MULTI-AGENT REINFORCEMENT LEARNING-BASED PEDESTRIAN DYNAMICS MODELS FOR EMERGENCY EVACUATION

FINAL REPORT

JUNE 2022

AUTHORS
Hyoshin Park

North Carolina A&T University

Dahai Liu, Sirish Namilae

Embry-Riddle Aeronautical University

DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the Department of Transportation, University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Multi-agent Reinforcement Learning-based Pedestrian Dynamics Models for Emergency Evacuation		5. Report Date 06/1/2022
		6. Source Organization Code
7. Author(s) Hyoshin Park, Dahai Liu, Sirish Namilae		8. Source Organization Report No. CATM-2022-R5-NCAT
9. Performing Organization Name and Address Center for Advanced Transportation Mobility Transportation Institute 1601 E. Market Street Greensboro, NC 27411		10. Work Unit No. (TRAIS)
		11. Contract or Grant No. 69A3551747125
12. Sponsoring Agency Name and Address University Transportation Centers Program (RDT-30) Office of the Secretary of Transportation–Research U.S. Department of Transportation 1200 New Jersey Avenue, SE Washington, DC 20590-0001		13. Type of Report and Period Covered Final Report: 4/2019-5//2022
		14. Sponsoring Agency Code USDOT/OST-R/CATM

15. Supplementary Notes:

16. Abstract

An efficient and safe evacuation of passengers is important during emergencies. Overcapacity on a route can cause an increased evacuation time. Decision making is essential to optimally guide and distribute pedestrians to multiple routes while ensuring safety. Developing an optimal pedestrian path planning route while considering learning dynamics and uncertainties in the environment generated from pedestrian behavior is challenging. While previous evacuation planning studies have focused on either simulation of realistic behaviors or simple route planning, the best route decisions with several intermediate decision points, especially under real-time changing environments, have not been considered. This project develops an optimal navigation model providing more navigation guidance for evacuation emergencies to minimize the total evacuation time while considering the influence of other passengers based on the social-force model. The integration of the optimal navigation model was ultimately able to reduce the overall evacuation time by 10.6%, compared to the use of only one modeling approach. This study further simulated robot agents to explore and evacuate from a dynamic environment with or without collaboration using the MARL Q-learning algorithm. The multi-agent collaboration method was found to perform better than the single-agent exploration regarding the evacuation time, death counts, and reward both in the static threats and dynamic threats environment. Results were discussed, and future directions were given in the end.

17. Key Words Reinforcement Learning, Multi-agent collaboration, emergency, airport evacuation		18. Distribution Statement Unrestricted; Document is available to the public through the National Technical Information Service; Springfield, VT.		
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified		21. No. of Pages 80	22. Price …

Form DOT F 1700.7 (8-72)   Reproduction of completed page authorized

Executive Summary

With the considerable development of the air transportation system in recent years, the demand for a more efficient evacuation system for airport security services has increased. The efficient and safe evacuation of passengers is important during emergencies. Overcapacity on a route can cause an increased evacuation time. Decision making is essential to optimally guide and distribute pedestrians to multiple routes while ensuring safety. Developing an optimal pedestrian path planning route while considering learning dynamics and uncertainties in the environment generated from pedestrian behavior is challenging. While previous evacuation planning studies have focused on either simulation of realistic behaviors or simple route planning, the best route decisions with several intermediate decision points, especially under real-time changing environments, have not been considered. The purpose of the study is to investigate the utilization of different machine learning methods using Deep Reinforcement Learning (DRL) to optimize evacuation in dynamic and complex environments such as fire, terrorist attacks, and more, particularly in an environment such as an airport.

