Urban life: a model of people and places

Andreas Züfle[1] · Carola Wenk[2] · Dieter Pfoser[1] · Andrew Crooks[3] ·
Joon-Seok Kim[1] · Hamdi Kavak[1] · Umar Manzoor[4] · Hyunjee Jin[1]

Abstract

We introduce the Urban Life agent-based simulation used by the Ground Truth program to capture the innate needs of a human-like population and explore how such needs shape social constructs such as friendship and wealth. Urban Life is a spatially explicit model to explore how urban form impacts agents' daily patterns of life. By meeting up at places agents form social networks, which in turn affect the places the agents visit. In our model, location and co-location affect all levels of decision making as agents prefer to visit nearby places. Co-location is necessary (but not sufficient) to connect agents in the social network. The Urban Life model was used in the Ground Truth program as a virtual world testbed to produce data in a setting in which the underlying ground truth was explicitly known. Data was provided to research teams to test and validate Human Domain research methods to an extent previously impossible. This paper summarizes our Urban Life model's design and simulation along with a description of how it was used to test the ability of Human Domain research teams to predict future states and to prescribe changes to the simulation to achieve desired outcomes in our simulated world.

1 Introduction

The purpose of the "Urban Life" agent-based model was to act as a sandbox environment for Human Domain research teams to assess different methods and tools for analyzing complex social phenomena in a simulated world. It was designed by a team of geographers, computer scientists, and computational social scientists from George Mason University, Tulane University and the University at Buffalo. The

---

✉ Andreas Züfle
  azufle@gmu.edu

Extended author information available on the last page of the article

model simulates an stylized urban setting on an Earth-like planet. Similar to Earth in the early 21st century, agents live, work, eat, and carry out recreational activities. This introduction gives an overview of the main technical and intellectual contributions of this simulation and provides references to sections and publications where additional details can be found.

1.1 Procedural city generation

Our procedurally generated world has buildings that are connected through a spatially explicit road network (Kim et al. 2018). Agents move across this network to find places (i.e., locations) to eat, to work, to find shelter (i.e., homes), to follow recreational activities, and to meet their friends. While agents are carrying out recreational activities, they can strengthen their friendship and make new friends. Details on how spatial road networks and places are generated to create a plausible urban environment for our agents to live in is described in Sect. 2.

1.2 Simulation of patterns of life

In our Urban Life model, the set of needs for the agents are based on a Maslow-like (Maslow 1943) model. Needs of agents include the need for shelter, food, sleep, financial safety, love, and esteem. Needs trigger behavior of agents to achieve goals that satisfy these needs, such as going to a restaurant for food or going home to sleep. This behavior leads to emerging patterns of daily human life (e.g., commuting to and from work, eating, sleeping, etc.) in a reflexive way. A brief description of our implemented needs that lead to emerging patterns of life can be found in Sect. 3.1. Additional details on how needs, triggers, actions, behavior, and goals of agents are implemented can be found in Kim et al. (2019a).

1.3 Scalability

One challenge we face when developing the model was that generally speaking, urban life simulations are usually complex and well-known agent-based modeling toolkits such as Wilensky (1998) or MASON (Luke et al. 2018) have scalability limitations, especially when the model and underlying spatial infrastructure is complex (i.e., large numbers of agents, detailed geometric structures). Therefore, to support large-scale urban life simulations, we designed a framework by integrating the multi-agent systems toolkit JADE (Bergenti et al. 2014) with the MASON agent-based modeling framework and its GIS extension, GeoMASON (Sullivan et al. 2010). Due to space limitations, we refer readers to Manzoor et al. (2021) for further details about this framework, but in essence, this framework allows us to simulate large areas with hundreds of thousands of agents for years of simulation time using

5-min simulation ticks without sacrificing the model generality. The model itself uses MASON and is written in Java

1.4 Complex agents

The purpose of the Urban Life model was to provide a sandbox environment in which social scientists (in our specific case, Human Domain research teams of the DARPA Ground Truth program) can investigate research questions by obtaining related data from simulations. For example, social scientists may seek to understand how agents decide which places to visit or how social connections are formed. Details on the ground truth model of the agents used in the Urban Life simulation are found in Sect. 3, including details on how agents choose places to visit (Sect. 3.2) and how agents tie and break social links (Sect. 3.3).

1.5 Advanced concepts of human behavior

The Ground Truth program challenged research teams to infer the ground truth causality of our Urban Life model to understand how our world functioned, how its future can be predicted, and how changes can be prescribed to improve the future. In our model, urban life included basic ground truth concepts such as hunger leading to agents seeking restaurants to eat, or working to earn money. While such causality can be easily guessed by transferring understanding from the real world, we implemented advanced concepts that are socially plausible but that are not as easily guessed. Specifically, we introduced the concepts of "ascension" and "flashmobs" (which are described in Sects. 3.4 and 3.5, respectively) to test if social scientists could explain the model dynamics based on simulation outputs as will be described further below and in Sect. 4. Following the ascension concept, agents that are wealthy and socially connected make their best effort to ascend from the world. The ascension process involves a nomination and voting schema. Following the flashmob process, agents aim to expand their social network (to receive more votes for ascension) by collectively visiting a site and imposing their collective traits on the site. Such a collective behavior is only triggered when agents with the same interest experience limited social contacts over a longer period of time.

1.6 Challenge design

For the simulation to be used as a sandbox for Human Domain research teams to test their understanding of the (unknown for them) causality, our model was instantiated into a simulation for which observable data was generated and challenges were defined to predict future states of the simulation and to inject interventions into the simulation to achieve a desired outcome (further details about this is provided in Sect. 4). By only providing observable data (and hiding the underlying simulation code), this sandbox allowed us to evaluate methods and tools for analyzing complex

social phenomena. Using the Urban Life model, we defined two challenges: (1) A predict test to challenge Human Domain research teams to gain sufficient understanding of our world to predict future states, and (2) a prescribe test to challenge Human Domain research teams to inject a limited number of interventions (changes) into the simulation to achieve a desired future outcome (see Sect. 4 for further details).

1.7 On-the-fly simulation interventions

As part of the DARPA Ground Truth program, human domain research teams were able to request additional information by submitting a wide range of research requests. Such research requests could include agent surveys, passive data collection mechanisms, additional journal data (i.e., locations visited, friends met), social network information, as well as experiments allowing to prescribe changes to the simulation and observe consequences. Changing simulation states on-the-fly is often conducted ad-hoc and entails manual code adjustments, which are time-consuming and error-prone. Therefore, to facilitate Human Domain research team simulation state change related requests, we developed an innovative injection mechanism to automatically inject prescribed changes into the simulation on-the-fly, for details see Kim et al. (2019b).

1.8 Roadmap

In the remainder of this paper, we provide the details of the aforementioned concepts. Section 2 explains how we generated the stylized urban environment using procedural city generation. In Sect. 3, we provide details about agents and their behaviors. Specifically, Sect. 3.1 describes the daily patterns of life driven by Maslowian needs that our agents, while Sect. 3.2 describes how agents choose their home location, work location, restaurants, and recreational sites. Section 3.3 discusses social network formation and Sect. 3.4 provides addresses the advanced concepts of ascension and flashmobs. In Sect. 4, we present two challenges: the Predict Test and the Prescribe Test. Both are used to test the capability of Human Domain research teams to predict future simulation states and to prescribe changes to the simulation towards a desired outcome. Section 4.4 provides simulation results showing that our simulation yields low variability for a viable Predict Test and provides red-teaming[1] results of the proposed Prescribe Test. We further added a disease model to our simulation described in Sect. 5, to create further challenges for Human Domain research teams to predict the spread of an infectious disease and to prescribe mitigation measures to stop the spread. Finally, Sect. 6 summarizes the contributions of this work.

---

[1] Red teaming is the practice of challenging tasks developed by a (blue) team as an adversary to approach the tasks from a more objective perspective. In short, our red teaming results present the results of our own efforts of solving our own challenges.

2 Urban environment and procedural city generation

Similar to many agent-based models, the activities of the agents in our model are subject to their surrounding environment along with other agents in the system (Crooks et al. 2019). For instance, the distance between an agent's home and workplace determines the commute time. If the commute time is too long, it may reduce participation in other social activities due to limited remaining time. Another example is social network formation. Agents are more likely to interact with their neighbors rather than with others living further away (cf. Sect. 3.3). Thus, the makeup of the urban environment, i.e., the built environment including transportation networks, plays an important role in our simulation (Crooks et al. 2015).

One of our goals is to provide a sandbox environment for social science research by generating an artificial world. The realization of urban environments consists of two steps: map generation and instantiation of urban components. The first step is the generation of synthetic transportation networks. Alternatively, real world maps can be used in the simulation (an example of which is shown in Sect. 5). The second step is to load maps and instantiate places associated with a synthetic agent population. In what follows, we explain each of these steps in more detail.

The urban environment used in this project involves generating both transportation networks and 2D building information to allow agents to have home and work locations and navigate the artificial world. The road networks are generated using procedural city generation based on L-systems (Lindenmayer 1968) along with considering different types of city morphologies. Unlike manual data generation that needs substantial human effort, procedural data generation is performed by a procedure to automatically generate content and data (Kim et al. 2018). Simply stated, an L-system is a string rewriting system that can be used to generate fractals with a dimension ranging from 1 to 2. Our rationale for this was that we wanted our generated networks to mimic those of actual cities (e.g., Batty and Longley 1994).

In terms of buildings, we consider *residential*, *commercial*, and *educational* types. We developed methods to distribute building types according to space syntax (Hillier and Hanson 1989). Notions from space syntax are incorporated in the generation of places, in the sense that the space syntax method attempts to explain how urban form impacts the accessibility and attractiveness of areas. The methods estimate a Central Business District (CBD) by analyzing spatial networks. The main algorithm is to build hierarchical graphs from networks (Kim and Li 2016). Different strategies for generating hierarchical graphs result in differently estimated CBDs. In the example of Fig. 1, yellow areas indicate low accessibility while red areas imply high accessibility [cf. results in Dubin and Sung (1987), Nelson (1973)]. We consider two types of metrics (i.e., control value[2] (Kim et al. 2012) and the degree of density[3]) in the hierarchical graph to determine building

---

[2] $C(i) = \sum_{k \in A} \frac{1}{C_n(K)}$ (i, K,: node, A: a set of nodes connected i node, $C_n(k)$: the number of k connected nodes connected K).

