Does a group's size affect the behavior of a crowd? An analysis based on an agent model

Carolina Crespi[0000-0003-1050-2453] and Mario Pavone[0000-0003-3421-3293]

Department of Mathematics and Computer Science
University of Catania
Viale A. Doria 6, I-95125 Catania, Italy
carolina.crespi@phd.unict.it
mpavone@dmi.unict.it

Abstract. The field of crowd simulations has gained increasing attention due to its numerous applications, such as emergency management, sociology, computer games, and path planning. Despite the increasing literature available, creating crowd models is still a challenging problem due to the complex and dynamic nature of crowd behavior and the lack of data available. In this paper, a mesoscopic model is proposed that combines agent-based models with swarm intelligence methods to investigate the effect of group size on the behavior of the population. Two types of analysis were carried out, an overall analysis and a type analysis, and three evaluation metrics were used: the number of agents that reached the exit, the exit times, and the cost of the paths taken to reach the exit.

Keywords: agent-based models · crowd simulation · collective behavior · swarm intelligence

1 Introduction

Crowd simulations have gained increasing attention in recent years due to their numerous applications, such as emergency management [11], sociology [14], computer games [22], and path planning [15] among many others. This has led to a wide range of models and techniques being developed in the field, with significant progress made possible by advancements in computer hardware and software. Despite the increasing literature available, creating crowd models is still a challenging problem due to several reasons. Firstly, there is a lack of data available for modeling crowd behavior, which can limit the accuracy of models [12]. Secondly, crowd behavior is a complex and dynamic phenomenon, that may be influenced by a wide range of factors such as age, environment, and personality [8]. In [13] the authors propose an agent-based model of emergency evacuation that takes into account panic behaviors. In [1] are considered the physical characteristics of pedestrians while in [9] a specific emergency is taken into account. It follows that there exist various crowd behavior models, and each one has a unique perspective on the issue based on the framework used. Consequently, the outcomes obtained from different frameworks vary, and selecting one over the other can be challenging as none outperforms others. These models can be split out in three main categories [20]: (1) microscopic models [23], in which individuals are considered as distinct entities, with unique traits whose interactions may produce unexpected collective behaviors; (2) macroscopic models [10] in which individuals are viewed as a continuous flow generally governed by adequate physical dynamics; and (3) mesoscopic models whose aim is to combine the strengths of both macro and micro techniques [21]. Indeed, although microscopic models are adept at studying collective behavior, they are computationally expensive and not well-suited for modeling large-scale scenarios. Conversely, macroscopic models struggle to capture individual interactions but can simulate thousands of individuals. In this paper, we present a mesoscopic model that combines agent-based models, known for their effectiveness in modeling individual decision-making and social behavior, with swarm intelligence methods, which have shown their usefulness not only for optimization purposes [5], but also in modeling crowd dynamics [4,24,16]. In this model, a population of virtual agents is tasked with reaching a specific location, referred to as *exit*, starting from a designated point. Agents may be divided in groups of different sizes and can adopt two distinct behavioral strategies: *collaboration*, which involves sharing information about paths and/or repairing damaged paths, and *defection*, which entails not sharing any information, destroying paths and/or nodes, but still benefiting from the help of collaborative agents. Despite their behavioral differences, the primary objective for each agent is to reach the exit point. The aim of this paper is to examine how the behavior of a population as a whole is influenced by the size of the groups that the initial population is divided into, as well as the behavioral strategies that are employed. The occurrence of group formation among pedestrian crowds in various circumstances has been frequently observed, and it has been shown that the presence of groups can have a significant impact on crowd movement [7] as well as their size and numbers may affect the evacuation time [19]. In [2] the authors considered a model with a crowd divided into several groups and have found that groups' presence may positively affect the information transmission among agents. In this context, two types of analysis were carried out: an *overall analysis*, which compares the performance of populations divided into groups of varying sizes, and a *type analysis*, which examines the performance of collaborators and defectors within the same groups. Three evaluation metrics are considered and simultaneously compared for both types of investigation, namely the *number of agents* that reach the exit; the *exit times*; and the *cost* of the paths crossed to reach the exit.

