427248e-01
174762) beta< 3.5 1920 4.450816e+02 4.807695e-01
349524) beta>=2.5 960 0.000000e+00 0.000000e+00 *
349525) beta< 2.5 960 1.294174e+00 9.615390e-01 *
174763) beta>=3.5 960 1.147459e+00 9.666354e-01 *
43691) beta>=5.5 960 1.241944e+00 9.676816e-01 *
10923) beta>=7.5 960 1.230477e+00 9.680533e-01 *
2731) beta>=9.5 960 4.347143e-01 9.759142e-01 *
683) beta>=11.5 960 7.310274e-01 9.768633e-01 *
171) beta>=13.5 960 4.018846e-01 9.824113e-01 *
43) beta>=15.5 960 4.912849e-01 9.757696e-01 *
11) beta>=17.5 960 8.237112e-01 9.707065e-01 *
3) citizenClass=Mean_neighbour 17250 1.247096e+02 9.647936e-01 *

```

and varImpPlot for this model

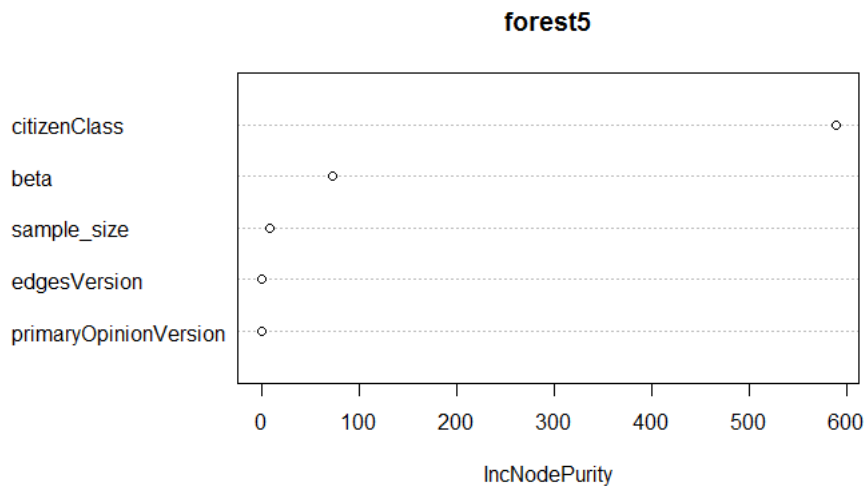


Figure 27 Random forest node purity at the end of simulation across the parameter sweep and simulations runs. Similarly to the previous graph it can be clearly seen that the preference concordance is determined by the type of opinion diffusion dynamics and the weight of an agent's own opinion.

The results from the linear model for this variable are:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.8770001	0.0080294	109.223	<2e-16 ***
citizenClassDominating_opin.	-0.4784667	0.0037345	-128.119	<2e-16 ***
beta	0.0076802	0.0003602	21.323	<2e-16 ***
sample_size	0.0002413	0.0001113	2.169	0.0301 *

The relative importance of variables in two models (red is period 5):

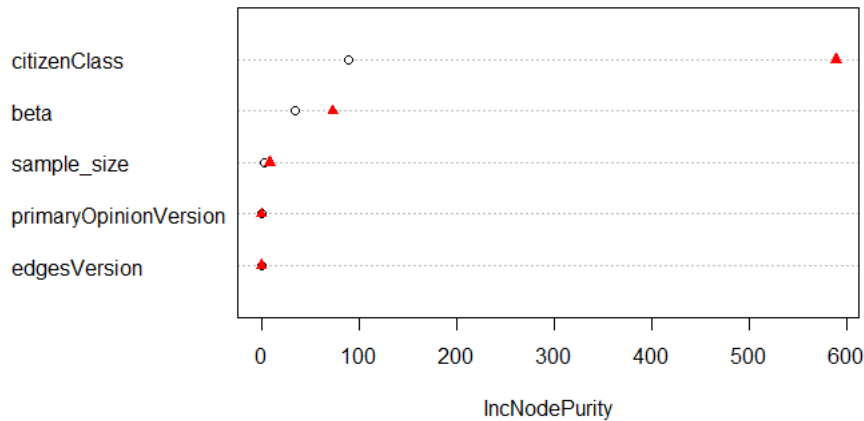


Figure 28 Random forest node purity at the end of simulation across the parameter sweep and simulations runs. The variable importance is presented at the beginning of simulation (black circle) as well as at the end (red triangle)

Explained variable: pop_extreme_PCT on results_f0

* denotes terminal node

- 1) root 34530 9.638692e-01 0.0009211423
- 2) beta>=1.5 32640 1.485548e-02 0.0001127322 *
- 3) beta< 1.5 1890 5.592976e-01 0.0148822600
- 6) sample_size>=39 1830 4.142965e-01 0.0135361300
- 12) citizenClass=Mean_neighbour 900 2.096831e-01 0.0066971830
- 24) sample_size>=59.5 420 5.757781e-03 0.0014050970 *
- 25) sample_size< 59.5 480 1.818704e-01 0.0113277600
- 50) sample_size< 53.5 240 1.540637e-02 0.0041901410 *
- 51) sample_size>=53.5 240 1.420102e-01 0.0184653800
- 102) sample_size>=57 90 9.796249e-03 0.0094209700 *
- 103) sample_size< 57 150 1.204345e-01 0.0238920200
- 206) sample_size< 55.5 60 1.903918e-02 0.0128873200 *
- 207) sample_size>=55.5 90 8.928502e-02 0.0312284800 *
- 13) citizenClass=Dominating_opin. 930 1.217830e-01 0.0201544800
- 26) sample_size>=77 150 4.828106e-03 0.0130469500 *
- 27) sample_size< 77 780 1.079201e-01 0.0215213100
- 54) sample_size< 51.5 210 9.168828e-03 0.0148725700 *
- 55) sample_size>=51.5 570 8.604798e-02 0.0239708400
- 110) sample_size>=60 420 4.605924e-02 0.0225486300
- 220) sample_size< 66 210 8.222162e-03 0.0181924900 *
- 221) sample_size>=66 210 2.986719e-02 0.0269047600
- 442) sample_size>=72.5 90 2.478675e-03 0.0194835700 *
- 443) sample_size< 72.5 120 1.871434e-02 0.0324706600
- 886) sample_size< 70 90 2.235915e-03 0.0261737100 *
- 887) sample_size>=70 30 2.203862e-03 0.0513615000 *
- 111) sample_size< 60 150 3.676051e-02 0.0279530500
- 222) sample_size< 57 120 5.088570e-03 0.0206983600 *
- 223) sample_size>=57 30 9.358262e-05 0.0569718300 *
- 7) sample_size< 39 60 4.054589e-02 0.0559389700 *

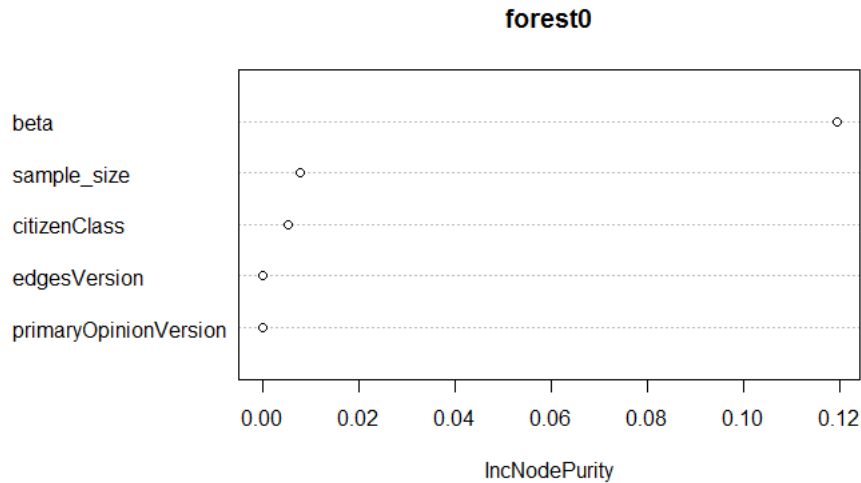


Figure 29 Random forest node purity at the end of simulation across the parameter sweep and simulations runs. Similarly to the previous graph it can be clearly seen that the preference concordance is determined by the type of opinion diffusion dynamics and the weight of an agent's own opinion.

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	3.508e-03	1.169e-04	29.999	< 2e-16 ***
citizenClassDominating_opin.	9.325e-04	5.439e-05	17.146	< 2e-16 ***
beta	-2.830e-04	5.245e-06	-53.957	< 2e-16 ***
sample_size	-5.943e-06	1.621e-06	-3.667	0.000246 ***

Explained variable: pop_extreme_PCT on results_f5

* denotes terminal node

- 1) root 34530 3514.5690000 0.135602700
- 2) citizenClass=Mean_neighbour 17250 141.8961000 0.008578036 *
- 3) citizenClass=Dominating_opin. 17280 2816.4900000 0.262406800
 - 6) beta>=7.5 10560 299.2775000 0.075341310
 - 12) beta>=9.5 8640 22.8239700 0.030702270 *
 - 13) beta< 9.5 1920 181.7631000 0.276217000
 - 26) beta< 8.5 960 0.0000000 0.000000000 *
 - 27) beta>=8.5 960 35.2750900 0.552434000 *
 - 7) beta< 7.5 6720 1566.9890000 0.556366800
 - 14) beta>=1.5 5760 1362.1890000 0.485206900
 - 28) beta< 2.5 960 0.0000000 0.000000000 *
 - 29) beta>=2.5 4800 1090.9790000 0.582248200
 - 58) beta>=3.5 3840 901.4245000 0.483007400
 - 116) beta< 4.5 960 0.0000000 0.000000000 *
 - 117) beta>=4.5 2880 602.8054000 0.644009900
 - 234) beta>=5.5 1920 442.9332000 0.478691300
 - 468) beta< 6.5 960 0.0000000 0.000000000 *
 - 469) beta>=6.5 960 2.9740720 0.957382600 *
 - 235) beta< 5.5 960 2.4499560 0.974647200 *
 - 59) beta< 3.5 960 0.4583384 0.979211400 *
 - 15) beta< 1.5 960 0.6300358 0.983326700 *

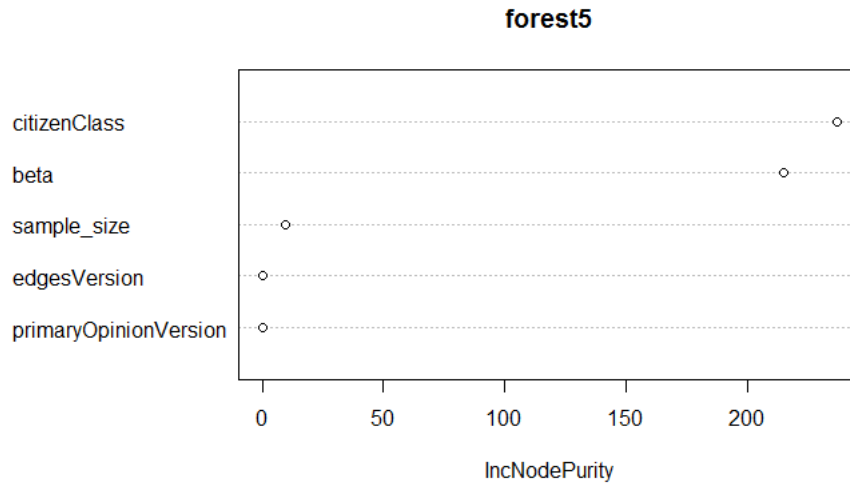


Figure 30 Random forest node purity at the end of simulation across the parameter sweep and simulations runs. Similarly to the previous graph it can be clearly seen that the preference concordance is determined by the type of opinion diffusion dynamics and the weight of an agent's own opinion.

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.163e-01	6.189e-03	34.950	<2e-16 ***
citizenClassDominating_opin.	2.535e-01	2.879e-03	88.060	<2e-16 ***
beta	-2.295e-02	2.776e-04	-82.646	<2e-16 ***
sample_size	1.736e-04	8.578e-05	2.024	0.043 *

Comparison of period 0 and 5 (red triangle is period 5)

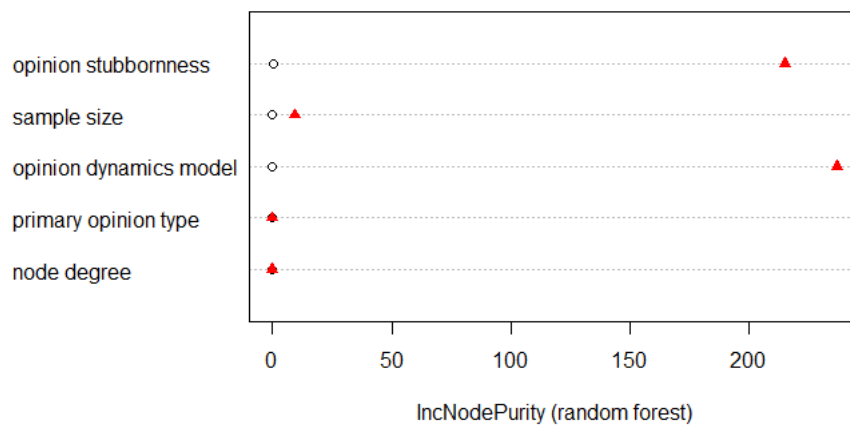
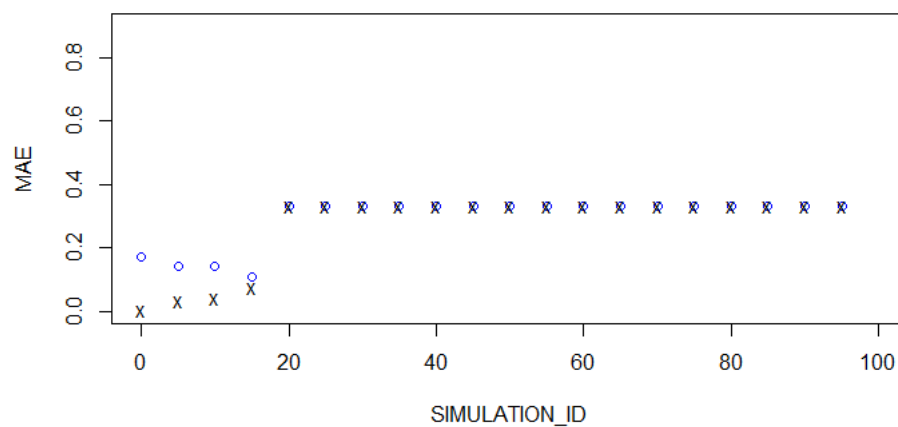


Figure 31 Random forest node purity at the end of simulation across the parameter sweep and simulations runs. The variable importance is presented at the beginning of simulation (black circle) as well as at the end (red triangle)



4.2.2.2 SIMULATION PHASE 2

Phase 2 includes simulation of the opinion dynamics and has the following sequence of steps:

Phase 1 includes building the network representation of the entire population and has the following sequence of steps:

- 1) Synthetic population generation – depending on the available census data (the attributes of citizens, the multi-way distributions available, the so-called seed-sample), the synthetic population is generated. The synthetic population consists of all the citizens that belong (are of interest) to the particular PA that uses SPOD. The citizens are mathematically represented by a set of socio-demographic features (attributes), e.g. age, gender, number of children, etc.
- 2) Links generation (to build the mathematical representation of the synthetic population as a graph) – the model of the probability of the link existence between a pair of the citizens in the synthetic population, that we use takes into consideration:
 - a. intrinsic, psychological features (“attractiveness”) that influence the propensity to build social links with other citizens, independent of socio-demographic features.
 - b. Homophily – the tendency of the citizens with the same or similar socio-demographic features to establish social relations (links) more often than citizens with different socio-demographic features.
 - c. The influence of mutual friends (geometrically weighted edgewise shared partners) Mathematical details are given in Appendix 1.
- 3) Sample selection from synthetic population – the sample (which represents the future subpopulation that will use SPOD) is characterized by socio-demographic attributes (the same as the generated in previous steps synthetic population) and the links between them. We do the snowball sampling using the following parameters:
 - a. a chosen number of initial citizens (so-called first users)
 - b. the number of the consecutive waves (1st wave consist of all the direct friends of the initial citizens, that were encouraged by the initial citizens to join the SPOD platform. 2nd wave consist of all the direct friends of 1st wave citizens, that were encouraged by 1st wave citizens to join SPOD platform, etc. In the simulation we limit the number of waves.)
 - c. the probability that a citizen is encouraged by her/his friend to join SPOD platform.
- 4) Model estimation – based on the selected (in previous step) sample we estimate 2 models
 - a. The logistic regression model, that gives the probability of the link between selected pair of the citizens
 - b. The trinomial model, that gives the probability of the initial opinion for each citizen, based on her/his socio-demographic features
- 5) Network representation of the synthetic population – the missing links between synthetic population citizens are estimated in a statistically efficient way. In such a manner, the synthetic population acquires a mathematical representation in the form of a network (graph). However, contrary to the graph representation of the entire synthetic population (see. step 2), the links are reconstructed based on the selected sample, (see step 3) and the estimated logistic regression model (see. Step 4). The graph consists of nodes (citizens and attributes) and links (edges) between citizens representing the fact that both citizens communicate and discuss relevant issues in real life.
- 6) Primary opinion reconstruction – the missing primary opinions of the citizens that are not in a sample (see step 3) are reconstructed based on the socio-demographic features and the estimated trinomial model (see. step 4).

Steps 1), 3), 4), 5) and 6) are common for both phases. The only is the modeling of the mutual friends influence on the probability of the social link between a pair of citizens. This however requires completely different statistical approach (Monte-Carlo Markov Chain).

The results are similar to the results of Phase 1 (the importance of the beta factors is more pronounced, compared to the way of opinion gathering and processing). However, one can notice that adding and considering more complex models of social interaction is possible within the SIM platform.

Sample_yes_pct for results_f0

* denotes terminal node

```

1) root 34560 422.491900 0.8605312
2) beta< 8.5 15360 116.944300 0.7673677
   4) citizenClass=Mean_neighbour 7680 50.172460 0.7142971
      8) beta< 6.5 5760 31.811760 0.6919163
         16) sample_size< 213.5 3570 20.473560 0.6683884 *
         17) sample_size>=213.5 2190 6.140514 0.7302698 *
      9) beta>=6.5 1920 6.819980 0.7814394 *
   5) citizenClass=Dominating_opin. 7680 23.510510 0.8204383
      10) sample_size< 175 4050 13.849980 0.7968919 *
      11) sample_size>=175 3630 4.909810 0.8467091 *
3) beta>=8.5 19200 65.578520 0.9350620
   6) beta< 12.5 7680 15.769160 0.8900292 *
   7) beta>=12.5 11520 23.851550 0.9650838
      14) citizenClass=Dominating_opin. 5760 9.807085 0.9301677 *
      15) citizenClass=Mean_neighbour 5760 0.000000 1.0000000 *

```

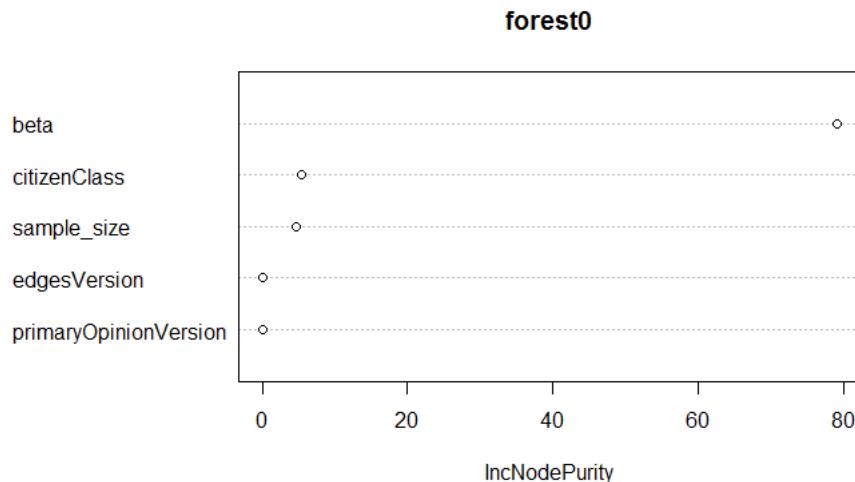


Figure 32 Random forest node purity at the end of simulation across the parameter sweep and simulations runs. Similarly to the previous graph it can be clearly seen that the preference concordance is determined by the type of opinion diffusion dynamics and the weight of an agent's own opinion.

