ODD - Egress

Duarte Almeida

2023

1 Purpose and patterns

The purpose of the model is to simulate non-urgent egress of visitors from mass gathering events happening in real-life urban public open spaces (e.g., squares, plazas), evaluating the impact of different built environments, hypothetical exit-choosing behaviors of the visitors, social retention and crowd density scenarios and number of individuals in a social group in the egress patterns, the main emergent phenomenon obtained from the model. The model should aid decision-makers to choose the appropriate public spaces for a specific event and to timely deploy public utility teams (e.g., cleaning, emergency, security). The patterns we believe our model must be able to reproduce are the following:

- **Pattern 1: Relation between built environment layout and egress times.** The layout of the obstacles and exits, total space geometry, the total available exit width, as well as the shape of the open space should affect egress rates. E.g., a closed environment with a single small exit should lead to larger egress times than the same environment, but with more, wider exits.
- **Pattern 2: Relation between exit choosing behavior and egress times.** Chaotic exit choosing should lead to higher egress times, as a greater number of collisions between erratic pedestrians should overall reduce the egress rates. Whereas, orderly behavior should minimize collisions, smoothing the flow of pedestrians toward the available exits.
- **Pattern 3: Relation between crowd density and egress times.** It is expected a significantly denser crowd egress a built environment with less efficiency due to the increasing expected collisions between pedestrians and bottlenecks at exits.
- **Pattern 4: Pedestrian collision avoidance** Pedestrians avoid bumping into other pedestrians and other obstacles, maintaining a certain distance between them and altering their routes in order to avoid them. This pattern is based on the Social Force Model by Helbing et al. [1]
- **Pattern 5: Group cohesion.** Members of a social group maintain spatial cohesion, i.e., maintain a certain proximity to each other. This pattern is based on group behavior studies [2], [3]

Patterns 1,2 and 3 are based on evacuation principles described in the *Society of Fire Protection Engineers (SFPE) Handbook* [4], which we believe can be observed in non-urgent egress as well.

2 Entities, state variables, and scales

2.1 Entities

There are 4 entity types partaking in this model: (i) the Model entity, (ii) environment entities, (iii) pedestrian agents, and (iv) group agents.

The Model entity only serves as an orchestrator of the domain entities, controlling the behavior of most agents as if the agents were autonomous. The use of this entity was chosen to test the parallelization features of the GAMA Platform.

The environment entities are composed of the GIS polygonal representations of the buildings that surround and delimit the open space, as well as the buildings inside the space (e.g., statues, monuments), the usable area of each public open space, which is modeled as a polygon enclosed by the surrounding buildings or borders with bodies of water (e.g., river boundaries, coast), where pedestrians will randomly spawn, and the exit geometries (or targets), which are modeled as polygons that cover the entrances to the exiting streets, bordering the open space polygons.

The pedestrian agents represent the visitors of the event, who egress from it. The group agents represent the social groups in which the pedestrians are organized (e.g., a family, a group of friends).

These were the chosen entities, as the model is intended to assess the impact of the interactions between these entities on egress times.

3 State variables

The environment entities are static and passive and serve as limits or targets for the pedestrian agents. Thus, the only state variables of these entities are their geometry configurations. In Table 1, the state variables of the remaining entities are presented.

Table 1: Model state variables

Entity	Variable	Description	Possible Values	Units
Pedestrian	speedX	The X-axis component of the instantaneous speed	-	m/s
	speedY	The Y-axis component of the instantaneous speed	-	m/s
	heading	The instantaneous angle of direction	-	Degrees
	exit_target	The exit the pedestrian wants to reach	Environment entity (exit)	-
	location	The spatial location of the agent	Location point	-
Group	id	The identification of the group	-	-
	members	The list of members of the group	List of pedestrians	-
	centroid	The center of mass of the group	Location point	-
	social_time	The time a group takes to socialize	-	s

4 Scale

This ABM represents both space and time. The space is two-dimensional and finite and is represented on a continuous scale. The dimensions of the environment depend on the represented open space. The elementary unit of time represented in the model is 0.1 seconds, which is the value of the time step in a simulation cycle. The scales were chosen in a compromise between realism (the more continuous space and time are, the more realistic the model is) and simulation execution time.

5 Process overview and scheduling

The whole process and schedule are fairly simple. At the start of a simulation, the model agent initializes the environment geometries, groups, and pedestrians. After the spawn, at each simulation time step, the agents behave as follows:

1. The group agents execute the group submodel (in an undefined order), calculating their centroids and managing group member activity.