This project develops an optimal navigation model providing more navigation guidance for evacuation emergencies to minimize the total evacuation time while considering the influence of other passengers based on the social-force model. The integration of the optimal navigation model was ultimately able to reduce the overall evacuation time by 10.6%, compared to the use of only one modeling approach. This study further simulated robot agents to explore and evacuate from a dynamic environment with or without collaboration using the MARL Q-learning algorithm. The multi-agent collaboration method was found to perform better than the single-agent exploration regarding the evacuation time, death counts, and reward both in the static threats and dynamic threats environment. Results were discussed, and future directions were given in the end. DRL has been widely used in intelligent transportation systems, especially under emergent situations, and can explore the dangerous environment and make optimal decisions to guide the evacuation process for human beings.

Publication and Award records

1. Y. Yupeng, Y. Jiahao and L. Dahai and L. Sang-A and N. Sirish and I. Sabique-UI and G. Huaxing and H. Park, Multi-agent collaboration for emergency evacuation using reinforcement learning for transportation systems. IEEE Journal on Miniaturization for Air and Space Systems. (In Press)
2. N. Pugh, H. Park, P. Derjany, S. Namalie, D. Liu. 2021. Deep adaptive learning for safe and efficient navigation of pedestrian dynamics. IET Intelligent Transport Systems, 15 (4), 538–548.
3. 2nd Winner (Honorable Mention) INFORMS 2020 Annual Meeting MIF student poster competition (Mr. Pugh), Safe and Optimal Pedestrian Navigation in a Dynamic Emergency Environment, https://www.informs.org/Recognizing-Excellence/Award-Recipients/Nigel-Pugh
4. N. Pugh, H. Park. Deep Adaptive Learning For Safe And Efficient Navigation Of Pedestrian Dynamics – Airport Evacuation. Poster Presentation at the INFORMS 2019 Annual Meeting, Seattle, WA, October 20-23, 2019.

The multiagent of this project is divided to human in part 1 and robot in part 2. The collaboration between human and robot is out of scope of this project.

Part 1. Deep Adaptive Learning for Safe and Efficient Navigation of Pedestrian Dynamics

Introduction

Pedestrian evacuation in emergencies is a significant and ubiquitous problem especially when non-optimal pedestrian evacuations strategies are considered. Non-optimal evacuations can cause unnecessarily prolonged evacuation times and increased injuries or fatalities. When an emergency occurs, human pedestrians are often required to make quick decisions under high time-sensitive pressure while dealing with uncertainty and unfamiliar surroundings (Neria et al. 2008; Alexander et al. 2013). Successful decision-making is largely dependent on the extent of information availability, accuracy of the information given, limited processing time (Proulx et al. 1991; Proulx 1993; Yoon et al. 2008; Hasan et al. 2011; Heliovaara et al. 2012; Knuth et al. 2013), and several interdependent human (e.g. decision making, cognition and perception) and environmental factors (e.g. layout, pedestrian density, visual range and obstacles, type of emergency etc.) (Purser et al. 2001 ; Cassidy et al. 2002 ; Cocking et al. 2009; Kobes et al. 2009 ; Vorst et al. 2010; Abolghasemzadeh et al. 2013 ; Mu et al. 2013). The uniqueness of the emergent situation makes computational modeling an essential tool for policy analysis and design (Shi et al. 2009; Miyoshi et al. 2012).

While there are numerous successful models like social force models (Helbing et al. 1995; Namilae et al. 2017), and agent-based models (Kirby et al. 2015; Chen et al. 2018) that address high density crowds, there is a glaring lack of effective modeling techniques targeted at low-to-medium-density pedestrian situations. Furthermore, previous studies have focused on either pedestrians' strategic route planning or pedestrians' physical movements without considering the interactions between these two levels. Most of the previous studies have focused on the simulation of realistic pedestrian dynamics, but the decision-making process has been static and hard to adapt to dynamic environment changes. Especially when there are multiple stationary obstacles (e.g.,

chairs, walls) and moving obstacles (e.g., other pedestrians), the optimal strategy keeps changing accordingly.