Parametric coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	6.537e-01	1.054e-03	619.99	<2e-16 ***
citizenClassDominating_opin.	2.024e-02	6.810e-04	29.73	<2e-16 ***
beta	1.703e-02	6.563e-05	259.51	<2e-16 ***
sample_size	1.824e-04	3.696e-06	49.34	<2e-16 ***

Sample_yes_pct for results_f5

* denotes terminal node

```

1) root 34560 1530.740000 0.8189200
2) beta< 7.5 13440 728.503100 0.7239504
4) citizenClass=Mean_neighbour 6720 172.240900 0.5928206
8) beta< 1.5 960 46.486200 0.3743445 *
9) beta>=1.5 5760 72.295060 0.6292333 *
5) citizenClass=Dominating_opin. 6720 325.161100 0.8550802
10) sample_size< 175 3570 203.340700 0.7943731 *
11) sample_size>=175 3150 93.752670 0.9238816 *
3) beta>=7.5 21120 603.879600 0.8793553
6) citizenClass=Dominating_opin. 10560 457.716000 0.8163862
12) beta>=16.5 1920 202.746400 0.5775716
24) beta< 17.5 960 2.502646 0.2554782 *
25) beta>=17.5 960 1.054926 0.8996651 *
13) beta< 16.5 8640 121.133600 0.8694561
26) sample_size< 143.5 3990 66.838010 0.8198493 *
27) sample_size>=143.5 4650 36.051770 0.9120219 *
7) citizenClass=Mean_neighbour 10560 62.420480 0.9423244
14) beta< 12.5 4800 20.267360 0.8731136 *
15) beta>=12.5 5760 0.000000 1.0000000 *

```

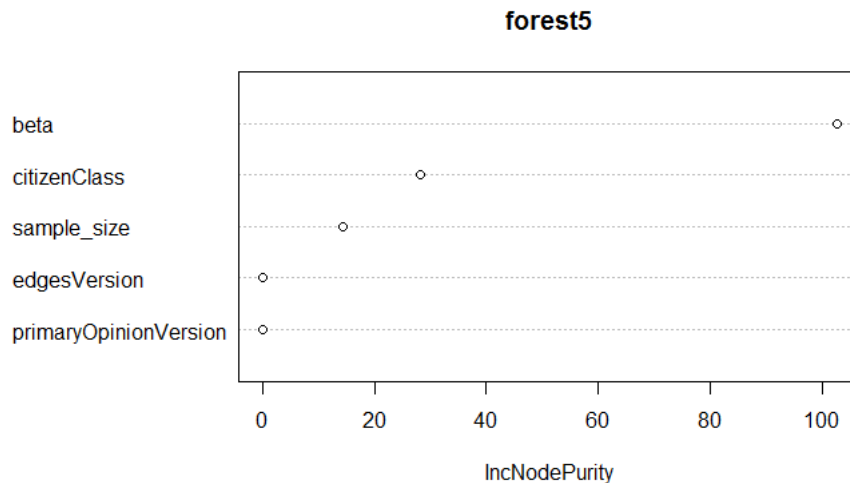


Figure 33 Random forest node purity at the end of simulation across the parameter sweep and simulations runs. Similarly to the previous graph it can be clearly seen that the preference concordance is determined by the type of opinion diffusion dynamics and the weight of an agent's own opinion.

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	6.256e-01	3.285e-03	190.42	<2e-16 ***
citizenClassDominating_opin.	2.490e-02	2.122e-03	11.73	<2e-16 ***
beta	1.277e-02	2.045e-04	62.44	<2e-16 ***
sample_size	3.114e-04	1.151e-05	27.04	<2e-16 ***

The simulation results show that the representativeness bias increases while opinions becomes more homogenous. The weight of a user's owns an opinion (beta parameter) is the most important factor in the elicitation of average preferences for the entire population. However, when we consider preference elicitation errors the most crucial parameter is the type of information diffusion dynamics.

4.2.3 SIMULATION RESULTS - CONCLUSIONS

In this section we have presented beta versions of the tools for optimal design of preference elicitation and aggregation systems with heterogeneity in citizens' geographical location and demographic structure. The tools are delivered as a multi-agent simulation model. In our modelling approach we have assumed that an economic system consists of interacting heterogeneous agents and hence we have considered a socioeconomic system with the agents representing members of the local community. The agent-based simulation model presented in the report has been implemented in Java using MASON for controlling the simulation. The applied (exponential random graph model (ERGM)) allow us to consider both individual intrinsic (psychological) features that influence how easily the given citizen becomes acquainted (discusses relevant issues) with other citizens and the role of mutual friends (neighbours) in helping to establish new relations (friendships) with other citizens.

The goal of the report is to present a method for preference elicitation that is robust to selection bias. In particular, we propose a method for analysing the dynamics of entire population preferences based on the observed preferences of a limited subpopulation. In the theoretical models we assume that the opinion diffusion process takes place across the entire population. However, a PA can only observe a sample subpopulation. In this approach, unobserved population members influence the observed information diffusion. Moreover, we assume that opinion diffusion has the same dynamics on the subsample as in the whole population.

The simulation tool has been calibrated with empirical data from the Prato population. In subsequent analysis we will incorporate data from another pilot as well as empirical data from the SPOD platform. The developed tool enables the finding and analysis of optimal mechanism design for sharing information on SPOD, required for the use of modelling tools that allow analysis of the heterogeneity of economic agents, their geographic location, virtual and real-world social networks and information flow (including data comprehension) within those networks.

In this report we have performed 7,680 simulation runs for an artificial population of 1,420 heterogeneous agents. Within those runs 256 virtual society parametrizations have been considered. The population structural properties have been calibrated with empirical data from one of the pilots - the Prato municipality. The simulation results show determinants for successful generalisation of preferences from a subpopulation onto the entire society. The simulation results show that the representativeness bias in a population increases when opinions becomes more homogenous. Moreover network connectivity and the importance of an agent's own opinion are major determinants of the quality of the preference elicitation process.

5 PA'S DECISION PROCESS VISUALISATION AND ANALYSIS FOR THE ODGM

SilverDecisions is a software for designing, presenting, analyzing and discussing decision process that takes place around the data within the PA by creating and analyzing decision trees. It is a community-driven project thus it might be used for teaching, research or any other community activities. Particularly the application may be used in public administration making decision process:

- 1) to communicate the rationale behind decisions made by PA to citizens and NGOs and link it to open data;
- 2) to collaborate between PA, citizens and NGOs in specification of decision making framework to help PA better understand all consequences of the decisions that are to be made.

SilverDecisions is used by the PA in My Space of the SPOD platform. SilverDecisions decision tree is a special type of SPOD Datalet.

5.1 SILVERDECISIONS' DEVELOPMENT PROCESS

5.1.1 PILOT'S REQUIREMENTS

The Software has been developed after discussion with pilots regarding ODGM for decisions making scenarios in Public Administration. Detailed discussion with Pilots and results of this discussion are presented in the Appendix A. The Appendix A also contains scenarios for utilizing the platform for Pilot's needs regarding decision making in the public administration.

5.1.2 DEVELOPMENT METHODOLOGY

SilverDecisions development uses the standard agile process. Software requirements are being formulated as results of continuous discussion with Pilots. The project is Open Source and managed via GitHub at:

<https://github.com/bkamins/SilverDecisions>

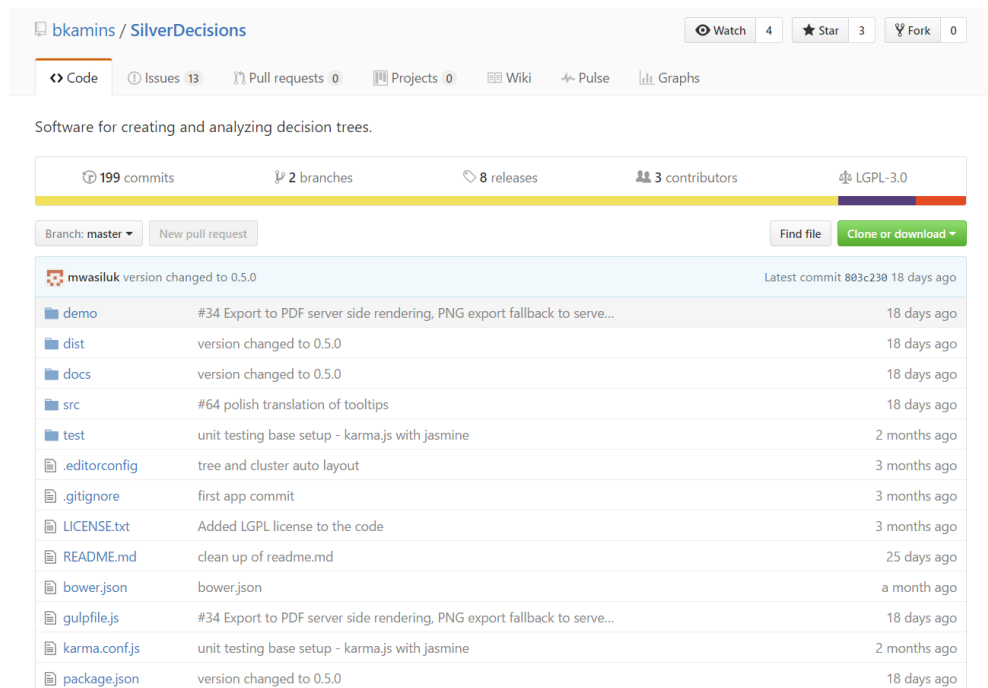


Figure 34 SilverDecisions' GitHub repository.

Code section includes project code developed in JavaScript where one may find “releases” part containing information on new featurers and enhancements in the subsequent project phases.

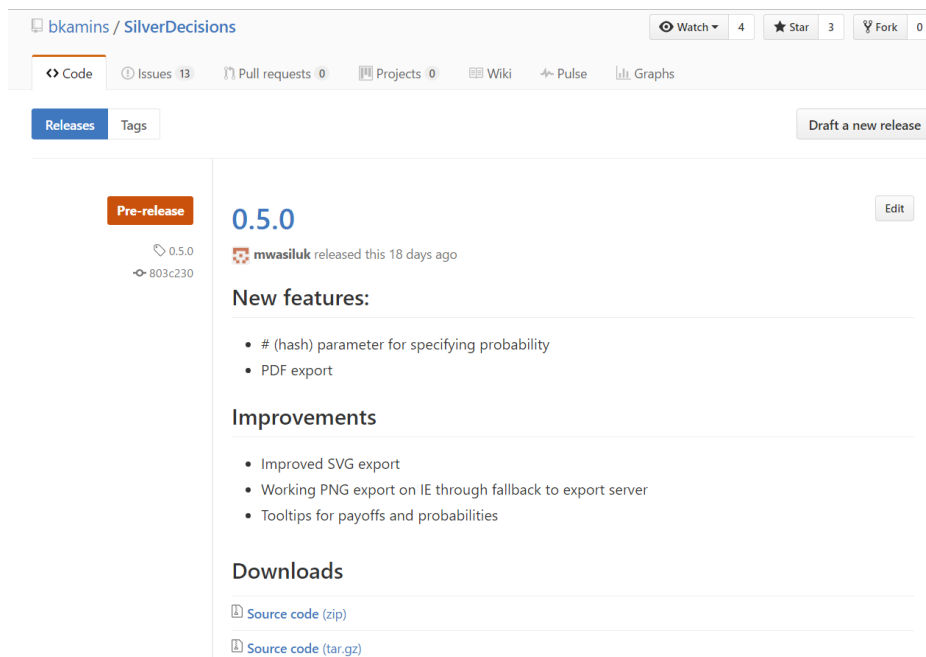


Figure 35 SilverDecisions' later release as of SIM beta version.

GitHub tracker is called *Issues* and it enables keeping track of enhancements, bugs and project development in general. Issues might be shared and discussed with the rest of the team as well as by other contributors. The following features help in issues filtering and categorization: *milestones*, *labels* and *assignees*. Milestones associate issues with given project phases. Currently the most up-to-date SilverDecisions' version available is 0.5.

There are 4 labels available in the project: “bug” (in case of a bug an issue may be filed), “enhancements” associated with the subsequent project phases, “needs docs” indicating project functionalities that need to be documented and “question”. Each Issue may be assigned to any project team member. On the Issue main page only open issues are visible but filtering and search options may be used to show closed issues too.

bkamins / SilverDecisions

Watch 4 Star 3 Fork 0

Code Issues 13 Pull requests 0 Projects 0 Wiki Pulse Graphs

is:issue is:open Labels Milestones New issue

Issue	Labels	Assignee	Sort
13 Open 61 Closed	Author Labels Milestones Assignee Sort		
1 Readonly seems not to work	question		
2 Add detection of Win7/IE11 on website	enhancement		2
3 Translation files	enhancement		4
4 Sensitivity analysis of decision trees	enhancement		
5 EVPI/EVII calculations	enhancement		
6 Multiple payoffs in decision tree	enhancement		1
7 Ideas for future functionalities	question		2
8 Custom keybindings on OS X	bug needs docs		7
9 Write SPOD integration documentation	needs docs		
10 Tree flipping algorithm	needs docs		7

Wiki page is the place where SilverDecisions documentation is being published – it is available at: <https://github.com/bkamins/SilverDecisions/wiki>

bkamins / SilverDecisions
Watch 3
Star 5
Fork 1

Code
Issues 8
Pull requests 0
Projects 0
Wiki
Pulse
Graphs

Home
Edit New Page

Bogumił Kamiński edited this page 13 hours ago · 19 revisions

Welcome!

SilverDecisions is a software for creating and analysing [decision trees](#).

You can run SilverDecisions by visiting its [website](#).

The fastest way to learn about it is to visit our [gallery of example decision trees](#).

If you have any comments or questions please write an e-mail to `silverdecisions [at] sgh.waw.pl` or create an [issue](#).

SilverDecisions is a community-driven project, so if you use it for teaching, research or any other activity that you would be willing to share please let us know. We would be glad to add a link to your activities on [SilverDecisions Community](#) page.

If you find a bug or have a feature request please fill an [issue](#). Before reporting bugs please make sure that you are using the latest production version of Silver Decisions and refresh the application by pressing `Ctrl-F5`.

The project is developed at [Decision Support and Analysis Division, Warsaw School of Economics](#).

This is a rewrite in JavaScript of old SilverDecisions that was developed using Microsoft Silverlight and is currently not maintained. If you liked the Silverlight version you can run it [here](#) and its

Pages 11

- Home
- Documentation
 - 1. Decision tree model
 - 2. Create your first decision tree with SilverDecisions
 - 3. Actions supported by application
 - 4. Application settings
 - 5. Usage tricks and known issues
- Community
- Gallery
- Developer's guide
- SilverDecisions in SPOD

Clone this wiki locally

<https://github.com/bkamins/>

Clone in Desktop

Apart from the documentation wiki includes the *gallery* of sample decision trees as well as the *developer's guide* where basic technical information is available. As SilverDecisions is a community-driven project, *community page* is also included within project wiki, which allows anyone to publish one's experiences and activities with SilverDecisions.

Gallery

Bogumił Kamiński edited this page on Nov 21, 2016 · 17 revisions

Edit New Page

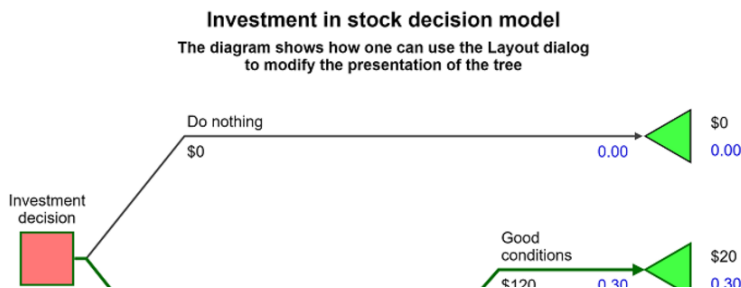
Submitting your work to the gallery of example decision trees

Users are welcome to submit proposals to publish their trees here. If you would like to have your tree published please write an e-mail to [Bogumił Kamiński](#) including: the JSON file of the tree you want to publish, author's name and contact information (this information will be published in the gallery under the terms of the GNU Lesser General Public License version 3.). It is recommended to provide a tree with a title and its short description.

Examples of decision trees

Investment in stock decision model

Author: [Bogumił Kamiński](#), publishing date: 2016/11/19, [open in SilverDecisions](#)



Pages 10

- Home
- Documentation
 - 1. Decision tree model
 - 2. Create your first decision tree with SilverDecisions
 - 3. Actions supported by application
 - 4. Application settings
 - 5. Usage tricks and known issues
- Gallery
- Developer's guide
- SilverDecisions in SPOD

Clone this wiki locally

<https://github.com/bkamins/>

Clone in Desktop

Developer's guide

Michał edited this page 18 days ago · 14 revisions

Edit New Page

SilverDecisions project is built with JavaScript ECMAScript 6 compiled to ES5 with [Babel](#), [npm](#) and [gulp](#).

Core dependencies are:

- [D3.js v4](#)
- [Math.js](#)
- [i18next](#)
- [lodash](#)

All dependencies are imported as ES6 modules and are prepackaged in the distribution files.

Quick start

1. Clone the repository
2. Make sure `nodejs` and `npm` is installed in your system ([nodejs.org](#)).
3. Install `gulp-cli` ([gulpjs.com](#)) and `bower` ([bower.io](#)) globally: `npm install --global gulp-cli bower`

Pages 10

- Home
- Documentation
 - 1. Decision tree model
 - 2. Create your first decision tree with SilverDecisions
 - 3. Actions supported by application
 - 4. Application settings
 - 5. Usage tricks and known issues
- Gallery
- Developer's guide
- SilverDecisions in SPOD

Clone this wiki locally

<https://github.com/bkamins/>

5.1.3 USER ACTIVITY MONITORING

The first SilverDecisions release took place on 14th Oct 2016. From that date till 19th Jan 2017 there were 4 669 unique application users (each user has usually several sessions of application usage). It should be noted that the number of users for the standalone application continuously increases – between 7th Jan 2017 and 19th Jan 2017 we could observe 70 around new application users a day.

The map below presents geographic distribution of number of user sessions of SilverDecisions. It can be observed that the software is used in all European Union Countries and also in almost all countries around the globe.

