

Emotion contagion model for dynamical crowd path planning

Andrej Sušnik, Timotej Zgonik, and Ema Leila Grošelj

Collective behaviour course research seminar report

January 10, 2025

Iztok Lebar Bajec | izredni profesor | mentor



An emotion contagion model for dynamical crowd path planning would allow for realistic and plausible simulation of crowd path planning and crowd behavior. Modeling agents' personalities using the OCEAN, or big five, personality trait model allows us to simulate a diverse crowd that reacts to the environment in different ways, while also affecting neighboring agents and their path planning. As an addition to the original model, we tried to implement a panic parameter to add an element of randomness to the agents' movements.

Dynamic crowd path planning | Crowd behavior | Simulation | Emotion contagion

1. Introduction

Efficient crowd path planning is crucial in various real-world applications, from urban transportation management to emergency evacuations. The *Emotion Contagion Model for Dynamical Crowd Path Planning* article presents an innovative approach by integrating emotional dynamics into crowd behavior modeling, offering a look into how individual emotions influence collective movement patterns.

This project aims to reproduce the results of the original study to validate its findings and explore potential improvements. We would like to see how adding an independent panic parameter, distinct from the already existing aggressiveness in that it causes agents to no longer follow the optimal path according to their personality, would affect the simulation.

2. Methods

Related work. The *Emotion Contagion Model for Dynamical Crowd Path Planning* article will be the main starting point for our work. The main approach to crowd path planning in the article is shown in Figure 1.

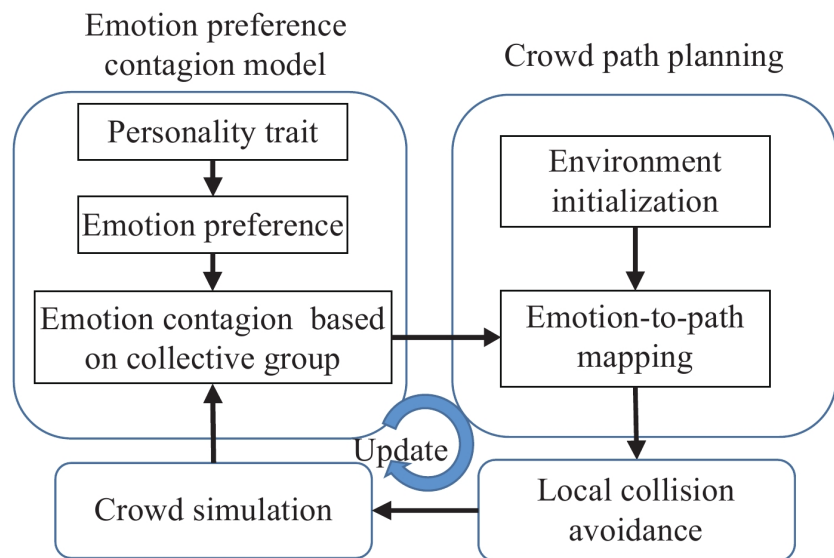


Figure 1. The emotion contagion model as proposed by the source article [1].

The source article proposes generating agents with distinct values for the five OCEAN traits: openness to experience, conscientiousness, extroversion, agreeableness, and neuroticism. Based on those factors, agents are then assigned an initial distance preference P_d and an initial velocity preference P_v . Agents have larger distance preferences if their personality factors are simple, aggressive, and fragile (corresponding to OCEAN factors: O-, E-, A+), and agents have larger velocity preferences if their personality factors are variable, energetic, and functional (corresponding to OCEAN factors: C-, E+, N+). If an agent has a large distance preference, its velocity preference is usually smaller, and vice versa. An agent with a larger velocity preference

will seek to find less crowded paths even at the cost of the distance being longer, while those with a larger distance preference will stick to a path even as it becomes crowded. An emotion contagion mechanism further enhances the model by allowing emotional states to spread among neighboring agents, mimicking the way emotions such as stress or calm influence a group. A dampening factor regulates this contagion to ensure an agent's inherent personality traits are at least partially preserved. The model incorporates a strategy where a least expected-time objective function is used to dynamically select paths, factoring in both environmental variables and agents' emotional states and preferences [1].

The other aspect of the article is emotion contagion. For that purpose, the simulation requires a method to define collective clusters among the agents to determine which of them will be mutually affected. Agents are defined as being collective neighbors if they share similar motion and have a mutual goal. Information is contagious between neighboring agents in the same group and is affected by the distance between agents and the difference between emotion preferences. Additionally, agents may be affected by environmental contagious sources, such as a fire disaster [1].

The *Simulation of crowd dynamics in pedestrian evacuation concerning panic contagion: A cellular automaton approach* article describes how panic could influence the path planning of the agent. The authors propose a model where panic is contagious and negatively impacts the ability of an agent to find an exit [2].

Proposed improvements.

