Automated Scalable Modeling for Population Microsimulations

Delving in suitable design decisions for microsimulations of influenza propagations

Johannes Ponge, Diego de Siqueira Braga, Dennis Horstkemper, Bernd Hellingrath, Stephan Ludwig
Westfälische Wilhelms-Universität Münster
Münster, Germany
{johannes.ponge, diego.siqueira, dennis.horstkemper, bernd.hellingrath, ludwigs}@uni-muenster.de

Fernando Buarque de Lima Neto
University of Pernambuco
Recife, Pernambuco, Brazil
fbln@ecomp.poli.br

***Abstract*—The propagation of diseases within the population is an ever-reappearing hot topic in news stories. Mutations of known diseases repeatedly infected large portions of the population. In order to support the select appropriate mechanisms to help taming the spread of an epidemic, several simulation approaches have been developed to forecast the propagation behavior of diseases. Agent-based micro simulations promise to create the most detailed and accurate forecasts, but require a high modeling effort. In this paper we propose an approach to lessen this modeling effort by introducing a method that automatically creates agents for representing groups within the population based on multiple data sources (e.g. census data, vaccinations records, etc.). Our approach also facilitates combining these heterogeneous data with geographic information systems as well as dealing with incomplete data for enabling automatic and scalable creation of epidemics models for different simulation purposes in epidemiology. Two test cases were used to assess the proposition.**

Keywords— Synthetic population, Microsimulation, Disease propagation, Grouping problems, Influenza Models.

I. Modeling Populations for Microsimulations

Influenza viruses are a common long known disease. Each year within the winter months throughout Europa and other continents these viruses cause potentially dangerous epidemics. Furthermore, spontaneous occurrences of newly mutated influence viruses can lead to pandemics which often cause severe health issues including multiple death cases. In order to introduce measures to limit the spread of such epidemics or pandemics, computer simulations have already been employed. Several approaches put forward are based on stochastic modeling paradigms [1],[2]. In these simulations information is gathered in intensive surveillance programs and from reconstructions of past epidemics. However, they often do not consider properly detailed factors heavily influencing the spread of influenza diseases, such as geography and demography as well as the social environment and workplaces of individuals.

Microsimulation approaches, in which the population is represented by agents that are used to model specific individuals, promise to overcome such limitation by consideration of more relevant factors. Thus, affording a much more detailed and suitable forecasting of propagation behaviors of diseases, e.g. such as influenza. Microsimulations, as a subset of all simulation approaches, focus on the interaction of singular objects often in a one-to-one manner. From these interactions an overall trend generally emerges, e.g. the propagation pattern of a disease within a population that is transferred via skin contact or airborne infection. However, such modeling approaches require much more detailed data and therefore have a considerably higher requirement in modeling efforts regarding distinct agents.

When the objects within a microsimulation represent members of a population, the entirety of these objects can be designated as a synthetic population [3]. The process of modeling such a synthetic population is called population synthesis. Since the situational context for the simulation of the propagation of a disease might change over time, researchers and practitioners frequently need to employ different modeling strategies, e.g. by expanding the considered region or agent types within the simulation. Therefore, approaches to automatically perform a population synthesis would greatly lessen the modeling effort for such microsimulations.

The seminal issue when creating a synthetic population is the pre-supposed close resemblance with the real population. However no concrete (and complete) information about the real population is usually the case. While national and regional governments possess and release census data, this information in most cases only contains highly aggregated data about persons and households in one area. In several places data protection is assured and so, no information about individual persons can be utilized. This means, that for the purpose of the population synthesis, all given bundled data needs to be disaggregated prior use into the modelling. Agents are the means to perform such individualization, whose characteristics should be as close as possible similar to the aggregated data

provided [4]. The aim of this paper is to introduce a method to attain this cause.

The remaining part of this paper is structured as follows: in Section 2 an overview of past attempts to tackle population synthesis are presented. Section 3 describes the proposed method for the creation of synthetic populations in an automatic and scalable matter, aiming at microsimulations. The results in Influenza Epidemics are presented and analyzed in Section 4. Finally, concluding remarks are made in Section 5.

II. Existing methods

Over the last 20 years a number of population synthesis models were developed. The applied methods themselves strongly depend on the available data since most of the algorithms were originally developed for specific use cases. For a general overview we categorize existing methods by three criteria: (i) Population Synthesis on individual- or household level, (ii) sample free or sample based methods and (iii) simultaneous or sequential generation of individuals and households.