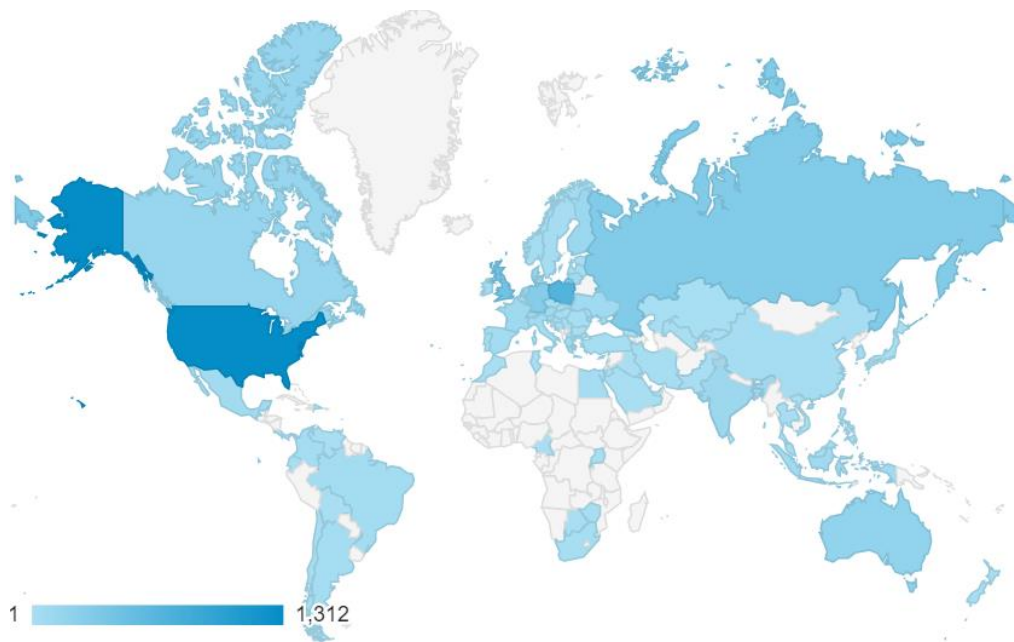
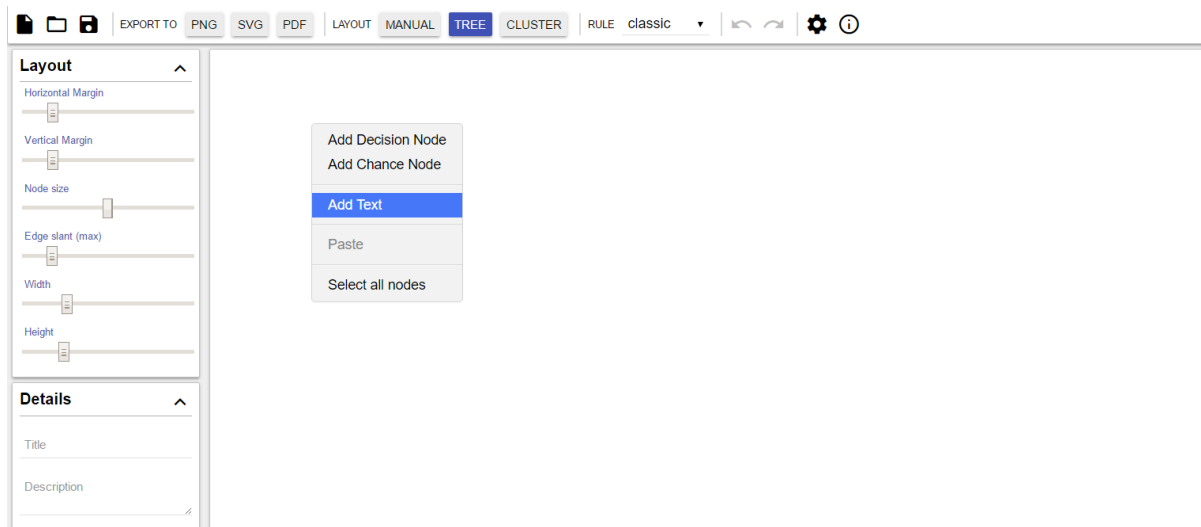


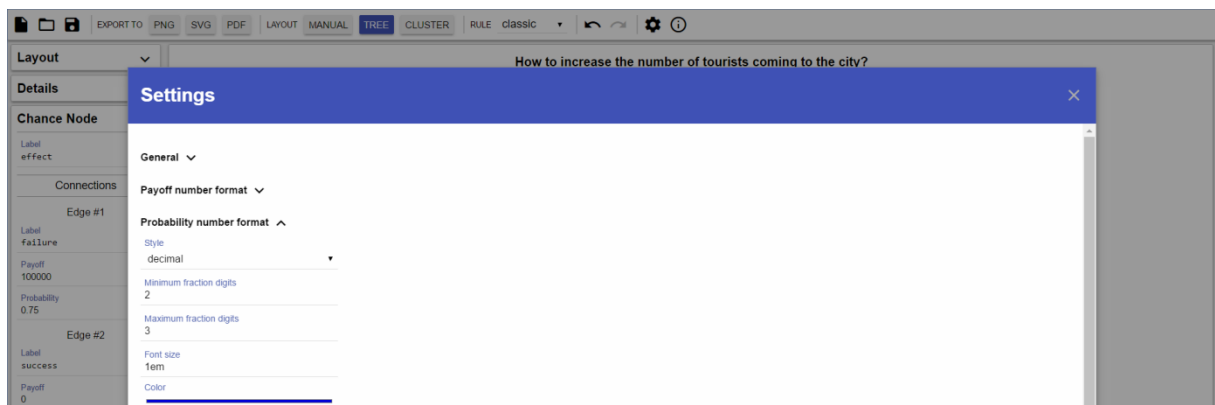
Figure 36 Users of SilverDecision software can be found on all continents around the globe .

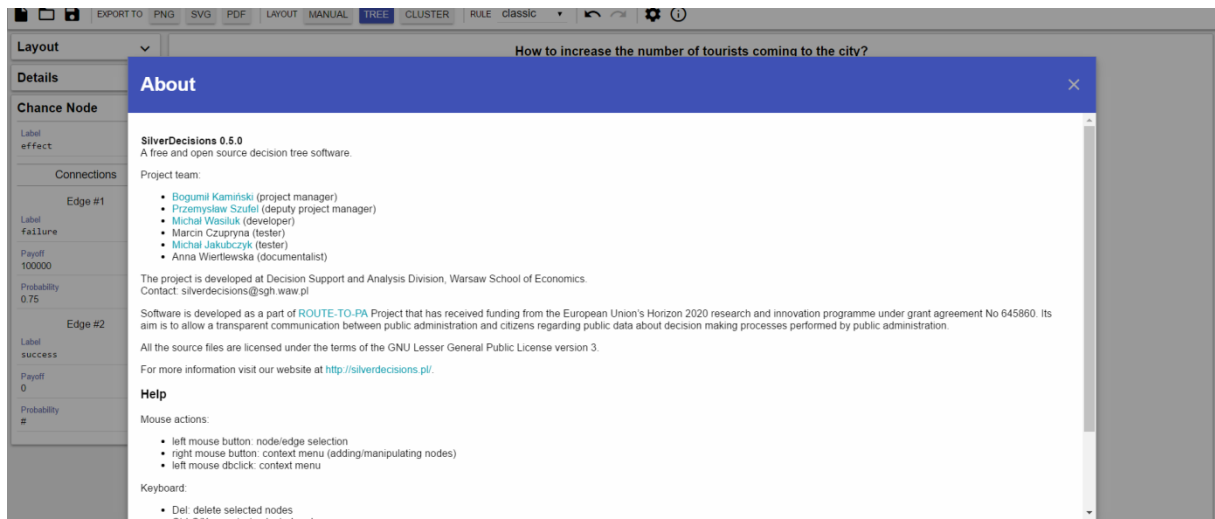
5.2 USER INTERFACE LAYOUT

Currently there are two available application languages: English and Polish. To start with SilverDecisions one needs to choose the language and Run the application. Below there is a screen of the main user application interface within which the following elements may be listed: *layout options* and *details panel* on the left side of the webpage, *context menu* on the white plotting canvas which is visible right after double/left mouse click, toolbar options in the top part with the following elements: the first three options from the left refer to *creating new diagram* (it clears current diagram), *opening diagrams* saved in JSON format and *saving diagram*. The next three buttons enable a graphical diagram *export* to PNG, SVG or PDF format. Then the *layout options* may be found and chosen from the following ones: manual, tree (default) and cluster. The next option enables decision rule selection: classic (default), maxi-min and max-max.

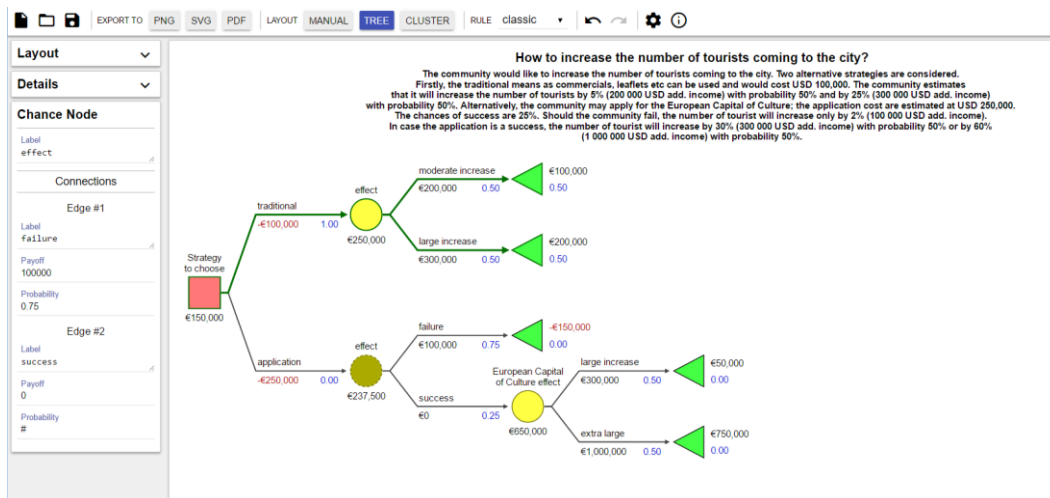


Then there are *undo* and *redo* buttons, *settings* panel fully described in the project documentation and *about* page containing some basic application instructions like mouse actions or keyboard shortcuts.





Options displayed in the left panel differs depending on the tree elements currently selected on the diagram. For instance on the below screen the *chance node* panel is displayed as this is the element currently selected on the tree graph.



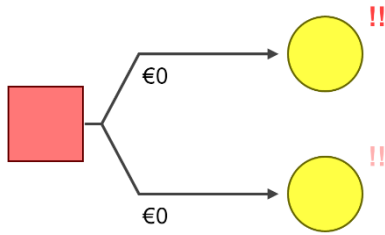
5.3 MODELLING DECISION TREES WITH SILVERDECISIONS

In this section we describe stand alone SilverDecisions application. However, all functionality is also integrated into SPOD as a datalet.

Getting started with SilverDecisions

Go to <http://www.silverdecisions.pl/> and Run the application.

To start creating the tree simply press right mouse button anywhere in the white plotting canvas. Choose *Add Decision Node* option from the context menu and you will see the first node on the canvas. Then add two *chance nodes* by right-clicking on the *decision node* and choosing *Add Chance Node* option from the context menu. Now you should see the following graph:



In the next step try to add two *terminal nodes* for each chance node in an analogous way. When you finish, left-click on the first node - the Decision node dialog has appeared on the left. *Decision node* label, as well as *edges'* labels and payoffs may be changed there. Then left-click on any of the *chance nodes* to see its dialog on the left. Right now you can see a new option for edges to be changed: it's probability. The decision tree you have just created will find the optimal decision path as soon as all the necessary parameters are set.

Fully detailed guide on creating decision trees with SilverDecisions including some advanced options as well as application functionalities may be found further in this documentation.

If you are a mobile/tablet user please go to [Mobile/Tablet support](#) first to see how to start with SilverDecisions, as some basic actions may be different from the ones on the PC.

Table of contents

1. Decision tree model
2. Create your first decision tree with SilverDecisions
3. Actions supported by application
4. Application settings
5. Usage tricks and known issues

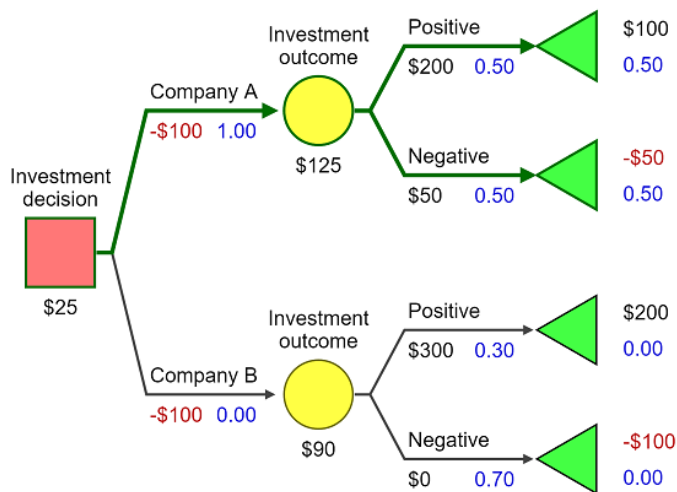
5.3.1 DECISION TREE MODEL

What is a decision tree?

Sequentiality and uncertainty are inherent in practical decision making. The former means that decision makers have to consider multi-staged strategies, encompassing several actions following one another, rather than just a single action. The latter means that decision makers' payoffs depend not only on actions but also on exogenous events (states of the world), which may often be perceived as random. The actions and reactions are usually intertwined, further complicating the picture. Decision trees are used as a model that helps to discover, understand and communicate the structure of such decision problems.

A simple decision tree created with SilverDecisions is presented below (you can run the SilverDecisions file containing this tree [here](#)).

A Simple Investment Decision Model



Decision tree model

A decision tree model describes and visualizes *sequential* decision problems under *uncertainty* in a tree-like diagram. This means that decision trees may be useful in such problems:

- the decision maker takes several actions following one another,
- the states of the world may differ based on the decisions that have already been made,
- some decisions may result in more accurate probability estimates of those states.

The tree-like diagram presents possible decisions to be made, independent events that may happen, and the outcomes associated with combinations of those decisions and events. There are two parameters that must be determined: the probabilities of events and the values. The former represents the probability that the specified state of the world happens. As the possible states of the world within one reaction are in fact competing events, the sum of their probabilities must be equal to 1. Then, *the values* stand for payoffs as the consequence of a decision or a state of the world. It might be either profit or loss. The decision tree model includes one more concept: expected value (or expected utility). It is calculated as the probability-weighted average of the values for competing decision-and-event sets. The expected value indicates how much one may earn or lose by making optimal decisions (this means such decisions that maximize gains and minimize losses). Finally, the *outcome associated with decisions and events* represents the total consequence of a set of decisions-and-events in the whole decision process. It might be interpreted as a decision maker's payoff - the result of both his decisions and the independent events that have occurred.

Decision trees, with their easy to interpret structure, are excellent tools for decision analysis problems. They enable investigation of the possible decision outcomes and they help to choose between certain courses of action. A primary goal of the decision tree model is to determine the best possible decision, which represents the greatest payoff or the smallest loss.

Decision tree structure

A decision tree is constructed using a directed graph from left to right, with a set of nodes that split into three disjoint sets:

- decision nodes - typically represented as squares,
- chance nodes - represented as circles,
- terminal nodes - represented as triangles.

The leftmost node is called root node and it is the first decision node (first red square from the left - see *A simple investment decision model* above). In *decision nodes* it is the decision maker who makes the choice, i.e. in selecting exactly one of the branches emanating from this decision node. Those branches represent the set of available decision alternatives (actions). In a *chance node* (yellow circles - a sample tree above) each of the edges stemming from it - a reaction - is selected randomly with a given probability of event. *Terminal nodes* (blue triangles on a sample tree above) represent the outcome of a sequence of actions/reactions from the root node to that particular terminal node. The terminal node is the endpoint: no decisions can be made and no events may occur afterwards.

In the SilverDecisions application, the probabilities of events and the values associated with those events or decisions are defined in edges. The expected values calculated for every set of decisions-and-events are displayed in each decision/chance node, while the terminal nodes show the outcomes and the probabilities that one ends up in a specified terminal node.

Note that each edge is combined with two nodes: the one on the left, from which the edge comes out is called the parent node and the second one, on the right, is called the child node. Subtree is another term associated with decision trees - it represents that part of the tree which starts in any child node and each of them together with any descendants form a subtree. For instance, the subtree starting in the root node is the entire tree.

5.3.2 CREATE YOUR FIRST DECISION TREE WITH SILVERDECISIONS

In this section you will learn how to build, edit and interpret decision trees with SilverDecisions. You will also see how to change the diagram layout and other more advanced options. Finally, you will be guided through the import and export options including saving the tree to disk in various formats and then loading it into the application again.

If you are a mobile/tablet user please consult the [Mobile/tablet support](#) section first, as there are some differences in application functionality between mobiles/tablets and PCs.

Before you begin

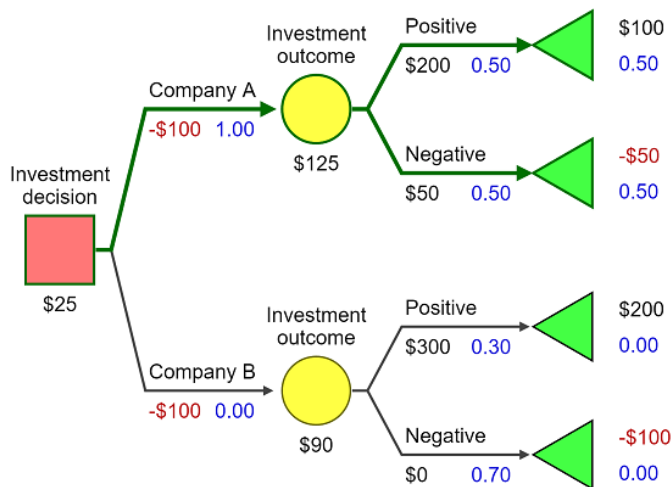
The application runs in the browser. Simply go to [SilverDecisions site](#) to run it.

The software has been tested under Windows, Linux and OS X and should work across all these platforms. Although the default browser we develop for is Chrome, it should also work in other major browsers (Firefox, Safari, Opera, IE, Edge). Also the application works on mobile phones and tablets under Android with Chrome (but in iOS). If you find a bug within your browser please file an [issue](#). Current problems with application functionality may be found in the [Known issues](#) section.

How to edit the tree

Here we outline the basic steps that lead to creation of *A Simple Investment Decision Model* tree presented below (the SilverDecisions file containing this tree may be found [here](#)).

A Simple Investment Decision Model



1. When you start the application, the plotting canvas is empty.
2. Press the right mouse button anywhere in the plotting canvas. The context menu will pop up. Select the *Add Decision Node* option. Note that the context menu may also be opened by double-clicking anywhere on the canvas.
3. Observe that there are **!!** marks at the top-right side of the decision node. This means that the tree is not proper (it is not allowed to have a decision node without child nodes).
4. Select a decision node by left-clicking on it. In the left panel, where the *Decision Node* dialog appears, enter the text "Investment decision" into it. Note that you can press **Enter** after the word "Investment" to add a new line to the text.
5. Now let us add the tree title. Left-click on the Details option in left panel and enter the text "A Simple Investment Decision Model" in the title box. A title of the decision tree appears in the plotting canvas. Observe that the decision node we have edited has now become deselected and its properties dialog has disappeared from the left panel.
6. In the next step we wish to add two chance nodes as children of the decision node. Child nodes may be added by either left-clicking on the decision node again and pressing **Ctrl-Alt-C** twice or right clicking on the decision node and selecting *Add Chance Node* option from the context menu.
7. Once you have understood how to add and edit the nodes of the tree, the next step is to learn how to edit the edges. There are two ways to achieve this. Note that the decision node still remains selected. Therefore, in the left panel you can see the list of all edges coming from it. You can edit their payoffs and labels. Type the label "Company A" and the payoff "-100" in the *Edge #1* boxes. Then complete the *Edge #2* boxes also. Note that you may use **TAB** to move to the next box. The other way to edit edge properties is to simply left-click on the edge you wish to edit. It then becomes highlighted and in the left panel an *Edge* dialog appears. The properties of this edge can be modified there.
8. Finally, you can finish creating the tree presented in this guide by adding the remaining terminal nodes and assigning proper values to the payoffs and probabilities of the edges:
 - As soon as you add terminal nodes to chance nodes you will see that the probability value on the left panel for any edge coming out of the chance node is by default set to the **#** sign. On the other hand, the probability displayed on the diagram is equal to 0.5. **#** is the default parameter for specifying probabilities: it is automatically assigned to each edge originating from the chance node. Its exact value is calculated in such a way that the probabilities of all the edges coming from a given chance node sum

up to 1. In our example, there are two edges coming from the chance node, so the probability for each of them is computed to be 0.5. You may of course enter your own values into the probability boxes. But if the # parameter is not replaced by the user's value, it is automatically computed and its calculated value is displayed on the diagram. On the other hand, on the panel there will be # sign as long as it is not replaced by the user.

- Note that arithmetic expressions are allowed when specifying probabilities and payoffs. For instance, instead of typing 200 as the payoff for the first terminal node from the top, the expression $4*100-200$ may be entered. Similarly, as for the # sign, expressions are presented in the boxes in the left panel, but computed values are displayed on the diagram.
- To check if the value on the diagram is a number, # sign, or expression, one may select the proper node/edge and look at the left panel or, alternatively, move the mouse cursor over the probability/payoff value on the diagram. More information about the probability and payoff format as well as on # parameter may be obtained [here](#).