2 Model

The framework of our model draws inspiration from the Ant Colony Optimization algorithm, which is a metaheuristic algorithm that emulates the behavior of real ants. This algorithm is commonly used to tackle various types of combinatorial optimization problems, including scheduling and routing problems [17], coloring [6], path problems [3], and others. In our study, we utilize the computational framework of the Ant Colony Optimization algorithm to model the behaviors of the agents and the environment, as well as how the agents interact with time in the environment. We represent the environment in our model as a weighted undirected graph $G = (V, E, w)$. Here, V denotes the set of vertices, $E \subseteq V \times V$ denotes the set of edges, and $w\colon V \times V \to \mathbb{R}^+$ is a *weighted* function that assigns a positive cost to each edge. Let $A_i = j \in V : (i, j) \in E$ denote the set of vertices adjacent to vertex i. At any given time-step t, an agent k visits a non-empty sequence of vertices $\pi^k(t) = (\pi_1, \pi_2, \ldots, \pi_t)$ (which may include repeated vertices), where $(\pi_i, \pi_{i+1}) \in E$ for $i = 1, \ldots, t-1$. A population of N agents begins at a given location and explores the environment to move to the target location as quickly as possible, possibly taking a less expensive route. It is divided into Γ groups, and each group starts its investigation at regular intervals. An agent k placed on a node i moves to one of its neighbour vertices j, with a probability $p^k_{ij}(t)$ defined by:

$$p^k_{ij}(t) = \begin{cases} \frac{\tau_{ij}(t)^\alpha \cdot \eta_{ij}(t)^\beta}{\sum_{l \in J^k_i} \tau_{il}(t)^\alpha \cdot \eta_{il}(t)^\beta} & \text{if } j \in J^k_i \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

with $J^k_i = A_i \setminus \{\pi^k_{t-1}\}$ all possible displacements of the agent k from vertex i. Furthermore, $\tau_{ij}(t)$ is the trace intensity on the edge (i, j) and $\eta_{ij}(t)$ is the desirability of the edge (i, j) at a given time t. The importance of trace intensity in relation to the desirability of an edge is determined by the two parameters α and β. The intensity of the trace τ_{ij} on a given edge (i, j) indicates the number of times that edge has been crossed and serves as a guide for agents in selecting their path. Essentially, agents make decisions based on the behavior of other agents. The trace is considered a passive source of information since it is left unintentionally by agents and its value is incremented by a constant K after each movement as follows:

$$\tau_{ij}(t+1) = \tau_{ij}(t) + K, \tag{2}$$

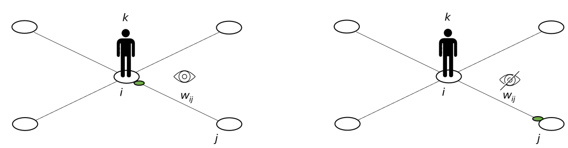
The value of K is determined by the user. According to Eq. 2, each agent unintentionally leaves a trace with a constant value after crossing an edge (i, j). The trace gradually diminishes over time, reflecting the impact of time on the environment. Specifically, after every T ticks[1], the amount of trace on the edges decreases based on the following rule:

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t), \tag{3}$$

with ρ evaporation decay parameter. The desirability $\eta_{ij}(t)$ in Eq. 1 determines how much an edge (i, j) is promising at a given time t. The agents do not know it in advance because it is connected to their state of knowledge of the environment. After crossing an edge, the agents intentionally release a piece of information about the weight of the crossed edge. However, agents do not directly see the