The first criterion indicates the synthesis' granularity means whether actual persons and households are to be treated as individual objects during the synthesis process or not. On the one hand there are known methods that only generate a population of households with attributes like "family income" or "number of rooms" [5]; that is, population synthesis methods on household level. Though, in the majority of cases, households are extracted from a representative sample of households. Individual persons can then be created from those samples if it contains information on the individuals that make up that household. So far it is not possible to combine an arbitrary group of persons into one household. These procedures can also be classified as methods on household level since persons and households are not treated individually during the synthesis process. Such approaches are actually implemented and tested [6][7]. On the other hand there are also known algorithms that synthesize households and persons individually, and simultaneously [8][9][10].

The second criterion specifies whether a sample is used to synthesize the population. For sample-based methods, there are two well established practices. One approach is to assign a weight to every entity in the sample depending on their combination of attributes. Those weights specify how many objects with distinct set of attributes must be included in the population to meet certain socioeconomic or demographic characteristics. After that, objects are drawn randomly (depending on their weights) from the sample until the requested population size is reached [6]. Another approach is based on the urn principle "drawing with replacement". In this case, single entities are drawn randomly from the sample until the requested population size is met. In a subsequent step, combinatorial optimization algorithms are applied to check if some individuals in the artificial population can be replaced by other entities from the sample to reach a better overall fitting [4][11][12].

The third criterion is only of interest if a method is capable of simultaneously synthesizing individual persons as well as households into groups of individuals. Therefore, it is possible to apply changes on both, individual and household level, in every iteration [9][10]. In contrast to that, there are sequential syntheses that generate households and persons in two separated steps [6][7]. To the best of the authors' knowledge all known methods create households first and then derive the single individuals from that. In contrast, the proposed method here is capable of handling persons and households completely independent one from another and can rely on multiple data sources even on household level (which is not the case in some current approaches). Section 3 explains in detail how the proposed method works.

The two most widely used general approaches for population synthesis are the Synthetic Reconstruction method (SR), and the Combinatorial Optimization (CO) [4]. The starting point of the SR method is a series of tables that specify the frequency that a particular characteristic of people is present within a population. For example, the number of people of a certain age group that compose the sample (for example, 0-5 years, 5-10 years ...). This information is referred to as aggregates, since they indicate only the absolute frequency of each attribute, but not the number of combinations. They can be taken mostly from public sources such as the census. For some combinations (e.g. age and sex) data is available, but much information is not published (because of privacy or low interest) and must therefore be estimated. The SR method uses the Iterative Proportional Fittings [13] in order to match a sample of data in a way that there is consistency against other data sources. Thus, a conditional probability can be determined for each combination of features a conditional probability, which designates basis of how the population is reconstructed. Although this method is the most widespread, it also involves many disadvantages as all operations are based on coincidences and probabilities, where a random error must always be taken into account. A further risk involves the fact that the next decision will always be based on a conditional probability, starting from the current situation taken. Thus, random deviations of the synthesized data arise. With the Combinatorial Optimization approach [11][12], a sample of households can be drawn from a random required number of households. Individual households can thereby be arbitrarily assigned. After that, it is determined how satisfactory this combination of households is regarding the desired demographic and socioeconomic factors. The main issue with this approach is, that due to the variety of combinations that can be generated from the sample, it is almost impossible to test all combinations. Hence, intelligent algorithms are required so they reach, in a reasonable time, reasonable solutions.

The use case of microsimulations of influenza propagations makes special demands to population synthesis models. For the

simulation of the virus transmission from person to person it is necessary to employ a population synthesis method that is operating on person level. In many cases national census data will provide a sound data baseline for the population generation. Additionally, it must be possible to assign individual disease related attributes to every entity, e.g. information about chronic illnesses or inborn disabilities that may affect the contagion probability of the disease. Population synthesis methods that work on household level or methods, which draw households including their inhabitants from samples, usually cannot provide this features. This is due to the fact that either this information is not contained in the underlying sample and/or it is not possible to equip the synthetic individuals with additional information once the population is created without losing its validation.