This is all you need to know in order to start using SilverDecisions. You will be able to learn all the advanced options of SilverDecisions by reading the descriptions below and subsequent sections of this documentation (there are a lot of options that speed up editing of the tree: copy/paste functionality, keyboard shortcuts, undo/redo, changing the look-and-feel of the tree, saving it as a JSON file or image, changing the layout of the tree, etc.).

How to interpret the output of the tree optimization

SilverDecisions supports 3 decision making rules and *maximization of expected value of decisions* is the default one. When you look at the tree, if it is proper then the optimal path in the tree is highlighted in green by default. In our example above it is the upper branch.

Under each decision and chance node of the tree there is a number representing the expected value of the payoff in this node. It is calculated by collecting all the expected values from the node's subtree under the assumption that the optimal decision is taken. For instance, in the leftmost decision node we have \$25, which means that under the optimal decision we can expect to obtain this value in our problem. Similarly, a value of \$90 under the lower chance node means that the expected payoff from the subtree starting in this node is 90.

Terminal nodes are annotated in a slightly different way. On their right there are two values: the higher is an aggregate *payoff* of the whole path of the tree, starting from the root, that would be collected if we end up at a given node (e.g. look at the third node from the top on our example tree: \$200 would be the payoff if one terminates at this node). The lower value is the probability that we will actually finish at this particular terminal node. For instance, the probability of 0.00 in the third node from the top means that this is not the optimal path (it will never be selected because the upper branch of the tree is preferred). On the other hand, the probabilities in the upper two terminal nodes are positive as they lie on the *optimal path*.

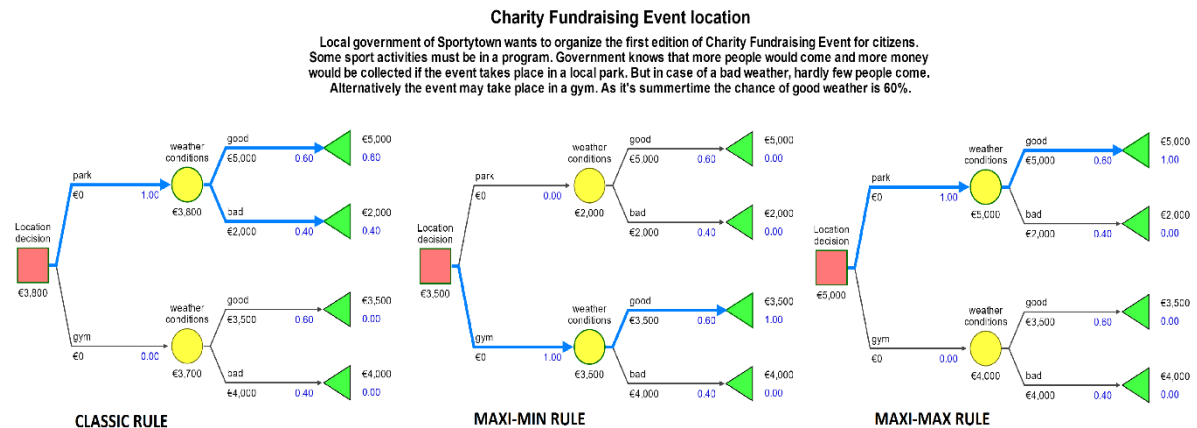
Finally, there are the values on the edges. They represent respectively the payoff of selecting the given edge and the probability that the edge will be selected from its parent node. These values can be edited in the left panel when an appropriate chance node/edge is selected. On the other hand, the values in decision nodes represent the result of an optimization process and obviously they cannot be directly edited.

Note:

- Payoffs and probabilities are always displayed on the tree diagram as numbers representing computed values (e.g. the result of arithmetic expressions or computed probabilities values for # parameters - both values are displayed as numbers). On the other hand, arithmetic expressions or the # sign may either be present in the left panel, when an appropriate node/edge is selected, or on the tree diagram after moving the mouse cursor over the probability/payoff value.
- If there are several branches coming out of a decision node that are equally good (have the same expected value) *all of them* will be highlighted in green by default as optimal and *all of the edges* coming from this decision node that are on the optimal path will have their probability set to 1.00.

The decision making algorithms

There are 3 decision making criteria available in SilverDecisions, which can be chosen in the Toolbar RULE section. **Classic** is a default rule based on expected value maximization. Two additional available criteria are based on a pessimistic **maxi-min** approach (in each chance node the *lowest* payoff gets chosen) or an optimistic **maxi-max** approach (in each chance node the *highest* payoff gets chosen). Below you will find 3 tree diagrams representing the same decision problem, but they have been solved based on different algorithms. The first one from the left is a *classic* criterion solution, the tree in the middle was solved using the *maxi-min* rule and the next one was solved using the *maxi-max* rule. The optimal path has been highlighted in blue.



It is readily apparent that the optimal path may be completely different, depending on the chosen decision making criterion.

How to change the layout of the tree

Once you know how to build the tree and how to interpret it, the next step is to take a look at the layout options. There are three layout options on the right of the toolbar: *manual*, *tree* (default) and *cluster*. Click on the first one. At this point you are able to manually place the nodes. Just left-click on any node, hold the button and move that node. Then, let us try to change the position of the whole tree. Place the mouse cursor somewhere on the canvas, click and hold down the left button of the mouse to select the entire tree. When the entire graph is selected, the colors of its nodes become darker. Now, just left-click on any node, hold the button and move the tree. Note that you are allowed to position the nodes and the tree only if the **MANUAL** layout is selected. **TREE** and **CLUSTER** are automatic layouts of the tree nodes aligned to the left or to the right respectively. By changing the layout again to **TREE** or **CLUSTER**, your decision tree will be automatically aligned once more.

Some other layout options may be found on the left in the *Layout* panel. Here you can change the horizontal and vertical margin by simply left-clicking on the slider and moving it to the left or right. If you move the *horizontal margin* slider to the right you will see that the left canvas margin gets wider. Try moving the *vertical margin* slider to the right - see how the top margin gets wider. Within the layout panel, the *node size* and *edge slant* can also be controlled. By moving the *node size* slider you can scale the nodes and by moving the *edge slant* slider you may control the maximum slant for plotting the sloping part of the edge. For the **TREE** and **CLUSTER** layout modes, you can also see the tree height and width settings in the *Layout* panel on the left.

Options regarding numerical formatting, currency type and colors can be found in the [Settings](#) panel in the toolbar.

Tree export and import

The first few toolbar actions correspond to creating a new diagram (note that this option clears the whole diagram), opening an existing diagram or saving the current one. By clicking on Open an existing diagram, you can load a previously created tree from your disk. Note that only those files saved in [JSON](#) format may be loaded

into the application. To save your tree, just click on the Save current diagram option - the tree is then automatically saved to disk in JSON format to a download folder specified in your browser. The next toolbar panel enables export of the graphical diagram to [PNG](#), [SVG](#) or [PDF](#) format. Note that the tree is saved in tree metadata in its layout mode and with its layout parameters.

In this section you have learned how to work with SilverDecisions. For further detailed information about all the application options and functionality please proceed to the following pages.

5.3.3 ACTIONS SUPPORTED BY APPLICATION

If you are a mobile/tablet user, please consult the [Mobile/tablet support](#) section first, as there are some differences in application functionality between mobiles/tablets and PC.

Mouse and keyboard actions

Mouse actions:

- left mouse button: node and edge selection (you can select multiple objects by region-selection: just left-click on the canvas and move the mouse)
- right mouse button: context menu (adding/manipulating nodes, copy-paste, adding text box, selecting/flipping the subtree)
- left mouse double click: context menu (adding/manipulating nodes, copy-paste, adding text box, selecting/flipping the subtree)

Keyboard shortcuts

- Del: delete selected nodes
- Ctrl-C/X: copy/cut selected nodes
- Ctrl-V: paste copied nodes as a subtree of a selected node
- Ctrl-Y/Z: undo/redo
- Ctrl-Alt-D/C/T: add new Decision/Chance/Terminal subnode of a selected node

On OSX also Ctrl should be used (not Cmd).

Actions in the left panel

Layout

- Horizontal margin: set margin from the left of the canvas
- Vertical margin: set margin from the top of the canvas
- Node size: set the node scaling
- Edge slant (max): set the maximum slant for plotting the sloping part of the edge
- Width: horizontal separation of tree nodes (shown only for TREE or CLUSTER layout mode)
- Height: vertical separation of tree nodes (shown only for TREE or CLUSTER layout mode)

Details

- Title: type the title of the diagram
- Description: type a description for the diagram - it is displayed right under the title in a smaller font size

Both of them can be made to span multiple lines by pressing **enter**.

Decision/Chance/Terminal Node

Shown only if a node is selected

- Decision Node
 - Label: type the *decision name* (span multiple lines by pressing **enter**)
 - Connections: Labels and payoffs of specified edges (*shown only if some edges emanate from the decision node*)

- Chance Node
 - Label: type the *event name* (span multiple lines by pressing **enter**)
 - Connections: Labels, payoffs and probabilities of specified edges (*shown only if some edges emanate from the chance node*)
- Terminal Node
 - Label: type the *endpoint name* (span multiple lines by pressing **enter**)

Edge

Shown only if an edge is selected

- Label: type *action name* or *reaction name* (span multiple lines by pressing **enter**)
 - Payoff: type the value of the specified action/reaction
 - Probability: type the probability of the specified reaction (*shown only for chance nodes*)
- Note that arithmetic expressions are allowed as probability and payoff inputs. The **#** parameter is a defaulted input for probabilities. For more information please read the description [below](#).

Floating text

Shown only if floating text is selected.

- **Text:** any text can be entered into this dialog. Note that text can be made to span multiple lines by pressing **enter**

Toolbar actions

- New diagram: clears canvas and starts an empty new diagram
- Open existing diagram: loads existing diagram from disk
- Save current diagram: saves current diagram to disk in [JSON](#) format
- PNG: saves current diagram in [PNG](#) format
- SVG: saves current diagram in [SVG](#) format
- PDF: saves current diagram in [PDF](#) format
- MANUAL: allow manual positioning of tree nodes
- TREE: default, automatic layout of tree nodes aligned to the left
- CLUSTER: automatic layout of tree nodes aligned to the right
- RULE: decision making criterion can be selected from the following options:
 - classic: expected value maximization rule
 - maxi-min: rule based on the worst-case scenario, pessimistic approach: in each chance node the lowest payoff gets chosen
 - maxi-max: rule based on the best possible scenario, optimistic approach: in each chance node the highest payoff gets chosen.
- Undo: undo last action
- Redo: redo action
- Settings: open diagram settings dialog box
- About: open dialog box containing concise information about SilverDecisions

Probability and payoff input format

- **# probability parameter**

The **#** parameter is used for specifying probabilities. By default, any edge emanating from the chance node has a probability set to **#**. The exact value of **#** is calculated in such a way that the probabilities of all the edges coming out of a given chance node sum up to 1.

The aim of the **#** parameter is to ensure in a simple way that probabilities in a chance node sum up to 1. In general: assume that for a given chance node some of the edges originating from it have a specified probability (e.g. 0.1, 0.2 and so on) which sum up to **p**. The remaining **m** edges have a default probability, **#**, which is calculated as $(1-p)/m$. So each occurrence of **#** will be replaced by the value equal to $(1-p)/m$.

Take the following example: assume that there is a chance node, to which two terminal nodes are added. The probability for every edge is automatically set to the **#** parameter, which is calculated as $1/2$ (the probability of a given chance node divided by the number of edges coming from it). Try adding two more terminal nodes: the probability for each edge is recalculated and its value is now set to 0.25 (1 divided by 4 edges). Now let us try to set a probability of 0.4 for one of the edges - the rest of them with probability **#** will be automatically recalculated to $(1-0.4)/3 = 0.2$ (1 decreased by probability 0.4 and divided by 3 edges).

- **Arithmetic expressions**

Arithmetic expressions are allowed when specifying probabilities and payoffs. For instance, one may type $(100+200)/5*0.3$ instead of 18 into any payoff's box.

- **Values displayed on the tree diagram**

Note that on the tree diagram only computed values (numbers) are displayed (e.g. the calculated value for **#** parameter or the result of an arithmetic expression). Arithmetic expressions as well as **#** signs are presented in the boxes in the left panel. To check if the value on the diagram is a number, **#** sign, or expression, one may select the proper node/edge and look at the left panel or, alternatively, move the mouse cursor over the probability/payoff value on the diagram - a pop-up will display the information from the left panel's box.

Linking to your decision tree

SilverDecisions provides a mechanism to share your decision tree with others through an interactive link. Please look at the examples below:

- open an editable tree [click](#)
- open a read-only (i.e. not editable) tree [click](#)

In order to accomplish such functionality the following steps need to be taken:

- create a decision tree
- export the decision tree as a JSON file (use the *Save current diagram* option)
- upload the JSON file to the Internet. In this example, let's assume that the uploaded tree is available at <http://mydomain.com/trees/tree.json>
- Construct a link with the following items:
 - <http://www.silverdecisions.pl/SilverDecisions.html>
 - `?LOAD_SD_TREE_JSON=`
 - address to the JSON file (e.g. <http://mydomain.com/trees/tree.json>)

Finally, the link should look as follows:

http://www.silverdecisions.pl/SilverDecisions.html?LOAD_SD_TREE_JSON=http://mydomain.com/trees/tree.json

Now the link can be shared with other people interested in the same decision tree.

If you want the link to be read-only, append it with `&readonly=True`. In the above example, the link will look as follows:

http://www.silverdecisions.pl/SilverDecisions.html?LOAD_SD_TREE_JSON=http://mydomain.com/trees/tree.json&readonly=True

Please note that while the user may edit the tree, they cannot save it on your web page. They can only save the tree locally on their own devices. If they wish to share it with others, they again need to follow the steps above (i.e. upload the file to some hosting website and reconstruct the link).

5.3.4 APPLICATION SETTINGS

Language selection

Currently, SilverDecisions is available in 2 languages: English and Polish. The language setting can be selected within the application [home page](#).

Settings panel overview

When you click the **SETTINGS** button in the top right corner, the following options are available:

- In the General section, the font family and size, in addition to the number format locale, may be changed by typing values into the appropriate boxes. Try typing 'Arial black' to *font family* box and '9px' to *font size* box. Note that the text on the tree will have changed. *Number format locale* requires typing the *language tag* to adjust to the desired format. *Font weight* and *font size* may be changed by selecting one of the options from the context menu.
- In the following two sections the *Payoff number format* and *Probability number format* may be adjusted, including *number style*, *fractions digits* or *color*. For payoffs, you can additionally change the *grouping separators* and *currency* options, including the way the currency is displayed (symbol/code/name). Probability numbers may be displayed as decimals or percentages in different sizes or colors.
- The Node part enables setting of different graphical options for different nodes: decision ones, chance ones and terminal ones. Moreover, a unique stroke width and color can also be adjusted for all the nodes on the optimal path.
- In the Edge section, analogous modifications can be made.
- The Diagram title section includes the font, margin and subtitle options.
- The Other options allow you to control how the tree is repainted (their main use is on system/browser configurations that have problems with correct handling of JavaScript repaint events)

Choosing language locale for number formatting

In SilverDecisions the numerical formatting is configured by the choice of locales. More specifically, [Intl](#) object is used by the web browser to format numbers with a given locale. A full [reference for available locales](#) and hence we provide a table below for your convenience

COUNTRY	LANGUAGE TAG	SAMPLE FORMATTING
United States (default setting)	en	2,121,000.25
Germany	de	2 121 000,25
France	fr	2 122 000,25
Ireland	en-IE	2,121,000.25
Italy	it	2.121.000,25
Netherlands	nl	2.121.000,25

COUNTRY	LANGUAGE TAG	SAMPLE FORMATTING
Russian Federation	ru	2 121 000,25
Poland	pl	2 121 000,25
United Kingdom	en-UK	2,121,000.25

Please note that currency formatting must be configured independently of the chosen language tag. Moreover, *number format locale* settings refer only to the text format displayed on the tree graph. In the left panel, a dot `.` is the expected decimal separator regardless of the chosen standard locales.

Floating text

Notes on the diagram can be made in floating text boxes, which may be added by selecting an appropriate option from the left-click/double-click context menu. Text can be made to span multiple lines by pressing `enter` and the text box can be placed anywhere in the canvas. Note that the text field can only be selected by clicking on it (region selection is not supported in tandem as it is not possible to select nodes AND text fields). Moreover, copy/paste of text fields is not supported either.

Settings panel

- **General**
 - Font family: typeface that will be applied to the text (on the diagram and in the title/subtitle); *sans-serif* by default.
 - Font size: size of the whole text on the diagram; *12px* by default.
 - Font weight: 4 options may be chosen from the context menu: *normal* (default), *bold*, *lighter* and *bolder*
 - Font style: 3 options may be chosen from the context menu: *normal* (default), *italic* and *oblique*
 - Number format locale: to adjust a given language format, *language tag* must be used - *en* is the default
- **Payoff number format**
 - Style: *currency* (default) or *decimal* can be chosen
 - Currency display: *currency symbol* (default), *code* or *name* can be chosen
 - Currency: a specified currency may be set by typing the currency code, *USD* by default (other typical currency codes are e.g.: *EUR*, *JPY*, *RUB*, *PLN*); display of the currency symbol depends on the chosen language locale
 - Minimum fraction digits: can be changed by typing the value or by clicking on the spinner on the right; *0* by default.
 - Maximum fraction digits: can be changed by typing the value or by clicking on the spinner on the right; *2* by default.
 - Use grouping separators: to remove the thousands separators, just uncheck the box
- **Probability number format**
 - Style: *decimal* (default) or *percent* may be chosen from the context menu
 - Minimum fraction digits: can be changed by typing the value or by clicking on the spinner on the right; *2* by default.
 - Maximum fraction digits: can be changed by typing the value or by clicking on the spinner on the right; *3* by default.
 - Font size: probability number font size may be set in [em unit](#); *1em* by default.
 - Color: probability number color can be chosen by clicking on the color stripe; *blue* by default.
- **Node**

- Stroke width: stroke width settings for all the nodes that do not lie on an optimal path; *1px* by default.
- Optimal
 - Stroke width: stroke width settings for all the nodes that belong to an optimal path; *1.5px* by default.
 - Color: color settings for all the node strokes that belong to an optimal path; *green* by default.
- Label
 - Label font size: label font size settings for all the nodes in [em unit](#); *1em* by default.
 - Label color: node label color; *black* by default.
- Payoff
 - Font size: payoff font size settings for all the nodes' in [em unit](#); *1em* by default.
 - Color: font color for positive payoffs in nodes; *green* by default.
 - Negative color: font color for negative payoffs in nodes; *red* by default.
- Decision node
 - Fill color: fill color for decision nodes; *red* by default.
 - Stroke color: stroke color for decision nodes; *brown* by default. Note that the stroke color for the optimal path may differ in accordance with the *optimal node* settings.
 - Selected fill color: fill color for selected decision nodes; *dark red* by default.
- Chance node
 - Fill color: fill color for chance nodes; *yellow* by default.
 - Stroke color: stroke color for chance nodes; *dark green* by default. Note that the stroke color for the optimal path may differ in accordance with the *optimal node* settings.
 - Selected fill color: fill color for selected chance nodes; *dark yellow* by default.
- Terminal node
 - Fill color: fill color for terminal nodes; *light green* by default.
 - Stroke color: stroke color for terminal nodes; *black* by default. Note that the stroke color for the optimal path may differ in accordance with the *optimal node* settings.
 - Selected fill color: fill color for selected terminal nodes; *dark green* by default.
 - Payoff
 - Font size: font size for terminal nodes' payoffs in [em unit](#); *1em* by default.
 - Color: font color for positive payoffs in terminal nodes; *black* by default.
 - Negative color: font color for negative payoffs in nodes; *red* by default.
- **Edge**
 - Color: color for all the edges; *black* by default. Note that the color of optimal edges may differ.
 - Stroke width: edge width settings; *1.5* by default. Note that the width of optimal edges may differ.
 - Optimal
 - Stroke width: optimal edges width settings; *2.4* by default.
 - Color: optimal edges color settings; *green* by default.
 - Selected
 - Stroke width: selected edges width settings; *2.4* by default.