[1] The time unit used

weight of each edge but they can know it in two ways: (1) by traversing the edge and storing its weight in their memory. This constitutes their **prior knowledge** $\bar{w}_p^k$ equal to $\bar{w}_p^k = \frac{1}{n}\sum_{i=1}^{n} w(\pi_i, \pi_{i+1})$; (2) if a piece of information is present on the nearest endpoint of the edge to its current node. In detail, an agent k placed on a node i can see the weight w_{ij} of the edge (j, i) only if this piece of information is present on the nearest endpoint to node i of the edge (j, i). This constitutes their **local knowledge** $\bar{w}_l^k$ equal to $\bar{w}_l^k = \frac{1}{m}\sum_{i=1}^{m} w(\pi_i, \pi_{i+1})$, where $\pi(t) = (\pi_i, \pi_2, \ldots, \pi_{i+1})$ is a generic path from node i to node $i+1$, n is the number of visited edges and m is the number of neighbors' edges to the position where is present the information. The path cost of the entire path from the starting to the destination point is defined as$\sum_{i=1}^{t-1} w(\pi_i, \pi_{i+1})$, with π_1 and π_t are the starting and destination points, respectively. Consequently, the value



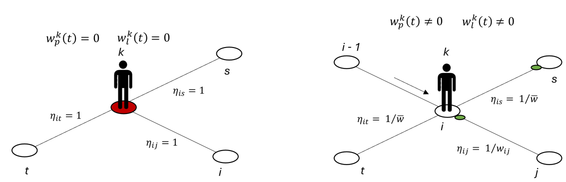
(a) (b)

Fig. 1: An agent k placed on a node i sees the information (green dot) about the weight of the edge (i, j) only if it is present on the nearest endpoint (a), otherwise it doesn't see it (b).

of the desirability $\eta_{ij}(t)$ will be determined as follow:

$$\eta_{ij}(t) = \begin{cases} \frac{1}{w_{ij}} & \text{if } \mathit{info} \neq 0 \text{ and } T \neq 0 \\ \frac{1}{\bar{w}} & \text{if } \mathit{info} = 0 \text{ and } T \neq 0 \\ 1 & \text{if } \mathit{info} = 0 \text{ and } T = 0 \end{cases} \tag{4}$$

where $\bar{w}$ is equal to: $\bar{w} = \frac{\bar{w}_p^k + \bar{w}_l^k}{2}$. To put it differently, when an agent k is situated on a node i, it will determine the desirability $\eta_{ij}(t)$ of an edge (i, j) by considering the weight of the edge. If information about the edge weight is available at either of the nearest endpoints to node i, the agent will evaluate the desirability as the inverse of the weight of the edge. In this situation, the lower the weight of an edge is, the higher its desirability will be. Fig 1 represents a common situation that happens in a simulation. Supposed the green circle to be the piece of information present on the endpoint of an edge (i, j). An agent k placed on the node i sees the information only if it is present on the nearest endpoint, otherwise, it doesn't see it. If information about the weight of neighboring edges (i, j) is missing or not visible but at least one piece of information is available, as represented in Fig 2b, the agent will estimate the desirability $\eta_{ij}(t)$ as the inverse of its global knowledge $\bar{w}$ about the environment. This estimation is obtained by averaging the agent's prior and local knowledge, as described in eq. **??**. However, it's possible for an agent to have no local knowledge $\bar{w}_l^k$ when no information is available about the weight of the neighboring edges. In such a scenario, the agent will evaluate the weight of each edge as the inverse of its prior knowledge $\bar{w}_p^k$. If both prior and local knowledge is zero, as in Fig 2a the agent will evaluate the desirability of each edge as equal to one. This situation can occur only at the beginning of exploration when the agent is positioned at the entrance of the environment and has yet to cross the first edge. In this



(a) (b)

Fig. 2: In (a) an agent k on a node i estimates the desirability of its adjacent edges as equal to 1 if and only if it is at the beginning of its exploration because it has not nor prior $w_p^k = 0$ neither local knowledge $w_l^k = 0$. In (b) an agent k arriving at a node i from a node $i-1$, estimates the desirability of its adjacent edges as equal to the inverse of its global knowledge if the information about their weights is missing or not visible. If the information is present and visible, it estimates the desirability as the inverse of the weight of the link.