Panic contagion. An improvement that we propose is adding **Panic contagion** to the simulation. Each agent is assigned a panic parameter and a panic susceptibility parameter, where panic inhibits the ability of the agent to move to their goal. As the panic parameter increases, the agent's movement becomes increasingly erratic and random.

The panic susceptibility parameter, in our approach, is calculated from the OCEAN traits of the agent, as shown in Equation 1. The subscript ₀ next to a given trait denotes the value being shifted by 0.5 from the original expected value, so that the new expected value is equal to 0. Openness, conscientiousness and agreeableness have negative correlation with panic susceptibility, while neuroticism has positive correlation. The distribution of panic susceptibility remains equal to the distribution of the traits: $N(0.5, 0.1)$. We decided that all traits are equally important: $w_O = w_C = w_A = w_N = \sqrt{1/4}$.

$$\begin{aligned} \text{panic_susceptibility} &= -w_O * O_0 - w_C * C_0 - w_A * A_0 + w_N * N_0 + 0.5, \\ &\text{where } w_O^2 + w_C^2 + w_A^2 + w_N^2 = 1 \end{aligned} \quad [1]$$

Each agent is assigned a current panic parameter. This parameter represents the probability that the agent will not behave according to their personality and will either freeze or move randomly. This parameter can be modified in three ways. Every agent seeks to reduce their panic to a baseline level, the return to the baseline level being dependent on their panic susceptibility parameter. The relationship is described in Equation 2 for every Δt .

$$\text{current_panic} -= (0.5 + (0.01 - 0.5) \cdot \text{panic_susceptibility}) * (\text{current_panic}) \quad [2]$$

There are two means by which the panic parameter of an agent may increase. Agents are susceptible to the panic of other agents in their cluster and their panic parameter value will accordingly change to match the cluster's average, with the speed of the change governed by their panic susceptibility, as described by Equation 3.

$$\begin{aligned} \text{panic_diff} &= \text{average_cluster_panic} - \text{current_panic} \\ \text{current_panic} &+= \text{panic_susceptibility} * \text{panic_diff} * 0.05 \end{aligned} \quad [3]$$

An agent's current panic may also increase if they are near a panic-inducing source, which is described by Equation 4.

$$\begin{aligned} \text{dist} &= \frac{1}{|\text{position} - \text{panic_source_position}|} \\ \text{current_panic} &+= \text{dist} * \text{panic_susceptibility} * 0.001 \end{aligned} \quad [4]$$

Clustering algorithm evaluation. It was also possible to explore how the **clustering algorithm** described in the article performs when compared to hierarchical clustering and Fast Label Propagation [3].

Enhanced navigation graph. Additional movement options have been added to the navigation graph, as described in Section 3.

Corrected error. We believe to have found an error in the source article with its equation:

$$d_{ori}(i, j) = |\arccos(vel_i) - \arccos(vel_j)|$$

which does not appear to be a operation that can be performed on a vector, so we instead propose:

$$d_{ori}(i, j) = |\text{atan2}(vel_{i_y}, vel_{i_x}) - \text{atan2}(vel_{j_y}, vel_{j_x})|$$

where we obtain the correct angle θ from the x -axis [4] for each of a pair of agents and then subtract the angles to obtain the difference between their orientations.

3. Results

Clustering. Figure 6 shows multiple runs of the clustering algorithm from article [1]. Even though the clusters were very different when we used a different algorithm, little to no effect was visible on the agent paths.

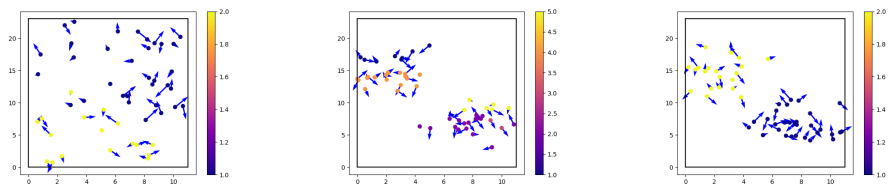


Figure 2. Example runs of clustering according to association with the closest neighbor with a higher degree. The leftmost sub-figure represents agents with positions initialized uniformly, while the two to the right represent agents with positions initialized with a bimodal Gaussian distribution.

Navigation graph. For each exit we constructed a navigation graph of shortest paths. Every such graph is a tree or a forest with roots among the nodes belonging to an exit, representing shortest paths from all possible positions on the grid to one of the exits. It was constructed using Dijkstra’s algorithm allowing movements to all 8 neighboring fields as well as to fields located two steps in one direction and one step in a perpendicular direction. An example graph is shown in Figure 3.

This navigation graph would, in theory, allow for modeling the movement of agents in a more natural way than the basic left-right-up-down version of it.

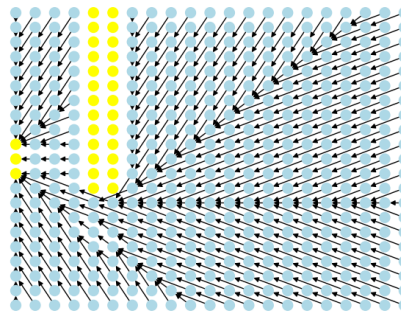


Figure 3. Example navigation graph for a grid of dimensions 10 times 10. Yellow nodes represent the exit of width 3 and the obstacle.