 - Color: selected edges color settings; *blue* by default.
 - Label
 - Font size: font size for edge labels in [em unit](#); *1em* by default.
 - Color: color for edge labels; *black* by default.
 - Payoff
 - Font size: font size for edges' payoffs in [em unit](#); *1em* by default.
 - Color: color for positive edges' payoffs; *black* by default.
 - Negative color: color for negative edges' payoffs; *red* by default.
- **Diagram title**
 - **Font size:** title font size; *16px* by default.

- **Font weight:** 4 options may be chosen from the context menu: *normal*, *bold* (default); *lighter* and *bolder*.
- **Font style:** 3 options may be chosen from the context menu: *normal* (default), *italic* and *oblique*.
- **Color:** title color; black by default.
- **Margin**
 - Top: top margin settings; 15 by default.
 - Bottom: bottom margin settings; 10 by default.
- **Subtitle (diagram description)**
 - Show: to hide the diagram description, just uncheck the box
 - Font size: subtitle font size; 12px by default.
 - Font weight: 4 options may be chosen from the context menu: *normal*, *bold* (default), *lighter* and *bolder*.
 - Font style: 3 options may be chosen from the context menu: *normal* (default), *italic* and *oblique*.
 - Color: subtitle color settings; *black* by default.
 - Margin top: margin between a title and a subtitle; 5 by default.

5.3.5 USAGE TRICKS AND KNOWN ISSUES

Mobile/Tablet support

SilverDecisions mobile/tablet functionality was tested on Chrome and it should work on properly in this browser. Please bear in mind the following functionality disparities between the PC and mobile/tablet touch actions:

- **LONG PRESS** on the canvas: open context menu (corresponds to double left-click/right click on PC)
- **DOUBLE TAPPING** anywhere on the canvas/on the left panel/on the toolbar: zoom
- **LONG PRESS AND MOVE:** selection brush (corresponds to region selection)
- **MULTITOUCH:** zoom. Please note that when multitouch pinch is detected, the selection brush becomes momentarily disabled. Then it is possible to scroll or zoom the canvas.
- **TAP AND MOVE:** scrolling. Please note that panning/scrolling the canvas is possible only when the selection brush is disabled. This is indicated by a pop-up warning in the top left corner of the canvas.
- **Tips:**
 - Tap on the probability/payoff value on the diagram to see the pop-up showing whether the value is a number, arithmetic expression or a # sign.
 - To enable/disable the selection brush, just multi-touch (e.g. double-tap with two fingers).

Important note about saving files

Note that trees are saved in tree metadata with their layout mode and options. All the files are saved/exported to the default downloads repository set in your browser.

- **File name**

Saving the diagram and export to PNG/SVG/PDF format use the default browser settings for saving files. The default saved filename format is `decisiontree@YYYY.MM.DD_HH.MM.SS.[JSON|PNG|SVG|PDF]`. If you wish to change this name (*save as* action), it is necessary to adjust your browser settings, e.g.:

- Google Chrome: go to the *Settings/Advanced settings/Downloads section* - the box marked *Ask where to save each file before downloading* must be checked
- Mozilla Firefox: go to the *Options/General/Downloads section* and check the box *Always ask me where to save files*
- Microsoft Edge: go to *Settings/Advanced settings/Downloads section* and check the box *Ask me what to do* before each download

The diagram name may be changed when the *File download* dialog box appears after clicking on the **save** button.

- **PNG/SVG/PDF export format**

- SVG is the recommended export format as it guarantees the exact same image as seen in the browser and the fastest way of export.
- PDF export is performed via [external converter](#). It is not guaranteed that the exported diagram looks exactly the same as in the browser: fonts which are **not** on the PDF renderer server will be substituted by alternatives in the exported PDF file.
- Exporting preserves the selection: if the nodes or edges are selected at the time of exporting the diagram, they will also appear selected in the exported file.

Error control

- Parent nodes (decision nodes or chance nodes) without child nodes are marked with **!!**. This sign indicates that the subsequent (child) nodes should be added.
- Chance nodes with probabilities which do not sum up to 1 are also marked with **!!**.
- Incorrect number format: if the number in an input field is entered in an incorrect format (e.g. negative probability value or empty field left), a red underline appears in the input field but no other error is shown globally. Note that after exiting the node edit dialog, the last correct value (the one before the error) is retained.
- Errors within the settings panel are not indicated at all. If one enters an incorrect value in any of the settings fields, it is silently replaced by a default value or the last known correct value. Therefore, before making any settings changes, ensure that they are correct.

Selecting, copying and pasting the nodes

- It is possible to move the whole tree or any of its nodes by simply selecting them and moving the mouse cursor.
- To copy or paste the nodes you can use keyboard shortcuts (**ctrl+c/ctrl+v**) or choose the corresponding options from the right-click mouse context menu.
- Any selected node gets copied along with all its subsequent nodes. This means that by copying any parent node, the whole of its subtree gets copied.
- It is possible to copy and paste multiple disconnected subtrees.
- Any selected nodes or subtrees can be copied both to the canvas (as single nodes/subtrees) or to the diagram structure. To paste copied nodes onto the canvas just right-click anywhere on the canvas (nodes become deselected) and choose the *paste* option from the left-click context menu. Please note that the **ctrl+v** shortcut works only for pasting the nodes/subtrees to any part of the tree graph. To do this, just select an appropriate node and click *paste* or press **ctrl+v**. The copied subtree should be added to this node.

Probabilities and payoffs format

- Note that arithmetic expressions are allowed when specifying probabilities and payoffs - e.g. try typing $(100+200)/5*0.3$ into any payoff's box.
- By default, any edge emanating from the chance node has a probability set to the # parameter. Its value is calculated in such a way that the probabilities of all the edges coming from a given chance node sum up to 1. More information on the # parameter may be found in the [Actions supported by application](#) section.
- The# sign, as well as arithmetic expressions, can be seen in the left panel. On the diagram there are always values (numbers) displayed as the result of probability/expression computing. More information may be found [here](#)

Other tips & tricks

- By default the diagram has a TREE layout. It is recommended to construct and develop the tree in this layout. In turn, the MANUAL layout is preferred when final corrections are made.
- To add subsequent chance/decision nodes, one may right-click on the node and select a suitable option from the context menu or, alternatively, left-click on the node and use `ctrl + alt + c` for chance node or `d` for decision node keyboard shortcuts, which speeds up editing of the tree.
- In the same manner (by choosing a suitable option from the context menu or using key shortcuts), chance or decision nodes may be injected directly into an edge. To do so, a given edge must be *selected*.
- **UNDO** and **REDO** toolbar actions are also recommended in tree editing.
- The diagram layout mode, as well as the layout parameters (e.g. margins), are saved in the tree metadata.
- Note that only standard colors are available in the color palette settings

Known issues

- Sometimes it is reported that the application may periodically become unresponsive within Internet Explorer; if this situation persists on your machine, consider switching to an alternative browser.
- Under some configurations (eg. Windows 7 and Internet Explorer 11) there is a problem with repainting of the tree; in the event of this occurring, consider toggling the options *Settings/Other/Disable animations* and *Settings/Other/Force full redraw of edges*; if the problem remains unresolved, consider switching to an alternative browser.
- Under Linux the `Ctrl-Alt-T` shortcut may conflict with *open terminal*;
- The application does not support iOS (but you can edit the tree under Android with Chrome).

6 CONCLUSIONS

The objective of this document was to present a beta version of *open data governance model* (ODGM). ODGM implements the idea of *evidence based policy making* and is designed to improve two way communication of preferences and intentions between Public Administration (PA) and citizens and NGOs. This goal is achieved in two perspectives:

- 1) *PA to citizen*: to help PA to explain and visualize the rationale between decision processes that take place around Open Data on the social platform and to collaborate with citizens and NGOs to make those decisions better informed in terms of data and preferences taken into account;
- 2) *citizen to PA*: to provide information about citizen activity on the SPOD platform, and in particular to design an efficient system for elicitation of social preferences in heterogonous communities and in consequence also make better informed decisions.

For example the *social platform for open data* (SPOD) allows citizens to monitor allocation and spending of financial resources. On top layer the idea is that higher control level of citizens over PA will lead to increased efficiency. In short *PA will make better decisions because they know that they are closely observed*.

ODGM provides a more direct, and in our opinion, more effective mechanism of leading to increased efficiency of decisions. When PA shares open data with citizens a two-way exchange of information about preferences and intentions takes place and thus PA actions can be directly shaped by a participatory society. In short *PA will make better decisions because they know that what citizens want and citizens will better understand rationale behind policy of PA*.

The tools developed in SIM module have been already positively validated by scientific community and general public. The results of work on simulation module was presented at 2 national and 4 international conferences. On the other hand, the policy support module, developed to facilitate collaboration of PA, citizens and NGOs, since its initial public release on 14th Oct 2016, till 25th Jan 2017 attracted 5 138 unique application users (each user has usually several sessions of application usage).

ROUTE-TO-PA team makes sure that SIM deliverable is known and used by the target community not only via standard dissemination channels (conferences, websites) but also by also having introduced them to teaching curriculum at undergraduate and MBA courses.

7 BIBLIOGRAPHY

1. Acemoglu, D. and A. Ozdaglar, (2011) Opinion Dynamics and Learning in Social Networks, *Dynamic Games and Applications*, p.3-49.
2. Ashraf, Q., Gershman, B. and Howitt, P. (2011). Market Organization and Macroeconomic Performance: An Agent-Based Computational Analysis. NBER Working Paper No. 17102.
3. Axtell, R.L. (2007). What economic agent do: How cognition and interaction lead to emergence and complexity. *Review Austrian Economics*, 20, 105-122.
4. Barrat A., M. Barthelemy, A. Vespignani (2008) *Dynamical Processes on Complex Networks*. Cambridge University Press.
5. Barthelemy J., P. Toint (2012) Synthetic Population Generation Without a Sample, *Transportation Science*
6. Barthelemy J. and T. Suesse (2016). mipfp: Multidimensional Iterative Proportional Fitting and Alternative Models. R package version 3.1. <https://CRAN.R-project.org/package=mipfp>
7. Barton R. R., Metamodels for simulation input-output relations, in: J. Swain, D. Goldsman, R. Crain, J. Wilson (Eds.), (1992) *Proceedings of the 1992 Winter Simulation Conference*, IEEE, 1992, pp. 289-299.
8. Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society, ser. B*, 36, 192 - 225.
9. Bratley P., Fox B.L. (1988); Algorithm 659: Implementing Sobol's Quasirandom Sequence Generator, *ACM Transactions on Mathematical Software* 14, 88–100
10. Butts C. (2003) Network inference, error and informant (in)accuracy: a Bayesian approach, *Social Networks* 25, pp.103-140
11. Corander, J., Dahmström, K., and Dahmström, P. (1998). Maximum likelihood estimation for Markov graphs. Research Report 1998:8, Department of Statistics, University of Stockholm
12. Darley V., Outkin A. (2007), *Nasdaq Market Simulation: Insights on a Major Market from the Science of Complex Adaptive Systems*, World Scientific Publishing Company.
13. DeGroot, M. H. (1977) Reaching a Consensus. *Journal of the American Statistical Association*, 69, 118–121
14. Dahmström, K., and Dahmström, P. (1993). ML-estimation of the clustering parameter in a Markov graph model. Stockholm: Research report, Department of Statistics.
15. Diao S-M., Y. Liu, Q-A Zeng, G_X Luo and F. Xiong, (2014) A novel opinion dynamics model based on expanded observation ranges and individuals' social influences in social networks *Physica A*, 220-228
16. Dosi, G., Fagiolo, G. and Roventini, A. (2006). An Evolutionary Model of Endogenous Business Cycles. *Computational Economics*, 27(1), pp. 3-34.
17. Fagiolo, G. 1998. Spatial interactions in dynamic decentralized economies: a review. In: P. Cohendet, P. Llerena, H. Stahn, and G. Umbhauer, ed., *The Economics of Networks: Interaction and Behaviours*, Springer Verlag, Berlin - Heidelberg.
18. Fagiolo, G., Windrum, P. and Monetaz, A. (2007). A Critical Guide to Empirical Validation of Agent-Based Economics Models: Methodologies, Procedures, and Open Problems. *Computational Economics*, 30(3), pp. 195-226.
19. Farine D., Strandburg-Peshkin (2015), Estimating uncertainty and reliability of social network data using Bayesian inference, *Royal Society for Open Science*
20. Farmer D.J., D. Foley (2009), The economy needs agent-based modelling. In *Nature*, vol. 460, pp. 685-686.

21. Fischhoff B., Manski C. F. (2000). Elicitation of Preferences. In *Journal of Risk and uncertainty*, vol. 19: 1-3.
22. Frank O. (1974) Survey sampling in graphs. *Journal of Statistical Planning and Inference*, vol. 1, pp. 235–64
23. Frank O., (1981), Survey of Statistical Methods of Graphs Analysis, *Sociological Methodology*
24. Frank, O. (1991). Statistical analysis of change in networks, *Statistica Neerlandica*, 45, 283 { 293.
25. Frank, O. and Strauss, D. (1986). Markov graphs. *Journal of the American Statistical Association*, 81, pp. 832 –842.
26. Frick M., K. Axhausen (2004), Generating Synthetic Populations Using IPF and Monte-Carlo Techniques: Some new Results, Conference Paper STRC 2004
27. Gaffeo, E., Gatti, D.D., Desiderio, S. and Gallegati, M. (2008). Adaptive Microfoundations for Emergent Macroeconomics, *Eastern Economic Journal*. 34(4), pp. 441-463.
28. Gajdos T., Tallon J.M., Vergnaud J. C. (2008). Representation and Aggregation of Preferences under Uncertainty. In *Journal of Economic Theory*, vol. 141, iss. 1, July 2008, pp. 68-99.
29. Geman, S., and Geman, D. (1983). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 721 - 741.
30. Geyer, C.J., and Thompson, E.A. (1992). Constrained Monte Carlo maximum likelihood for dependent data. *Journal of the Royal Statistical Society, B* 54, 657 - 699.
31. Gilbert, N.: *Agent-Based Models*, SAGE Publications (2008)
32. Giovanni, D., Fagiolo, G., and Roventinic, A. 2010. Schumpeter Meeting Keynes: A Policy-Friendly Model of Endogenous Growth and Business Cycles. *Journal of Economic Dynamics and Control* Volume. 34(9), pp. 1748–1767.
33. Goodreau, S.M., Kitts, J.A. & Morris, M. *Demography* (2009) 46: 103. doi:10.1353/dem.0.0045
34. Guo J., C. Bhat, (2007) Population synthesis for microsimulating travel behaviour, *Transportation research record*
35. Handcock M., K. Gile (2010), Modeling Social Networks from Sampled Data, *The Annals of Applied Statistics*, Vol 4., No.1, pp.5-25
36. Handcock M, Hunter D, Butts C, Goodreau S, Krivitsky P and Morris M (2016), *ergm: Fit, Simulate and Diagnose Exponential-Family Models for Networks*, The Statnet Project (<URL: <http://www.statnet.org>>). R package version 3.6.0, <URL:<http://CRAN.R-project.org/package=ergm>>.
37. Hastings, W.K. (1970). Monte Carlo Sampling Methods Using Markov Chains and Their Applications. *Biometrika*. 57 (1): 97–109
38. Haung Z., P. Williamson (2001) A comparison of synthetic reconstruction and combinatorial optimization approaches to the creation of the small-area microdata, Working Paper 2001/2, University of Liverpool
39. Holland, P. W. and Leinhardt, S. (1981) An exponential family of probability distributions for directed graphs, *Journal of the American Statistical Association*, 76, 33–65.
40. Hongming Xi L., M. Yong D. (2015) An evidential opinion dynamics model based on heterogeneous social influential power. *Chaos, Solitons & Fractals* 73, 98–107
41. Hunter, D. R. and Handcock, M. S. (2006) Inference in curved exponential family models for networks, *Journal of Computational and Graphical Statistics*
42. Hunter DR, Handcock MS, Butts CT, Goodreau SM, Morris and Martina (2008). *ergm: A Package to Fit, Simulate and Diagnose Exponential-Family Models for Networks.*, *Journal of Statistical Software*, 24 (3), pp. 1-29.
43. Lazega, E., van Duijn, M., 1997. Position in formal structure, personal characteristics and choices of advisors in a law firm: a logistic regression model for dyadic network data. *Social Networks* 19, 375–397.

44. Kamiński B., (2015) Interval metamodels for the analysis of simulation Input–Output relations, *Simulation Modeling Practice and Theory*, 54, s. 86-100
45. Kamiński, B. (2012). Multi-agent approach for market modelling. Methods and applications (in Polish: *Podejście wieloagentowe do modelowania rynków. Metody i zastosowania*) SGH Warsaw School of Economics Press.
46. King, G., and L. Zeng. (2001). Logistic regression in rare events data. *Political Analysis* 9(2): 137-163.
47. Kirman A.P. (1992). Whom or What Does the Representative Individual Represent?. In *The Journal of Economic Perspectives Publication*, 6(2), pp.117-136.
48. Kirman, A.P. 1997. The Economy as an Interactive System. In: W.B. Arthur, S.N. Durlauf and D. Lane, ed., *The Economy as an Evolving Complex System II*, Santa Fe Institute, Santa Fe and Reading, MA, Addison-Wesley.
49. Kleijnen J. P., R. G. Sargent, A methodology fitting and validating metamodels in simulation, *European Journal of Operational Research* 120 (1) (2000) 14-29
50. Krause U., (2000), A Discrete Nonlinear and Nonautonomous Model of Consensus Formation, in *Communications in Difference Equations*, S. Elaydi, G. Ladas, J. Popena, and J. Rakowski (eds.) Gordon and Breach, Amsterdam, 2000.
51. Law, A.: (2006) *Simulation Modeling and Analysis*, McGraw-Hill
52. le Cessie, S., van Houwelingen, J.C. (1992). Ridge Estimators in Logistic Regression. *Applied Statistics*. 41(1):191-201
53. Leijonhufvud, A. 2006. Agent-based macro. In: L. Tesfatsion and K. Judd, ed., *Handbook of Computational Economics*, volume 2 of *Handbooks in Economics*, North Holland, Amsterdam, pp. 1625-1637.
54. Lengnick. 2013. Agent-based macroeconomics: A baseline model. *Journal of Economic Behavior & Organization*. 86, pp. 102–120.
55. Lenormand M., G. Deffuant (2013). Generating a Synthetic Population of Individuals in Households: Sample-Free vs Sample-Based Methods, *Journal of Artificial Societies and Social Simulation*
56. Leskovec, J. & Faloutsos (2006) C. Sampling from Large Graphs. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 631–636
57. Lorenz J., (2005) A Stabilization Theorem for Dynamics of Continuous Opinions, *Physica A*, vol. 355, pp. 217-223
58. Lovasz L.,(1993) *Random Walk on Graphs: A Survey*, Bolyai Society Mathematical Studies
59. Marsden, P. V. (1988). "Homogeneity in Confiding Relations." *Social Networks* 10:57–76.
60. Miller, J.H., Page, S.E.: *Complex Adaptive Systems*, Princeton University Press (2007)
61. Newman M., A-L Barabasi, and D. Watts (Eds.). *The Structure and Dynamics of Networks*. Princeton University Press. 2006.
62. Nooy W. A. Mrvar, and V. Batagelj. *Exploratory Social Network Analysis with Pajek*. Cambridge University Press. 2005.
63. Oechslein, C., Klügl, F., Herrler, R., Puppe, F.: (2002) UML for Behavior-Oriented Multi-agent Simulations, *Lecture Notes in Computer Science*, vol. 2296/2002 pp. 742-743
64. Oeffner Marc (2009) Agent - Based Keynesian Macroeconomics - An Evolutionary Model Embedded in an Agent-Based Computer Simulation. MPRA Paper No. 18199, posted 31. October 2009
65. Pattison P., G. L. Robins, T. A.B. Snijders, P. Wang, (2013), Conditional estimation of exponential random graph models from snowball sampling designs, *Journal of Mathematical Psychology* 57 (2013) 284–296
66. Pyka, A. and Fagiolo G. (2005). Agent-based modelling: A methodology for Neo–Schumpeterian economics. University of Augsburg, Discussion Paper Series No. 272.