Furthermore, most known methods can only consider more than one data source in very limited use-cases. When data is incomplete or only partially available, it is often almost impossible to create a population with them. To give an example, let us assume, that we are investigating how an influenza disease propagates in Germany. We focus on the age and the fact whether an individual is vaccinated or not. A comprehensive record about the age distribution in Germany is provided by the Census 2011 [14]. Additionally, the Robert Koch Institute [15] tracks data about vaccinations. Unfortunately, only children get examined on a regular basis during their primary school enrollment. Therefore, we do only have exact data for a specific age range. The residual distribution has to be estimated. Thus, the natural recommendation is to use a population synthesis method that is capable of processing known and estimating unknown data. Yet, as mentioned before, no known approach does provide this feature.

Lastly, it is important to mention that households are only treaded as "containers" for grouping people in the current use case. This abstraction is important in order to reproduce the contact behavior of a family, e.g. to accurately simulate a disease transmission. Thus, households may not need any specific attributes themselves. To sum the requirements up, a population synthesis method that allows researchers to generate single persons and groups of persons as households individually is required. Moreover, a sample-free approach is needed, since in most cases samples that contain all required information do not exist.

III. METHOD FOR THE CREATION OF SYNTHETIC POPULATIONS

In this section we introduce a scalable and flexible two step modeling approach of population synthesis for a microsimulation use case. We also show how incorporating multiple data sources to reach reasonable results in manageable timeframes.

The terminology used is the following: the most important part of a population model are the individuals' attributes, e.g. 'age', 'gender' etc. A set A of all attributes is defined as the vector $\langle A_1, \dots, A_n \rangle$ where N is the overall number of attributes. The vectors' elements $A_1, \dots, A_n$ represent sets of feature characteristics $A_n = \{d_{n1}, \dots, d_{nm}\}$ of the attributes. For example, let A_1 be the attribute 'gender', then d_{11} and d_{12} might be 'male' or 'female'. Numerical attributes like 'age' are also defined as a set of feature characteristics. This is not a limitation anymore, since most data sources like the census only provide aggregated data for age ranges. If a more precise definition is required, the user is able to define more fine-grained feature sets. In an ongoing example, let A_2 be the attribute 'age' with feature characteristics d_{21}= 'children (0-14)', d_{22}= 'adults (14-65)' and d_{23}= 'seniors (65-100)'.

The set P represents the basic population of all persons. A person $p \in P$ is a vector $\langle D_1 \subseteq A_1, \dots, D_n \subseteq A_n \rangle$ with $|D_1|, \dots, |D_n| = '1`$ of length N. Every person holds exactly one feature characteristic for every attribute.

The set H represents the basic population of all households of a synthesized population. A household $H_i \in H$ is a set with cardinality $|H_i| = g_i$ where g_i is the number of inhabitants of this specific household which is defined a priori. Inhabitants are represented as persons $p \in P$. The overall number of households is defined as X. Furthermore, we constitute that any person from P can only be assigned to one household from H and that all persons have to be assigned to any household. We can derive the following implications:

$$H_1 \cup \dots \cup H_X = P \text{ and}$$

$$H_i \neq \emptyset \text{ with } H_i \cap H_j = \emptyset\,,\ i \neq j$$

FIRST STEP – CREATING THE POPULATION

The presented method is of strict sequential nature, meaning that the synthesis of population P and their grouping to households can be done independently from each other. Any known method can be used to create P. Unfortunately, for modelers, up to this article, we did not know any algorithm that is capable of treating persons and households individually while incorporating multiple data sources that may be even incomplete. Thus, next, we propose a new approach for that.

We start out by deriving all population restrictions from the data source and store it in set W. A population restriction is what we call any information from the data source about the number of occurrences of an attribute's specific feature characteristics and their combinations in a particular population. Such restrictions are tuples (t, k) consisting of an integer target value t and a vector $k = \langle D_1 \subseteq \{A_1, \emptyset\}, \dots, D_n \subseteq \{A_n, \emptyset\} \rangle$ with $max(|D_1|, \dots, |D_n|) = 1$ of length N. The vector contains one element for every attribute in A. It holds one feature characteristic of the associated attribute. A restriction $w_1 = \langle 10, \langle \{d_{11}\}, \{d_{21}\} \rangle \rangle$ would imply that ten male children from age 0 to 14 have to be in the synthesized population. Beyond that, an element in k can the null set to specify that any feature characteristic of the attribute fits in. Restriction $w_2 = \langle 10, \langle \emptyset, \{d_{21}\} \rangle \rangle$ would therefore specify that ten people in the population have to be of age between 0 and 14 regardless of which gender they are. We are certainly aware that most data sources will not provide this data format right away. But since we aim at put forward a universal approach to generate baseline populations from arbitrary data sources, transforming the data into these uniform restrictions is a mandatory step.