2. In an undefined order, the pedestrian agents execute the egress submodel, in which they move toward their chosen exits.

3. At each simulation second (10 steps), the model updates the output files with the number of remaining pedestrians in the simulation.

A simulation ends when the number of active pedestrians reaches a defined percentage of the number of spawned pedestrians.

6 Design concepts

7 Basic principles

The model is set over the concept of pedestrians attending mass gathering events in urban open spaces, and leaving the venue after the end of the event, without urgency. It appears this particular system has not been yet addressed in the state of the art. Some studies and ABM have been developed addressing egress flow dynamics similar to the ones described in the model, but represent a much smaller scale (usually room-scale), and different contexts [5]. At the agent level, the interactions between the crowd members follow the established Social Force Model (SFM) first presented by Helbing et. al. [1], extended with social group behavior in [2]. The social force models represent repulsive and attractive forces that act upon the psychology of the pedestrians, dictating an actual velocity to be chosen by a pedestrian, in order to avoid obstacles/other pedestrians and reach the desired destination. The social group extension adds group cohesion forces, as pedestrians from a group want to be close to each other [2]. As opposed to scenarios of evacuation, where the chosen exit will depend on multiple factors, such as panic states, distance to exit, etc., the egress in non-urgent scenarios will depend on the will of the pedestrians. Some may go to nearby bars, others may want to enter public transportation to return to their homes. This behavior is hard to predict due to a multitude of factors, and as such, the exit choice is modeled as a random process, weighted by the relative attractivity of exits (e.g., an exit that leads to more popular amenities, public transportation, or parking lots could have more attractivity; a less popular street will have less attractivity than a more popular one). The choice is made as a group, as it is assumed all members of a group want to go to the same location.

The model tries to approximate reality by setting a continuous space, a fine-grained time scale, and the microscopic representation of all individual pedestrians.

8 Emergence

The main emergent phenomenon obtained from the model is the egress pattern, which is represented by the number of remaining pedestrians in the venue at each time step. It emerges from the interaction between the pedestrian agents, due to social forces, and their spatial displacement towards the chosen exit. The model does not include imposed results.

9 Adaptation

The pedestrian agents adapt their desired velocity during each step according to the social forces acting upon them, which come from the other neighboring agents. They also adapt when they encounter obstacles in their desired paths, walking around and avoiding them.

10 Objectives

The adaptation of the agents does not have clear decision-making objectives. The agents move towards the chosen exit or around an obstacle in their path to the exit.

11 Learning

Learning is not implemented in this model.

12 Prediction

Prediction is not implemented in this model.

13 Sensing

Each pedestrian agent senses its neighboring pedestrians, obstacles, and its group center of mass, evaluating their distance and calculates the acting social forces.

14 Interaction

The main interaction between agents is through the calculation of the social forces they apply to each other.

15 Stochasticity

The stochasticity is applied in this model in the processes of (i) pedestrian and group spawning, (ii) exit choosing, and (iii) group social retention time. The pedestrians are randomly clustered around their group centers (within a certain distance from the group center), which are randomly picked locations inside the open space polygon with uniform random distribution, the exits are chosen with weighted random distribution, where the weight is the exit's attractivity.

16 Collectives

The collectives implemented in the model are the pedestrians' social groups. The social groups are modeled as entities, as they keep track of group members and the group centroid (calculated each cycle). The pedestrians will use the state variables of their group agents for the motion submodel.

17 Observation

The only observed variables during and after the model are the count of the current pedestrians in the model (the ones who did not egress at a given time step), and the number of pedestrians who left at each time step, which are calculated dynamically by counting the agents.

18 Initialization

The model is intended to simulate multiple environmental and behavioral scenarios. Each scenario is defined in an experiment, which in GAMA represents an execution of a model, being initialized with key values to represent the scenario. The environment is built from shapefiles containing relevant geometries. For a given open space, 3 files must be passed as parameters to parameters are *buildings_file, targets_file, and start_area_file* which represent the shapefiles with the geometries of, respectively, the surrounding and internal buildings of the open space, the geometries representing the available exits and the area in which the pedestrians will spawn.