model, there are two types of agents: collaborators (denoted by C) and defectors (denoted by D). When collaborators cross an edge (j, i), they leave behind a piece of information $\eta_{ij}(t)$ to aid other agents in selecting promising paths. Before making a decision on where to go, collaborators may also attempt to repair a damaged edge or vertex with probabilities P_e^C and P_v^C, respectively. On the other hand, defectors do not leave any information behind when crossing edges and may accidentally cause damage to an edge or vertex after crossing it, with probabilities P_e^D and P_v^D, respectively. A damaged edge becomes inaccessible, while a damaged vertex remains reachable but cannot be crossed. To elaborate, collaborators primarily engage in actions that support other agents in reaching the exit point as quickly as possible. They proceed cautiously through edges and vertices, taking care not to cause any harm and making repairs if needed. Furthermore, they leave information about edge costs $(\eta_{ij}(t))$ so that other agents can use it to make decisions. In contrast, defectors tend to behave hastily and carry out actions that could potentially disrupt the environment. After crossing a node or edge, they might accidentally damage it, which reduces the chances of other agents exploring the environment. This behavior could affect not only collaborative agents, but also themselves, especially if the disrupted path is critical to reaching the exit. Whenever agents traverse an edge, the time they take is directly proportional to the weight of that particular edge.

The presented investigation aim is to assess how the behavior of two agent types and the number of agent groups impact the crowd. To achieve this, we used the number of agents reaching the exit, path cost, and exit time as comparative measures. These metrics are concurrently analyzed since the destruction or repair of nodes and edges make the environment dynamic. Indeed, due to this different agents behavior, periodic destruction and/or restoration cause the environment's structure to change, making it dynamic.

3 Experimental Results

Two types of analyses was performed: *overall analysis* in which the system's performance are investigated when divided into groups of different sizes, and *type analysis* in which collaborators' performance and defectors present in the same groups are compared. For each analysis, as said, we simultaneously compare three evaluation metrics: number of agents reach the exit; exit times; and paths cost.

The proposed model was developed using the NetLogo framework [18], a well-known programmable modeling environment for multi-agent systems. The environment is modeled as a grid network where each node may be connected to up to eight neighbors, and where to each edge is assigned a weight that is a real number randomly selected uniformly from the range $]0,1]$. Specifically to this investigation, an environment with $|V| = 225$ vertices and $|E| = 501$ edges was put up. Further, a scenario with $N = 1000$ agents split into Γ groups has been simulated, where the Γ value vary among the following set: $1, 2, 4, 5, 10, 100$. The distribution of the types of agents inside groups is determined by the real user-defined parameter $f \in [0,1]$, called *collaborative factor*: the value assigned to f determines the ratio of collaborative and defector agents. Specifically, f represents the fraction of collaborative agents, and $(1-f)$ of defector ones. As a result, a group may have either one type of agent or a mix of both: when $f = 0.0$, only defector groups are formed; when $f = 1.0$, only collaborative groups are formed; when, for instance, $f = 0.6$, each group will have 60% of agents as collaboratives and 40% as defectors. Any group starts its exploration after a fixed time of $T_e = |V|$ has elapsed since the preceding group's start: i–th group begins its exploration at time $t = T_e \times (i-1)$. A time limit is set for the entire crowd to reach the exit, and it is defined as $T_{max} = c \times \Gamma \times |V|$, with c constant factor set to 5. Group membership of an agent determines the time window within which it must reach the exit, and i.e. $T_i = T_{max} - (T_e \times (i-1))$, where i represents the group to which the agent belongs. Follows that agents of the early groups have a more extended time frame to explore the environment than those in the later groups. This means that agents belonging to the same group may exit at different time. The impact of time in this model is represented as a gradually reduction over time of the trace in the environment, considering a fixed degradation interval of $T_d = |V|$. As a result, the Eq. 3 rule is applied every T_d ticks with an evaporation rate of $\rho = 0.001$. In the beginning, the trace on all edges is set to $\tau_{ij}(t = 0) = 1.0$. Furthermore, the destruction and repair probabilities are equal for both types of agents,($P_e^C = P_e^D = 0.02$ and $P_v^C = P_v^D = 0.02$). To pursue our goal, i.e. investigate how different sizes of groups affect the whole system, several experiments at the f varying have been performed (from 0.0 to 1.0 with 0.1 increment) and for each simulation, 10 independent runs were performed. These experiments have been conducted for all Γ values considered.