After all restrictions have been set, we start generating the population in an iterative process. Thereby we always exploit the restrictions with the lowest target value and most restrictive conditions. For a simple example, we assume that restrictions $w_1, \dots w_4$ were derived from a data source:

$$w_1 = \langle 100, \langle \emptyset, \emptyset \rangle \rangle;\ w_2 = \langle 20, \langle \{d_{11}\}, \{d_{21}\} \rangle \rangle$$

$$w_3 = \langle 40, \langle \{d_{11}\}, \emptyset \rangle \rangle;\ w_4 = \langle 15, \langle \emptyset, \{d_{23}\} \rangle \rangle$$

For visualization purposes we then transfer all restrictions to a matrix (Table 1). The algorithm can be applied for a bigger number of attributes; this table represents only a didactic visualization.

Table 1: Matrix of feature characteristics combinations

		age			
		0-14y	*14-65y*	*65-100y*	*total*
gender	*male*	20	i_{21}	i_{31}	40
	female	i_{12}	i_{22}	i_{32}	i_{a2}
	total	i_{1g}	i_{2g}	15	100

Since we could not find any method, that is capable of creating individuals and households sequentially and integrate multiple data sources at the same time, we developed this completely new approach in order to do so. To generate the actual population we propose the following algorithm:

1. Assign 0 to all cells for restrictions $w \in W$ with target value $T = 0$. If there is no restriction left with a target value $T > 0$, set all remaining cells to 0 and terminate.
2. Pick the restriction $w \in W$ with the lowest target value $T_{min} > 0$. When this restriction is exploited no other restriction can be violated since it has the lowest target value.
3. Find all cells I, that are affected by the picked restriction w and do not already have a numerical value assigned. Chose the set of cells I^* from I that satisfy most restrictions.
4. Assign the value $\left\lfloor \frac{T_{min}}{|I^*|} \right\rfloor$ resp. $\left\lceil \frac{T_{min}}{|I^*|} \right\rceil$ to all cells in I^*. That will portion out the target value to all considered cells in I^* and lead to a maximum heterogeneous population.
5. Check for all cells $i \in I^*$ which of the remaining restrictions satisfy that cell and decrease their target value by the value of i. After that, go back to step 1.

In our current example, we have solely restrictions with target values $T > 0$, therefore we immediately jump to step two. Now w_4 is chosen to be exploited next, since it has the lowest target value. Step three identifies $I = \{i_{31}, i_{32}\}$ as the cells that are affected by w_4 and do not already have a numerical value assigned. We see that i_{31} is the only feature characteristic combination that also satisfies w_1 besides w_4. Therefore we set $I^* = \{i_{31}\}$. In the fourth step, the target value of w_4 is equally portioned out to all members in I^*. Since I^* only has one member, i_{31} gets assigned with the target value 15. Table 2 shows the matrix of feature characteristics after one iteration.

The last step updates the restrictions:

$$w_1 = \langle 85, \langle \emptyset, \emptyset \rangle \rangle;\ w_2 = \langle 20, \langle \{d_{11}\}, \{d_{21}\} \rangle \rangle$$

$$w_3 = \langle 25, \langle \{d_{11}\}, \emptyset \rangle \rangle;\ w_4 = \langle 0, \langle \emptyset, \{d_{23}\} \rangle \rangle$$

Table 2: Feature characteristics after one iteration

		age			
		0-14y	*14-65y*	*65-100y*	*total*
gender	*male*	20	i_{21}	15	40
	female	i_{12}	i_{22}	i_{32}	i_{a2}
	total	i_{1g}	i_{2g}	15	100

The algorithm terminates after four iterations with a completely filled matrix which contains the exact number of individuals that should have a specific feature characteristic combination in the synthesized population.

As already mentioned, this procedure is capable of processing multiple data sources and even only partially available data that does not provide perfect marginal distribution for all attributes.

SECOND STEP – AGGREGATING THE HOUSEHOLDS

After the population P has been created, the subsequent task is to aggregate individuals to households. As stated before, we could not discover any previously developed method to fulfill this task while treating persons and households individually and incorporate arbitrary data sources. Hence, we designed a new approach related to the work on grouping problems [16].