Experiment. We performed an experiment where we manually set the velocity preference of a single agent to a significantly higher value than the other agents. The results can be seen in Figure 4.

Panic. We discovered that the panic parameter has great influence on the paths, as shown in Figure 5.

Testing of hierarchical clustering algorithm. We tested the hierarchical clustering algorithm and found that the results are no different than with the clustering algorithm proposed by the source article.

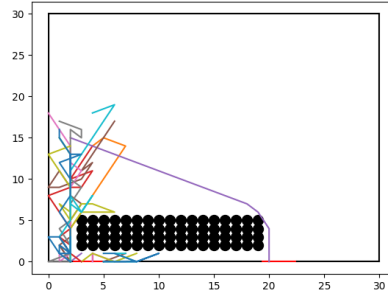


Figure 4. Paths of the agents in the case of a single agent assigned a very strong velocity preference, forcing them to choose the farther exit. The black dots represent an arbitrary obstacle.

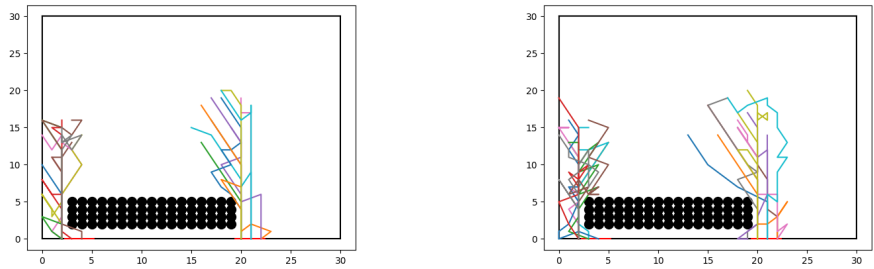


Figure 5. Example runs without a source of panic (left) and with a source of panic (right) included. The more erratic movement as a result of the panic is clearly seen in the right part of the figure.

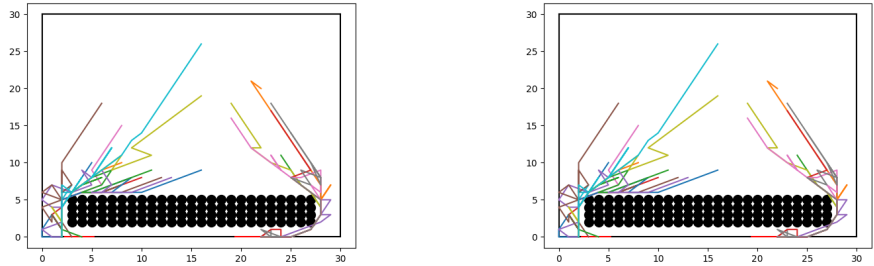


Figure 6. The left image represents the paths of the agents that were clustered using hierarchical clustering and the image on the right represents the paths of agents based off the clustering algorithm proposed by the source article.

4. Discussion

We have implemented a simulation workflow where we can define the environment in the text format and then run the simulations. We tried out three different clustering algorithms, but the change of clustering algorithm appears to have no visible influence on the result. The introduction of panic parameter was a fruitful decision, as panic like behavior can be seen in the paths of agents.

Many aspects of the implementation had to be reinvented, as the authors of article [1] at some points did not elaborate on their solutions. One such instance was with the implementation of the navigation graph generation, as for small environment sizes in the range of 100×100 tiles were reasonably fast, but larger environments were not feasible as the time to construct the graph was excessive. To allow for greater precision of agent movement, it would be preferable to optimize the navigation graph generation or at least cache it for repeated use.

Future work could explore alternatives for the navigation graph that would allow for the agent positions to be points instead of real numbers. Additionally, the agent paths could be curves parametrized by continuous time parameters, which would allow for continuous time instead of discrete time steps.

CONTRIBUTIONS. **AS** worked on the code (environment representation, agent movement) and on the report, **TZ** worked on the report and its editing as well as proofreading, and oversaw project management, and **ELG** worked on the code (contagion, navigation graph, clustering,

visualization) and on the report.

Bibliography

1. Wu Y, Huang X, Tian Z, Yan X, Yu H (2024) Emotion contagion model for dynamical crowd path planning. *International Journal of Network Dynamics and Intelligence* 3(3):100014.
2. Wang GN, Chen T, Chen JW, Deng K, Wang RD (2022) Simulation of crowd dynamics in pedestrian evacuation concerning panic contagion: A cellular automaton approach. *Chinese Physics B* 31(6):060402.
3. Traag VA, Šubelj L (2023) Large network community detection by fast label propagation. *Scientific Reports* 13.
4. Organick E (1966) *A FORTRAN IV Primer*. (Addison-Wesley).