The building geometries come from the OpenStreetMap (OSM) initiative, a volunteer-based mapping project to map the entire world, offering geographic data for free use. This data source was chosen due to its growing support, free-use data, and wide and detailed coverage of accurate geographical data across the world. The geometries of exit and start areas are created manually with the QGIS software, using as a reference the data from OSM. The exits must have a field called *id*, representing its identification number. The main model initialization parameters are the following:

- *nb_people* is an initialization parameter representing the number of pedestrians that will spawn in the simulation.
- *group_distribution_values* is a list containing the relative frequency a group size should be created (e.g., if a group size of 4 has 0.5, 50% of the total groups should have 4 elements). The sum of the values must be equal to 1.
- *exit_attractivity_values* is a list with the relative attractivity weights of every modeled exit (from 0 to 100), where the sum of the values must be equal to 100. The number of the position of the list element maps to the id of the exit (e.g., the first element of the list is the attractivity of the exit with id 1).
- *group_speed_values* is a list containing the group speeds for each group size.
- *egress_interval_period* represents the expected time it takes for an event to be fully empty of visitors. This value should be determined by analyzing the type of event and its characteristics.

Based on the set values of *nb_people* and *group_distribution_values* at the start of the simulations, the number of groups is calculated. At the beginning of a simulation, for each obtained group, a group center location is randomly created to guarantee a uniform distribution across the environment area. The pedestrians who are members of a specific group are then spawned with two-dimensional random uniform distribution within a set distance to the center of their respective groups. The group members are then assigned to their group agents, having mutual access to each other. Then, all members of a group choose the same exit, randomly distributed by the weight of the chosen exit's attractivity.

19 Input data

The model does not use input data to represent time-varying processes.

20 Submodels

The submodels in this model are implemented in the behavior of the pedestrian agents and group behavior. They consist of the (1) egress submodel and (2) group auxiliary submodel.

20.1 Egress submodel

This submodel describes the egress action of the pedestrians. They pursue the chosen exit for their group, moving through space until the desired exit is reached, avoiding strangers and obstacles, and maintaining group cohesion. The motion is implemented with adapted versions of the classic SFM by Helbing et. al [1], with an adaptation of the extension to adopt group behavior, presented by Moussaïd et al [2]. The implementations are based on the implementation found in the GitHub repository [1], adapted for the GAMA platform.

The parameters of this model are presented in Table 2. The parameters for the classic SFM were set based on the simulations made in the original study [1]. The parameters for the group SFM component were chosen based on multiple simulations, until group behavior provided more realism, through discussion followed by group consensus among the authors. This *expert-panel* approach was required due to the lack of empirical studies for high-density group behavior. Further calibration of these parameters should be done in future work.

Table 2: Egress submodel parameters

Parameter	Description	Unit
V0	Desired walking speed of a pedestrian	m/s
A_ped	The pedestrian repulsion strength parameter of the classic SFM	-
D_ped	The pedestrian range of repulsive forces parameter of the classic SFM	m
A_obs	The obstacle repulsion strength parameter of the classic SFM	-
D_obs	The obstacle range of repulsive forces parameter of the classic SFM	m
ped_influence_radius	The range within which a pedestrian exerts social forces	m
obs_influence_radius	The range within which an obstacle exerts social forces	m
Tr	The relaxation parameter of the classic SFM	s
shoulder_length	The shoulder width of the pedestrians	m
B_group_attr	The intra-group attraction strength parameter of the extended group SFM	-
B_group_rep	The intra-group repulsion strength parameter of the extended group SFM	-
group_rep_threshold	The range within which group members exert repulsive forces	m
group_attr_threshold	The distance from group center from which attractive forces occur	m

For each simulation cycle, the submodel for each pedestrian agent acts as follows:

1. If the simulation time has not passed the social retention time of the pedestrian's group, the pedestrian's speed is reduced to 0. The following behaviors still apply, as pedestrians must be able to move to make way for other pedestrians who want to egress.

2. The agent checks if there is an obstacle near enough, between the agent and its desired destination. If that verifies, the heading becomes a perpendicular angle to the heading to the nearest point of the obstacle. The agent will then want to walk left or right of the obstacle. The agent goes to step 4.

3. The heading is calculated through the current instant speed components, using the *atan2* operation. If both speed components have a non-zero value, the heading is the angle toward the chosen exit.

$$heading = atan2(speedY, speedX)$$

   Else, the heading is calculated through the direction vector between the pedestrian agent's location and the centroid of its desired destination using the *heading(from,to)* operation, where *from* is a location from where the heading points, and *to*, where the heading points to:

$$heading(from, to) = atan2(to.y - from.y, to.x - from.x)$$

[1] urlhttps://github.com/chraibi/SocialForceModel

In steps 2,3, and 4, all the forces acting upon the pedestrian will be calculated. The X and Y components of the forces are summed in variables sum_forces_X and sum_forces_Y, respectively.