First, a number of restrictions R is derived from the data sources, similar to the person restrictions. A household restriction $r \in R$ is a tuple (T, E), consisting of a target value T and a number of inhabitant restrictions. The target value T specifies the number of households in Y that are supposed to satisfy restriction r. An inhabitant restriction $e_i \in E$ defines properties that a number of individuals living in that household should match with. The variables $Q_{min_{e_i}}$ and $Q_{max_{e_i}}$ state how many inhabitants must satisfy a certain restriction. Attribute feature characteristics that a distinct person must provide in order to match with an inhabitant restriction are represented by vector $v_i = \langle D_1 \subseteq (A_1 \cup \emptyset), \dots, D_n \subseteq (A_1 \cup \emptyset) \rangle$ of length N. This means, that there is a set of valid feature characteristics for every attribute which can also be the null set. In this case, any feature characteristic is valid. A short example will clarify the basic principle.

The exemplary restriction r_1 from Table 3 implies that there are 15 households to be created in the synthetic population that satisfy the profile "single mother". It consists of inhabitant restriction e_{11} and e_{12}. Those constraints declare that there has to be exact one individual in the household that is female (d_{12}) and either of age 14 to 65 (d_{22}) or 65 to 100 (d_{22}). Furthermore, there can be a number from one to four individuals that are 0 to 14 years old (d_{12}) regardless of their gender. If any subset of P meets those demands, this subset is considered as a "single mother" household. Because of $Q_{min_{e_i}}$ and $Q_{max_{e_i}}$ the number of persons of one household is not fixed previously. This is due to the fact that many data sources, especially in a census, only provide a number of household profiles like "single mother". A fixed set of inhabitants and their properties is not given in the data.

To check whether a household h as a subset of P satisfies a given household restriction r, we defined an indicator function $I_r(h)$ that outputs 1 if the conditions are met and 0 if not.

Table 3: household restriction

household restriction r_1 ("single mother")			
inhabitant restriction	Q_{min}	Q_{max}	valid feature characteristics
e_{11}	1	1	$\langle\{d_{12}\},\{d_{22},d_{23}\}\rangle$
e_{12}	1	4	$\langle\emptyset,\{d_{21}\}\rangle$
target value $T_1 = 15$			

A big advantage of our introduced procedure is the possibility to combine multiple data sources on person as well as on household level. Moreover, it is possible to apply nested restrictions, meaning, that one household can meet several conditions. A simple example would be the household profile "family with two children" that also fits the profile "student living with parents".

The optimal assignment of the individuals in P to the households in H provides all characteristics to be classified as a grouping problem as defined by Falkenauer [16], who states that most of the grouping problems are NP-hard. For the current use case, the individual appearance of the problem instance strongly depends on the combination of household restrictions. In fact, our tests have shown that it is actually NP-hard in the cases tested. In addition to that, we assume, that most of the problem instances will be too large to be solved by exact algorithms. Since this synthesis method is designed to create artificial populations with several thousand, perhaps even several million individuals, the resulting number of possible household combination exceeds the computational capabilities of any known machine. Therefore, we decided to soften our constraints and do not apply exact algorithms. Hence, we derived a fitness function to assess any household combination for the population P:

$$minimize\ U = \frac{\sum \delta_r \left(\frac{F_r}{T_r + 1}\right)^2}{\sum \delta_r}.$$

First of all, we compute the total error

$$F_r = \left(\sum_{h \in H} I_r(h)\right) - T_r$$

for every restriction $r \in R$ and divide it by $(T_r + 1)$ to get the proportional error for r. The divisor is incremented by one to avoid undefined zero division in restrictions with a target value of zero. Additionally, restrictions are weighted by δ_r. The default value of δ_r would be one. Finally, the sum of the weighted squared errors is computed and divided by the sum of weights. That makes the range of U independent from the number of household restrictions and fitness of different models comparable.

As shown in the literature [11][12], combinatorial optimization seems to be the method of choice when trying to reconstruct a population from a given sample. Though our approach does not employ a sample, it provides a fully reconstructed population P after the first step. Assuming that a complete (representative) population is a data base at least as a good as a representative sample, we decided to apply combinatorial optimization algorithms for optimal household allocation as well. In contrast to sample based approaches where single households can be drawn with replacement, the current application requires every person in P to be assigned to exactly one household. Therefore, no replacement is possible in order to respect the population restrictions introduced in the first step.