67. Ribeiro D., B. Towsley (2010) Estimating and Sampling Graphs With Multidimensional Random Walks, IMC'10 Melbourne
68. Robbins, H., and Monro, S. (1951). A stochastic approximation method. *Annals of Mathematical Statistics*, 22, 400 - 407.
69. Santos I. R., P. R. Santos, Simulation metamodels for modeling output distribution parameters, in: S. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. Tew, R. Barton (Eds.), *Proceedings of the 2007 Winter Simulation Conference*, IEEE, 2007, pp. 910-918
70. Shang Y., (2014), Consensus Formation of Two-Level Opinion Dynamics, *Acta Mathematica Scientia*, 1029-1040
71. Smith J. A., M. McPherson, and L. Smith-Lovin, Lynn, (2014), Social Distance in the United States: Sex, Race, Religion, Age, and Education Homophily among Confidants, 1985 to 2004, Sociology Department, Faculty Publications. Paper 246. <http://digitalcommons.unl.edu/sociologyfacpub/246>
72. Snijders T., (2002) Markov Chain Monte Carlo Estimation of Exponential Random Graph Models, *Journal of Social Structure*, Vol. 3
73. Snijders, T.A.B., P.E. Pattison, G.L. Robins, and M.S. Handcock. (2006). New Specifications for Exponential Random Graph Models. *Sociological Methodology* 36:99–153.
74. Stivala A., J. H. Koskinen, D. A. Rollsa, P. Wang, G. L. Robins, (2016) Snowball sampling for estimating exponential random graph models for large networks, *Social Networks* 47, 167–188
75. Strauss, D. and Ikeda, M. (1990). Pseudolikelihood estimation for social networks. *Journal of the American Statistical Association*, 85, 204 –212.
76. Tesfatsion L. (2002) Agent-Based Computational Economics: Growing Economies From the Bottom Up. In *Artificial Life*, vol. 8, no. 1, MIT Press Journals, pp. 55-82.
77. Thompson S. K., O. Frank, (2000) Model-based estimation with link-tracing sampling designs, *Survey Methodology*, June 2000 , Vol. 26, No. 1, pp. 87-98 Statistics Canada, Catalogue No. 12-001
78. Wang G. G., S. Shan, Review of metamodeling techniques in support of engineering design optimization, *Journal of Mechanical Design* 129 (4) (2006), 370-380
79. Wang, Y. J., Wong, G. Y., (1987) Stochastic blockmodels for directed graphs, *Journal of the American Statistical Association* 82 , 8-19.
80. Wasserman S., K. Faust (1994) *Social Network Analysis: Methods and Applications*. Cambridge University Press. 1994.
81. Wasserman S., P. Pattison (1996), Logit Models and Logistic Regressions for Social Networks: I An Introduction to Markov Graphs and p^* , *Psychometrika* 61, 401-425
82. Windrum, P. (2005). Heterogeneous preferences and new innovation cycles in mature industries: the camera industry 1955-1974. *Industrial and Corporate Change*, 14(6), pp. 1043-1074.
83. Windrum, P., Fagiolo, G., and Moneta A. (2007). Empirical Validation of Agent-Based Models: Alternatives and Prospects. *Journal of Artificial Societies and Social Simulation* 10(2), 8.
84. Zhou X., B.Chen, L. Liu, L. Ma and X. Qju, (2015), An Opinion Interactive Model Based on Individual Persuasiveness, *Computational Intelligence and Neuroscience*

Appendix

8 APPENDIX A – FEEDBACK RECEIVED FROM PILOTS

8.1 ODGM DECISION MODELLING PLATFORM VISUALIZATION FEATURE PROPOSAL

This part presents proposal and results of discussion with pilots of considered tool for interactive presentation and visualization of Open Data decision process for public administration. The main assumption was that the tool will make it possible for the citizens and PA to graphically present decision problems considered on the SPOD platform.

8.1.1 CONCLUSIONS FROM DISCUSSION WITH PILOTS

ROUTE-TO-PA project Pilots (Prato & Dublin) have agreed that such approach will bring a new level of transparency to SPOD decision making. The PA can present variant scenario of decision situation and subsequently citizens can supply their estimates and opinions to enhance the decision making process. This visualization layer brings a new level to Open Data transparency and participative decision making by the community of SPOD platform users.

The overall aim of the SPOD platform is to allow to transparently communicate open data between PA and citizens. The holistic scope of open data covers:

- tabular-type structured data regarding the facts (e.g. historical expenditures, planned budget);
- information about planned decisions of PA and their consequences (e.g. possible options how the investment budget can be allocated and planned impact of these decisions on the community).

Currently SPOD platform contains a wide variety of means for visualization of tabular-type data (tables, graphs etc.). However, the only way how currently data about decision processes can be shared is by providing their textual description in an unstructured way. Moreover, the citizens can also add comments regarding the possible decisions and their consequences also using unstructured text.

It is well established that adding structure to the representation of decision processes improves the communication of the problem and facilitates discussion, e.g. by avoiding ambiguity of the unstructured textual description. A standard tool allowing for structuration of open data about decision processes are decision trees (https://en.wikipedia.org/wiki/Decision_tree).

Decision trees not only allow to visualize the open data about the decision making problem, but also to simulate the consequences the decision and give recommendation of the best course of action.

Adding a component allowing to visualize open data about decision making process using decision trees bring the following benefits for the communication between PA and citizens:

- PA can clearly communicate the structure and consequences of the open data about the decision making problem in a simple and structured way;
- Citizens can comment/correct the assumptions of the PA regarding the structure of the problem as well as about the possible consequences in a consistent way;
- The tool can simulate the overall consequences of the decisions (taking into account assumptions of PA and feedback of citizens) and give a recommendation of the best decision under various criteria for evaluation of the best alternative;

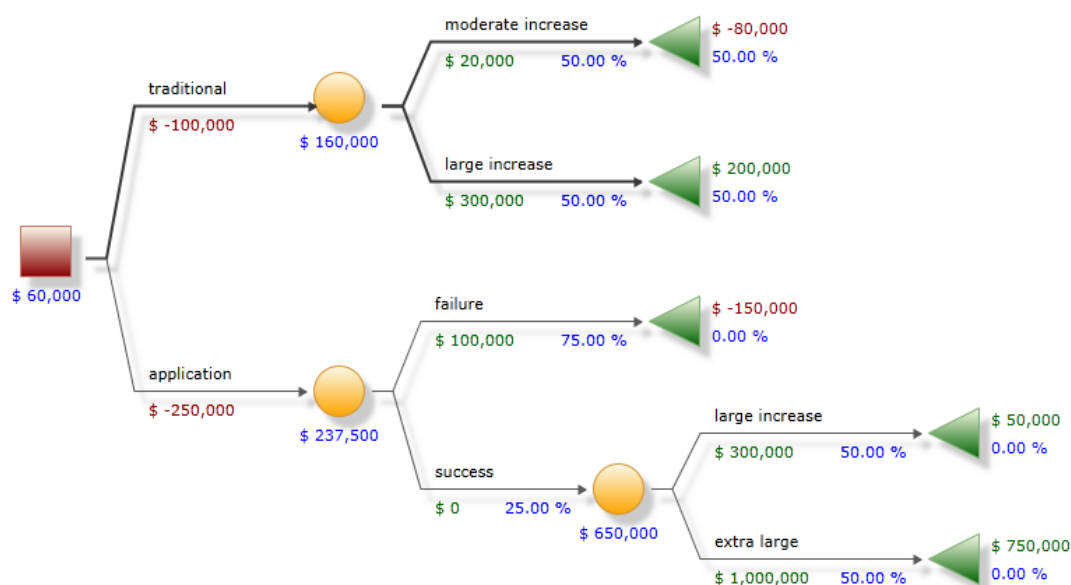
In the following section we provide example use cases of such a tool.

8.1.2 USE CASES SCENARIOS DISCUSSED WITH PILOTS

Below we describe the Open Data decision visualization platform by presenting 2 simplified use cases in decision making for local communities. Please note that the below discussion took place before the software development took place. Hence the decisions trees presented to the PAs have been drawn using Open Source product not developed within the ROUTE-TO-PA project

Case 1

The community would like to increase the number of tourists coming to the city. Two alternative strategies are considered. Firstly, the traditional means as commercials, leaflets etc can be used and would cost USD 100,000. The community estimates that it will increase the number of tourists by 5% (200 000 USD add. income) with probability 50% and by 25% (300 000 USD add. income) with probability 50%. Alternatively, the community may apply for the European Capital of Culture; the application cost are estimated at USD 250,000. The chances of success are 25%. Should the community fail, the number of tourist will increase only by 2% (100 000 USD add. income). In case the application is a success, the number of tourist will increase by 30% (300 000 USD add. income) with probability 50% or by 60% (1 000 000 USD add. income) with probability 50%.

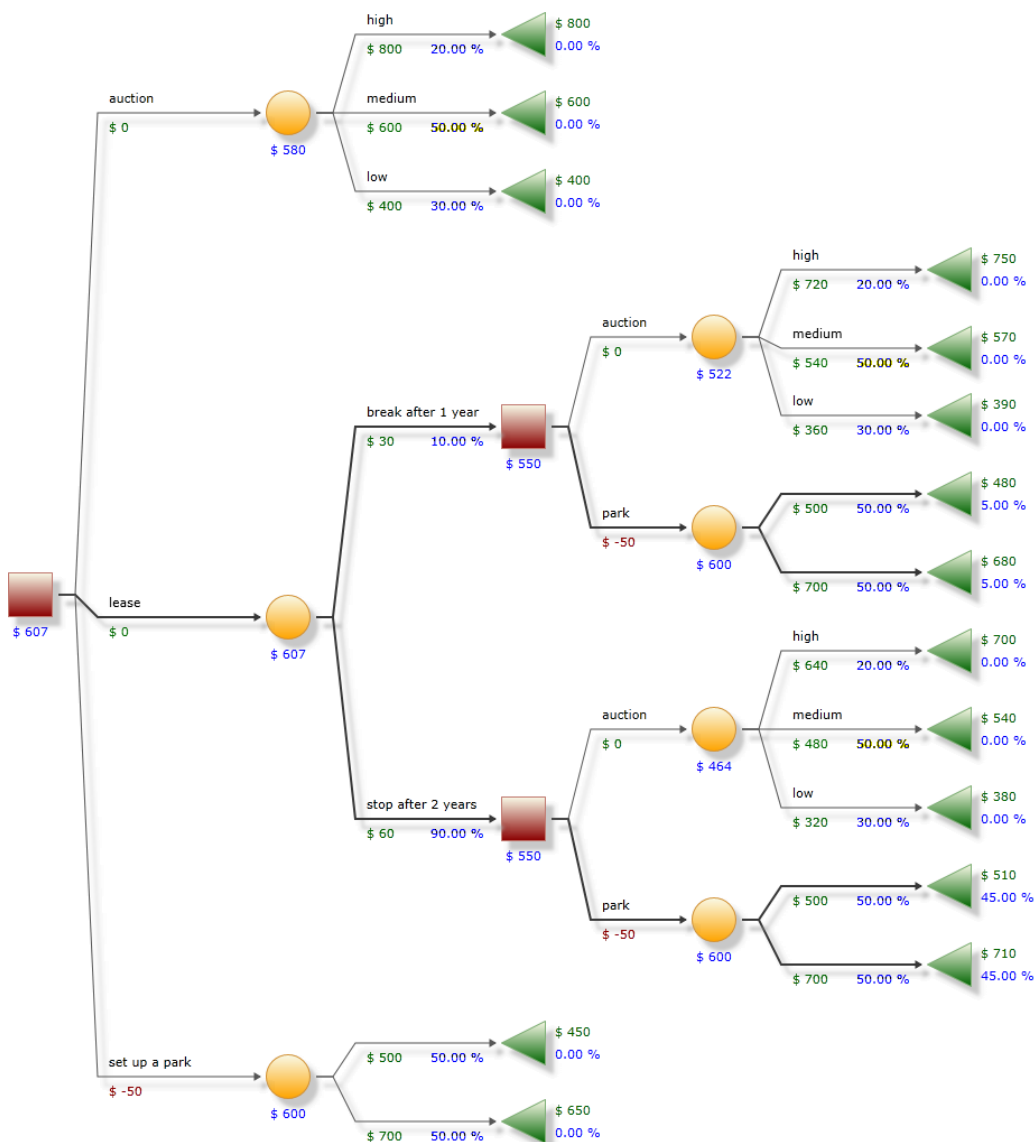


Case 2

Municipal authorities have a land plot available and they want to make the best use of it. There are three general alternatives available. Firstly, the authorities can set up an auction to sell the land. The analysis of past cases suggests that as high a price as EUR800,000 could be obtained with probability 20%, there is a risk (30%) of obtaining only EUR400,000, and it is most likely (50%) that the plot would be sold for EUR600,000.

Secondly, the authorities can lease the land for two years, for EUR40,000 per year. Standard contracts require that the lessee can break the contract after each year. The estimated risk of such an event is 10%, every year.

Thirdly, the authorities can set up a small children playground, at a cost of EUR50,000. The societal value of playground is either EUR500,000 (50%) or EUR700,000 (50%). After the lease has ended (perhaps prematurely) the authorities can set up an auction, identical to the one considered immediately (except for the fact that future earnings should be decreased by 10% per year due to observed market trends), or still set up a park.



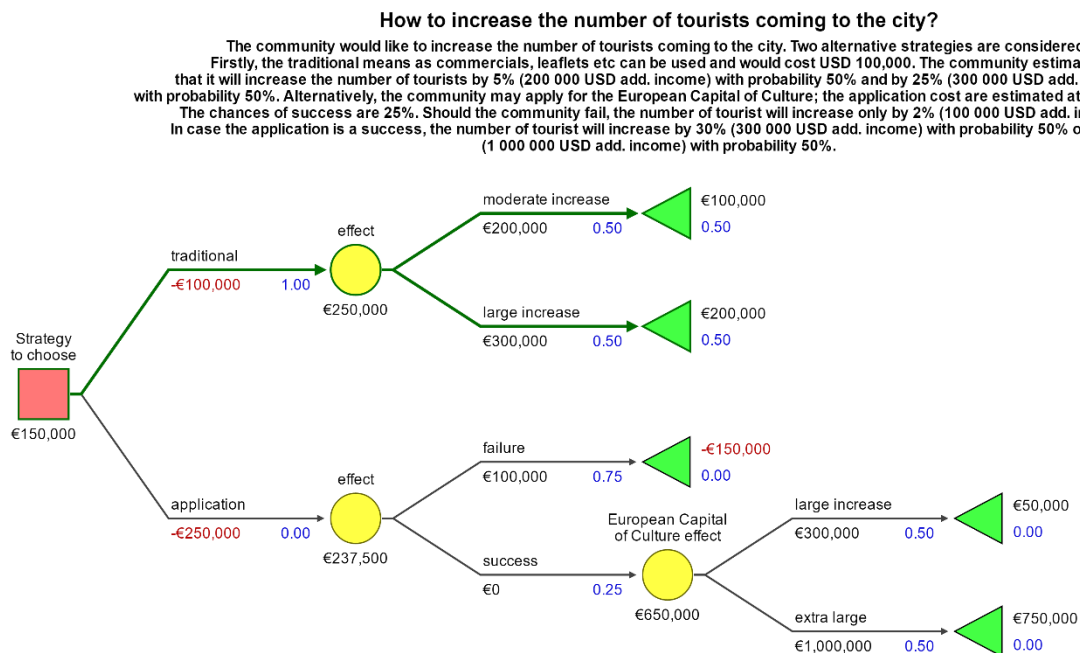
8.2 SCENARIOS FOR SUPPORTING PA IN PRESENTING DECISION MAKING PROCESS TO CITIZEN WITHIN THE ODGM MODEL

These are scenarios presented and discussed with the Pilots during the ROUTE-TO-PA project meeting on January 9th and 10th at the University of Salerno. The scenarios are illustrated with trees generated with the beta version of SilverDecisions software developed for the PA within the SIM module.

The below examples contain four different use cases for modelling Open Data decision making in the Public Administration.

CASE I – How to increase the number of tourists coming to the city?

The community would like to increase the number of tourists coming to the city. Two alternative strategies are considered. Firstly, the traditional means as commercials, leaflets etc. can be used and would cost USD 100,000. The community estimates that it will increase the number of tourists by 5% (200 000 USD add. income) with probability 50% and by 25% (300 000 USD add. income) with probability 50%. Alternatively, the community may apply for the European Capital of Culture; the application cost are estimated at USD 250,000. The chances of success are 25%. Should the community fail, the number of tourist will increase only by 2% (100 000 USD add. income). In case the application is a success, the number of tourist will increase by 30% (300 000 USD add. income) with probability 50% or by 60% (1 000 000 USD add. income) with probability 50%.

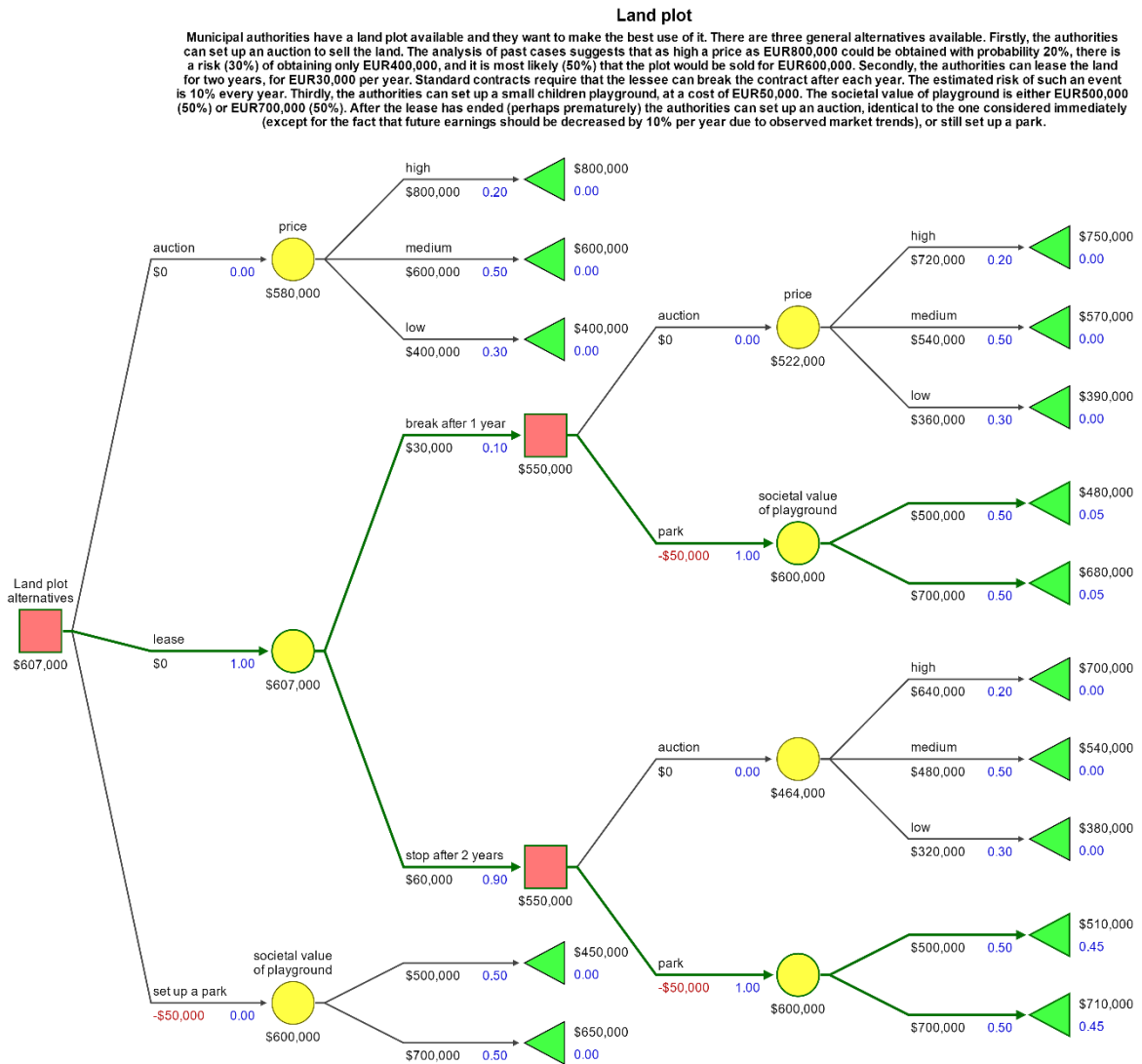


CASE II – Plot of land

Municipal authorities have a land plot available and they want to make the best use of it. There are three general alternatives available. Firstly, the authorities can set up an auction to sell the land. The analysis of past cases suggests that as high a price as EUR800,000 could be obtained with probability 20%, there is a risk (30%) of obtaining only EUR400,000, and it is most likely (50%) that the plot would be sold for EUR600,000.

