Proceedings of the 2021 Winter Simulation Conference
S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo, and M. Loper, eds.

A GENERALIZED NETWORK GENERATION APPROACH FOR AGENT-BASED MODELS

Kristina Heß
Jan Himmelspach

Department of Computer Science
NORDAKADEMIE
Köllner Chaussee 11
25337 Elmshorn, GERMANY

Oliver Reinhardt
Adelinde M. Uhrmacher

Institute for Visual and Analytic Computing
University of Rostock
Albert-Einstein-Straße 22
18059 Rostock, GERMANY

ABSTRACT

For their initialization, many agent-based models require a population which corresponds in its essential characteristics to the examined real population. It should reflect the real distribution of attributes of interest, e.g., age, gender, or income, as well as the social network between the agents. Since a disaggregated data set with all required information is rarely available, a synthetic population must be created. Methods that assign realistic attribute values to agents are well studied in the literature. In contrast, the generation of plausible social networks has been less extensively researched, but several comprehensive adhoc models have been developed. The focus of this work is to introduce a reusable, generalized approach for the generation of synthetic social networks. Symbolic regression is used to automatically train generation rules based on a network sample, instead of having to define rules a priori. Manually specified constraints are taken into account to avoid implausible relationships.

1 INTRODUCTION

In Agent-Based Modeling (ABM), single individuals are modeled on the micro level as agents with heterogeneous characteristics and behaviors. These independent agents are able to act and interact with each other based on their individual state and their environment and will evolve during the simulation. The behavior of the system as a whole is composed of the micro level dynamics (Michel et al. 2009, p. 9). To initialize the simulation, many of these models need a population which reflects the real distribution of the social population of interest, e.g., age, gender, marital status or income, as well as the social network between the agents. The network includes for instance the assignment of partners, children, and friends, or the formation of households. For the creation of synthetic populations, detailed data is required which is difficult to collect and often limited due to privacy concerns (Chapuis and Taillandier 2019). Thus, such a synthesis is typically based on data from different sources which exists in differing aggregation levels and cannot be used directly to create such a population.

Consequently the generation of artificial initial data for models is a well-known challenge (cf. Section 2). In the research area of social science simulation, a number of researchers tried to approach a solution by exploiting a variety of methods. The creation of a population based on not-interconnected individuals is described in Section 2.1. In Section 2.2 existing approaches to generate relationships among individuals are shown. Based on the existing concepts we describe a novel approach to generate such initial (used at the start of a simulation) networked populations exploiting a constraint genetic programming algorithm.

1.1 Motivation

Within the research field of social science, Agent-Based Social Simulation (ABSS) is widely used to model complex systems and to gain insights on (artificial) societies in a controlled, laboratory-like environment

978-1-6654-3311-2/21/$31.00 ©2021 IEEE

(Michel, Ferber, and Drogoul 2009, p 10). Example applications are the prediction of care needs (Noble et al. 2012), the research of migration decisions (Klabunde et al. 2017) and the investigation of epidemics and the effectiveness of intervention measures (Kerr et al. 2020). Since the behavior of agents in such simulations is primarily modeled based on their attribute values, it is essential that the synthetic population, the basis of the model, reflects the real population under study (Chapuis and Taillandier 2019).

In social simulations a trend towards more complex and data driven models can be observed and has been introduced by Edmonds and Moss (2005) under the slogan "Keep it Descriptive, Stupid" (KIDS). Since then, several papers have discussed the KIDS approach and the need for more realistic models to leverage the strengths of ABM to analyze external and internal mechanisms (Pastrav and Dignum (2020); Kraan et al. (2019); Conte and Paolucci (2014)). More complex models also require populations that represent the distribution of a large number of attributes and satisfy several constraints regarding the social network. The complexity discussions in recent publications therefore also imply the relevance of efficient population synthesis approaches.

1.2 The Challenge

The proper initialization of models containing a population of arbitrary size, based on a sample of a population (see Figure 1a), implies the need to add artificially generated population members by keeping the ratios and correlations of every single attribute of individuals. In addition, for networked populations, the relationships within the generated population (see Figure 1b) have to match the original linkage probabilities. The data available may contain information about individuals (health status, age, education, ...) and the networks they are part of (marriage, parent-child, job, household, ...).

distributions of individual properties
distributions of individual properties
linkage propabilities

(a) without network information (b) with network information

Figure 1: Population synthesis creating larger populations from smaller samples.

In general every individual in a population could be connected to every other individual for each of the relevant network types (household, job, family, ...) leading to *n* categorized (named) edges between the individuals of the network.

Thereby the attributes as well as the existing edges (and their weights) have an influence on all individuals and their outgoing and incoming edges. Each of the agents has a number of attributes (e.g., age, interests, education, working place, job profile, parental status, ...) which influences the strength, type, and number of relationships. Further on, a number of relationships can never occur, e.g., a 5 year old female cannot be the mother of a 90 year old person.

2 RELATED WORK

As each population model and simulation requires a proper initialization several approaches have already been developed. Generally we can differentiate between non-network based approaches (cf. Section 2.1) and those focusing on the generation of networks (cf. Section 2.2).

2.1 Non-Network Based Approaches

Various methods to create populations of individuals are described in literature. These methods rely on a sample (i.e., statistical information about the sample) which is then used to create individuals in a larger population by keeping the same ratios as in the sample. Methods developed so far can be categorized (see Fabrice Yaméogo et al. (2021), Sun et al. (2018)) by the main population/distribution fitting method used.