[3] The degree of density is computed by aggregating degrees in a hierarchical graph. That is, the degree of density is the total number of edges connected from the parent nodes of one node in the hierarchical graph.

(a) Price based on space syntax methods

(b) Price based on network degree

Fig. 1 Rent price distribution. (Color figure online)

types and properties (Dubin and Sung 1987). The degree of density impacts the determination of building types (e.g., commercial, residential, and educational), capacity, rent, etc. Further information about our procedural city generation and its role in agent-based modeling can be found in Kim et al. (2018). We provide several cities that differ in size for use by the research community such as those interested in exploring algorithms for location-based social networks (Kim et al. 2020a; Kavak et al. 2019).

As noted above, in order for agents to have home, work, and recreational places to visit and to generate patterns of life, places with a 2D polygonal shape, which mimics building footprints, are also generated along with the spatial road network. For the sake of simplicity, the locations of sites within each polygon are evenly distributed. Also, this helps to analyze the distribution of agents visually. The number of sites is proportional to the area of the building to which the sites belong (which is a simulation parameter). Site locations are recursively generated up to the given number of sites. These flexible parameters can adjust the density of places given in by the static maps. In total, our initial environment has 861 buildings and 4704 road network edges. From the degree distribution of the generated maps, we identified three aggregation levels: census tract, block group, and block. These three levels are a unit used to provide census data and they are color-coded in Fig. 2.
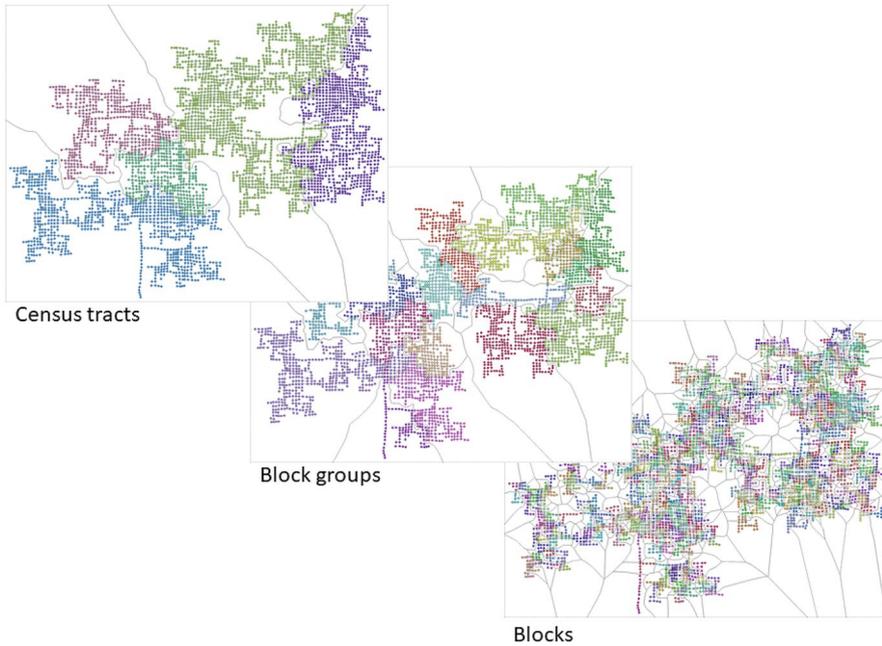
Fig. 2 Our generated maps colored based on different aggregation levels. (Color figure online)
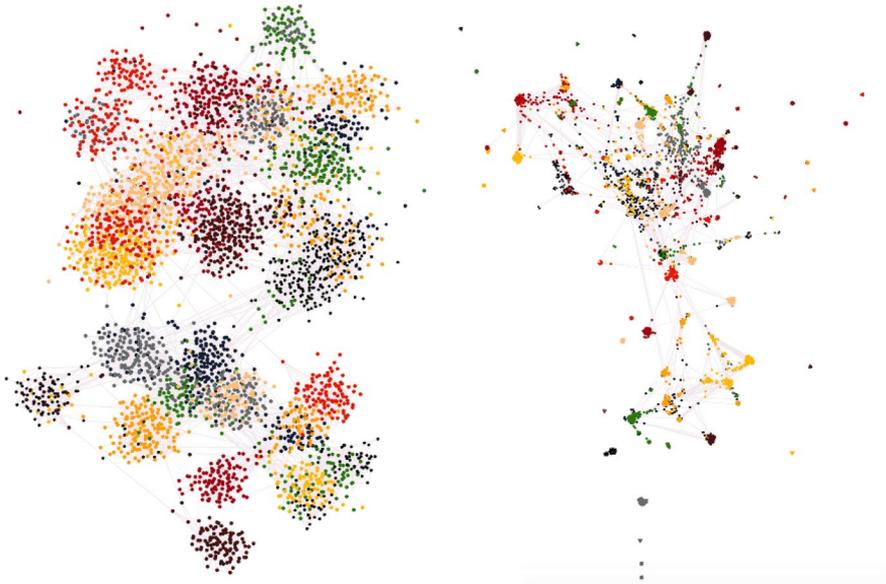
3 Agents

Agents in our model simulate the patterns of life for agents as demonstrated in
Kim et al. (2020a). Agents have attributes that are listed in Table 1. As described
in Sect. 3.1, agents follow a simple daily life cycle such as (1) go to work during
the day to earn money for financial safety, (2) go to recreational sites to make/meet
friends, (3) eat when hungry, (4) sleep when it is time to sleep (5) seek shelter at
night, etc. Agents meet other agents at sites leading to friendship between agents
and creating a social network, which is described in more detail in Sect. 3.3. Each
agent has the ultimate goal of getting "ascended", which is explained in Sect. 3.4.
A screenshot of our graphical user interface (GUI) of the simulation is shown in
Fig. 3. This figure shows the location of agents on the spatial network on the top
left. In this figure, the spatial network is represented as black lines, which shows
roads that are surrounded by yellow and pink colors which denote residential and
commercial areas, respectively along with the location of the agents coded by the
food status of hungry (red) and not hungry (blue). We show the distribution of the
number of friends (Fig. 3 bottom middle) showing a realistic long-tail distribution
of friends (Hill and Dunbar 2003). We also show the social network in the top mid-
dle of Fig. 3 where each node corresponds to an individual agent. The color of the
node represents the agents' interest attribute and edges between nodes correspond to
friend relationships in the social network. Sine this simulation screenshot was only
taken 1 day after simulation start, the social network is only starting to evolve. Fully
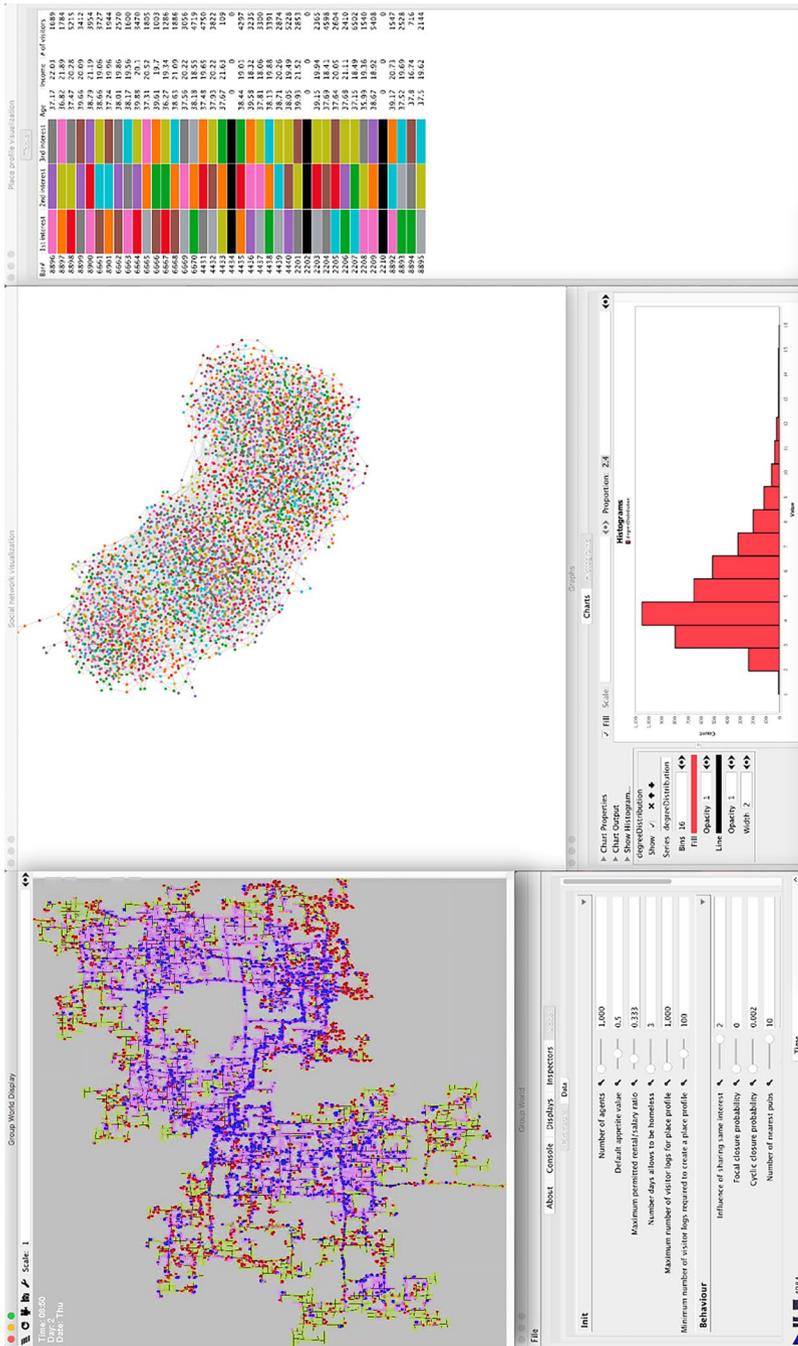
Fig. 3 A screenshot of the graphical user interface from a representative model run. Top-Left: The spatial network and agents. Bottom left: Simulation parameters that can be specified prior to simulation start. Top-middle: the social network. Bottom-middle: Summary statistics of the simulation during run-time such as friendship. Right: Profiles of recreational sites. (Color figure online)