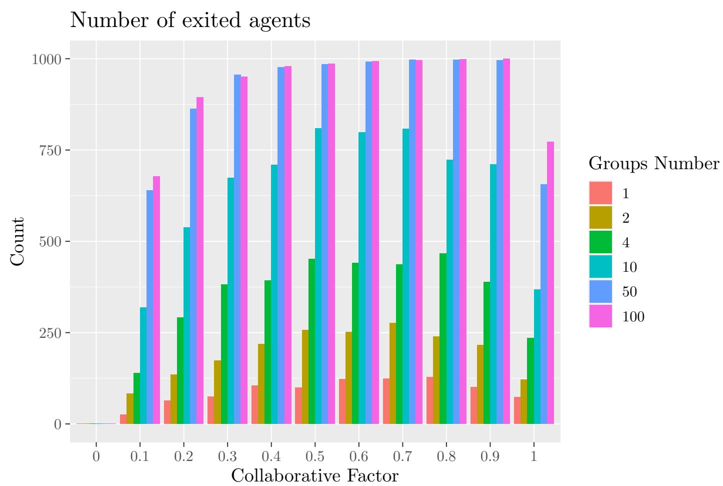


Fig. 3: Overall number of exited agents

The first investigation focuses on analysing the system as a whole observing the contributions made by both collaborators and defectors. The purpose is to figure out how group size affects the entire system. Fig 3 shows the number of exited agents at different group values, and at collaborative factor f varying. Each colored tile in the legend indicates the number of groups Γ that the crowd has been split into. As the number of groups increases, the number of agents who reach the exit also increases, particularly when the crowd is separated into 50 or 100 groups, almost all of the agents in the crowd reach the exit when $0.6 \leq f \leq 0.9$. Crowd performance worsens as the Γ decreases, as well as when the crowd is entirely collaborative ($f = 1.0$), with the worst configuration being one in which all crowd agents belong to a single group. In conclusion, when the crowd is split into several small groups, many more agents reach the exit compared to when it is divided into few very large groups or a single group that contains the entire crowd. The figures 4 and 5 display the exit times and path costs, respectively. These metrics have been normalized based on the group success rate, which is the percentage of agents in a group that reaches the exit point. Interestingly, the crowd appears to exit faster when divided into a few

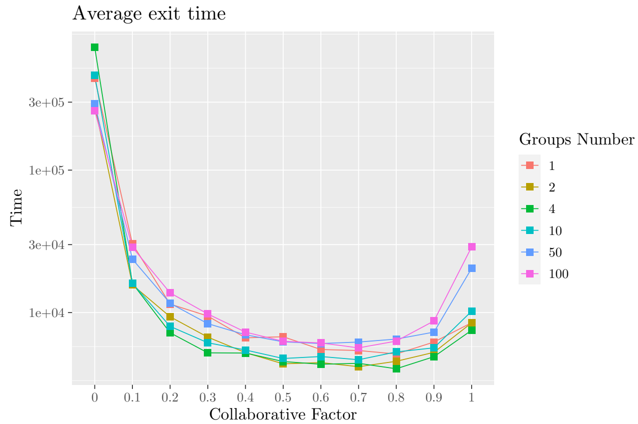


Fig. 4: Overall exit time.

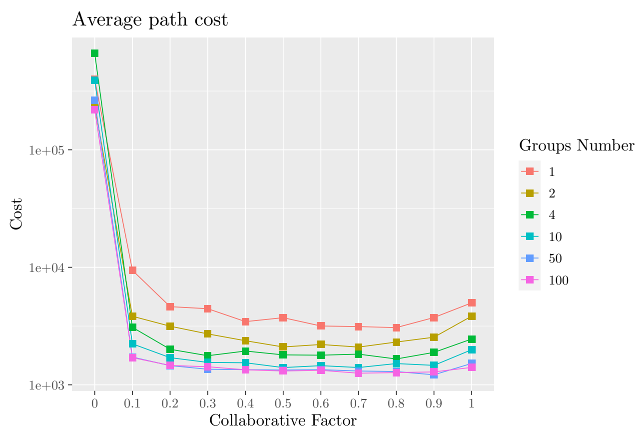


Fig. 5: Overall path cost.