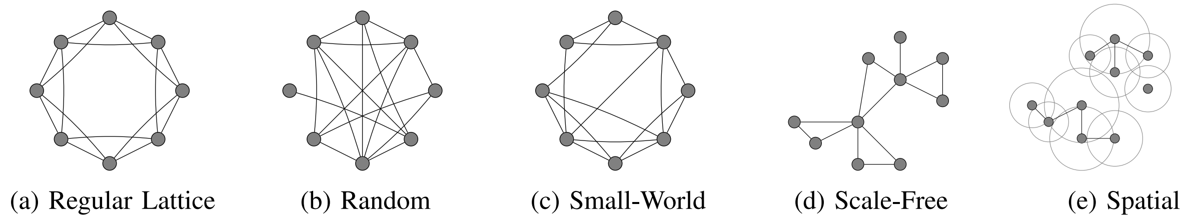
The "fitting" based approaches reweight a disaggregated sample to meet aggregated control totals. The general procedure can be divided into a fitting and a generation stage. The fitting stage is used to estimate the joint distribution of person level attributes (e.g., age, interests, ...) and household level attributes (e.g., number of individuals, type of residence, ...) based on the disaggregated and aggregated input data. During the subsequent generation phase, the original sample entries will be reweighted in accordance to the calculated joint distribution to generate the target population. (Müller 2017)

Like the fitting approaches, "Combinatorial Optimization" (CO) methods rely on a disaggregated sample and aggregated control totals to create suitable populations for multiple small regions (Williamson, Birkin, and Rees 1998). However, CO uses a different procedure to reweight the sample records. The population is not generated by applying weights, but by repeatedly modifying its composition until a quality criterion is reached (Chapuis and Taillandier 2019). While the previous approaches iteratively switch between the constraints, here all constraints are checked simultaneously at each step (Harland et al. 2012).

The term "Synthetic Reconstruction" (SR) has already been used by Williamson et al. (1998) to classify approaches which generate imaginary individuals and households, instead of cloning and reweighting existing data. Those early SR approaches used a sequential procedure to successively assign attribute values by sampling from known univariate probability distributions. Since these methods reconstruct data from statistical data, they are sometimes referred to as Statistical Learning (SL) (Sun et al. 2018; Fabrice Yaméogo et al. 2021), probabilistic simulation-based approaches (Fournier et al. 2020) or Markov process-based methods (Saadi et al. 2016).

2.2 Network Based Approaches

The fitting, CO and SR approaches described previously are suitable to represent synthetic populations with two hierarchy levels, such as individuals and households. In most cases, however, these population generation methods do not consider the interdependencies between household members and are not able to include other types of relationships. Since agent based models simulate interactions between individuals, it became common to describe the structure of these interactions as a social network (Thiriot and Kant 2008). The social network is often built on top of a synthetic population, which was generated with a method from Section 2.1. Table 1 provides an overview of existing social network generation approaches. The methods are divided into three categories. Abstract networks that follow "simple rules", which do not distinguish between individual characteristics. Typical types are regular networks (Figures 2a), full random networks (2b), neighborhood based networks (2c), and networks where properties are based on mathematical laws, e.g., degree distributions following power laws (2d). In "spatial networks", the distance between agents (2e), its physical or social space, determines the network. "Adhoc network models" summarizes all approaches that are strongly related to a specific modeling purpose. These cannot be reused easily for different use-cases.



(a) Regular Lattice (b) Random (c) Small-World (d) Scale-Free (e) Spatial

Figure 2: Comparison of network models.

Table 1: Overview of Network Generation Approaches

Abstract Network Models	• Follow simple rules • Do not distinguish between individual characteristics	
	Characteristics	**References**
Regular Lattices	• Derived from cellular automata • Each agent has an equal number of links	• Leydesdorff (2001)
Random Networks	• All links are created randomly • Link density is configurable	• Erdös and Rényi (1959)
Small-World Networks	• Random network with a reduced average path lenght	• Watts and Strogatz (1998)
Scale-Free Networks	• Preferential attachment model • Nodes with a higher degree are more likely to receive further links	• Barabasi et al. (2000) • Newman et al. (2002)
Spatial Network Models	• Consider spatial or social space • A cost is associated with the distance of agents	
	Characteristics	**References**
Spatially Extended Abstract Models	• Effects of space constraints on the structure of abstract networks	• Barthélemy (2011)
Social Circles Model	• Only individuals within reach of their social circles get linked	• Hamill and Gilbert (2009)
Adhoc Network Models	• Based on rules regarding the modeled system • Highly use case specific and rarely reusable	
	Characteristics	**References**
Hierarchic Networks	• Model hierarchical networks like an organizational structure	• Kim (2009)
Kinship Networks	• Focus on the formation of couples • Complete family structures are often created during a simulated warm-up period	• Noble et al. (2012) • Klabunde et al. (2017)
Activity-Based Approaches	• Advanced methods especially from the field of epidemiology • Based on physical co-location of indiviuals	• Eubank et al. (2004) • Kerr et al. (2020)
Rule-Based Approaches	• Use a Bayesian Network to model the probability of being linked in dependence on the characteristics of individuals	• Thiriot and Kant (2008)
Empirical Networks	• Use real network data with a similar size as the target population	• Cointet and Camille (2007)

3 A GENERALIZED SYNTHESIS APPROACH

The aforementioned approaches (see Section 2.1) provide the base for the novel network synthesis approach described here. So far the existing network approaches (Section 2.2) need manual adaptions to create a population for a concrete simulation study. In contrast, our generalized synthesis approach shall overcome this limitation by being reusable and thus reduce the efforts required.