Table 1 Attributes of agents

Need attributes	Other
Shelter need (current shelter)	Age
Food need (status, appetite)	Has family
Sleep need (status, time to sleep)	Socialness
Financial safety need (wealth, job, budgets, coworkers)	Education level
Love need (social status, friends)	Interest
Esteem need (ascension, nomination)	

Fig. 4 Social networks generated with two different simulation settings. Each unique color represents an agent with different interest value. Left: Our default simulation settings after 40 simulation days. Right: A simulation with a decreased chance of focal closure (making friends with strangers that share no common friends). (Color figure online)

evolved social networks after many simulation days are shown in Fig. 4. The simulation also provides default parameters (Fig. 3 bottom left) and summary statics during the tune-time of the simulation, which in this case shows the distribution of the number of friends (degree of the social network) at the current time. This window can also be used to provide time series, such as the average social network degree over time (Fig. 3 Bottom-middle). Finally, the GUI also provides profiles of recreational sites, including the Top-3 color-coded interests of visitors of these sites along with the average age, average income, and the total number of visitors of each site. It should also be noted that some recreational sites may not show these statistics due to not being visited yet (Fig. 3 Right).

Initially, we generate the agent population and attribute values sampled from certain value ranges. Agent *age* is initially between 25 and 50. Although the agents are gender-less in our model, they have a *family status* attribute, which indicates they have a spouse or children, but this only impacts housing selection and education cost. *Education level* is a static attribute and assigned one of the four values (Low, High School-College, Bachelor's, Graduate) based on percentages which are provided as parameters to the simulation. The *socialness* of an agent indicates how much an agent desires to make friends or make money. Initially, in the first phase of the program, we made socialness a discrete value (i.e., social, non-social, balanced) while in the later phases we made it a continuous variable. *Interest* is a static attribute representing the agent's hobby. We generated ten interests and each agent is assigned with one initially. The number of agents in the simulation can vary, however, for Challenge 1, we simulated 10,000 agents. You can see a screenshot from our simulation highlighting the spatial placement and color-coded hunger status of agents, social network snapshot, and degree distribution along with recreational place profiles. Our patterns of life mechanism and the inner workings of the model are explained in the following subsection.

3.1 Patterns of life

In our model, agents follow a pattern of life type of a daily cycle based on the augmentation of Maslow's hierarchy of needs (Maslow 1943). Each agent implements the first four levels of the hierarchy. In the first level, **physiological needs** contain several concepts including shelter, food, and sleep. In the second level, **safety needs** are captured in terms of financial security. In the third level, **love need** is expressed in the form of social status and friendship. And finally, **esteem need** is represented using the ascension mechanism. Agents prioritize these needs based on the need's placement in the hierarchy and seek to fulfill them as the goal of their lives. While fulfilling their needs, agents travel on the spatial network with a constant walking speed of 1.4 m/s and visit places that allow them to fulfill their needs.

Physiological needs are essential to survive in our Urban Life model and captured in three basic forms. (1) *Shelter need* makes agents find and rent an affordable apartment in the environment so that they can fulfill this innate need. There are three possible forms of living within the bounds of this model. Agents can live alone, live with other agents (i.e., become roommates), or live with their family (if they have one). Agents with families must live by themselves while other agents can share an apartment as long as there is enough room. Once an agent rents an apartment, the first month's rent is paid upfront. (2) *Food need* is another innate desire for agents to fulfill. We represented this status of food need as a variable with values ranging from 0 (very hungry) to 100 (very full). Once it goes beyond a threshold, the agent immediately goes to eat food (at home or a restaurant) and increases the fullness level. The increase and decrease in fullness are captured in the form of time-dependent functions which are informed by the agent's appetite. If the appetite is high, it means the agent eats more often. (3) *Sleep need* makes an agent go home and become unavailable until wake up time. Typically, agents follow a 24-h circadian rhythm while

the length of an agent's sleep is between 7 and 9 h. The start of the sleep time is dependent on the work start and commute times. Since physiological needs are the most basic needs, they must be first satisfied for agents in order to fulfill higher levels of needs such as safety, love, and esteem.

Safety need is used in ensuring that the agent has financial stability. Initially, agents start with a non-zero initial wealth and find a job (i.e., position) in workplaces based on the required education level. For simplicity, we assume that jobs have an 8-h fixed daily work schedule five times a week. Workdays are fixed but can change from job to job. Once an agent's daily work schedule is over, the agent gets paid based on the hourly rate of the job. This increases the wealth of agents. In terms of safety needs, agents aim to have a stable financial situation, meaning the agent's wealth should cover 1 month of expenses such as shelter and food. If an agent's finance is not stable, then, the agent moves to a cheaper place and reduce food spending.

Once the first two levels of needs (physiological and safety) are fulfilled, agents aim to expand upon the next two levels. **Love need** captures an agent's belongingness and social status. Recall that the socialness parameter of the agent makes the agent focus on maximization of happiness, which is related to making new friends, or maximization of the accumulation of wealth, which is related to job choice and expenditure. For some agents, socialness is the parameter that balances between maximizing happiness and maximizing wealth. Finally, **esteem need** captures agents' desire for reputation within this artificial society. These two last levels are explained in more detail below. Social network formation is indeed the process that changes the fulfillment of the love need.

3.2 Mobility

Each day at midnight, agents plan which places to visit on their next day. Places include their own home, their workplace, restaurants, and recreational sites. If the next day is a workday for the agent (each job has a work location and we assume a 5-day work week that is Monday to Friday for most agents but not all), the agent plans to work for 8 h. For any other time, depending on the social status of the agent, the agent plans to visit recreational sites (to increase social status) or work more (to improve financial safety). Visits to restaurants are not planned but occur during the day when the agent's food status reaches zero (the speed of which depends on the agent's appetite value). During the day, agents take activities prioritized by their needs as follows: Shelter and sleep need have the highest priority, causing agents to go home in the evening, around 10 pm to 1 am (depending on conflicting needs that may require them to work or recreate long hours). Once an agent's food status reaches zero, they go to their nearest restaurant to eat. If this occurs while they are working, the time spent getting food does not count towards the required 8 h of work per day. Only once an agent returns to work, they resume working. Once agents finish working, both financial safety need and love need compete for an agent's decision-making. Both are guided by the agent's financial safety (which is a function of the agent's current wealth, plus daily salary minus projected daily expenses), and the

agent's social status (which is a function of the number of friends of an agent). An agent's prioritization of financial safety versus love need is weighted by the agent's socialness. Agents with a high socialness prioritize socializing at recreational sites (unless financial safety becomes too critical) and agents with a low socialness prioritize making money (unless their social status becomes too critical). As socialness is a constant initialized uniformly in [0,1], all agents prioritize these two needs differently.

When agents change locations, they use the spatial network to go from their current location to a new one using the shortest path at a constant speed of $5\frac{km}{h}$. While traveling on the spatial network, agents do not interact with other agents. We compute shortest paths using the A^* algorithm for shortest path search (Hart et al. 1968). However, since our simulation requires a very large number of shortest path calculations each simulation day, we memorize all previously computed shortest paths in a look-up table to avoid re-computation of the same path. The different places an agent can visit are described in more detail in the following.

Home The home location of an agent is chosen randomly at initialization. Each home location has a rent price, depending on the in-betweenness of the network edge the home is located at. Edges having a higher in-betweenness have a higher price. Figure 1 shows an example distribution of rent prices for a generated road network. Each home also has a capacity of agents that can live together in the same home (as "roommates", since our simulated society is genderless and there are no families). If multiple agents live in the same home, they share the rent equally. Each day, agents project their income (through their job) and expenses (rent, restaurants, recreational sites). If an agent projects having higher expenses than income, they change their home to reduce their rent. Agents also consider the commute time between their home and their job (weighted by their hourly rate at work), such that a higher rent having a closer distance to their job location may be advisable for the agent. If an agent runs out of money and is not able to find a new home that matches their budget in three consecutive days (during which agents have critically low financial safety compelling them to work more and earn more money) an agent is removed from the simulation. This case usually only occurs during the first few days of the simulation, the initialization phase. Agents who share the same home meet daily, thus increasing their social network strength. Thus, agents living in the same home are expected to have extremely social ties (indeed resembling a family-like relationship, which however can change when agents change their home).

Work The work location of an agent is chosen at initialization. Each work location has a required education level, with higher education jobs paying a higher salary. Agents are not able to take a job for which they do not meet the minimum education level. Once per week, agents search for open jobs to find an available job that has a higher income than the agent's current job. Agents also consider the distance between home and work in the projected income of the job. Thus, changing home location may cause agents to later change their work location to converge to a lower salary. Changes in work location mainly occur in the initialization phase and converge to a stable state in which changes are rare, but may still occur, for example due, to jobs opening due to agents becoming "ascended" (see Sect. 3.4). Agents are paid at work continuously each tick (5

min) while located at their workplace. Each job requires agents to work for 8 h a day, 5 days a week. Most jobs, but not all, have a Monday to Friday workweek. When at work, agents continue working until they have completed their 8 h per day. If an agent is no longer required to work (either having worked 8 h, or it is not a workday), agents may still go to work to earn more money.

Restaurants When agents are hungry (critical food status), agents travel to their nearest restaurant (using network distance) to eat. While eating, food status is recovered (depending on the appetite of agents) and money is paid for the food. At restaurants, agents don't talk to (make friends with) strangers. However, agents may meet (and thus, increase friendship) with existing friends and co-workers. Whenever an agent enters a restaurant they scan the set of other agents at the same restaurant and pick a random friend or co-worker to meet. If no such other agent exists, agents do not meet anyone and eat in solitude.