To solve the problem, we applied two combinatorial optimization algorithms, namely a hill climbing algorithm, which is the most basic approach in combinatorial optimization, and a grouping genetic algorithm as introduced by [16]. The former starts out by assigning all persons to a household randomly. After that, the fitness of this initial candidate solution is computed. A number of persons is then drawn from the candidate solution and reassigned randomly. If the new solution is of higher fitness (respectively a smaller error), the changes will be adopted or discarded, otherwise. Our tests on different problem instances revealed, that the optimal number of individuals to switch every iteration can be approximated by the absolute value of a normal distribution with variance $10^{-4,42}$. Yet this value might change depending on the size of the problem instance. It is advisable to use a normal distribution rather than a fixed number of persons to switch. This way, it is theoretically possible to switch all persons at once and therefore "reach" every possible combination at point in time. On the one hand, the number of switches should not be too high, to prevent an again randomization of the solution. On the other hand, with a fixed number of switches it might be possible that certain combinations can never be reached and the solution is only optimized locally. A normal distribution with mentioned variance induced the fastest overall improvement of the candidate solution while assuring a global optimization.

The grouping genetic algorithm, originated from the class of evolutionary algorithms. Therefore, we always hold a pool of candidate solutions from which we combine new, hopefully better solutions. These new solutions replace their parents in the candidate solution pool and contribute to a better overall fitness of all solutions. The idea is to keep good solutions and merge promising parts of different solutions. However, this comes at the cost of quite some overhead since all persons must be assigned to a household exactly once. Thus some additional modifications had to be made after every crossover in order to make a solution feasible again. Another feature adopted is to mutate existing solutions, just as hill climbing algorithm does, to create new candidate solutions and add new information to the population of solutions.

IV. Exemplary application

To test our algorithm, we set up two test instances: `Hard600` and `ZensusMuenster`. The `Hard600` instance contains 100 single, 100 two-person and 100 three-person households. All persons are equipped with an ID-attribute that

make them fit only into one distinct household. This instance represents the most restricted possible use case.

The `ZensusMuenster` instance incorporates real demographic data that was imported from the German census 2011 database. All persons in the population have the attribute age that was split into 19 age group and the attribute gender. The size of the problem instance represents 10% of the actual population of the city of Muenster in north-western Germany (i.e. approximately 30000 inhabitants). Six restrictions were applied on household level regarding the number of: couple without children, couple with children, single parent, seniors exclusive, seniors and younger people and no-senior households.

We applied both combinatorial optimization techniques on all three instances with arbitrary runtimes until we terminated the algorithms. Figure 1 and Figure 2 show the results.