The starting point for our algorithm is a population generated by one of the non-network based approaches mentioned before. This population will be enriched with a synthetic social network. The method is in principle independent of the particular non-network based approach chosen. But since the attributes of the synthetic individuals are used to decide which individuals are connected, the quality of the network also depends on the quality of the prior generated population. The approach is built upon the work of Menezes and Roth (2014), who use symbolic regression to automatically identify generative network models. Generative network models are used to describe rules that create networks with specific characteristics. Those models, like for example the abstract network models (Figure 2), are often built based on intuition or on established theories of relationship formation. Symbolic regression is a genetic programming technique which searches over the space of mathematical formulas to identify the one that describes a given dataset best. By describing a generative network model in the form of a mathematical formula, Menezes and Roth (2014) apply this method to automatically identify potentially counter intuitive network generation rules. They compare the symbolic regression with an "artificial scientist" who is able to discover models which generate real world network topologies that can differ significantly from the abstract ones (Figure 2). The approach was kept very generic and used to analyze social networks, as well as networks from other domains, such as a network of protein interactions.

The generalized approach introduced in this chapter combines ideas of the adhoc network models with the concept of automatic model identification based on a representative network sample. In contrast to the adhoc models, the novel network generator is not designed for a specific simulation target or domain, which makes it applicable for different scenarios. The symbolic regression procedure was transferred to the problem of population synthesis. For this purpose, the original approach was extended to identify network generation rules that consider person attributes. Thus, not only the structure of the social network is taken into account, but also the characteristics of the individuals who are linked to each other. Previously, only topological network properties like the node degree were incorporated. In addition, the user can specify constraints to avoid implausible relationships. The rules identified by the approach can now be applied to another potentially larger population with different characteristics, since not only abstract nodes with incremental IDs are processed like before, but persons with attributes. Table 2 summarizes the characteristics of the generalized approach in comparison to the aforementioned network synthesis approaches (Table 1).

Table 2: Characterization of the generalized synthesis approach.

	Characteristics
Generalized approach	• Automatically identify rules based on a network sample (genetic programming) • No a priori hypotheses necessary • Link individuals based on their characteristics • Constraints can be specified externally to enforce only plausible relationships

3.1 A General Algorithm

Figure 3 illustrates the steps within the symbolic regression procedure. The basic idea behind the algorithm is that a mathematical weight function is used to calculate how likely two persons of the synthetic population are linked. The symbolic regression is used to automatically discover this weight function based on a sample network. The most important steps, performed iteratively within the procedure, are the mutation of a previous weight function, the subsequent generation of a network by applying the new function, and finally the evaluation whether an improved network could be obtained. The network generation step is executed during each iteration to be able to assess the fitness of a new weight function. The similarity between the generated network and the sample network is measured based on several criteria regarding the network topology, as well as the attribute distributions of linked individuals. The optimization procedure

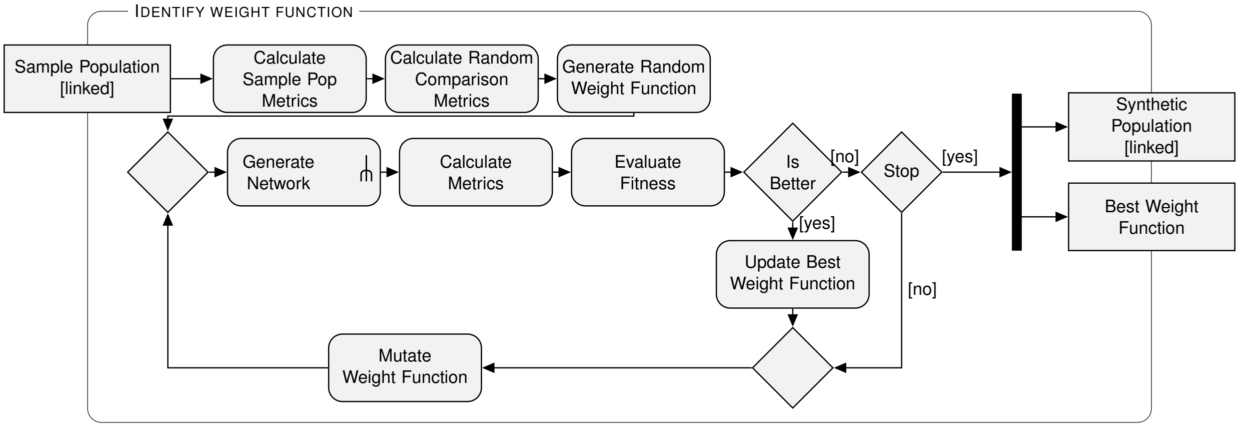

IDENTIFY WEIGHT FUNCTION
Sample Population [linked]
Calculate Sample Pop Metrics
Calculate Random Comparison Metrics
Generate Random Weight Function
Generate Network
Calculate Metrics
Evaluate Fitness
Is Better
[no]
Stop
[yes]
[yes]
Update Best Weight Function
[no]
Mutate Weight Function
Synthetic Population [linked]
Best Weight Function

Figure 3: Symbolic regression.

is stopped if within a configurable number of iterations no further weight function could be found that generates a network with an improved fitness. The result of the symbolic regression is on the one hand the best discovered weight function and on the other hand the social network that was generated using this function.

The network generation subroutine (Figure 4) is used within the symbolic regression, but can also be applied independently of it. After training a weight function based on the network sample, the identified function can be used to generate the network for a larger target population. For each edge to be generated, the same processing steps are performed. First, a subset of potential edges is chosen randomly, since for a large number of nodes the weights cannot be calculated for all pairings at every step. Next, it is ensured that the edge candidates do not violate any externally specified constraints. For all valid candidates, the weight function is evaluated. The ratio of a single weight to the sum of all weights is interpreted as the occurrence probability of one edge. One edge is selected randomly according to these probabilities. The procedure is repeated until the desired number of edges is present.