groups with a large number of agents, but at the same time appears to find cheaper paths (as well as have much more agents that reached the exit) when divided into a large number of groups with few agents. This indicates that the exit times are obtained by a small number of agents and, therefore, evaluating all metrics simultaneously shows that the crowd's performance cannot be positively assessed when it is divided into a few groups with a lot of agents. Indeed, in this case, the path costs are worse and this suggests that the optimization cost process is driven primarily driven by the groups rather than the number of agents itself. Generally, except for the exit times, the more the number of groups into which the crowd is divided, the better its performance is. Furthermore, when the crowd is solely composed of collaborative agents ($f = 1.0$), its exit times and path costs are worse for all values of the groups' number.

The second analysis conducted addresses the investigation of the system's performance by separately inspecting the performance of collaborators and defectors. The goal is to identify any disparities between the two types of performance and determine which type of agent would perform better. Figure 6 displays the number of agents that exited, categorized by type, for various group values as the collaborative factor, f, changes. The number of exited collaborators and defectors both increase as the collaborative factor increases and reaches their maximum at different f values based on Γ. The most notable difference between the two is that collaborators' maximum is achieved at higher f values, and the trend is nearly linear for $\Gamma > 10$ with a slight drop at $f = 1.0$. In contrast, the defectors' maximum is attained at lower f values, and the trend is also mostly linear for $\Gamma > 10$, but with a decreasing trend. The two figures, Fig 7 and Fig 8,

Fig. 6: Number of exited agents per type C and D

display the exit times and path costs for both agent types at different values of f and Γ. The results confirm the overall analysis, which suggests that collaborators and defectors exit faster when there are fewer groups with many agents, while they find cheaper paths with more groups containing fewer agents. Collaborators exhibit different best exit times for different values of Γ, with $f = 0.9$ for $\Gamma \geq 50$ and $f = 0.7$ for $\Gamma \leq 50$. When $f = 1.0$, the performance worsens, as seen in the previous metric. Defectors show smoother behavior, with exit times improving slowly as f increases. It is worth noting that there is a performance jump between $\Gamma = 50$ and $\Gamma = 10$, which may be due to a significant difference in values. These observations apply to the path costs as well, although the performance differences are less pronounced. Collaborators and defectors find better paths as Γ and f increase, and their behavior is more stable as Γ increases. However, in this case, collaborators do not perform better when $f = 1.0$