To safely reduce the congestion effect among other agents and ultimately produce a shorter evacuation time, an adaptive routing strategy per individual passenger is required. In this study, while previously developed social force model (Chen et al. 2018) are trained by real-world data to provide a realistic simulation of pedestrian behaviors under emergent situations, a dynamic routing model is developed to suggest the best options to evacuate faster. A performance of evacuation times will be compared against our optimally guided path planning model and social force model without optimal path guidance.

Figure 1
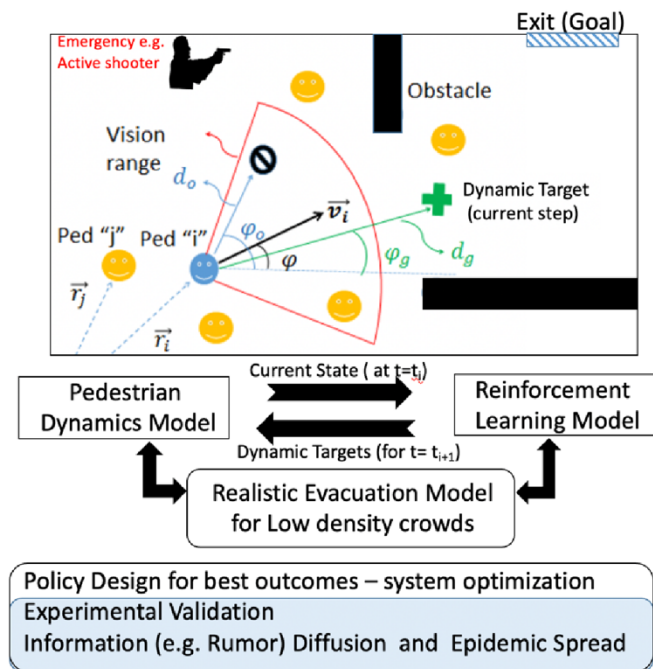
The framework for our Optimally Guided Path Planning Model Safe and Efficient Navigation of Pedestrian for Evacuations. The social force model (Pedestrian Dynamics Model) provides density input for the optimally guided path planning model. The two models integrated together provide a realistic evacuation model for low density crowds. The integrated model can be expanded for policy designs for Information Diffusion and Epidemic Spread.

In this study, we present a computational modeling framework expanded from previously developed social force model applied to an evacuation simulation. The pedestrian dynamics patterns are integrated into the path planning model solved by Markov Decision Process (MDP). While there are numerous limitations from previous approaches, the contribution of this project is summarized as follows:

1. Model the process of pedestrian making choices by identifying a realistic decision and integrating with local pedestrian movements in a complex environment in an emergency evacuation.

2. Integrate social force modeling with the optimally guided path planning model for emergency evacuations.

3. Simulate newly integrated optimally guided path planning model applied to a pedestrian evacuation.

4. Evaluate and compare the performance of average pedestrian evacuation time with and without optimal path guidance.

Literature Review

In prior related work (Namilae et al. 2017; Chunduri et al. 2018; Derjany et al. 2018; Namilae et al. 2018; Liu et al. 2018) we developed a particle dynamics pedestrian movement model that tracks the movement of individual passengers in airplanes, airports, and combined it with stochastic infection dynamics. In Liu and Namilae et al. (2018), human panic behaviors were considered, and several simulations models were developed based on the review of previous studies, using social forces models and agent-based models. These factors include pedestrian density, environmental barriers, layout, numbers of evacuees, age, and gender. The significant effects of these factors on the efficiency of evacuations were identified and these results laid out a solid foundation for data collection, mathematical formulation, and simulation analysis for addressing the human behaviors under emergency. Other recent pedestrian evacuation studies using social force models such as Zhou et. al (Zhou et al. 2019) have studied route choice of pedestrians based on density, distance, and capacity factors. The study provided valuable insight at understanding effective strategies that result in quick evacuation response times depending on the number of evacuating pedestrians.