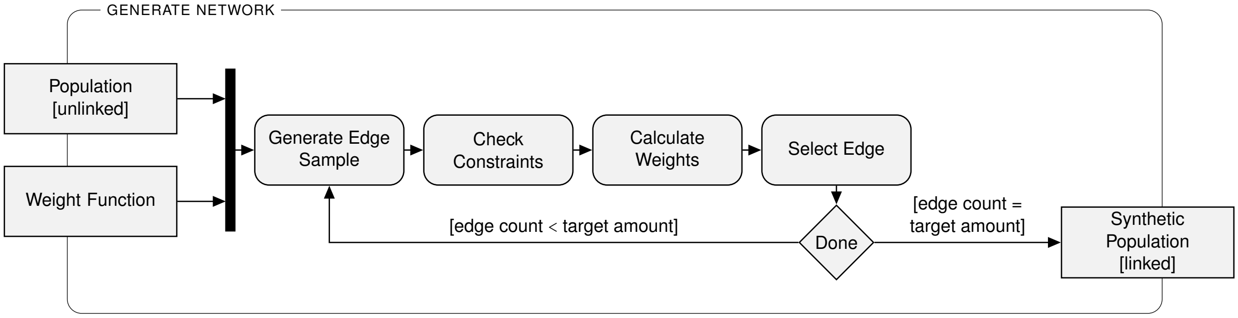


Figure 4: Network generation.

3.2 Weight Function

The weight function is a generalized representation form of the rules that can be applied to generate a social network. The function is a simplified representation of the link probability distribution in the multidimensional space of all incorporated attributes and topological variables of two individuals. The advantage compared to other forms of representation, such as multiple probability distributions organized in a Bayesian Network, is that it consolidates all relevant information in a single mathematical function. As in Menezes and Roth (2014), the weight function is represented internally as a symbolic expression tree. The tree leaves consist of person level attributes, topological variables, and constants, while all other nodes represent operations. These operations can be mathematical functions (e.g., `+`, `−`, `LOG`, `MIN`, `MAX`), or logical operations (e.g., `>`, `<`, `AND`, `NOT`). Numeric attributes can be used in mathematical functions or for comparisons and alphabetic attributes can be checked for equality. These building blocks can, for example, consider age differences or a common interest. The root of the tree must consist of a mathematical function, a numeric attribute or a numeric constant to evaluate the function to a number that can be interpreted as the weight. An `IF` condition can be used to combine mathematical and logical operations.

Weight functions are generated by randomly selecting operations, variables and constants. The randomness is limited by a configurable maximum size of the function and by a restriction that only the same variable type is allowed to be used at the same level of a subtree, as indicated by the blue frame in Figure 5. This prevents calculating directly with different types of variables and units. Evolutionary algorithms are characterized by the survival of superior solutions and the emergence of new ones through mutation and recombination. The tree representation enables the mutation of a weight function by randomly choosing a position in the tree and exchanging the subtree below it (Figure 5). During the symbolic regression, a new weight function is created by either mutating a previous function that achieved a good fitness, or by randomly creating a completely new function to achieve a higher diversity. Two types of functions survive during the optimization procedure. One of these is the function that generated the network with the best fitness so far. In addition, the shortest function, whose fitness is only within a configurable tolerance range below the best function, is retained. This principle is used to prevent that too complicated functions evolve. If two functions generate equally well networks, the shorter one is preferred as it is easier to understand. Large functions are very likely to contain bloat, i.e., building blocks that have no effect on the evaluated weight.

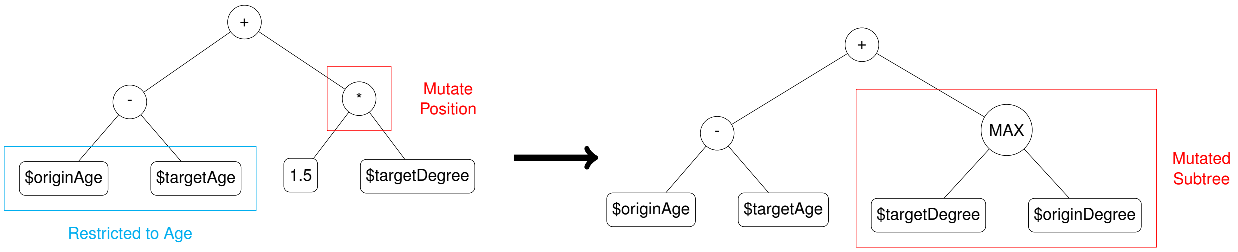


Figure 5: Mutating a weight function.

3.3 Fitness Evaluation

The weight function is trained to create a social network with similar statistical properties as the sample network. The similarity is measured based on several frequency distributions regarding the network topology, as well as the attributes of linked individuals. The distance of the degree distributions and the difference in the occurrence of triadic relations were adopted from Menezes and Roth (2014) to compare the network topologies. To evaluate how well the attributes of linked persons comply with the sample network, for each attribute the frequency distribution of all possible pairwise combinations is calculated, e.g., comparing the number of relationships between two women, two men, and a woman and a man for the attribute gender. To avoid having to calculate the frequencies for too many combinations, numeric attributes are grouped into categories, e.g., using groups that span 10 years for the attribute age. The distance between two attribute frequency distributions is calculated using the Earth Mover's Distance. As an additional indicator, the adjacency lists of both networks are compared to determine how many edges match exactly. To compare all attribute and topology related metrics on a standardized level, the metrics are set in relation to the average distances of multiple randomly generated networks. These relative metrics express the percentage to which a synthetic network is closer to the original statistics than a randomly generated network. The target of the symbolic regression is either to minimize the average relative distance or the worst relative distance.