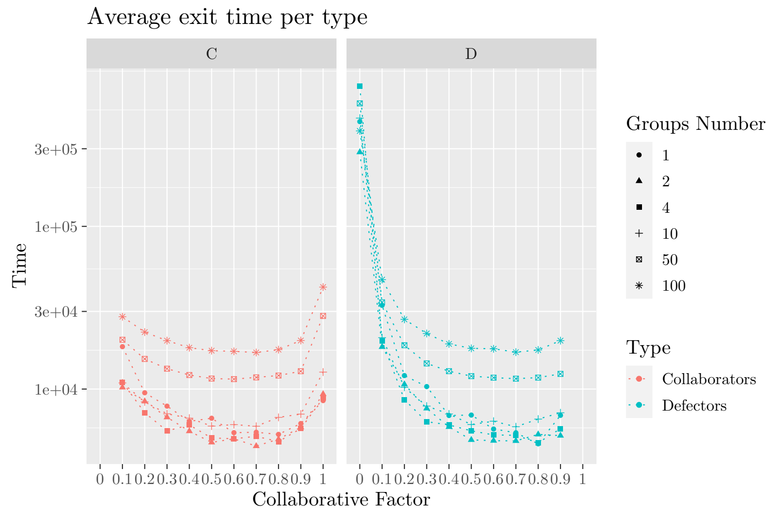


Fig. 7: Exit time per type C and D

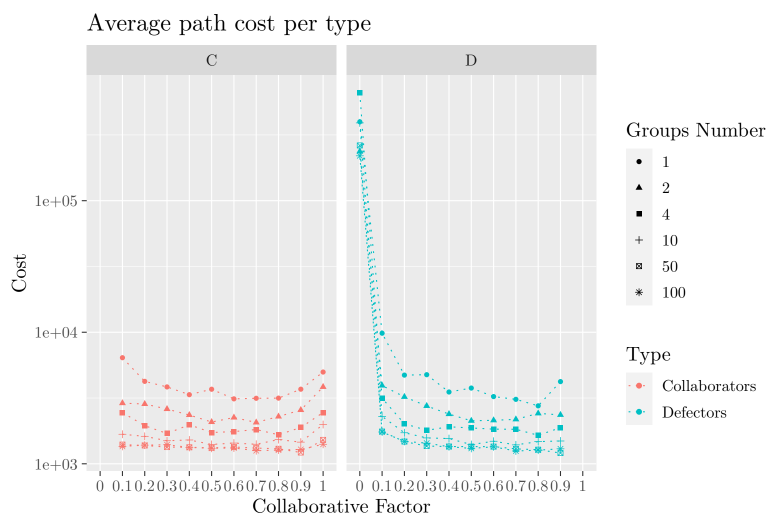


Fig. 8: Path cost per type C and D

4 Conclusions

In this research work, we introduced an agent-based model to simulate crowds and investigate the effects of dividing the population into Γ groups of varying sizes. The model is based on the Ant Colony Optimization algorithm (ACO) and includes two types of agents: collaborators and defectors. The goal of both agents is to reach the exit of a virtual environment starting from a designated entrance. Collaborators help other agents reach the exit by providing information about the paths they have taken and repairing damaged paths. Defectors, on the other hand, do not provide information and may accidentally damage paths. The proportion of collaborative and defector agents is controlled by a parameter called the collaborative factor (f), which is expressed as a fraction of the total population. To evaluate the performance of the model, we used metrics such as the number of agents that successfully exited, exit times, and path costs. We conducted two types of analyses: an *overall analysis*, where we varied the collaborative factor and group sizes, and a *type analysis*, where we separately evaluated the performance of collaborators and defectors. Our results show that when the crowd is divided into fewer, larger groups, the exit times are faster. However, when the population is divided into many smaller groups, more agents can exit, and they are able to find cheaper paths. We also found that, overall, collaborative agents outperform defectors, particularly in terms of the number of agents that successfully exit. However, defectors have a slight advantage in terms of exit times and path costs, as they are better able to take advantage of the presence of collaborative agents. Our findings suggest that the size and number of groups can significantly impact crowd behavior and should be considered when designing crowd management strategies.

Acknowledgements This research is supported by the project Future Artificial Intelligence Research (FAIR) – PNRR MUR Cod. PE0000013 - CUP: E63C22001940006

References

1. Aghabayk, K., Parishad, N., Shiwakoti, N.: Investigation on the impact of walkways slope and pedestrians physical characteristics on pedestrians normal walking and jogging speeds. Safety Science **133**, 105012 (2021)
2. Crespi, C., Fargetta, G., Pavone, M., Scollo, R.A.: An agent-based model to investigate different behaviours in a crowd simulation. In: Bioinspired Optimization Methods and Their Applications: 10th International Conference, BIOMA 2022, Maribor, Slovenia, November 17–18, 2022, Proceedings. pp. 1–14. Springer (2022)
3. Crespi, C., Scollo, R.A., Fargetta, G., Pavone, M.: How a different ant behavior affects on the performance of the whole colony. In: Metaheuristics: 14th International Conference, MIC 2022, Syracuse, Italy, July 11–14, 2022, Proceedings. pp. 187–199. Springer (2023)
4. Huang, Z.M., Chen, W.N., Li, Q., Luo, X.N., Yuan, H.Q., Zhang, J.: Ant colony evacuation planner: An ant colony system with incremental flow assignment for multipath crowd evacuation. IEEE Transactions on Cybernetics **51**(11), 5559–5572 (2021)
5. Khamis, N., Selamat, H., Ismail, F.S., Lutfy, O.F., Haniff, M.F., Nordin, I.N.A.M.: Optimized exit door locations for a safer emergency evacuation using crowd evacuation model and artificial bee colony optimization. Chaos, Solitons, Fractals **131**, 109505 (2020)