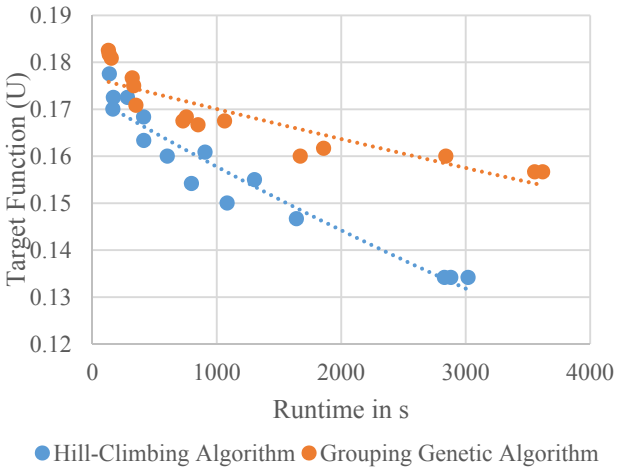

Figure 1: Test Series for Hard600 Instance

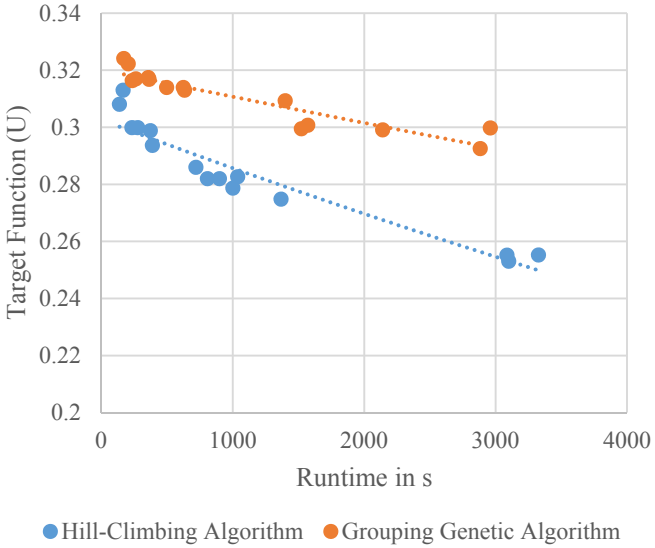

Figure 2: Test Series for ZensusMuenster Instance

The tests have shown that we could already reduce the error for the Hard600 instance to 13.5% in manageable time with the hill climbing algorithm. Our designated goal is to fine-tune this approach to ultimately lower the error to under 5% and therefore keep up with state of the art synthetic population creation techniques. Surprisingly, the grouping genetic algorithm that was developed for this specific class of problems, did not perform as well. This might be due to the fact, that the additional effort to reconstruct a feasible solution after a crossover slows down the whole process drastically. As seen in Figure 2, the error for our real world data scenario is a little higher. Though, this test instance is not a constructed case like the previous one. Census data rely on statistical estimations. Furthermore, some definitions of specific household types can be very broad. For example, when three elderly people live together, whereas one is the child of the others, this household still counts as "parents with children", at least for the German census [14]. Additionally, some households in the underlying data are based on special attributes of the inhabitants that are not captured in the statistics for the individuals due to privacy reasons. If multiple data sources are combined, it will get even harder to guarantee their compatibility. But since a population synthesis is always only an estimation based on the available data, it is inevitable to work with incomplete data and accept a certain amount of deviation. Due to those reasons we cannot assume that there is a perfect solution with $U = 0$ for all real world population synthesis scenarios.

This paper focuses on the creation of synthetic populations that match real world demographic characteristics. However we want to spend a few words on how we use this approach in context of a microsimulation of influenza propagations. The population creation process returns a set of households and persons they contain. These households are placed on a 2D map. Every person is represented by an agent in our simulation and it provides a distinct set of attributes. The values of these attributes which were computed in the population synthesis. Additionally, the agents are equipped with certain movement patterns to realistically simulate human behavior. Yet, the explanation of these patterns and how to create them would exceed the scope of this paper.

After the initial setup, the simulation can be executed. Whenever agents, following their individual patterns, meet at a particular geographical location, we can compute the probability of a disease transmission, if one of the agents is infected. For this situation, we believe that the simulation results will benefit from our population synthesis approach. Many simulations of influenza propagations rely on a restricted (almost fixed) probability of transmission. Yet, it is general knowledge, that the probability transmission is highly dependent on the individual's immune system which is influenced by its vital parameters. These parameters may be age, gender, body mass, basal immunity level, pregnancy, immune suppression diseases and many more. Whereas the first two attributes can be obtained from census data, it is inevitable to incorporate multiple data sources to create baseline populations with all these parameters. Our population synthesis

approach allows us to break down the diseases structure and explore its properties in order to compute realistic transmission probabilities, rather than working with a black-box disease, assuming a fixed rate. Having these opportunities we can proceed to investigate the diseases effects on certain parts of the population, test countermeasures or simulate new mutations of a disease.

V. Summary & Further Research

In this paper we introduced a novel method for generating baseline populations for microsimulations tailored mainly for disease propagation related use cases. In contrast to common practices, our two step modeling approach allows us to work with multiple data sources and even incomplete data sets. We derive the available data into a unified format as a set of restrictions. A major advantage of this approach is, that it makes the applied algorithms independent from the data source. Moreover, we can consider persons and households individually and create both of them sequentially, which allows for a scalable and flexible creation of synthetic populations. Whereas we have way more fine-grained modeling options on the one hand, the computational costs do also rise on the other hand, since we cannot exploit data source dependent properties. Especially with regards to the assignment of persons to households, there has to be more research on finding better aggregation procedures. During our tests with combinatorial optimization strategies, we noticed that for good solutions, many households share the exact same or at least very similar set of persons, based on their feature characteristics combination. An algorithm that can exploit those similarities could avoid the computational effort of combining redundant households. Instead, a "pattern" mechanism could be employed to clone households with similar properties to exhaust the global restrictions. This, and other speed-up ideas are for sure candidate of further research, but first prototypical implementations may have to suggest promising results.