3.4 Constraints

To ensure the network consists only of plausible relationships, constraints need to be considered beside the link probabilities. The formulation of constraints requires domain knowledge and assumptions about the network to be modeled. Therefore, they need to be specified externally by using the same symbolic expression language as the weight functions. For example, for a mother-child relationship it can be defined that the mother must be female and older than the child: `(AND (EQ $originSex "female") (> $originAge $targetAge)`. Further conditions can be added to formulate the constraints as strict as required. In contrast to the weight functions, constraints must have a logical root operation to evaluate to a truth value. Constraints are always checked before any weights for potential edges are calculated. Only pairings that do not violate any constraints are further considered.

4 REALIZATION

The generalized approach has been implemented based on the work of Menezes and Roth (2017) in Java (the code can be accessed at https://doi.org/10.5281/zenodo.5043126). The class diagram (Figure 6) illustrates the structure of the main components. The `SymbolicRegression` is the starting point of the application and responsible to train a weight function. From here, the creation of weight functions, the application of these functions to generate networks and the assessment of the fitness are coordinated. To perform these steps, a `SymbolicExpressionFactory`, a `NetworkGenerator` and a `FitnessCalculationService` are used.

The `SymbolicExpressionFactory` is responsible to generate random weight functions and to mutate existing ones. The factory is able to consider every numeric or alphabetic person level attribute of the input population "out of the box". It is only necessary to specify which attributes should be taken into account. All mathematical operations can be performed on numeric attributes and alphabetic attributes can be tested for equality. If a specific characteristic is not directly evident from an attribute, a custom variable can be introduced. A related evaluation function needs to be implemented to teach the generator how to evaluate this variable regarding two potentially related individuals (i.e., an `EdgeCandidate`). For example, this could be a function that evaluates during the network generation if two persons have a mutual friend. The information is based on the relationships that already exist at that point in the generation process. The fitness function will automatically consider all person attributes and custom variables by calculating the frequency distributions. For example, for the custom variable that determines mutual friendships it is calculated how many friendships exist between persons that have a mutual friend and how many do not have a mutual friend. The weight function supports three types of expressions: numeric, logical, and alphabetic. The common characteristics of these types were summarized in the abstract classes: `AbstractLogicalExpression`, `AbstractNumericExpression` and `AbstractAlphabeticExpression`. The expression type describes which data type is returned when the expression is evaluated. An operation consists of further expressions, which represent the operands. A condition, for example, is numeric and consists of both logical and numeric operands: `(IF AbstractLogicalExpression AbstractNumericExpression AbstractNumericExpression)`. Each specific expression defines which type of operands it contains, except for variables and constants that do not contain any child elements. To evaluate an expression, an `EdgeCandidate` is passed to the root node. Then each node in the tree delegates the evaluation to its children, until a variable or constant is reached. The results of the subtrees are then consolidated upwards back to the root.

The `NetworkGenerator` receives an `AbstractNumericExpression` as a weight function and applies it to generate the desired number of edges for a given population. The sample ratio defines how many potential edges should be considered during each iteration. To be able to evaluate custom variables, the generator stores the related evaluation functions. Regardless of the weight function, the generator can be provided with a list of `AbstractLogicalExpressions` as the constraints. The user input for weight functions and constraints is done via the prefix notation. Expressions can thus be read and parsed from a file without having to extend the code. In addition to a trained weight function, manually set up functions or a manually adjusted function can be applied. In this way, various constraints and rules can be tested with little manual effort.

5 EVALUATION

To evaluate our approach, we set up a test scenario based on the initialization of a model of migration from Senegal to Europe by Klabunde et al. (2017). The authors examine the migration decision processes of potential migrants in Senegal. This decision process is conditional on the potential migrant's personal situation (e.g., income and cost of living), environmental factors (e.g., border control), and their social network. They will be more willing to migrate, if they have friends with positive migration experiences, or if migration would lead to family reunification. On the other hand, having children, or having friends

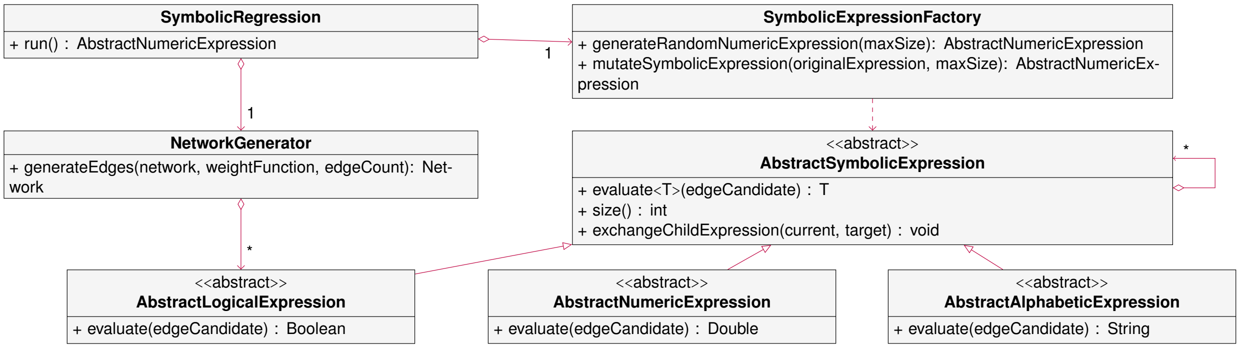

SymbolicRegression
+ run() : AbstractNumericExpression
SymbolicExpressionFactory
+ generateRandomNumericExpression(maxSize): AbstractNumericExpression
+ mutateSymbolicExpression(originalExpression, maxSize): AbstractNumericExpression
1
1
NetworkGenerator
+ generateEdges(network, weightFunction, edgeCount): Network
<<abstract>>
AbstractSymbolicExpression
+ evaluate<T>(edgeCandidate) : T
+ size() : int
+ exchangeChildExpression(current, target) : void
*
*
<<abstract>>
AbstractLogicalExpression
+ evaluate(edgeCandidate) : Boolean
<<abstract>>
AbstractNumericExpression
+ evaluate(edgeCandidate) : Double
<<abstract>>
AbstractAlphabeticExpression
+ evaluate(edgeCandidate) : String

Figure 6: Selected classes of the network generation system.