Recreational Sites The majority of social activities happen at recreational sites. Each tick an agent spends at a recreational site, they attempt to start a meeting. Until a meeting is started, an agent loops through all other agents present at the same recreational site and try to reinforce existing friendship relations or make new friends (details described below under Social Network Formation). Like in restaurants, the food status of agents increases while the agents stay at a recreational site (although, not as fast as at a restaurant) and an agent has to pay a fee to stay at the recreational site. Since most of the social activities happen at recreational sites, an important aspect of the simulation is the choice of recreational sites. To choose a recreational site, an agent chooses among the *k*-nearest (*k* is a parameter) recreational sites from the agent's current location. Among these recreational sites, an agent considers the distance as well as compares the profile of the recreational site (which is obtained from the agent attributes of previous visitors of the recreational site) to the agent's own attributes. In total, an agent considers (a) the normalized distance weighted exponentially, (b) age similarity, i.e., normalized difference between agent age and the average age of visitors of the recreational site, (c) income similarity, i.e., the normalized difference between agent income and the average income of visitors of the recreational site, and (d) interest similarity, which is defined as 1.0, 0.5, 0.25, if the agent's interest is the first, second, third, respectively, most common interest among visitors of the recreational site, and zero otherwise. As an example of interest profiles, Fig. 3 (right-most panel) shows the Top-3 color-coded interests of each recreational site. These four weights are added for each recreational site and the agent chooses a recreation site as their destination randomly with probabilities weighted accordingly. For example, if $k = 3$ and the weights of the nearest three recreational sites of an agents are (3.0, 0.3, 1.7), then the probability of visiting these three sites are $\frac{3.0}{3.0+0.3+1.7} = 0.6$, $\frac{0.3}{3.0+0.3+1.7} = 0.06$ and $\frac{1.7}{3.0+0.3+1.7} = 0.34$, respectively.

The following specifies parameters of our model used to define the mobility of agents in our simulation:

- Minimum number of visitor logs required for site profile calculation: 100
- Maximum number of latest visitor logs used for site profile calculation: 1000

– Charge per hour (recreational): 6
– Charge per visit (restaurant): 3
– Number of nearest sites to consider visiting: 10
– Spatial Proximity Site Choice Exponential Decay Constant (factor multiplied to the spatial proximity coefficient per kilometer distance): 0.55 (example: a recreational site at a distance of 2.3km would have a weight of $0.55^{1.3} = 0.46$)
– Site choice coefficients

   1. proximity: 1.0
   2. age: 1.0
   3. income: 1.0
   4. interest: 1.25

3.3 Social network formation

Following Maslow's Hierarchy, once agents satisfy their physiological and safety needs, they can look for higher levels of needs corresponding to belongingness and friendship. These needs are satisfied by making/maintaining friends through visiting a recreational site for socialization. The choice of the recreational site depends on many factors including cost, proximity, and the profile of the people who visit that site. The site profiles are generated based on the logs of visitors involving age, income, and interest. As such, in the long term, recreational sites are generally expected to be visited by a dominant group of people based on age, income, or interest. However, this can be disrupted with the flashmob concept explained in Sect. 3.5.

There are two main methods of friends formation at recreational sites: cyclic closure and focal closure (Kumpula et al. 2007). When agents visit a recreational site (e.g., bar), they first check if they have any friends from their existing network. When they find a friend, they meet and strengthen their ties. When an agent meets with a friend who is also meeting with other friends, the newly joining agent forms connections with these friends of the friend. This implements the cyclic closure. When an agent can not see any friends, there is also a low possibility that the agent forms a connection with another lone agent or a group of agents (if a limited number of people are meeting). This implements the focal closure concept. In this respect, co-location increases the chances for agents to make new friends and also helps to maintain existing friendships. On the other hand, the lack of co-location leads to the lack of social interaction, thus, decreases friendship strength. Once friendship strength becomes too weak it leads to the friendship disappearing (Murase et al. 2015).

Examples of resulting social networks can be found in Fig. 4 for different social parameters. Here agents are color-coded by the interest attribute. The left of Fig. 4 shows the social network of 1000 agents for our default setting after 40 simulation days. This is a result of agents preferring to choose recreational sites that match their own interest attribute, thus causing agents of similar interest to be co-located more frequently. This represents agents having moderate focal closure probability (to meet strangers and turn them into friends) and twice as high cyclic closure probability (to

make friends with a stranger who has a common friend). It should also be noted that our agents exhibit no sense of homophily and are equally likely to become friends with any other agent, regardless of matching their interest. We also observe multiple clusters of each color, corresponding to groups of agents having the same interest and living/working spatially close thus allowing them to visit the same recreational sites. Many of the groups overlap and interconnect.

On the right of Fig. 4 we see the social network of a setting have a lower focal cluster probability (making friends with strangers that share no common friends) but having a higher cyclic closer probability (as a new friend is more likely to be a cyclic closer, i.e., closing a social triangle among existing friends) with much fewer social connections outside of these clusters (due to agents having a lower chance to befriend strangers). Both the increased inter-cluster connectivity and the resulting lack of cross-cluster connectivity causes clusters to be much denser and have fewer connections to agents outside their cluster. In this case, agents are more likely to become friends with those who already have common friends, and less likely to meet new people.

Parameters of our model related to social network formation and evolution together with their default parameters:

- Initial network edge weight (when agents become friends): 50
- Maximum network edge weight: 250
- Network edge decay factor (factor multiplied with edge weights every night at midnight): 0.6 (corresponds to a 40% reduction)
- Network edge deletion threshold (below which an existing edge is deleted): 1
- Network edge weight strengthening rate (factor by which an existing edge is increased when existing friends meet again): 2.08
- Cyclic closure probability (probability that strangers become friends if they have at least one friend in common): 0.01
- Focal closure probability (probability that strangers become friends if they have no friends in common): 0.0025

With the default parameters, Fig. 5 shows the changes of the average social network degree over time for varying populations (1 K, 3 K, and 5 K). After a warm-up period of about 20,000 steps (approximately 70 days), we observe emerging weekly patterns of the degree bounded by 20 and 30. When population increases, agents are more likely to interact with others and build more friendships as shown in Fig. 5.

3.4 Ascension

In addition to the first three levels of Maslow's Hierarchy, our model considers an "ascension" scenario that represents the esteem need of agents, that is, a desire for reputation. Periodically, agents perform a democratic election in which each agent votes for their best friend among candidates. Agents can nominate themselves by paying a monetary nomination fee. One agent receiving the most votes is elected to leave the world and "ascends" to a better world. To achieve the goal of ascension,
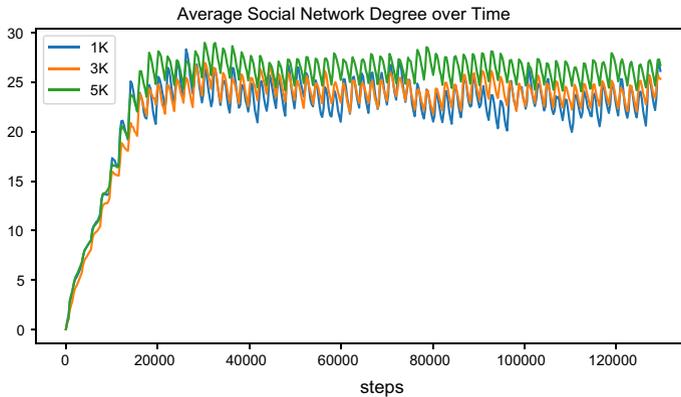
Fig. 5 Average Social Network Degree over Time for varying populations. (Color figure online)

agents aim to have as many friends as possible. In other words, agents follow strategies like expanding their social network to receive more votes from their friends and saving money to self-nominate. Although the number of agents that are removed is comparatively small, i.e., a few agents weekly, the fact that they are well-liked removes also an important hub in the social network and somewhat breaks up the social fabric.

This concept of ascension is implemented as follows. Each day (at midnight, when agents plan for their day), agents nominate themselves for ascension if their "social status" attribute (which is a love need) reaches a threshold of 0.9. Social status is an agent attribute that increases when an agent makes a new social tie (friend), decreases when a social tie is broken, and also decays over time. Intuitively, the social status of an agent is high if they were able to make many new friends recently. As most agents do not work on weekends and visit recreational sites instead, the social status generally spikes on weekends and decays during the workweek. When agents nominate themselves for ascension, they pay a nomination fee based on their financial safety, i.e., the amount of money that agents project they can spend. If the money an agent is willing to spend is below a minimum threshold, their nomination is declined and their money refunded. This minimum threshold is to ensure that agents don't nominate themselves every day, and it also ensures that financial safety and high social status are required to reach ascension. Thus, agents with a high attribute value of socialness are likely to have sufficient social status (as their needs guide them to visit recreational sites to meet friends more often), and agents with low socialness are likely to have sufficient financial safety to afford the nomination (as such agents are more likely to spend their time working and earning money). Yet, a balance of socialness is required for agents to consistently nominate themselves. If the nomination fee exceeds the minimum threshold, it is accepted and the money is deducted from the agent's balance and added to an ascension money pool. If the amount of money that the agent is willing to pay exceeds a maximum threshold, the agent is refunded any money above the minimum threshold. The latter is to ensure that agents with low socialness (who prioritize working versus socializing) do not

spend all their savings on a single nomination and can re-try multiple times. Once the total amount of money in the ascension money pool exceeds a threshold, nominations are stopped and agents vote for one nominated agent to ascend. For this purpose, each agent votes for the candidate to which the agent has the strongest social tie. If an agent has no social ties to any of the nominees, the agent votes uniformly at random. The candidate having the largest number of votes is chosen for ascension (with ties broken arbitrarily at random). The chosen agent is removed from the simulation, with all their social ties broken immediately, their job becoming vacant, and their home shelter becoming available. As agents dynamically change their income and social ties, we change the price of self-nominations for ascension dynamically after each ascension based on the demand for ascensions and the supply of agent money. Whenever two ascensions happen on consecutive days (meaning the demand for ascension is high due to many agents having high social status and/or the price for ascension is very low), we multiply the minimum and maximum nomination fees and the ascension money threshold by a constant called Ascension nomination fee change rate. If no ascension happens for more than 7 days (due to insufficient agents nominating themselves), we divide the minimum and maximum nomination fees and the ascension money threshold by the same constant, to make the next ascension easier to afford.