4. The pedestrian agent obtains its group's centroid from the respective entity. Using the centroid, if the agent's distance to the centroid is more than the defined group attraction threshold, the pedestrian calculates the group cohesion force components using the following equations:

$$groupAttrForceX = B\_group\_attr \times cos(heading(self, centroid)) \times (1 - sin(heading(self, centroid) - heading)))$$

$$groupAttrForceY = B\_group\_attr \times sin(heading(self, centroid)) \times (1 - sin(heading(self, centroid) - heading)))$$

5. The pedestrian queries its neighbors, considering the influence radius, and for each one : If the neighbor pedestrian is a member of its group, and the distance between the agent and the neighbors is less that the group repulsion threshold, the group collision avoidance forces are calculated using the equation:

$$groupRepForceX = B\_group\_rep \times cos(heading(neighbor, self)) \times (1 - sin(heading(neighbor, self) - heading)))$$

$$groupRepForceY = B\_group\_rep \times sin(heading(neighbor, self)) \times (1 - sin(heading(neighbor, self) - heading)))$$

Else, if a neighbor pedestrian is not part of the agent's group, the repulsive forces from the classic SFM are calculated with the equations:

$$pedRepForceX = A\_ped \times e^{(shoulder\_length - distance(self, obstacle))/D\_ped} \times cos(heading(neighbor, self)) \times (1 - sin(heading(neighbor, self) - heading)))$$

$$pedRepForceY = A\_ped \times e^{(shoulder\_length - distance(self, obstacle))/D\_ped} \times sin(heading(neighbor, self)) \times (1 - sin(heading(neighbor, self) - heading)))$$

6. The repulsive force components from the neighboring obstacles within the obstacle influence radius are calculated, using the equations:

$$obsRepForceX = A\_obs \times e^{(shoulder\_length - distance(self, obstacle))/D\_ped} \times cos(heading(obstacle, self)) \times (1 - sin(heading(obstacle, self) - heading)))$$

$$obsRepForceY = A\_obs \times e^{(shoulder\_length - distance(self, obstacle))/D\_ped} \times sin(heading(obstacle, self)) \times (1 - sin(heading(obstacle, self) - heading)))$$

7. Having the total sums of the X and Y force components, the instantaneous speed components are calculated as follows:

$$speedX = speedX + step \times (sum\_forces\_x + (V0 \times cos(hd) - speedX)/Tr)$$
$$speedY = speedY + step \times (sum\_forces\_y + (V0 \times sin(hd) - speedY)/Tr)$$

8. Having the speed components, the pedestrian tries to move, obtaining its next location:

$$next\_location = (location.x + speedX \times step, location.y + speedY \times step)$$

If the next position is a valid position, the agent's position is set as that position. Else, the agent does not move, and both speed components are set to 0. If the next position is inside the agent's desired exit, it is removed from the simulation, else, the submodel repeats for the agent.

20.2 Group submodel

This submodel consists in updating the group characteristics each cycle, according to the locations and behaviors of pedestrians. After the group and pedestrian initialization, at each cycle, a group agent behaves as follows:

1. The group agent checks if a group member has left the simulation. If that happens, it is removed from the group's list of members.
2. The agent calculates the group's centroid based on the current position of all its current members.
3. If there are no members left, the group agent is removed from the simulation. Else, return to step 1.

References

[1] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Physical Review E*, vol. 51, no. 5, pp. 4282–4286, 1995, ISSN: 1063651X. DOI: 10.1103/PhysRevE.51.4282. arXiv: 9805244 [cond-mat].

[2] M. Moussaïd, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz, "The walking behaviour of pedestrian social groups and its impact on crowd dynamics," *PLoS ONE*, vol. 5, no. 4, pp. 1–7, 2010, ISSN: 19326203. DOI: 10.1371/journal.pone.0010047.

[3] L. Huang, J. Gong, W. Li, *et al.*, "Social force model-based group behavior simulation in virtual geographic environments," *ISPRS International Journal of Geo-Information*, vol. 7, no. 2, 2018, ISSN: 22209964. DOI: 10.3390/ijgi7020079.

[4] M. J. Hurley, D. T. Gottuk, J. R. Hall Jr, *et al.*, *SFPE handbook of fire protection engineering.* Springer, 2015.

[5] A. K. Wagoum, A. Tordeux, and W. Liao, "Understanding human queuing behaviour at exits: An empirical study," *Royal Society open science*, vol. 4, no. 1, p. 160 896, 2017.