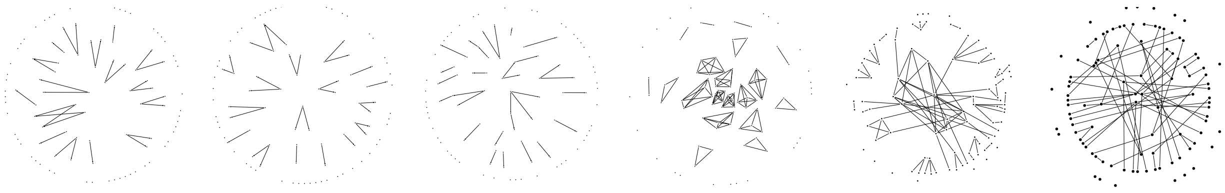
who failed their attempt to migrate, might impact their attitude to migration negatively, or might even make it completely infeasible. Hence, the social structure of the simulated population, including the initial population, has significant impact on the simulation.

For their initialization, Klabunde et al. (2018) produced an initial population, with the individual agents sampled from survey data. The individuals were then linked, forming married couples, assigning children to parents, creating household, and friendship networks. We followed the authors' documentation to produce two synthetic networked populations of different sizes (500 (Pop_A) and 2000 (Pop_B) individuals). We use the smaller one (Pop_A), to train our generator, which is then used to synthesize a network for the individuals of the larger one. The network synthesized by our generator is then compared to the original network of the larger population (Pop_B).

The different relationship types were trained independently of each other. As these social networks have very diverse topological characteristics (e.g., marriages between 2 individuals in contrast to friendships with larger clique formations), they are well suited as test scenarios for a network generator. Furthermore, a distinction can be made between undirected (e.g., friendships) and directed relationships (e.g., child-parent relationships). The friendship network was trained without using any restrictions, while reasonable constraints were specified for the parental relationships. The results for the 2000 individuals population were compared visually with the original and randomly generated networks (e.g., Figure 7) and examined based on the fitness statistics (e.g., Figure 8 and Figure 9). The test scenario showed that the novel network generator is able to detect individual characteristics that have an effect on the formation of relationships. As an example, the results are illustrated for the child-parent and the friendship network.

Both the synthetically generated child-parent and the friendship network differ from the randomly generated networks and approximate the original ones (Figure 7). At the child-parent network, the influence of the constraints can be well observed. The generator has been restricted so that each child can be assigned to at most two parents and circular relationships were excluded by age restrictions. In the random graph (Figure 7c), also unwanted relationships occur, e.g., children with three parents. The original friendship network, on the other hand, consists of many cliques (Figure 7d). The synthetic network (Figure 7e) also shows some smaller friendship cliques, but these are not as separated from each other as in the original network. Regarding the frequency distributions, some attributes could be reflected significantly better than others. The weight function identified for the child-parent network weights edge candidates stronger if a child has already been assigned to a father (`(IF (NOT $hasFather) 0.5509 20.28)`). This leads to a very good replication of single mother and two parents relationships, while single fathers were not observed in the sample network (Figures 8a and 8b), but occured often at the random network. The node degree distributions (Figures 8c and 8d) likewise show that the synthetic and original networks are more similar in terms of number of parents per child and number of children per parent

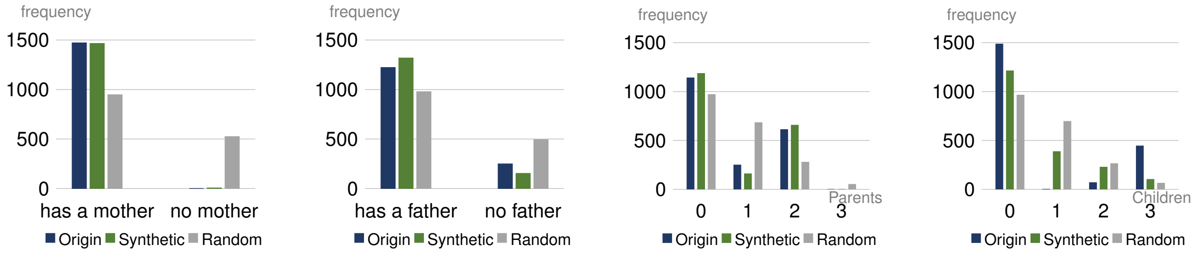
The weight function discovered for the friendship network weights edges among neighbors stronger (`(IF $areNeighbors 116.0 (^ 0.9345 116.2395))`), whereby this attribute statistic could be reproduced very well (Figure 9a). In contrast, friendships with mutual friends were generated only slightly



(a) Origin (b) Synthetic (c) Random (d) Origin (e) Synthetic (f) Random

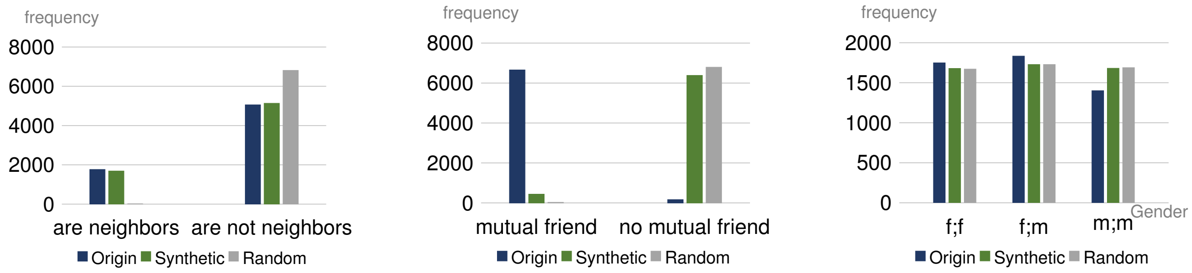
Figure 7: Comparison of two origin 2000 individual networks with synthetically and randomly created networks of the same size. Figure 7a, 7b, and 7c represent directed child-parent networks, while 7d, 7e, and 7f are undirected friendship networks. Extracts are shown for 100 randomly selected nodes (preferring nodes linked to already selected ones), arranged by the node degree.