In summary, the parameters of our simulation of "ascension" (with default parameter values) include:

– Initial Social Status of agents: 0.5
– Social Status Increase for new social tie: 0.07
– Social Status Decrease for broken social tie: 0.03
– Social Status Decay Factor at midnight: 0.8 (reduction of 20%)
– Ascension social status threshold: 0.9
– Initial ascension money threshold: 1000
– Initial min nomination fee: 50
– Initial max nomination fee: 100
– Ascension nomination fee change rate: 2.0

3.5 Flashmobs

Another aspect driven by relationships is the flash mob. Here, agents aim to expand their social network (i.e., to receive more votes for ascension), "voice" their frustration by "raiding" a recreational site when they experience limited social contact over long periods of time. That is, agents that have nothing in common besides being isolated come together at an arbitrary site and as such impose their collective traits on this site. Again, this allows for a re-shuffling of established relationships and social fabric.

Flashmobs are implemented as follows. Each night at midnight, an agent checks if their social status is below a given threshold. If this is the case for three consecutive nights then the agent attempts to participate in a flashmob on the next day. For each interest of agents (as explained in Sect. 3, interest is a nominal agent constant

that is initialized at random and corresponds to the "hobby" of agents), the simulation checks if there are at least a minimum number of agents that are attempting a flashmob having this attribute. If this is the case, the mean location of all these agents is computed and the nearest recreational site to it is computed as the flashmob destination. All agents that participate in the flashmob prioritize (as their love need) to visit this recreational site, if other needs allow time for it. (Agents that must work to feed themselves may not be able to join the flashmob). The idea of flashmobs is to cause socially unhappy agents to all get together in a potentially large congregation in which many new social ties can be established to increase the social status of these agents.

Parameters within our model for "flashmobs" with their default parameter values include:

– Number of interests: 10
– Social Status Threshold to participate in flashmobs: 0.1
– Minimum Flashmob Size: 3
– Social Status Days (number of consecutive days to have a social status below the threshold to qualify): 3

4 Predict and prescribe challenges

We created two challenges for social scientist teams (Human Domain research teams of the DARPA Ground Truth program) to predict future states of our Urban Life simulation, and to implement changes to the simulation to improve future states. For each of these challenges, observable simulation data is described in Sect. 4.1. The *Predict Test* requires Human Domain research teams with predicting future states of the simulation, such as determining which recreational sites will be busy in the future or what the social network will look like. This challenge was repeated for two scenarios, the null scenario in which the simulation continues without changes, and a scenario in which a number of recreational sites close. The details of this Predict Test are given in Sect. 4.2. The *Prescribe Test* challenge requires Human Domain research teams to select a sub-population of 200 agents such that the average network degree of the social network is maximized. This challenge requires an understanding of how friendship is formed and which agents are most likely to become friends. It is described in Sect. 4.3.

4.1 Data collection

This section describes how the teams were able to interact with the simulation to obtain observable data in support of solving challenges. An overview of this *data rollout plan* is given in Fig. 6. First, not shown in this figure, is an initial period of 6 months of simulation time for "warm-up". As the simulation initially starts with empty social networks, this warm-up period is to ensure that all observable data results from a simulation state and is not an artifact of initialization. The first
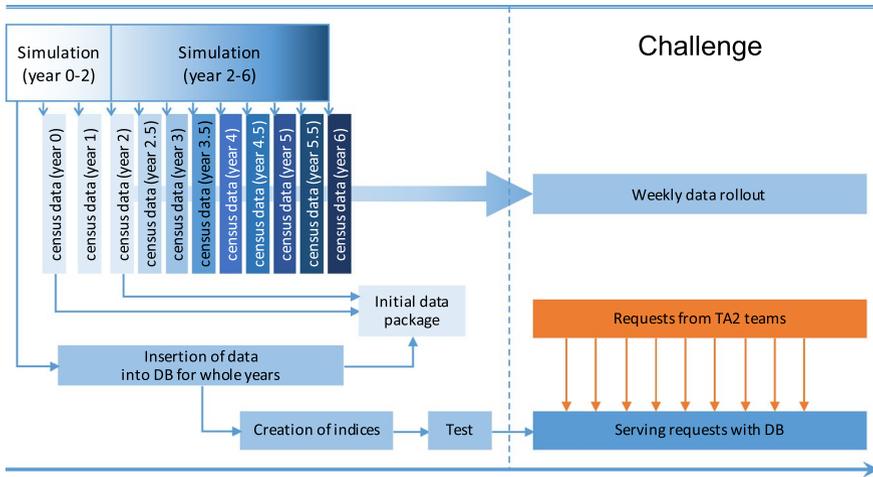
Fig. 6 Data roll out plan and explain test

observable day starts after this initialization period and the simulation continues running for an additional six years of simulation time. As shown in Fig. 6, all data is stored in a relational database for efficient access. This is necessary as the simulation creates terabytes of data that do not fit into conventional main memory. Once inserted, selected data of the first two simulation years was provided to teams in an *initial data package*. This initial data package included:

1. Environment data including the complete road network and locations of buildings and sites.
2. High-detail data from ten recreational sites selected spatially stratified, including check-ins and check-outs (including agent IDs) of all agents having entered the site and time. This data also includes data on all meetings that have occurred at these recreational sites, as well as IDs of all individual agents that have attended a meeting.
3. Low-detail data from 50 recreational sites selected randomly, including the number of agents located at these sites at any time. No agent IDs or meeting information is provided for these sites.
4. High-detail journal data for 100 agents selected randomly, including a detailed travel log (check-ins and check-outs at points of interest), financial transactions, and social interactions with other agents.
5. Census data reported each simulation year including population density, average income, and other statistics aggregated to small areal units encompassing 10–50 agents each.

In addition to the initial data package, Human Domain research teams were able to request additional information from the simulation by submitting *research requests*. Such research requests could include agent surveys, passive data collection

mechanisms, additional journal data, social network information, as well as experiments allowing to prescribe changes into the simulation and observe consequences. Since Human Domain research teams did not have direct access to the simulation database, research requests were submitted to us in natural language, and we implemented them as SQL queries and reported the resulting data. For all research requests, we enforced plausibly in a sense that only observable data was collected, and that the sampling strategies were not overly invasive. For example, survey requests were restricted to at most 20% of the population, whereas detailed journal requests were restricted to 100 agents only. Research requests were processed weekly for each Human Domain research team, and a weekly data roll-out was provided.

4.2 Predict test

After six years of simulation, we define a specific simulation date, namely simulation time 2022-07-15 at 0:00 am as P-Day, which denotes the starting point of time to predict. Teams were allowed to request data prior to and on P-Day, but any day after P-Day was strictly unavailable. The first challenge required to predict future simulation states 30 days after P-Day. This prediction included two scenarios:

– *Test Scenario 1 (the Null Scenario)* In the Null Scenario, nothing happens. The simulation is continued beyond P-Day without any exogenous events. The purpose of Test Scenario 1 is to test the basic ground truth understanding of our simulation and our agents' patterns of life.
– *Test Scenario 2 (the "Close Recreational Sites" Scenario)* In this scenario, one-third of the recreational sites in our world are closed permanently (20 out of 60) on P-Day. Therefore, the selected 20 Recreational Sites are permanently flagged as closed and can no longer be visited by agents. The purpose of Test Scenario 2 is to test the understanding of how agents react to a situation where their choices (of choosing sites) become limited.

For each of these scenarios, teams were challenged to predict the following aspects of the simulation on system-level, group level, and individual agent level:

– For five specified recreational sites predict the number of agents that will visit the site,
– For the same five specified sites predict the number of meetings taking place at the site each day,
– For a specified set of 100 agents (chosen randomly) predict the (unique) set of recreational sites that the agent will visit (at least once), and
– Predict the average number of recreational site visits per day.

To give more insights on this challenge, Sect. 4.4 provides an example evaluation of the challenge of predicting the average number of recreational site visits per day.

(a) Mock test

(b) Timeframe of mock test
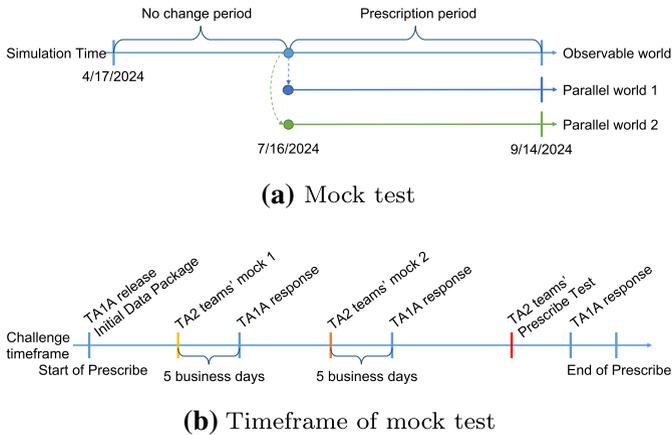
Fig. 7 Mock tests for Prescribe test

4.3 Prescribe test

The Prescribe Test challenge for Human Domain research teams required to combine all of what they have learned about our world and our people. Our Challenge 2 Prescribe Test represents that select a subset of 200 agents to keep in the world in order to maximize friendship and social interactions. Thus, 200 agents remain in the world and all other (4000+ individuals) are removed. The simulation continues running with only the 200 selected agents for one simulation month from P-day to 2022-08-14. On each day, we measure the average number of friends of our agents. The average over the last 7 days (08/08-08/14) determines the final score of a team. Teams are allowed up to four mock tests per team. A mock test allows teams to submit a test answer. For each mock test, we return the following data packages:

- Data Package 1: Detailed data from randomly selected recreational sites for 60 days before, and 30 days after P-Day
- Data Package 2: Low-detail information for selected recreational sites for 60 days before and 30 days after P-Day
- Data Package 3: High-detail journal data reported by 100 individuals for 60 days before and 30 days after P-Day
- Data Package 4: Basic statistics of social networks for 30 days before and for 60 days after P-Day

Mock tests are beneficial to help teams test their hypotheses. Teams have two opportunities to prescribe their changes to our simulation and observe the consequences of their choices. Figure 7 illustrates the simulation time and challenge timeframe for mock tests.

