Evac Sim: Fall 2020 CSS600

Justin Downes and Chris Smith

December 2020

Abstract

Simulating large scale evacuation is cost-prohibitive in terms of realism and required man-hours. Agent-based simulation supports laying out a floorplan, modeling behaviors and at least capturing some flavor of how a real evacuation might play out. This research uses NetLogo to vary floorplans and capture the average escape times for the agents.

1 Introduction

2 Background

2.1 Previous Work

The literature abounds with previous efforts in this area. [1] [3] [2] [5] [7]

This paper is key as it is extremely similar and a NetLogo implementation. We should know this paper and incorporate into our paper [6]

3 Methodology

The simulation itself is straightforward, and makes novel use of the Behavior Space facility, which we wrapped in a Python driver making use of pytest which is laid out in scripts/tests/test$_{f}ire_{s}im.py$

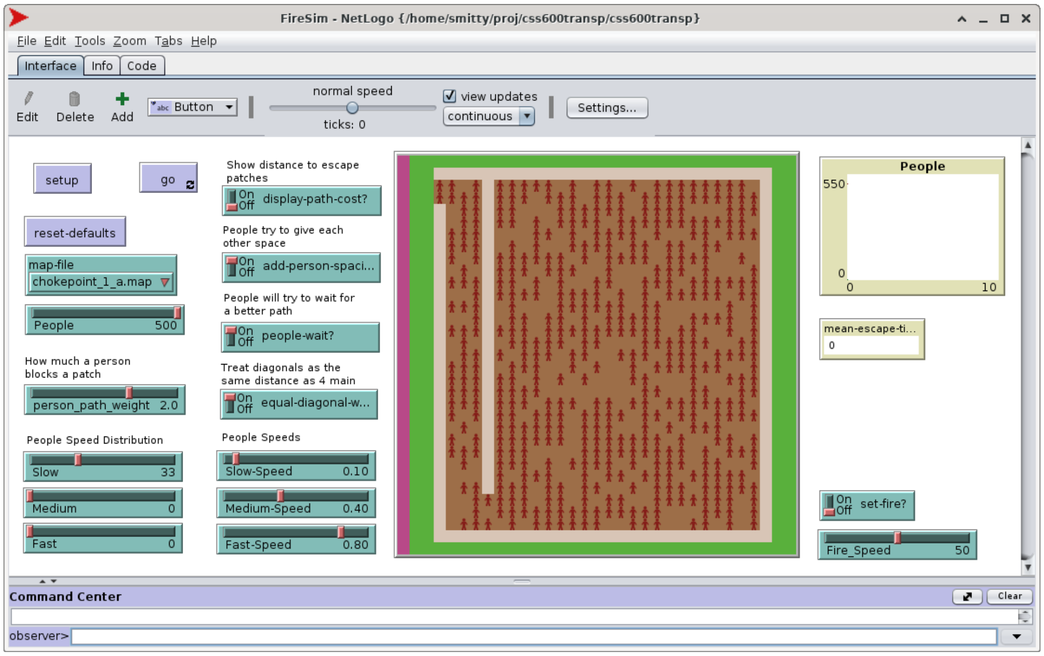

Figure 1: FireSim UI

3.1 Environment

For the map files in use, there were 10 runs against the map files, capturing mean-escape-time for the agents.

map-file	Name of map file to set up.
People	Number of people. Held constant at 500.
person$_{p}ath_{w}eight$	Agent blockage factor for patch
Slow	Percentage moving at this rate, which we set to 100
Medium	Set to 0
Fast	Set to 0
Slow-Speed	0.3 patches
Medium-Speed	0.75 patches
Fast-Speed	1.0 patches
add-person-spacing?	true
equal-diagonal-weight?	true
display-path-cost?	false
people-wait?	true
set-fire?	false
Fire$_{S}peed$	50
mean-escape-time	output

need to talk about patch parameters

3.2 Movement Mechanisms

this section will discuss agent and fire movement algorithms

The cost algorithm, where P_s is a safety patch, where A_w is the person path weight, a weight that a person adds to a patch due to blocking (this is configurable by the user)