Additionally, we have shown that our approach supports more sophisticated microsimulations of influenza propagations, since we can exploit specific properties of the diseases structure. Our population synthesis algorithm, was evaluated against the available input data and provided reasonable results. The subsequent task for us is now to examine, how a more precise baseline population effects the quality of microsimulations of influenza propagations. It will be one of the major issues of our further research.

Acknowledgment

The authors gratefully acknowledge the support of the *Nationale Forschungsplattform für Zoonosen,* grant to the *Institut für Molekulare Virologie*, and the Chair for Information Systems and Supply Chain Management at the University of Münster.

References

[1] Chao, D., Halloran, M. E., Obenchain, V. J. and Longini Jr, I.M. (2010) FluTE, a Publicly Available Stochastic Influenza Epidemic Simulation Model. PLOS Comp Biol, doi: 10.1371/journal.pcbi.1000656.

[2] Van den Broeck, Wouter, et al. "The GLEaMviz computational tool, a publicly available software to explore realistic epidemic spreading scenarios at the global scale." BMC infectious diseases 11.1 (2011): 1.

[3] Moeckel, Rolf, Klaus Spiekermann, and Michael Wegener. "Creating a synthetic population." Proceedings of the 8th International Conference on Computers in Urban Planning and Urban Management (CUPUM). 2003.

[4] Ryan, Justin; Maoh, Hanna; Kanaroglou, Pavlos (2009): Population Synthesis: Comparing the Major Techniques Using a Small, Complete Population of Firms. In: Geographical Analysis (41), S. 181–203.

[5] Melhuish, Tony, Marcus Blake, and Susan Day. "An evaluation of synthetic household populations for census collection districts created using optimisation techniques." Australasian Journal of Regional Studies, The 8.3 (2002): 369.

[6] Beckman, Richard J., Keith A. Baggerly, and Michael D. McKay. "Creating synthetic baseline populations." Transportation Research Part A: Policy and Practice 30.6 (1996): 415-429.

[7] Ma, Lu; Srinivasan, Sivaramakrishnan (2015): Synthetic Population Generation with Multilevel Controls: A Fitness-Based Synthesis Approach and Validations. In: Computer-Aided Civil and Infrastructure Engineering 30 (2), S. 135–150. DOI: 10.1111/mice.12085.

[8] Pritchard, David R.; Miller, Eric J. (2012): Advances in population synthesis: fitting many attributes per agent and fitting to household and person margins simultaneously. In: Transportation 39 (3), S. 685–704. DOI: 10.1007/s11116-011-9367-4.

[9] Gargiulo, Floriana, et al. "An iterative approach for generating statistically realistic populations of households." PloS one 5.1 (2010): e8828.

[10] Ye, Xin; Konduri, Karthik; Pendyala, Ram M.; Sana, Bhargava; Waddell, Paul (2009): A methodology to match distributions of both household and person attributes in the generation of synthetic populations. In: 88th Annual Meeting of the Transportation Research Board, Washington, DC.

[11] Voas, David; Williamson, Paul (2000): An evaluation of the combinatorial optimisation approach to the creation of synthetic microdata. In: International Journal of Population Geography 6 (5), S. 349–366.

[12] Huang, Zengyi; Williamson, Paul (2001): A comparison of synthetic reconstruction and combinatorial optimisation approaches to the creation of small-area microdata. In: Department of Geography, University of Liverpool.

[13] Deming, W. Edwards; Stephan, Frederick F. (1940): On a Least Squares Adjustment of a Sampled Frequency Table When the Expected Marginal Totals are Known. In: The Annals of Mathematic Statistics 11 (4), S. 427–444.

[14] Bayerisches Landesamt für Statistik und Datenverarbeitung (2013): Zensusdatenbank des Zensus 2011. Available online at: https://ergebnisse.zensus2011.de/, checked on 25.07.2016.

[15] Robert Koch Institut (2013): RKI - Impfquoten. Available online at: http://www.rki.de/DE/Content/Infekt/Impfen/Impfstatus/impfstatus_node.html, zuletzt aktualisiert am 11.02.2013, checked on 25.07.2016.

[16] Falkenauer, Emanuel (1994): A new representation and operators for genetic algorithms applied to grouping problems. In: Evolutionary computation 2 (2), S. 123–144.