4.4 Experimental results

This section shows initial results for challenges defined in Sect. 4. Specifically, for the Predict Test of Sect. 4.2 we show the resulting average number of recreational site visits per day after multiple simulation runs and provide confidence intervals of these results in Sect. 4.5. For the Prescribe test of Sect. 4.3, we show red-teaming results of using random prescriptions as well as simple heuristics to select agents that maximize their social network in Sect. 4.6.

4.5 Predict test

To understand the variability of our simulation, and thus, the difficulty of accurately predicting the future of our simulation, we ran our simulation 11 times each for Test Scenario 1 and Test Scenario 2.

We computed the mean and standard deviation of the number of individuals in recreational site #2626 observed in the 11 simulation runs. As shown in Fig. 8, the resulting confidence bound use ± 1 standard deviation intervals (68 % confidence intervals). The confidence values of this confidence interval assume that the number of visitors on a day follows a Gaussian distribution. We observe that the 68 % confidence intervals, denoted by the boxes are quite small (note: The last day has incomplete data and should be disregarded). In many cases, we also observe outliers (denoted by the whiskers). Most of these whiskers point downwards. These are caused by flashmobs (Sect. 3.5). As flashmobs attract a large number of agents to another site, much fewer agents are available to come to this site. In summary, we see that the recreational site #2626 has a consistent number of visitors each day, subject to a reasonable amount of noise, which may increase due to our flashmob events. It is still possible to get good prediction results without predicting flashmobs. Readers might also notice that in Fig. 8 that there are periodic patterns, these are weekly patterns caused by the fact that agents are more likely (to have time) to go to recreational sites on Saturdays and Sundays. Thus, the number of visitors to recreational sites is higher during these days.

4.6 Prescribe test

As described in Sect. 4.3, we are asking teams to select a set of 200 (out of 4000+) agents. Their goal is to select this set such that the average number of friends of the remaining agents, after 1 month of additional simulation, is maximized. In order to make sure that this Prescribe Challenge is both feasible and non-trivial, we have carried out red teaming on this. Figure 9 shows the preliminary results from different methods used for red teaming. The x-axis and y-axis denote the number of days past after prescription and the average degree of the social network, respectively. For 30 days after the 200 agents are selected, this figure shows the average social network degree (i.e., the average number of friends) of each prescription represented as a curve. A greater value of degree indicates better performance. As instructed in our Prescribe Test, the average over the last 7 days constitutes the Prescribe Test result.
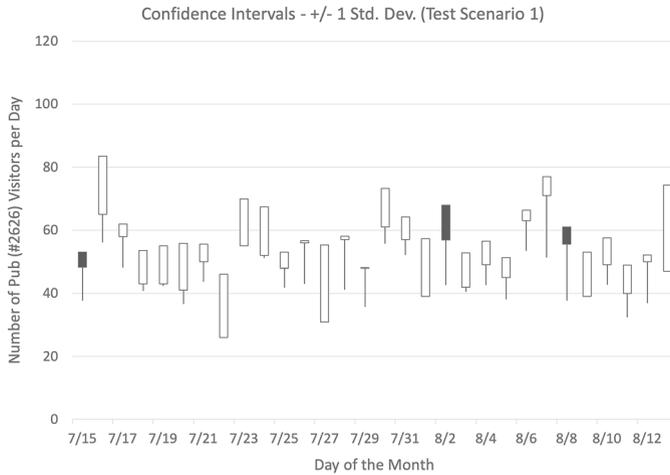
(a) Test Scenario 1
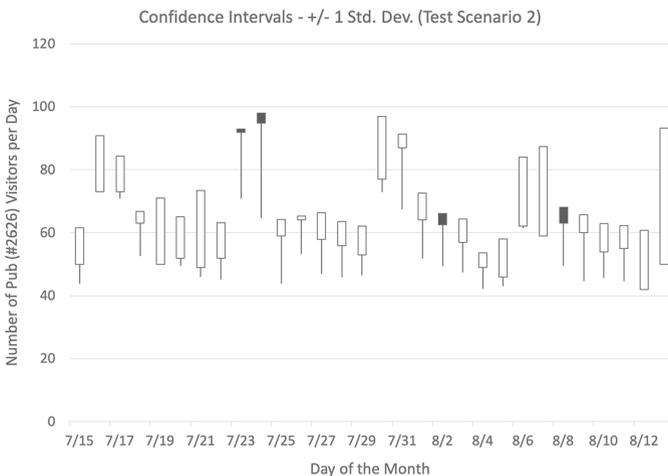
(b) Test Scenario 2

Fig. 8 Confidence intervals

To put these results into a quantitative context, we show the context of two baseline solutions that we implemented.

Red Teaming Baseline 1 (Random) Our red-teaming has shown that choosing random agents is terrible (labelled as Rnd1, Rnd2, and Rnd3 in Fig. 9). The average number of friends drops from 20 to 1 and hardly recovers. The reason is that the remaining random 200 agents are dispersed through the whole city. They may go to recreational sites to meet friends, but the chance is high that their recreational site of choice may be empty. Furthermore, many of the selected agents may not be
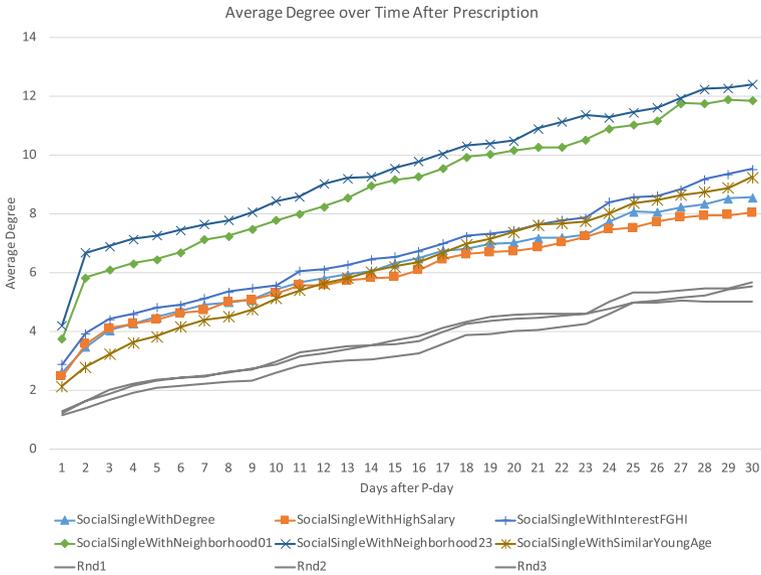
Fig. 9 Comparison of Prescribe test results. (Color figure online)

very social, and simply not care about making friends. We observe that the social network begins to regenerate over time, as agents make new friends with the other "survivors." The first two settings (*Rnd1* and *Rnd2*) are the settings provided in the initial data package. We note that the difference between different random choices is quite insignificant.

Red Teaming Baseline 2 (Socialness) As another baseline, we simply chose the 200 most social agents (with prefix 'Social-' in Fig. 9), without considering their location. This baseline makes sure that all selected agents want to make friends, but yet again, they are too dispersed to meet each other in large groups. Socialness defines how important it is for an agent to make friends. Agents with high socialness prefer to go to recreational sites to meet friends, while agents with low socialness prefer to work longer to make more money. For the socialness attribute, each agent has one of the following values: "high", "medium", "low". This baseline chooses exclusively "High" socialness agents. We considered similarity between agents including education levels (*SocialSinglelWithDegree*), salary (*SocialSingleWithHighSalary*), interest (*SocialSingleWithInterestFGHI*), neighborhood (*SocialSingleNeighborhood01*, *SocialSingleWithNeighborhood23*), and age (*SocialSingleWithSimilarYoungAge*).

Figure 9 shows three groups: "bottom" (*Rnd1*, *Rnd2*, *Rnd3*), "middle" (*Social-SinglelWithDegree*,   *SocialSingleWithHighSalary*,*SocialSingleWithInterestFGHI*, *SocialSingleWithSimilarYoungAge*),   and   "top"   (*SocialSingleNeighborhood01*, *SocialSingleWithNeighborhood23*) lines. The 3 worst lines, which are clearly separated on the bottom, are the same random baselines. All of the other approaches choose exclusively high socialness agents. Since there are more than 200 high socialness agents, different solutions are used to break the ties. The 4 approaches

in the middle break these ties by the attributes of the agents. For example, one chart chooses only high socialness agents that have a Graduate Degree, one chart shows the result of selecting high socialness agents with high salary, another chart select high socialness agents of similar age, similar interests, and so on.

Using high socialness agents yields a clear improvement over choosing random agents. The reason is that high socialness agents simply gain more friends. However, these four solutions have the problem that the spatial location is ignored. That is, agents are selected all over the city, making it very unlikely to meet many friends in this "sparse world". After all, the remaining 200 agents are distributed over the 60 recreational sites. Since not everyone is at a recreational site at any time, it is quite possible that an agent may find themselves completely alone at a site, unable to make any friends. Understanding socialness is but one concept that needs to be understood for this Prescribe Challenge.

The top two lines (*SocialSingleWithNeighborhood01* and *SocialSingleWith-Neighborhood23*) also select agents with high socialness, but only within one spatial region (called Neighborhoods). For these two lines, we again see a most significant increase in result score. The reason is that agents are not only looking to make friends, but they are also more densely placed in the same area, and thus more likely to visit the same recreational sites, as spatial proximity is one of the ground truth reasons to choose a site. To summarize, understanding that agents must be at the same place to make friends, in addition to understanding the socialness of agents, yields an additional increase in score.