Secondly, the authorities can lease the land for two years, for EUR30,000 per year. Standard contracts require that the lessee can break the contract after each year. The estimated risk of such an event is 10%, every year.

Thirdly, the authorities can set up a small children playground, at a cost of EUR50,000. The societal value of playground is either EUR500,000 (50%) or EUR700,000 (50%). After the lease has ended (perhaps prematurely) the authorities can set up an auction, identical to the one considered immediately (except for the fact that future earnings should be decreased by 10% per year due to observed market trends), or still set up a park.

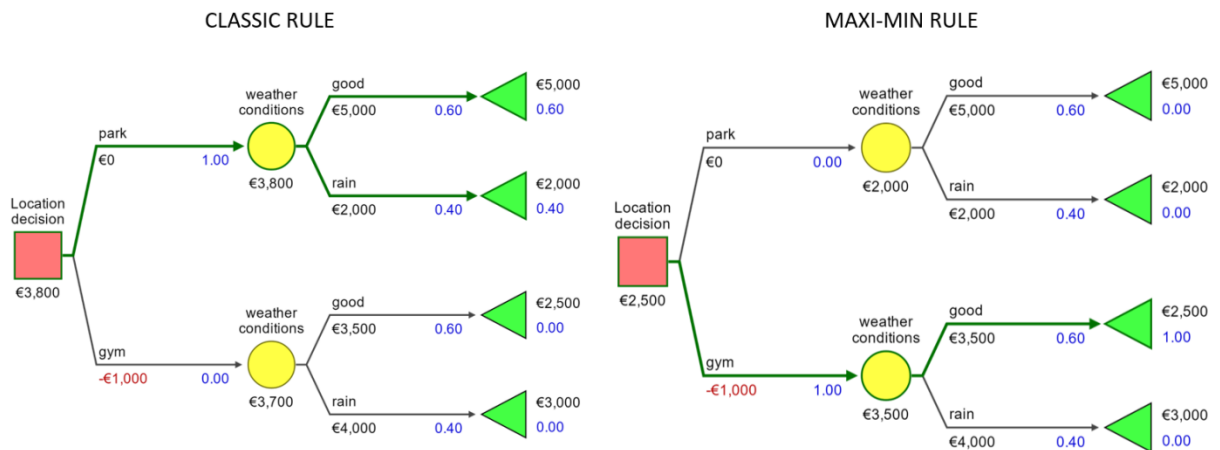


CASE III – Charity Olympic Games location

Public Administration wants to organize the first edition of Charity Olympic Games for local citizens. PA knows that more people would come and consequently more money would be collected if the event takes place in a local park. But in case of a rain, hardly few people come. Alternatively the event may take place in a gym. As it's July in Poland the chance of rain is 50%.

PARK	GYM
In case of good weather more money would be collected	In case of good weather less money would be collected
Rain may totally spoil the event	Rain shelter
No renting costs	Rent cost is €1000

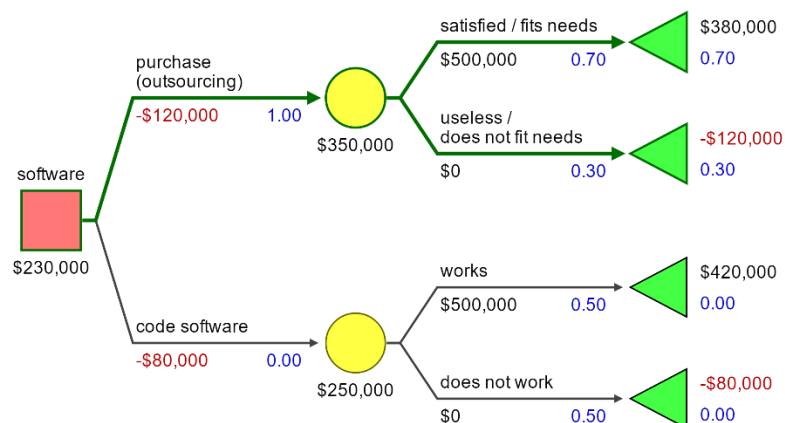
Charity Olympic Games location



CASE IV – City council management software

City council needs a new software system for effective schedule and employees management – such investment would reduce the management cost by as much as \$500 000 within the next 3 years. Such software might be chosen from the ones already available in the market – purchase cost is \$120 000. City council estimates that such software would fulfill its need with probability 70%. Unfortunately, there is still the chance that the system purchased in the market would not fit the city council's needs and consequently would turn out to be useless (the chance is 30%). On the other hand, the city council may outsource the external company to code the software for \$80 000. Such software for sure will fulfill Public Administration's needs, but there is the risk estimated at 50% that it would not work properly.

City council managment software



8.3 PILOT'S REACTIONS

On January 9th and 10th, 2017 a ROUTE-TO-PA project meeting took place at the University of Salerno. During the meeting initial reaction of Pilots to the new functionality for SIM decision presentation within the ODGM model has been collected. Pilots were wondering how to estimate the decision tree model parameters (like the probabilities or satisfied citizens fraction) as well as how to apply *a discussion with the citizens* to the decision tree diagram. A new application functionality, they were particularly interested in, was the simple *sensitivity analysis* which would enable decision tree parameters changes analysis: how much a certain parameter needs to be changed to have an influence on the final optimal decision delivered by the application? What is more, various decision process optimization measurements were also discussed, like the cost, profit or the percentage of citizens satisfied with certain decisions. This Pilot's comments will be addressed in the forthcoming releases of SilverDecisions software.

Pilots have identified several PA decision making problems which could be presented, discussed and solved with the SilverDecisions SIM SPOD plugin. The following list presents some of the cases proposed during the meeting:

- Public City-Centre WI-FI hotspot provider choice.
- The population of a certain town declines and consequently the healthcare service, education quality as well as living conditions get worse there. The following issues are among the main problems facing the town: hospitals and schools are being closed and there are no fast internet optical fiber cables available, as the providers are not willing to invest in that region. Fast Internet is crucial for both: small companies and the people living there. With no fast Internet there might be no new investments in the town at all. No new companies' investments means no job for the citizens. So the question arises: should the Public Administration participate in the Internet infrastructure investments to make the town more attractive for the companies and citizens?
- New Public Administration management software needed: which decision is better: to purchase the software from the solutions available in the market or to outsource the software coding?
- Healthcare: as the population declines in a certain region, public healthcare facilities maintenance gets very unprofitable and consequently the healthcare quality gets worse. The citizens would like to organize themselves by investing in healthcare facilities: should the Public Administration participate in such an investment?
- Public transport (buses/trams) vehicles purchase.
- City Marketing: How to promote tourism in the city in the most efficient way?
- Training programs funding: should the Public Administration invest in various training programs for the citizens? If so, which one should be chosen?
- Recycling - circle economy problem: the recycling process kept within the region generates cost, but in the same time it creates new job positions and could be an another source of energy: should the Public Administration invest in the circle economy?

The development of selected PA decision scenarios presented above will continue with the support of PA.

9 APPENDIX B – SIMULATION ALGORITHM DETAILS

This appendix contains the detailed description of multi-agent simulation for elicitation of preferences presented in Chapter 4. The logic in agent-based simulation models can be fully exposed fully through the sourced code. The authors in agent-based modelling literature agree that the source code of a multi-agent model is an important part of it's documentation. A detailed discussion of the role of source code in documenting agent-based simulation models can be found for example in Gilbert (2008), Law (2006) and Miller (2007). In this report we take the same approach and fully represent simulation model source code.

In the remainder of this section we explain details of steps 1 – 6 from the outlined procedure in Chapter 4.

Steps 1 and 2.

Reconstruction of edges is carried out between each agent $v \in V^{NS}$, i.e. between each agent who is not a platform user, and all the other agents $u \in V^P$, such that $u \neq v$, i.e. both those agents, who are not platform users, and between agent that is not a platform user and an agent who is on a platform. Therefore, edges that are known *a priori*, i.e. edges (edges between those agents, who are both platform users) in E , remain as they empirically are and are not reconstructed.

Model M_E is estimated on the basis of data available for agents $v \in V^S$, and then it predicts probabilities $p_{v,u}$, that an edge (v, u) exists between an agent $v \in V^{NS}$ and an agent $u \in V^P$ such that $u \neq v$. Probability of such an edge between v and u is inferred using data on $d(v)$ and $d(u)$. Several approaches can be chosen as applied in Chapter 4. We applied a exponential random graph models based approach:

$$\text{logit}(p_{ij}) = \mu_i + \mu_j + \beta \times |x_i - x_j| + \gamma \times e^\lambda (1 - (1 - e^{-\lambda})^k) \quad (1)$$

where $y_{v,u} = P((v, u) \in E^*)$ denotes a probability, that a link between agent v and u is formed, which, in the process of estimation, is approximated by:

$$y_{v,u} = \begin{cases} 1, & \text{if } (v, u) \in E \\ 0, & \text{if } (v, u) \notin E \end{cases}$$

i.e., by a binary variable which indicates which agents in V are connected by an edge.:

More specifically k is the number of edgewise shared partner which is $k_{ij} = \sum_k y_{ik} y_{jk}$ with decay parameter λ .

The model takes into consideration the intrinsic psychological factors (tendency to acquaintance with other people) of citizens i and j by parameters μ_i and μ_j , degree of homophily with parameters β as well as the influence of the common friends on the potential links between citizens whereas the marginal influence of the next common friend decreases (changing of common friends number from 0 to one has higher impact on the probability of existence of the common link between citizens i and j than increase e.g. from 20 to 21)

Such formulation lead to the following network statistics: degree distribution, sum of $|x_i - x_j|$ for all edges in the network and geometrically weighted edgewise shared partner distribution which is defined as $e^\lambda \sum_{k=1}^{n-2} (1 - (1 - e^{-\lambda})^k) \times EP_k$, where EP_k is the number of edges having exactly k shared partners.

The chosen function enables us to take into consideration the degree the similarity influences the probability of the link existence between citizens, the intrinsic individual features of the citizens and the social influence e.g. number of common friends .

As the parameter estimation using Monte-Carlo Markov Chain maximum likelihood estimation (MCMCMLE) method is computationally intensive and may be a burden in case of the large networks , we proceed as proposed in the literature, in case of large networks, namely we assume in the model (1) the average number of common friends for all the agents. This simplifies the estimation process and allows us to apply the maximum likelihood estimation (MLE) method and the logistic regression in particular. Thus population simulation and consecutive estimation process can be divided into the following variants:

Variant	Population simulation	Population estimation	Considered
Phase 1	Simplified (MLE)	Simplified (MLE)	+
Phase 2	Full (MCMC)	Simplified (MLE)	+
Phase 3	Full (MCMC)	Full (MCMCMLE)	-

Figure 37 Population simulation and estimation methods

Once a logistic regression model has been estimated, predicted probabilities $p_{v,u}$ can be derived. Than for all agents $v \in V^{NS} = V^P \setminus V^S$ the edges of the form (v,u) , such that $u \in V^{NS}$ and $u \neq v$ are reconstructed according to the estimated probabilities. The process is simulated many times.

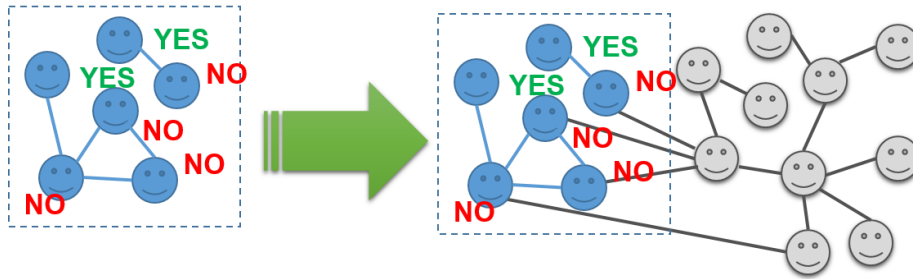


Fig. 38. Using data on platform users (blue agents), edges between agents who don't use the platform (grey), as well as edges between agents who don't use the platform and agents who use the platform are reconstructed (these are represented by solid grey lines).

For the logistic regression we use the ridge variant, see. le Cessie and Houwelingen (1992) implemented in WEKA (which is a dedicated data mining software) and set the ridge parameter to 1.

To consider the technical parameters of the computer machines available for the simulation purposes we may select the subsample of the sample using the snowball sampling (selecting however all of the linked citizens). Similar method called conditional Markov chain Monte Carlo maximum likelihood estimates (CMCMCMLEs) was proposed by Pattison et. al. (2013). The sampling can be repeated many times and the results combined as proposed by Stivala et al. (2016) Furthermore we reweight the number of observations with and without edge to avoid the rare events bias as described by King and Zeng (2001) so that the number of observations where there is no edge is not higher than 10 times number of observations where there is an edge (that represents the relation between citizens). The bias is related to the fact that the true population distribution is approximated more accurately when the sample number is higher. Thus when observations with and without edges are not balanced one compare the more exact distribution with less exact (more random) distribution.

We also recalibrate the original results to the true population average link density (we assume that such information is available based on the sociological studies and surveys). In the simulation we apply the generated synthetic population value.

Steps 3 and 4.

We assume 2 types of the citizens: the second type (continuous opinion type) admits opinion in an continuous way (e.g. she/he can support the idea to a different extent) Such an opinion $o(v, r)$ has a mathematical representation in form of the rational number for the $[-1, 1]$ interval., the first type (discrete opinion type) admits the opinion $o(v, r)$ that can have a threefold value, i.e. -1, 0 or 1.

For each agent v model M_o predicts three probabilities (with the sum of 1), corresponding to each of the three possible values of $o(v, r)$. Let us denote these probabilities by $p_v(x)$, where $x \in \{-1, 0, 1\}$. Model M_o is considered as an *a priori* opinion constructor in the sense, that it does not take into account any information on the network structure (on connections between agents), but it uses only information on an agent's characteristics, as contained in $d(v)$. If data in $d(v)$ is indeed discriminative with respect to $o(v, p)$, a model M_o can be a good predictor for opinions of agents in V^{NS} , i.e. for platform non-users.

As an opinion constructor we use a 3-nomial logistic regression model of the form:

$$y_{v,p}(k) = \frac{\exp(\gamma_k x_{v,p})}{\sum_{k \in \{-1, 0, 1\}} \exp(\gamma_k x_{v,p})}$$

where: $y_{v,p}(k) = P(o(v, r) = k)$ denotes a probability that agent's v opinion in round r on post p is k , where $k \in \{-1, 0, 1\}$, which, in the process of estimation, is approximated by:

$$y_{v,p}(k) = \begin{cases} 1, & \text{if } o(v, p) = k \\ 0, & \text{if } o(v, p) \neq k \end{cases}$$

i.e. by a binary variable which indicates what opinion is formed by an agent $v \in V$ on a post p , and:

$$x_{v,p} = f(d(v))$$

where $f: R^n \rightarrow R^{m_o}$ is a function, which constructs an m_o -dimensional vector of explanatory variables for agent v . More specifically, we assume that:

$$f(x) = (x^T, 1)$$

and that we have $m_o = n + 1$ and $\gamma_k \in R^{m_o}$, $k \in \{-1, 0, 1\}$, are vectors of parameters to be estimated using a maximum a posterior estimation method.

The primary opinion generation for the synthetic population generation (using the trinomial model with assumed parameters) follows 3 consecutive steps:

1. Trial opinion which is randomly drawn according to the probabilities calculated with the estimated trinomial model

2. The trial opinion is additionally polarized in a random way. Namely the polarized opinion is selected uniformly from the respective interval $<-1, -0.334>$ for negative trial opinion, $(-0.334, 0.334)$ for neutral trial opinion and $<0.334, 1>$ for positive trial opinion.
3. For the discrete opinion type of the agents the continuous opinion is transformed into discrete value in a following way : $<-1, -0.334> \rightarrow -1$, $(-0.334, 0.334) \rightarrow 0$ and finally $<0.334, 1> \rightarrow 1$.

Step 2 adds the variability to the simulated primary opinion models (making them more realistic by adding the explicit random component that represents the intrinsic psychological agent inclinations not related to her/his socio0demographic features).

We apply Step 3 in case of the trinomial model estimation for the continuous opinion type of the agents.

To attach an agent her/his opinion, we repeatedly simulate the primary opinions according to the estimated probabilities.

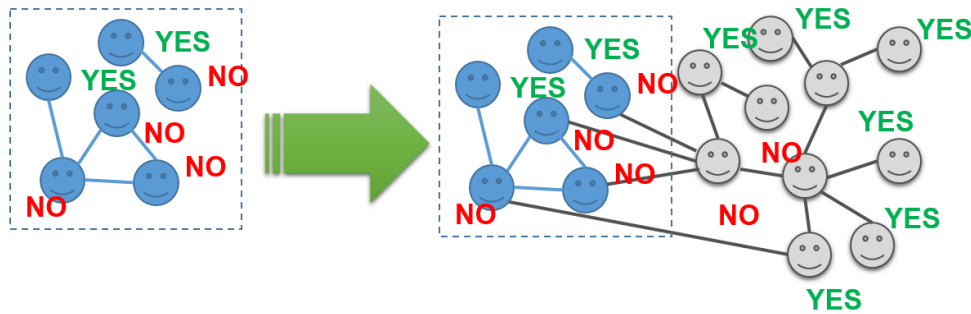


Fig. 2. Using data on platform users (blue agents), initial opinions of agents who don't use the platform (grey) are reconstructed.

Step 5.

At this stage for all agents in V^P and edges, or lack of thereof, is assumed between any pair of agents in V^P , therefore the structure of G^P has been fully built. Note, that primary opinions for agents in V^S are assumed not using the model M_o , but using empirical data on opinions $o(v, 0)$ for $v \in V$. Also edges between any pair of agents in V , i.e. edges $(v, u) \in E$, are not determined by the model M_E , but are assumed using empirical data on E^S . As a consequence, after the reconstruction of opinions and edges has been carried out, the empirical network structure G^S is extended by a predicted network structure and these two networks become interconnected. Therefore, the implied structure of G^P consists partially of empirical data and partially of data which was predicted or reconstructed by models M_E and M_o .

Once opinions have been assigned to agents and connections between them have been established, we simulate the synthetic population represented by G^P and $o(v, r), \forall r \in 1, \dots, n$ using an opinion diffusion algorithm which represents:

- a) the way in which an opinion of any agent $v \in V^P$ is influenced by an opinion of any agent $u \in V^P, u \neq v$, such that $(v, u) \in E^P$,
- b) the way in which an opinion of any agent $u \in V^P$ influences an opinion of any agent $v \in V^P, u \neq v$, such that $(u, v) \in E^P$.

The algorithm is implemented in according to the following rules, defined below. We also assume that the vector of parameters $\beta = [\beta_0, \beta_1, \beta_2]$, such that $\beta \in (0, 1)$ (homogenous for all the agents in the current

implementation) and $\sum \beta_i = 1$ represents the weight of the agent's own opinion in the relation to the opinions of the neighbours, the agents that the agent is connected to and the average opinion in the population (on SPOD) . We simulate for different values of β .

In particular, for consecutive elements of \mathbf{w} , update opinion of agent $v_j, j = 1, 2, \dots, |\mathbf{w}|$, according to the following rules:

$$o(v_j, r+1) \leftarrow k^*$$

$$\text{And } o(i, t+1) = \beta_0 \times o(i, t) + \beta_1 \times \overline{o_{\text{linked}}(t)} + \beta_2 \times \overline{o_{\text{obs},-i}(t)}$$

Where $\overline{o_{\text{linked}}(t)}$ is the average opinion of the agents that the agent i is linked to and $\overline{o_{\text{obs},-i}(t)}$ is the average opinion of all the observed agents different from agent i . For the population simulation purposes we assume that of all the observed agents different from agent i is just the average of all the agents in the population. For the estimation purposes we use the same approach for the agents outside the social platform and use the average of the population average opinion and platform average opinion for the agents present on the platform.