more frequently than in the random case (Figure 9b). The gender distribution is an example that some attributes have only a marginal influence and an uniform distribution approximates it already well (Figure 9c). In general, it is challenging to find a single weight function that approximates all fitness criteria equally well. Tests with a single fitness criterion showed that the symbolic regression can approximate a single attribute distribution clearly better than multiple simultaneously.



(a) Child - Mother (b) Child - Father (c) Number of parents (d) Number of children

Figure 8: Statistics regarding the child-parent networks.



(a) Neighborhood (b) Mutual friends (c) Gender

Figure 9: Statistics regarding the friendship networks.

6 SUMMARY AND FUTURE WORK

Extending previous work, we presented a novel generalized algorithm based on genetic programming. The network generator has been developed to provide a reusable approach which reduces the efforts required to adapt the existing adhoc approaches for another simulation study. The experiments conducted so far show that the approach is capable to discover generative network models that consider individual attributes as well as the network topology. Parts of the fitness function are formulated generically to measure the distribution of available attributes and can be used without adjustments. Further we have been able to approximate the manually generated networks of Klabunde et al. (2018) by applying a trained weight function to a larger population, which shows that the principal approach is suitable for different kind of relationships.

The difficulty in genetic programming lies in the high number of available parameters to optimize the learning process and the challenge to find suitable fitness criteria for multiple use cases. More experiments need to be done to examine the sensitivity of the algorithm regarding these parameters and to determine which and how many fitness criteria lead to the best results. Further on, the quality of the generated networks should be compared to existing adhoc approaches and the saved efforts need to be quantified. As the algorithm uses statistical means, the robustness of the generation procedure, when applying the same weight function several times, should be evaluated.

The generalized algorithm is able to detect simplified relationship formation effects and to comply with predefined constraints. Heterogeneous networks consisting of subgraphs with different characteristics or effects between several network layers, such as the influence of having children on the formation of friendships, were out of scope for this study. A disadvantage of the symbolic regression approach is that a representative network sample is required. However, since the function is trained only according to the statistical properties of the sample, further experiments should be conducted to evaluate to use scattered statistics from different sources for the fitness function instead of a sample network.

7 ACKNOWLEDGMENTS

This research received funding from the German Research Foundation (DFG) via the research grant Modeling and Simulation of Linked Lives in Demography (UH–66/15).

REFERENCES

Barabasi, A.-L., R. Albert, and H. Jeong. 2000, June. “Scale-Free Characteristics of Random Networks: The Topology of the World-Wide Web”. *Physica A: Statistical Mechanics and its Applications* 281:69–77.

Barthélemy, M. 2011, February. “Spatial Networks”. *Physics Reports* 499(1):1–101.

Chapuis, K., and P. Taillandier. 2019, September. “A Brief Review of Synthetic Population Generation Practices in Agent-Based Social Simulation”. Technical Report 15-009, Network Dynamics and Simulation Science Laboratory, Virginia Tech.

Cointet, J.-P., and R. Camille. 2007, June. “How Realistic Should Knowledge Diffusion Models Be?”. *Journal of Artificial Societies and Social Simulation* 10(3):1–5.

Conte, R., and M. Paolucci. 2014, July. “On Agent-Based Modeling and Computational Social Science”. *Frontiers in Psychology* 5(668):1–9.

Edmonds, B., and S. Moss. 2005. “From KISS to KIDS – An ‘Anti-Simplistic’ Modelling Approach”. In *Multi-Agent and Multi-Agent-Based Simulation*, edited by P. Davidsson, B. Logan, and K. Takadama, 130–144. Berlin, Heidelberg: Springer Berlin Heidelberg: Springer Berlin Heidelberg.

Erdös, P., and A. Rényi. 1959. “On Random Graphs”. *Publicationes Mathematicae* 6:290–297.

Eubank, S., H. Guclu, V. S. Anil Kumar, M. V. Marathe, A. Srinivasan, Z. Toroczkai, and N. Wang. 2004, May. “Modelling Disease Outbreaks in Realistic Urban Social Networks”. *Nature* 429(6988):180–184.

Fabrice Yaméogo, B., P. Gastineau, P. Hankach, and P.-O. Vandanjon. 2021, November. “Comparing Methods for Generating a Two-Layered Synthetic Population”. *Transportation Research Record: Journal of the Transportation Research Board* 2675(1):136–147.

Fournier, N., E. Christofa, A. P. Akkinepally, and C. L. Azevedo. 2020, February. “Integrated Population Synthesis and Workplace Assignment Using an Efficient Optimization-Based Person-Household Matching Method”. *Transportation* online:1.

Hamill, L., and N. Gilbert. 2009. “Social Circles: A Simple Structure for Agent-Based Social Network Models”. *Journal of Artificial Societies and Social Simulation* 12(2):1–3.

Harland, K., A. Heppenstall, D. Smith, and M. Birkin. 2012. “Creating Realistic Synthetic Populations at Varying Spatial Scales: A Comparative Critique of Population Synthesis Techniques”. *Journal of Artificial Societies and Social Simulation* 15(1):1.