5 Infectious disease simulation

We also included a disease model in our simulation to enable additional challenges of predicting the spread of a hypothetical disease and prescribing actions to mitigate the spread. We designed an SEIR compartmental disease representation (Kim et al. 2020c). The SEIR acronym refers to a "Susceptible" individual who has the potential to be "Exposed" to a disease. Once exposed, people are potentially "Infected". Finally, infected individuals transition to a "Recovered" state. For each unique disease, the entire population is initially assumed susceptible. To test different characteristics of diseases, we not only vary exposure and infection times but also person-to-person transmission rates. By doing this, it becomes possible to study many diseases ranging from those that quickly disappear to those that infect a large part of the population. We introduce restaurants as a place where new diseases are initiated and susceptible individuals are exposed based on an environmental spread probability. This spread probability decays daily assuming that the disease pathogen weakens and disappears over time. After being exposed, individuals do not immediately spread the disease but they have to become infectious first.

Figure 10 shows a screenshot of the simulation using the road network of the French Quarter, New Orleans, LA, USA on the left and the social network of agents on the right. In both networks, agents are color-coded by their current disease state.
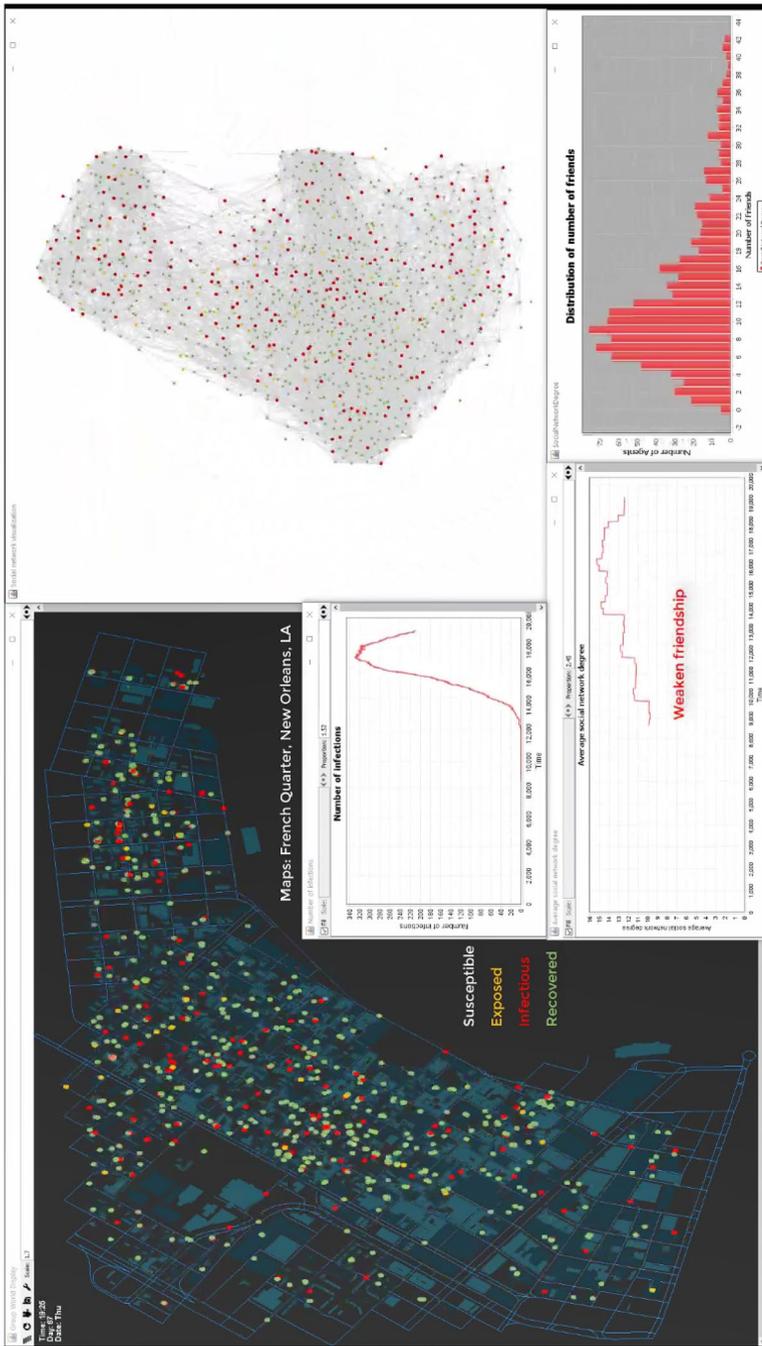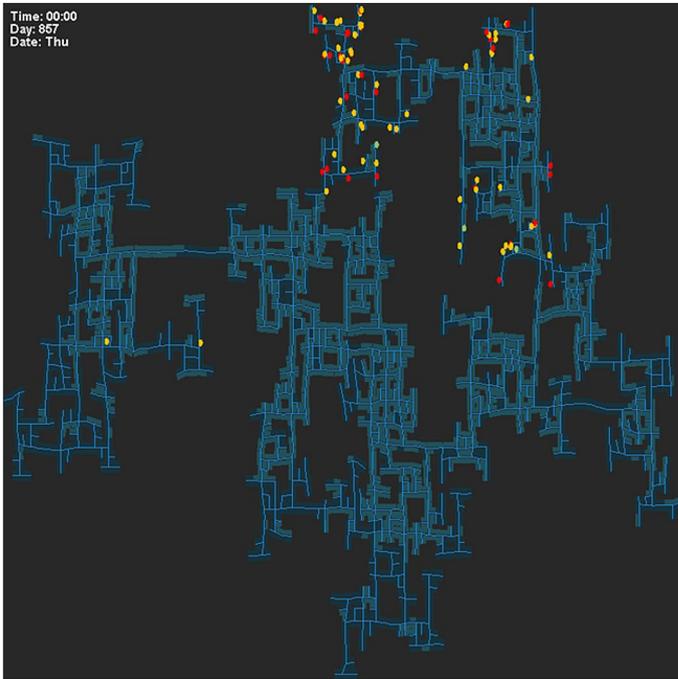
Fig. 10 Screenshot of the epidemic simulator depicting the French Quarter, New Orleans, LA, USA. (Color figure online)

Fig. 11 Agents recovered (green), exposed (yellow) and infected (red) at P-Day. (Color figure online)

The screenshot was taken shortly after the epidemic peak was reached. A full video of this simulation can be found at the project website[4].

5.1 Disease predict test

Using this disease model implemented on top of our simulation model, we further challenged Human Domain research teams to predict the future spread of a new infectious disease. For this purpose, we first generated four years of simulation data in which 58 diseases have been observed. As each disease has different parameters, this data allows teams to learn how different diseases progress in order to classify new diseases. We defined a new P-Day at a time when a new disease has just been observed (when about 10 of 6000 agents have entered the infectious state). We challenged teams to predict the disease spread 2 weeks into the future. What makes this task challenging is that teams were only given data on agents that were currently in the infectious state. However, data on agents that were in the exposed state (pre-symptomatic) was not disclosed. This fact requires teams to trace contacts of the

---

infected agents to identify agents having a high probability of having been exposed to the disease.

Figure 11 shows the locations of exposed and infected agents on P-Day. Teams can only observe infected agents (red dots), showing a hot spot in the North area of the simulated artificial urban area. As teams are not able to directly observe exposed cases (yellow dots) the challenge includes tracing and predicting the exposures and the corresponding disease hotspots they will cause.

5.2 Disease prescribe test

To mitigate the spread of this disease, we challenge teams to prescribe actions to the simulation to minimize the number of infections. Such actions include[5]:

– Quarantine agents, which is equivalent to remove agents from the simulation. Only two agents can be quarantined per day.
– Vaccinate agents, which transitions currently susceptible agents to the recovered state. Vaccinations are ineffective if an agent is already exposed or infectious. Only ten agents can be vaccinated per day.
– Equip agents with masks. We assume that agents equipped with masks have a 50% reduced chance of becoming infected. Only 20% of the population can be equipped with masks at any time.
– Force agents to work from home. Such agents continue to earn money but do so from home. Only 10% of the population can be forced to work from home on any day.

Participating Human Domain research teams provided us with a list of actions and corresponding agent IDs, which we injected into our simulation starting on P-Day. The goal of this prescribe test is to minimize the number of agents that will have become infectious within 14 days after P-Day. More details on this disease simulation as well as the effect of different actions on disease curves can be found at https://geosocial.joonseok.org/p/epidemic.html.

6 Conclusions

Our Urban Life model has been used in the Ground Truth program as testbeds for social science research methods. It served as functional testbeds for data collection methods and provided abundant data and information on causal structures, potential future states, and the results of policy prescriptions and system manipulations. This abundance of data, produced under conditions of controlled complexity and known ground truth allowed for explicit validation of hypotheses of Human Domain

---

[5] We note that this is a hypothetical alternative world and the simulated disease is a hypothetical disease. Our ground truth of 50% reduced spread using masks is not meant to reflect the real world, and the simulated contagion is not related to COVID-19 or other real world diseases.

research teams. To test the ability of Human Domain research teams to infer the ground truth of our simulation, we have designed challenges to predict behavior and social ties in our simulation. Our Predicts Tests challenged teams to predict future simulation states, both, in the case where a change was prescribed to the simulation, as well as for the counterfactual cases with no prescribed changes. Human Domain research teams were challenged to predict the future mobility of agents and to predict the spread of infectious diseases. Our Prescribe Tests challenged Human Domain research teams to select a subset of agents that will produce the strongest social network 2 weeks after removing all other agents from the simulation and to implement policies to mitigate the spread of an infectious disease.