6. Kole, A., De, D., Pal, A.J.: Solving graph coloring problem using ant colony optimization, simulated annealing and quantum annealing—a comparative study. In: Intelligence Enabled Research: DoSIER 2021, pp. 1–15. Springer (2022)
7. Köster, G., Seitz, M., Treml, F., Hartmann, D., Klein, W.: On modelling the influence of group formations in a crowd. Contemporary Social Science **6**(3), 397–414 (2011)
8. Li, C., Lv, P., Manocha, D., Wang, H., Li, Y., Zhou, B., Xu, M.: Acsee: Antagonistic crowd simulation model with emotional contagion and evolutionary game theory. IEEE transactions on affective computing **13**(2), 729–745 (2019)
9. Li, Z., Huang, H., Li, N., Zan, M.L.C., Law, K.: An agent-based simulator for indoor crowd evacuation considering fire impacts. Automation in Construction **120**, 103395 (2020)
10. Matkovic, F., Ivasic-Kos, M., Ribaric, S.: A new approach to dominant motion pattern recognition at the macroscopic crowd level. Engineering Applications of Artificial Intelligence **116**, 105387 (2022)
11. Sharbini, H., Sallehuddin, R., Haron, H.: Crowd evacuation simulation model with soft computing optimization techniques: a systematic literature review. Journal of Management Analytics **8**(3), 443–485 (2021)
12. Siyam, N., Alqaryouti, O., Abdallah, S.: Research issues in agent-based simulation for pedestrians evacuation. IEEE Access **8**, 134435–134455 (2020)
13. Trivedi, A., Rao, S.: Agent-based modeling of emergency evacuations considering human panic behavior. IEEE Transactions on Computational Social Systems **5**(1), 277–288 (2018)
14. Vezzoli, M., Zogmaister, C.: An introductory guide for conducting psychological research with big data. Psychological Methods (2023)
15. Wang, Q., Liu, H., Gao, K., Zhang, L.: Improved multi-agent reinforcement learning for path planning-based crowd simulation. IEEE Access **7**, 73841–73855 (2019)
16. Wang, S., Liu, H., Gao, K., Zhang, J.: A multi-species artificial bee colony algorithm and its application for crowd simulation. IEEE Access **7**, 2549–2558 (2019)
17. Wang, Y., Wang, L., Chen, G., Cai, Z., Zhou, Y., Xing, L.: An improved ant colony optimization algorithm to the periodic vehicle routing problem with time window and service choice. Swarm and Evolutionary Computation **55**, 100675 (2020)
18. Wilensky, U.: Netlogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL (1999), http://ccl.northwestern.edu/netlogo/
19. Xie, W., Lee, E.W.M., Li, T., Shi, M., Cao, R., Zhang, Y.: A study of group effects in pedestrian crowd evacuation: Experiments, modelling and simulation. Safety Science **133**, 105029 (2021)
20. Yang, S., Li, T., Gong, X., Peng, B., Hu, J.: A review on crowd simulation and modeling. Graphical Models **111**, 101081 (2020)
21. Yang, X., Wang, Q.: Crowd hybrid model for pedestrian dynamic prediction in a corridor. IEEE Access **7**, 95264–95273 (2019)
22. Yücel, F., Sürer, E.: Implementation of a generic framework on crowd simulation: a new environment to model crowd behavior and design video games. Mugla Journal of Science and Technology **6**, 69–78 (2020)
23. Zhou, Z.X., Nakanishi, W., Asakura, Y.: Route choice in the pedestrian evacuation: Microscopic formulation based on visual information. Physica A: Statistical Mechanics and its Applications **562**, 125313 (2021)
24. Zong, X., Yi, J., Wang, C., Ye, Z., Xiong, N.: An artificial fish swarm scheme based on heterogeneous pheromone for emergency evacuation in social networks. Electronics **11**(4) (2022)