An MDP is a framework that has shown promise with the ability to solve path planning problems (Berlet et al. 2004; Ferguson et al. 2004; Alterovitz et al. 2007; Achour et al. 2010; Sidrane et al. 2018; Nardi et al. 2019). MDPs have also been used for modeling problems such as crowd simulation (Loza and Hernandez 2017), optimal dispatching (Keneally et al. 2016; Jenkins et al. 2018), dynamic vehicle routing (Ulmer et al. 2016), e-bike drivers decision-making during signal change (Dong et al. 2019), evacuation planning, and dispatching (Jenkins et al. 2018). Sidran et. al. (2018) solved an MDP using Mixed Integer Programming for an evacuation route planning problem in a closed loop. This method achieved to perform 90% performance of an optimal MDP problem. Jeong et. al. (2014) solved a mission planning problem involving multi-UAV surveillance by formulating an MDP (Zebala et al. 2012). Burlet et. al. (2004) presented an MDP-based planning tool for mobile robots. Nardi et. al (2019) proposed an uncertainty augmented MDP for path planning on road networks that considered the uncertainty of a robot's location. The approach taken was able to trade off safety and travel time utilizing the information on the robot's uncertainty. Achour et. al. (2010) was able to achieve improved execution time of MDP in a robot path planning example. Allamraju et. al. (2014) provided a non-stationary MDP based model for Unmanned Aerial Systems to avoid the potential of a UAV being seen by a human for privacy reasons. The goal was to plan a route that avoided the anticipation of where the human population would be based on time of day, year, and season.

This study integrates a social force model and our optimally guided path planning model. The optimally guided path planning model navigates a group of pedestrians on a separate guided route that avoids the social force model path. The information on the local environment and human behavioral characteristics are formulated into a reward matrix to re-plan the pedestrians' path or adapt to the changes in the environments. This approach incorporates modeling of the nonlinear characteristics of the human decision-making processes beyond simple rule-based models. The main goal is to decrease the overall evacuation time with the incorporation of the optimally guided path planning model.

Methodology

Social Force Model

We model each mobile pedestrian as a particle, and immobile objects like walls as groups of stationary particles. The evolution of pedestrian *i* and their interaction with other pedestrians and stationary objects are modeled by molecular dynamics like a social force model (Helbing and Molnar 2000). The net force acting on the pedestrian *i* can be defined as:

$$\bar{f}_i = \frac{m_i}{\tau}\left(\bar{v}_0^i(t) - \bar{v}^i(t)\right) + \sum_{j \neq i}\bar{f}_{ij}(t) = m_i\frac{dv_i}{dt} \tag{1}$$

with the pedestrian *i*'s position at a given time obtained by integrating $r^{-i}(t) = v^{-i}(t)dt$ (the desired velocity of pedestrian *i*), $v^{-i}(t)$ (that of the actual velocity), m_i (the mass of pedestrian *i*) and τ (the evolution time constant). The momentum generated by a pedestrian *i*'s intention, denoted by $\frac{m_i}{\tau}\left(\bar{v}_0^i(t) - \bar{v}^i(t)\right)$, results in a self-propulsion force that is balanced by a repulsion force to obstacles in the direction of motion. We introduce location dependence to the desired velocity in the self-propulsion term as:

$$v_{0i}(t) = (v_A + \gamma_i v_B)\left[1 - \frac{\delta}{\min\left(\|\vec{r}_i - \vec{r}_j\|; i \neq j\right)}\right] \tag{2}$$

where $e_v = \cos(\phi)\hat{\imath} + \sin(\phi)\mathbb{Z}$ is the direction of desired motion. b

v_A and $\gamma_i V_B$ are the deterministic and stochastic components of desired velocity, δ is a distance constant such that at distance δ between i^{th} and k^{th} pedestrians the desired velocity of ith pedestrian is zero. The active particle evolution based on these equations in certain instances will be modified to match targeted behavior like in agent-based models. The walking speeds of individuals vary by age, group, and sex (Fang et al. 2004). In crowded locations like airports and mass gatherings, pedestrians often need to circumvent stationary and mobile obstacles. While the social force model works in simple situations, modifications to equations of motion are required for trajectories involving obstacle avoidance. While the social force model works in simple situations, modifications to equations of motion as shown in equation (2) are required for trajectories involving obstacle avoidance and formation of pedestrian queues.