To create this testbed, our Urban Life model simulates an alternate world in which agents follow patterns of life such as going to work to earn an income, going home to sleep, going to restaurants to eat, and going to recreational sites to meet friends. Our simulation can use either synthetic cities generated through a procedural city generation algorithm or real world road transportation networks and infrastructure. Our simulated agents are connected by realistic social networks, which are causally grounded based on agents' location and co-location patterns. As co-locations lead to friendship, a feedback loop is simulated by allowing existing friends to plan meetings to further increase their social ties. Our simulation software is available at a public git repository[6], and datasets of generated data can be found at https://osf.io/e24th/.

To create broader impacts beyond the DARPA Ground Truth program, the Urban Life team has organized international workshops on Geosimulation (Kim et al. 2019c, 2020b, 2021), on modeling and Understanding the Spread of COVID-19 (Anderson et al. 2021), and on Spatial Computing for Epidemiology (Züfle et al. 2021). The workshops brought together scientists from computer science, epidemiology, and social scientists to improve our understanding of geosimulation and how they can be used to fight emerging epidemics or even pandemics.

References

Anderson T, Yu J, Züfle A (2021) The 1st ACM sigspatial international workshop on modeling and understanding the spread of covid-19. SIGSPATIAL Spec 12(3):35–40

Batty M, Longley PA (1994) Fractal cities: a geometry of form and function. Academic press, New York

Bergenti F, Caire G, Gotta D (2014) Agents on the move: JADE for Android devices. In: Proceedings of the 16th workshop from objects to agents. CEUR-WS.org

Crooks A, Pfoser D, Jenkins A, Croitoru A, Stefanidis A, Smith D, Karagiorgou S, Efentakis A, Lamprianidis G (2015) Crowdsourcing urban form and function. Int J Geogr Inf Sci 29(5):720–741

---

[6] Source code: https://github.com/gmuggs/pol.

Crooks A, Malleson N, Manley E, Heppenstall A (2019) Agent-based modelling and geographical information systems: a practical primer. Sage, London

Dubin RA, Sung CH (1987) Spatial variation in the price of housing: rent gradients in non-monocentric cities. Urban Stud 24(3):193–204

Hart PE, Nilsson NJ, Raphael B (1968) A formal basis for the heuristic determination of minimum cost paths. IEEE Trans Syst Sci Cybern 4(2):100–107

Hill RA, Dunbar RI (2003) Social network size in humans. Hum Nat 14(1):53–72

Hillier B, Hanson J (1989) The social logic of space. Cambridge University Press, Cambridge

Kavak H, Kim JS, Crooks A, Pfoser D, Wenk C, Züfle A (2019) Location-based social simulation. In: Proceedings of the 16th international symposium on spatial and temporal databases, pp 218–221

Kim JS, Li KJ (2016) Location k-anonymity in indoor spaces. Geoinformatica 20(3):415–451

Kim JS, Han Y, Li KJ (2012) K-anonymity in indoor spaces through hierarchical graphs. In: Proceedings of the fourth ACM SIGSPATIAL international workshop on indoor spatial awareness, pp 21–28

Kim JS, Kavak H, Crooks A (2018) Procedural city generation beyond game development. SIGSPATIAL Spec 10(2):34–41

Kim JS, Kavak H, Manzoor U, Crooks A, Pfoser D, Wenk C, Züfle A (2019a) Simulating urban patterns of life: a geo-social data generation framework. In: Proceedings of the 27th ACM SIGSPATIAL international conference on advances in geographic information systems, pp 576–579

Kim JS, Kavak H, Manzoor U, Züfle A (2019b) Advancing simulation experimentation capabilities with runtime interventions. In: SpringSim, pp 1–11. IEEE

Kim JS, Kavak H, Züfle A, Crooks A, Manzoor U, Torrens PM (2019c) Geosim 2018 workshop report the 1st ACM sigspatial international workshop on geospatial simulation. SIGSPATIAL Spec 10(3):28–29

Kim JS, Jin H, Kavak H, Rouly OC, Crooks A, Pfoser D, Wenk C, Züfle A (2020a) Location-based social network data generation based on patterns of life. In: IEEE MDM, pp 158–167. IEEE

Kim JS, Kavak H, Pfoser D, Crooks A, Wenk C, Wise S (2020b) Geosim 2019 workshop report: the 2nd ACM sigspatial international workshop on geospatial simulation. SIGSPATIAL Spec 11(3):20–22

Kim JS, Kavak H, Rouly CO, Jin H, Crooks A, Pfoser D, Wenk C, Züfle A (2020c) Location-based social simulation for prescriptive analytics of disease spread. SIGSPATIAL Spec 12(1):53–61

Kim JS, Anderson T, Shashidharan A, Züfle A (2021) The 3rd ACM sigspatial international workshop on geospatial simulation: geosim 2020 workshop report. SIGSPATIAL Spec 12(3):11–14

Kumpula JM, Onnela JP, Saramäki J, Kaski K, Kertész J (2007) Emergence of communities in weighted networks. Phys Rev Lett 99(22):228701

Lindenmayer A (1968) Mathematical models for cellular interactions in development. J Theoret Biol Parts I and II 18(3):280–299

Luke S, Simon R, Crooks A, Wang H, Wei E, Freelan D, Spagnuolo C, Scarano V, Cordasco G, Cioffi-Revilla C (2018) The mason simulation toolkit: past, present, and future. In: International workshop on multi-agent systems and agent-based simulation. Springer, pp 75–86 (2018)

Manzoor U, Kavak H, Kim JS, Crooks A, Pfoser D, Züfle A, Wenk C (2021) Towards large-scale agent-based geospatial simulation

Maslow AH (1943) A theory of human motivation. Psychol Rev 50(4):370

Murase Y, Jo HH, Török J, Kertész J, Kaski K (2015) Modeling the role of relationship fading and breakup in social network formation. PLoS ONE 10(7):e0133005

Nelson RH (1973) Accessibility and rent: applying becker's time price' concept to the theory of residential location. Urban Stud 10(1):83–86

Sullivan K, Coletti M, Luke S (2010) Geomason: geospatial support for mason. Department of Computer Science, George Mason University, Tech. rep

Wilensky U (1998) NetLogo. http://ccl.northwestern.edu/netlogo/. Accessed 12 March 2021

Züfle A, Anderson T, Kavak H, Kim JS, Yu J, Roess A (2021) In: 2nd ACM SIGSPATIAL international workshop on spatial computing for epidemiology (SpatialEpi 2021). https://dataoceanlab.github.io/spatial-epi-2021/

Andreas Züfle is a tenured associate professor of spatial computing at George Mason University (GMU). He received his Ph.D. in Computer Science, summa cum laude, under the supervision of Dr. Hans-Peter Kriegel at Ludwig Maximilan University of Munich, Germany (LMU) in 2013. Dr. Züfle's research focuses on mining spatiotemporal data with applications in transportation, epidemiology, social science, and urban computing. He is a computer scientist who is passionate to work interdisciplinary and has a special interest in Geosimulation as a new paradigm to combine large-scale data generation with data mining to improve decision-making.

Carola Wenk is a professor in the Department of Computer Science at Tulane University, in the School of Science and Engineering. She also has an adjunct appointment in the Department of Mathematics. Her research interests are Algorithms, in particular Computational Geometry and Shape Matching, Topological Data Analysis, as well as applied areas such as Geospatial Data Analysis and Biomedical Applications.

Dieter Pfoser is a professor and chair of the Dept. of Geography and GeoInformation Science at George Mason University. He received his Ph.D. in computer science from Aalborg University, Denmark in 2000. At GMU he teaches courses related to geospatial data management, Linked Data, Web application development using open-source software, and data visualization. His research interests include data management, data mining for spatial and spatiotemporal data, graph algorithms for dynamic networks, and user-generated content, e.g., map-matching and map construction algorithms.

Andrew Crooks is a professor of Geography within the Department of Geography and a faculty member in the RENEW Institute at the University at Buffalo. Andrew's research Integrates agent-based modeling (ABM) and geographic information systems (GIS) to explore human behavior. More specifically, exploring and understanding the natural and socio-economic environments like urban areas using GIS, spatial analysis, social network analysis (SNA) and ABM methodologies.

Joon-Seok Kim is a Postdoctoral Research Fellow at the Department of Geography and Geoinformation Science at George Mason University. His research topics include data modeling, data management, data mining, location privacy, agent-based modeling, 3D Geoinformation, indoor LBS, geo-referenced media, and virtual reality. Recently, he has been working on an urban simulation framework on a DoD funded project.

Hamdi Kavak is Assistant Professor at the Computational and Data Sciences Department at George Mason University (GMU) and Co-Director of the Center for Social Complexity at GMU. His research focuses on the intersection of simulation modeling and data science; and he regularly applies his research to the study of human mobility, disease spread, and cybersecurity.

Umar Manzoor is Senior Lecturer at the Department of Computer Science and Technology at University of Hull. His research topics include multi-agent Systems, agent-based Modeling, machine learning, natural language processing, artificial intelligence and robotics.

Hyunjee Jin is a Postdoctoral Research Fellow at the Department of Geography and Geoinformation Science at George Mason University. Her research studies optimization in agent-based models. Recently, she has optmized agent-based simulations of infectious disease spread to minimize spread while also minimizing socioeconomic cost.

Authors and Affiliations

Andreas Züfle[1] · **Carola Wenk**[2] · **Dieter Pfoser**[1] · **Andrew Crooks**[3] · **Joon-Seok Kim**[1] · **Hamdi Kavak**[1] · **Umar Manzoor**[4] · **Hyunjee Jin**[1]

Carola Wenk
cwenk@tulane.edu

Dieter Pfoser
dpfoser@gmu.edu

Andrew Crooks
atcrooks@buffalo.edu

Joon-Seok Kim
jkim258@gmu.edu

Hamdi Kavak
hkavak@gmu.edu

Umar Manzoor
u.manzoor@hull.ac.uk

Hyunjee Jin
hjin6@gmu.edu

1   George Mason University, Fairfax, VA 22020, USA

2   Tulane University, New Orleans, LA 70118, USA

3   University at Buffalo, Buffalo, NY 70118, USA

4   University of Hull, Hull HU6 7RX, UK