The opinion dynamics parameter vector $\beta = [\beta_0, \beta_1, \beta_2]$ can be interpreted as follows. β_0 measures how strong the agent remains with his own opinion, β_1 measures how strong is the influence of direct friends opinions on the opinion of an agent i and β_2 measures how strong is the influence of the opinions expressed on the social platform and in a population a general (general mood) on the opinion of citizen i .

The steps a)-d) are repeated for each round r .

Step 6.

At this step we compare simulated opinions dynamics (for each value of parameter β and each of opinion updating versions generated opinions observable on the $v \in V^S$.

Similarity measures can be defined in the following way:

$$\rho_1 = \sum_{v_j \in V^S} 1(o(v_j, n) = o^{\text{observed}}(v_j, n))$$

Or

$$\rho_2 = \sum_{r=1, \dots, n} \sum_{v_j \in V^S} 1(o(v_j, r) = o^{\text{observed}}(v_j, r))$$

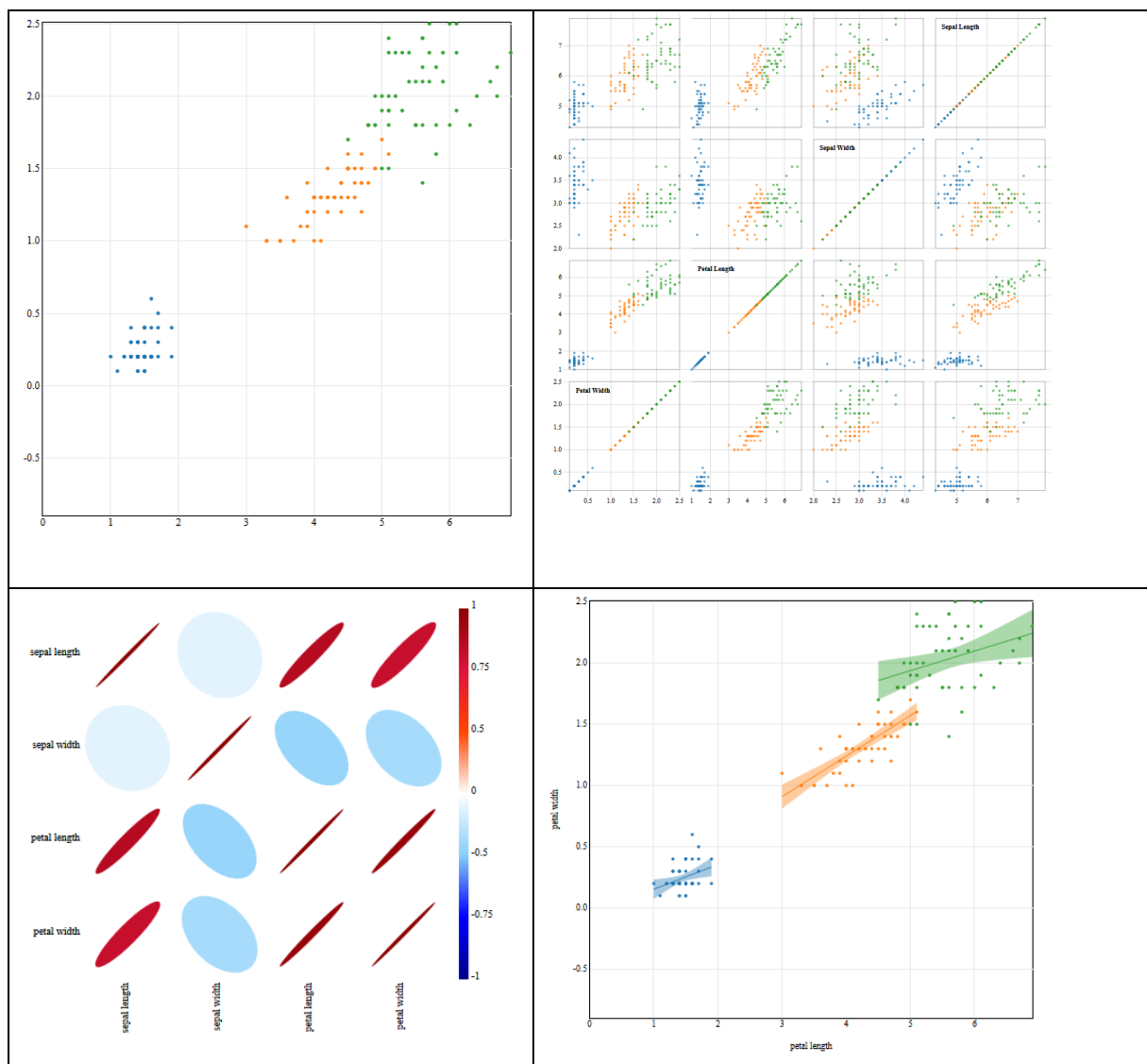
Where $1(.)$ represents standard identity operator and $o^{\text{observed}}(v_j, r)$ denotes the opinion expressed by the citizen v_j and observed on the platform in round r . The continuous opinions (continuous opinion type of the agents) are first make discrete using the same approach as in step 3.

A version of opinion dynamics and parameter value θ is chosen that maximizes the similarity measures ρ_1 or ρ_2 . So we choose such opinion dynamics that the pair wise concordance of the observed opinions and the simulated opinions observed on the sample of SPOD users is the highest.

For this particular dynamics on the whole synthetic population $v \in V^S$ we take $o(v_j, n)$ - which represents final reconstructed opinion as the generalization of the opinions observed in the sample.

10 APPENDIX C – TOOLS FOR ODGM DATA VISUALISATION FOR SPOD

Results of discussion with Pilots show a need to develop tools to visualize various aspects of SPOD usage dynamics along with opinion representativeness. In order to achieve this goal we have developed within SIM a set of open data visualization tools. The tools are open licensed and are currently available at the following address: <https://github.com/mwasiluk/ODC-d3>. The visualisation tools focus on presenting dependencies within the data. So far for the SIM module tools include a scatter plot, scatter plot matrix, correlation matrix, non-continuous heatmaps, boxplots and linear regression modelling (see Figure 39).



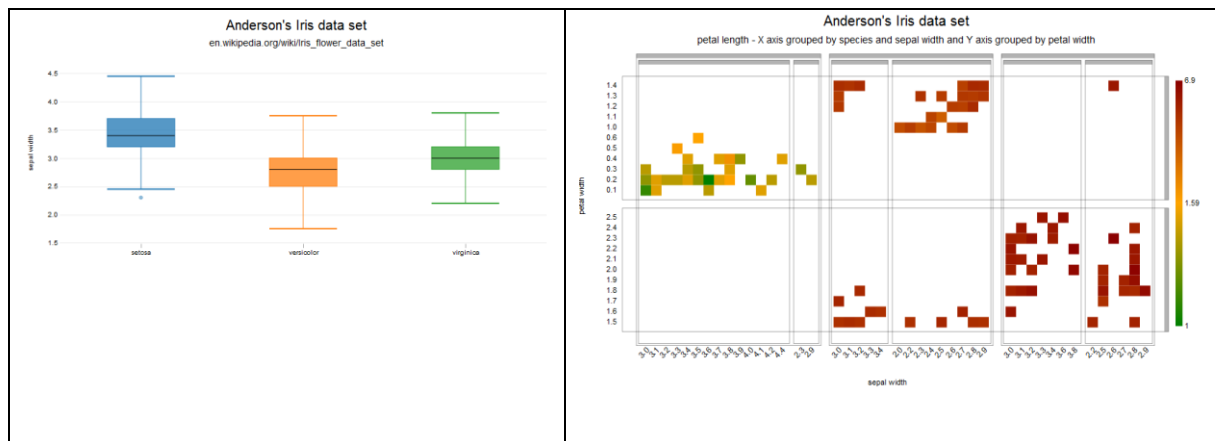


Figure 39 Sample data dependence visualisation currently under development within the SIM module. The visualisations will be used to illustrate preference structure within a virtual society.

11 APPENDIX D – SILVERDECISIONS' DEVELOPMENT GUIDE

SilverDecisions is a browser-based application. It can be used as a standalone webapp or it can be easily included in any HTML web page.

The project is built with JavaScript ECMAScript 6 compiled to ES5 with [Babel](#), [npm](#) and [gulp](#).

11.1 DEPENDENCIES

SilverDecisions was created without any heavy javascript and css frameworks (e.g Angular, Bootstrap) to ensure maximum portability and ease of use in existing web applications.

Core dependencies are:

- D3.js v4
- Math.js
- i18next
- lodash

All javascript dependencies are imported as ES6 modules and are prepackaged in the distribution files. List of all prepackaged dependencies may be found in package.json file.

Icons used in the app come from Material Icons and are included via Google Web Fonts.

11.2 PROJECT STRUCTURE

- demo - demo usage of SilverDecisions
- dist - distribution folder generated with gulp

- docs - website of SilverDecisions
- src - source files directory
- test - unit tests

11.2.1 SOURCE FILES DIRECTORY STRUCTURE

- i18n - internationalization files (i18n is implemented using [i18next](#)); there is a tool `check_trans.py` in this directory that checks if all internationalization files are valid and consistent
- model - data model implementation classes
- objective - objective rules
- styles - [Sass scss](#) styles source directory, compiled to css with [gulp-sass](#) plugin
- templates - HTML template partials, compiled with [lodash template function](#)
- tree-designer - decision tree designer files
- validation - validators

11.2.2 MAIN GIT REPOSITORY BRANCHES:

- master - contains current stable production version
- dev - contains development version.

11.3 QUICK START

1. Clone the repository
2. Make sure `nodejs` and `'npm'` is installed in your system ([nodejs.org](#)).
3. Install `gulp-cli` ([gulpjs.com](#)) and `bower` ([bower.io](#)) globally: `npm install --global gulp-cli bower`
4. Run `npm install` to install project's required modules

Build project with `gulp`

Watch changes and rebuild project automatically with `gulp watch`

11.4 BUILDING AND WORKING WITH PROJECT

11.4.1 GULP TASKS

- default - build project
- watch - watch changes and rebuild project automatically
- build-js - build and minify js
- build-css - build and minify css
- docs-gen - update silver-decision files in docs
- test - run [jasmine](#) unit tests with [karma test runner](#)

You can run `default`, `watch`, `build-js`, `build-css` tasks with `--dev` argument to skip files minification, e.g.:

```
gulp watch --dev
```

11.4.2 SAMPLE USAGE

SilverDecisions application is created with a **SilverDecisions** constructor function:

```
var container = "#app-container";
var config = { //optional config object
```

```

    ...
};
var data = { ... }; // optional object/json string with saved diagram

var app = new SilverDecisions(container, config, data);

```

11.4.3 CONFIGURATION

The **SilverDecisions** function takes 3 parameters:

1. id, selector or DOM element of app container element
2. optional config object
3. optional object/json string with saved diagram

Configuration object properties:

- **readOnly** - boolean (default: false)
- **lng** - GUI language key ('en')
- **buttons** - an object with the toolbar buttons visibility configuration:
 - **new** - new diagram button (true)
 - **save** - save current diagram button (true)
 - **open** - open existing diagram button (true)
 - **exportToPng** - export to PNG button (true)
 - **exportToSvg** - export to SVG button (true)
- **exports** - an object with the diagram export options:
 - **show** - show 'export' toolbar group (true)
 - **serverUrl** - url of the export server ('<http://export.highcharts.com/>')
 - **pdf** - PDF export configuration object:
 - **mode** - export mode, available values: 'client', 'server', 'fallback' ('server')
 - **png** - PNG export configuration object:
 - **mode** - export mode, available values: 'client', 'server', 'fallback' ('fallback')
- **rule** - default objective rule name ('expected-value-maximization')
- **format** - an object with default [number format options](#):
 - **payoff** - payoff number format options:
 - **style** - ('currency')
 - **currency** - ('USD')
 - **currencyDisplay** - ('symbol')
 - **minimumFractionDigits** - (0)
 - **maximumFractionDigits** - (2)
 - **useGrouping** - (true)
 - **probability** - probability number format options:
 - **style** - ('decimal')
 - **minimumFractionDigits** - (2)
 - **maximumFractionDigits** - (3)
 - **useGrouping** - (true)
- **jsonFileDownload** - enable/disable json file download after the 'save' button is clicked (true)
- **title** - default diagram title ('')
- **description** - default diagram description ('')
- **treeDesigner** - an object with the default tree designer options, described below

TreeDesigner configuration object properties

- **margin** - an object containing default plotting canvas margins
 - **left** - (25)
 - **right** - (25)
 - **top** - (25)
 - **bottom** - (25)

- **layout** - default layout settings
 - **type** - available values: 'manual', 'tree', 'cluster' ('tree')
 - **nodeSize** - node symbol size (40)
 - **gridHeight** - grid height (75)
 - **gridWidth** - grid width (150)
 - **edgeSlantWidthMax** - maximum slant for plotting the sloping part of the edge (20)
- **fontFamily** - ('sans-serif')
- **fontSize** - ('12px')
- **fontWeight** - ('normal')
- **fontStyle** - ('normal')
- **node** - [node settings](#)
 - **strokeWidth** - stroke width ('1px')
 - **optimal** - settings for nodes that belong to an optimal path
 - **stroke** - stroke color ('#006f00')
 - **strokeWidth** - stroke width ('1.5px')
 - **label**
 - **fontSize** - label font size ('1em')
 - **color** - label font color ('black')
 - **payoff** - settings for payoffs in nodes
 - **fontSize** - font size ('1em')
 - **color** - font color ('black')
 - **negativeColor** - font color for negative payoffs ('#b60000')
 - **decision** - settings for decision nodes
 - **fill** - fill color ('#ff7777')
 - **stroke** - stroke color ('#660000')
 - **selected** - settings for selected decision nodes
 - **fill** - fill color ('#aa3333')
 - **chance** - settings for chance nodes
 - **fill** - fill color ('#ffff44')
 - **stroke** - stroke color ('#666600')
 - **selected** - settings for selected chance nodes
 - **fill** - fill color ('#aaaa00')
 - **terminal** - settings for terminal nodes
 - **fill** - fill color ('#44ff44')
 - **stroke** - stroke color ('black')
 - **selected** - settings for selected terminal nodes
 - **fill** - fill color ('#00aa00')
 - **payoff** settings for terminal nodes' payoffs
 - **fontSize** - font size ('1em')
 - **color** - font color ('black')
 - **negativeColor** - font color for negative payoffs ('#b60000')
- **edge** - [edge settings](#)
 - **stroke** - stroke color for the edges ('#424242')
 - **strokeWidth** - stroke width ('1.5')
 - **optimal** - settings for optimal edges
 - **stroke** - ('#006f00')
 - **strokeWidth** - ('2.4')
 - **selected** - settings for selected edges
 - **stroke** - stroke color ('#045ad1')
 - **strokeWidth** - stroke width ('3.5')
 - **label**
 - **fontSize** - label font size ('1em')
 - **color** - label font color ('black')
 - **payoff** - settings for edges' payoffs
 - **fontSize** - font size ('1em')
 - **color** - font color ('black')
 - **negativeColor** - font color for negative edges' payoffs ('#b60000')
- **probability** - probability font options

- **fontSize** - font size ('1em')
 - **color** - font color ('#0000d7')
- **title** - diagram title settings
 - **fontSize** - font size ('16px')
 - **fontWeight** - font weight ('bold')
 - **fontStyle** - font style ('normal')
 - **color** - font color ('#000000')
 - **margin**
 - **top** - title top margin (15)
 - **bottom** - title bottom margin (10)
- **description** - diagram subtitle settings
 - **show** - show description on diagram as subtitle (true)
 - **fontSize** - subtitle font size ('12px')
 - **fontWeight** - subtitle font weight ('bold')
 - **fontStyle** - subtitle font style ('normal')
 - **color** - subtitle font color ('#000000')
 - **margin**
 - **top** - subtitle top margin (5)
 - **bottom** - subtitle bottom margin (10)
- **disableAnimations** - disable all diagram animations (false)
- **forceFullEdgeRedraw** - force full redraw of edges (false)

11.4.4 CUSTOM JAVASCRIPT EVENTS

- **SilverDecisionsSaveEvent** - an event fired after the 'save current diagram button' was clicked by the user. Event object's **detail** property value is set to a JSON string with saved diagram data. Example usage:

```
document.addEventListener('SilverDecisionsSaveEvent', function(evt) {
    console.log(evt.detail);
});
```

11.5 JSON FILE FORMAT

SilverDecisions JSON object has following properties:

- **SilverDecisions** - version of the SilverDecisions app used to save diagram file (**required**)
- **trees** - decision trees data; list of the root nodes (**required**, may be an empty array)
- **lng** - GUI language key
- **title** - diagram title
- **description** - diagram description
- **format** - number format options (same as described in the [Configuration section](#))
- **treeDesigner** - tree designer options (same as described in the [Configuration section](#))
- **texts** - list of floating texts

Data Model for Decision Tree representation

Point

- **x**
- **y**

Node

- **type** - 'decision', 'chance' or 'terminal'

- **childEdges** - list of edges going out of the node
- **name** - label of the node
- **location** - point
- **computed** - computed values

Edge

- **parentNode** - parent node
- **childNode** - child node
- **name** - edge label
- **probability**
- **payoff**
- **computed** - computed values

Text (floating text)

- **value** - text value
- **location** - point

12 APPENDIX E – RESPONSIBLE RESEARCH AND INNOVATION CRITERIA IN ICT (RRI-ICT)

The SIM module allowing for communication with citizens has an impact on end-users. Hence, it is important to address Responsible Research and Innovation (RRI) criteria about the work described in this document.

RRI has acquired prominence by its status as a cross cutting issue of the EU framework program for R&I, Horizon 2020 in the EU Commission document by Roger, Strand, et al. titled "Indicators for promoting and monitoring responsible research and innovation: report from the expert group on policy indicators for responsible research and innovation." (2015) . In this document, RRI is defined *"a transparent, interactive process by which societal actors and innovators become mutually responsive to each other with a view on the (ethical) acceptability"*. The objective should be trying to align process (and therefore outcomes) to the values, needs and expectation of our society.

1. Public engagement

The goal of SIM module is to increase the public engagement in the decision making process in the public administration by providing a clear, readable representation of the decision making process.

The SilverDecisions software development within the SIM was carried out with cooperation and in response to a need of Public Administration pilots to clearly communicate their decision processes to citizens.

The SIM module code is being released as Open Source. In order to collect the wide feedback from a maximum number of users we made the tool to be available not only as a SPOD plugin but also outside SPOD platform as an in-browser application for decision modelling available at silverdecisions.pl. By collecting users feedback we know how to develop the software.

2. Gender equality

Does not affect the SIM contribution.

3. Science education

The SIM module offers a quantitative approach to decision problem modelling. Decision tree model developed within the SIM module promotes a holistic, mathematic approach to decision problem modelling and solving. The software is also being promoted independently of SPOD as a tool to use for teaching decision analysis at MBA courses in various educational institutions.

4. Open Access

In general, as with the other “Public” deliverables, this deliverable will be available in its final format at the www.routetopa.eu/public-deliverables/. Publications related to these results will also be made available through the various open access archives of the respective research and academic institutions.

The source code and all materials is freely available as Open Source.

5. Ethics

Does not affect the SIM contribution.

6. Governance

The project Ethical advisor will review all processes involved in the engagement and collection of information from participants in workshops and other activities related to user stories elicitation and requirements specification.

7. Sustainability

The project is released as Open Source. One of the goals is to build a very broad user base of the SilverDecisions module. Since it is available on GitHub the large user base guarantees that the software will be sustained by its users once the ROUTE-TO-PA project is completed.

8. Social Justice/Inclusion

The SIM module supports personalization of the user interface (see SilverDecision documentation). Moreover, whenever possible both colors and shapes are being used to represent application logic (e.g. green triangles for final nodes, yellow circles for chance nodes and red squares for decision nodes).

We try to support a wide array of different web browsers to increase software’s accessibility.