Pedestrian Dynamic Agent Based Simulation

We developed a simulation model of a midsize Airport (KTAR) by incorporating the social force model (Section 3.1) in agent-based simulation software AnyLogic(Kobes et al. 2009; Mu et al. 2013). As seen in Figure 2, KTAR has six doors for evacuation and in the model, it was assumed that the middle exit is blocked by danger and not accessible. All the passengers would use the other five doors equally (i.e., each door had a 16.67% chance to be chosen by passengers to evacuate). Passengers were expected to evacuate in the following six steps: (1) leave the aircraft and enter the terminal through the aerobridge at the second floor; (2) proceed to the escalator or stairs that connect the second floor with the first floor; (3) make a choice between using escalator or stairs; (4) reach the first floor via the choice they made; (5) choose one of the available doors to exit; (6) proceed to the selected door and leave the terminal. However, it should be noted that only the passengers inside the airport were considered; consequently, the time taken by the passengers to evacuate the aircraft and reach the airport was not considered for this study.

The walking speed of the pedestrians in this study was set between 1.2 and 1.8 m/s. According to (Martinez et al. 2012), in the emergency evacuation simulation, the passenger's walking speed was uniformed distributed between 1.2 and 1.8 m/s. In addition to the sources of data mentioned above, the average speed of pedestrians on escalator and stairs were also provided by previous studies (Chen et al. 2018). For the data collection device, this study was collecting the total evacuation time from when the first passenger emerged on the second floor to when the last passenger left the terminal.

One thousand passengers were generated in the gate area on the second floor of the airport, these passengers then ran to escalator and stairs down to the first floor to evacuate, simulating an emergency scenario. On the first floor of the airport terminal. There are six doors in the first-floor terminal, it was assumed the middle exit is blocked and not accessible. Thus, passengers can only use the other five doors to evacuate the airport. The authors in this project previously worked on this airport evacuation simulation. Our current work expands the previous simulation by incorporating the optimally guided path planning model to better navigate the evacuation pedestrians.

Figure 2

An Emergency evacuation simulation of 1000 passengers using Anylogic software. Pedestrians were placed on the second level and required to exit on the first floor. The first floor has 6 exits location with one exit not accessible. The model runs 50 times and the average time to evacuate is 2115.53 seconds and standard deviation is 47.20 seconds.

[Upper Level Terminal of Airport Building]
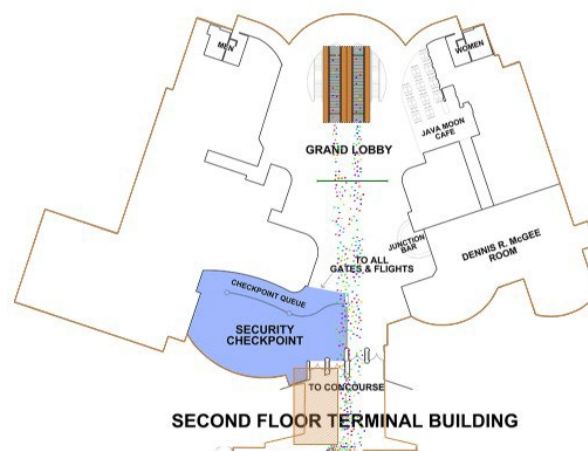
[Lower Level Terminal of Airport Building]
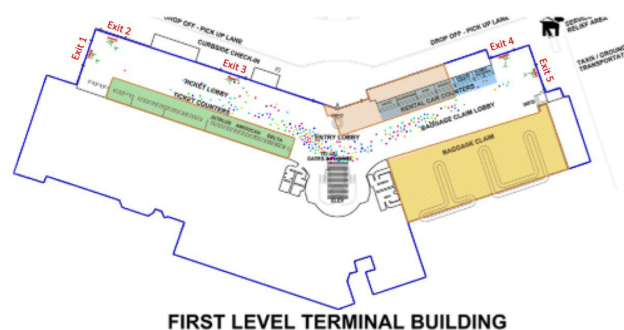
Optimally Guided Path Planning Model