Kerr, C., R. Stuart, D. Mistry, R. Abeysuriya, G. Hart, K. Rosenfeld, P. Selvaraj, R. Nunez, B. Hagedorn, L. George, A. Izzo, A. Palmer, D. Delport, C. Bennette, B. Wagner, S. Chang, J. Cohen, J. Panovska-Griffiths, M. Jastrzebski, and D. Klein. 2020, May. “Covasim: An Agent-Based Model of COVID-19 Dynamics and Interventions”. Preprint, medRxiv.

Kim, W.-S. 2009, June. “Effects of a Trust Mechanism on Complex Adaptive Supply Networks: An Agent-Based Social Simulation Study”. *Journal of Artificial Societies and Social Simulation* 12:1.

Klabunde, A., S. Zinn, F. Willekens, and M. Leuchter. 2017. “Multistate Modelling Extended by Behavioural Rules: An Application to Migration”. *Population Studies* 71(sup1):51–67.

Klabunde, A., S. Zinn, F. Willekens, and M. Leuchter. 2018. “Multistate Modeling Extended by Behavioral Rules (Version 1.6.0)”. Model, CoMSES Computational Model Library. https://www.comses.net/codebases/5146/releases/1.6.0/, accessed March 17th, 2021.

Kraan, O., S. Dalderop, G. J. Kramer, and I. Nikolic. 2019. “Jumping to a Better World: An Agent-Based Exploration of Criticality in Low-Carbon Energy Transitions”. *Energy Research & Social Science* 47:156–165.

Leydesdorff, L. 2001, June. “Technology and Culture: The Dissemination and the Potential ’Lock-In’ of New Technologies”. *Journal of Artificial Societies and Social Simulation* 4:1.

Menezes, T., and C. Roth. 2014, September. “Symbolic Regression of Generative Network Models”. *Scientific Reports* 4:1–7.

Menezes, T., and C. Roth. 2017, May. “Automatic Discovery of Families of Network Generative Processes”. In *Dynamics On and Of Complex Networks III*, 83–111. Springer Proceedings in Complexity. Springer, Cham.

Michel, F., J. Ferber, and A. Drogoul. 2009, May. “Multi-Agent Systems and Simulation: A Survey from the Agent Community’s Perspective”. In *Multi-Agent Systems: Simulation and Applications*, edited by A. M. Uhrmacher and D. Weyns, 3–52. CRC Press.

Müller, K. 2017. *A Generalized Approach to Population Synthesis*. Ph. D. thesis, ETH Zurich / SNF / ETH Zurich, Zurich.

Newman, M. E. J., D. J. Watts, and S. H. Strogatz. 2002. “Random Graph Models of Social Networks”. *Proceedings of the National Academy of Sciences* 99(suppl 1):2566–2572.

Noble, J., E. Silverman, J. Bijak, S. Rossiter, M. Evandrou, S. Bullock, A. Vlachantoni, and J. Falkingham. 2012, June. “Linked Lives: The Utility of an Agent-Based Approach to Modeling Partnership and Household Formation in the Context of Social Care”. In *Proceedings of the 2012 Winter Simulation Conference*, edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Pastrav, C., and F. Dignum. 2020. “Norms in Social Simulation: Balancing Between Realism and Scalability”. In *Advances in Social Simulation*, edited by H. Verhagen, M. Borit, G. Bravo, and N. Wijermans, 329–342. Cham: Springer International Publishing: Springer International Publishing.

Saadi, I., A. Mustafa, J. Teller, B. Farooq, and M. Cools. 2016, August. “Hidden Markov Model-Based Population Synthesis”. *Transportation Research Part B: Methodological* 90:1–21.

Sun, L., A. Erath, and M. Cai. 2018, August. “A Hierarchical Mixture Modeling Framework for Population Synthesis”. *Transportation Research Part B Methodological* 114:1.

Thiriot, S., and J.-D. Kant. 2008, September. “Generate Country-Scale Networks of Interaction from Scattered Statistics”. In *ESSA 2008, European Social Simulation Association Conference*. ESSA.

Watts, D. J., and S. H. Strogatz. 1998, June. “Collective Dynamics of ‘Small-World’ Networks”. *Nature* 393(6684):440–442.

Williamson, P., M. Birkin, and P. H. Rees. 1998. “The Estimation of Population Microdata by Using Data from Small Area Statistics and Samples of Anonymised Records”. *Environment and Planning A: Economy and Space* 30(5):785–816.
_eprint: https://doi.org/10.1068/a300785

AUTHOR BIOGRAPHIES

KRISTINA HEß is a masters student of Software Engineering in the Department of Computer Science at the NORDAKADEMIE. Her research interests are in mathematical and algorithmic approaches for modeling and simulation. Her email address is kristina-hess@live.de.

OLIVER REINHARDT is a Ph.D. student in the Modeling and Simulation Group at the University of Rostock. He holds an MSc in Computer Science from the University of Rostock. In his research, he is concerned with domain-specific modeling languages and the methodology of agent-based simulation. His email address is oliver.reinhardt@uni-rostock.de.

JAN HIMMELSPACH is a professor in the Department of Computer Science at the NORDAKADEMIE. He earned his Ph.D. in Modeling & Simulation from the University of Rostock. His research interests include software engineering and modeling & simulation. His email address is jan.himmelspach@nordakademie.org.

ADELINDE M. UHRMACHER is Professor at the Institute for Visual and Analytic Computing of the University of Rostock and head of the Modeling and Simulation Group. She holds a PhD in Computer Science from the University of Koblenz and a Habilitation in Computer Science from the University of Ulm. Her email address is adelinde.uhrmacher@uni-rostock.de.