$$cost(P) = distance(P_s, P) + Agent(P) * A_w$$

$$Agent(P) = \begin{cases} 1, & \text{Agent Present} \\ 0, & \text{Agent Not Present} \end{cases} \tag{1}$$

this can be configured through equal diagonal weight flag. if False the Distance is the Manhattan distance [1]. If true the distance is the Chebyshev Distance [2]

$$distance(P_1, P_2) =$$

$$equal - diagonal - weight? = \begin{cases} max(|x_1 - x_2|, |y_1 - y_2|), & True \\ |x_1 - x_2| + |y_1 - y_2|, & False \end{cases} \tag{2}$$

[1]https://www.sciencedirect.com/topics/mathematics/manhattan-distance

[2]https://en.wikipedia.org/wiki/Chebyshev_distance

patch to move to algorithm, where A_p is the patch for a given agent, P_x & P_y are the X & Y coordinates for a given patch

$$\begin{aligned} neighbors_4(P) = \{&patch(P_x - 1, P_y - 1), patch(P_x + 1, P_y - 1), \\ &patch(P_x - 1, P_y + 1), patch(P_x + 1, P_y + 1)\} \end{aligned} \tag{3}$$

$$move(A) = min(cost(neighbors_4(A_p))) \tag{4}$$

Person will wait for a better patch. This is configurable by the user. If it is on then a person will wait for a patch that is less cost than it's current

$$people - wait? = \begin{cases} min(cost(A_p), move(A)), & True \\ move(A), & False \end{cases} \tag{5}$$

add person spacing algorithm, people try to avoid each other, configurable through the $add - person - spacing?$ flag and the $person\_path\_weight$, A_w, parameter. here, A_w, is scaled by a factor of 10 since it is not the weight of being in the same square as another but of being next to another person.

$$add{-}person{-}spacing? = \begin{cases} cost(P) \sum Agent(neighbors_4(P)) * A_w/10, & True \\ cost(P), & False \end{cases} \tag{6}$$

[4] [3]

3.3 Experiments

describe our experiment harness that we built, netlogo behavior space, sql alchemy

3.3.1 experiments that are based on layouts.

-exit dims maps: see if placement has an effect on mean and max escape time. line plot for each number connecting a,b, and c points where each point is the average of multiple (10) runs . separate plots for mean and max

- choke point escape: same as above

[3]http://www.cs.us.es/ fsancho/?e=131

- need to run experiment on using a.map that plots mean escape time as we increase number of agents. probably 50 to 500 step size 10. we would expect to see this scale linearly but if not then there may be an interesting story.

3.3.2 experiments that are based on agent mechanisms

- vary experiments of person path weighting and agent speeds/distributions

3.3.3 experiments that are based on patch parameters.

- this may be obe by the exclusion of the fire mechanism

3.3.4 experiments that replicate emergent crowd behaviors from other papers

- maiinly from this paper [1] .

- herding and flocking

-arching and crowding

if we can show that we achieve similar results even though we use a simplified pathing algorithm and abm environment i think that would be insightful

4 Results

5 Conclusion

References

[1] João E. Almeida, Rosaldo J. F. Rosseti, and António Leça Coelho. Crowd Simulation Modeling Applied to Emergency and Evacuation Simulations using Multi-Agent Systems.

[2] E.D.Kuligowski and S.M.V.Gwynne. The need for behavioral theory in evacuation modeling.

[3] Angelika Kniedl, Dirk Hartmann, and Andre Borrmann. Using a multi-scale model for simulating pedestrian behavior.

[4] Maysam Mirahmadi and Abdallah Shami. A Novel Algorithm for Real-time Procedural Generation of Building Floor Plans.

[5] Fangqin Tang and Aizhu Ren. Agent-based evacuation model incorporating fire scene and building geometry.

[6] Eileen Young. Prioritevac: An agent-based model of evacuation from building fires.

[7] Jibiao Zhou, Yanyong Guo, Sheng Dong, Minjie Zhang, and Tianqi Mao. Simulation of pedestrian evacuation route choice using social force model in large-scale public space: Comparison of five evacuation strategies. 14(9):e0221872–e0221872.