A Markov chain is a random process in discrete time that is a sequence of a random variable. We can describe a Markov chain tuple as follows $< S,A,P,R >$. There are

a set of finite states $S = \{S_1, S_2,...,S_N\}$, N being the number of possible states. A is a finite set of actions $A = \{A_1, A_2,...,A_n\}$, n is the number of possible actions. Actions allow agents to transition from one state to another. A state transition $P : SXAXS^0 - > [0,1]$, is the transition function of probabilities of transition from state S to state S'. R is defined as the Reward function. The transition probability (P_{ij}) matrix is a (I x J) matrix that can be denoted as:

$$P = \begin{bmatrix} p & p_{21} & ... & P_{1J} \\ p_{21} & p_{22} & ... & P_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ p_{11} & p_{12} & ... & P_{IJ} \end{bmatrix} \qquad (3)$$

R (S, A, S') is the reward vector, defined as the reward received for a transition from state S to State S'. This model rewards state is determined by the pedestrian movements taken by the social force model (this will be discussed in detail in Section 5.1).

We developed a reinforcement learning technique to learn local navigation behaviors and simulate dynamic pedestrian behaviors when there is an emergency at airport buildings. This model determines intermediate goals for each pedestrian, which is a key input for the time evolution of pedestrian trajectories. Previous reinforcement learning studies have used desired velocity and the proximity to other pedestrians, when the intermediate goal is moved along with the environment, previously learned information interferes with the task of finding the new goal. The sub-agent model we developed separately. This introduces stochastic and dynamic terms (e.g., self-propulsion) to provide a more realistic learning process and simulation.

To guarantee global goal-seeking in complex dynamic environments, we formulate the problem, with the main objective of maximizing the expected accumulative reward r of with paired action and space (s, a). The Q look-up table provides training of the best action $a \in A$ in a finite action space, based on pedestrian's state $s \in S$, for optimal learning. The discount factor, γ, balances the immediate reward (exploration) and future reward (exploitation). A negative reward is given for crashing into stationary objects or another pedestrian at the airport in an evacuation scenario to learn a proper action.

$$Q: Q(s_t, a_t) = (1 - \alpha)Q(s_t, a_t) + \alpha(r_{t+1}, +$$
$$\gamma \max_a \{Q(s_{t+1}, a) - Q(s_t, a_t)\} \tag{4}$$

The pedestrian's state $S = \left(v_p, a_v, d_q, v_p^t, d_p^t, a_p^t, d_p^f, a_p^f\right)$ has high dimensional

space with features: v_p (velocity of the current pedestrian as an out from the social force

model); a_v (angle of the velocity vector relative to the reference line), d_g (distance to the

goal), v_p^i (velocity of the nearest pedestrian i in current pedestrian's visible range); d_p^i

(distance to the nearest pedestrian i), a_p^i (angle of the nearest pedestrian i location relative

to the reference line), d^j_p (distance to the stable object j (chairs and walls), a^j_p (angle of

the stable object j location relative to the reference line). As shown in Figure 1, the initial

pedestrian dynamic model provides the velocity v_p (t=1) as an input to the reinforcement

learning model in the initial stage ($s_{t=1}, a_{t=1}$). In each time step, the optimal action of the

reinforcement learning model serves as an input to velocity model for more accurate

velocity information v_p (t=2) in the dynamic environment. The velocity is used to update

the next time stage state and action ($s_{t=2}, a_{t=2}$) in the reinforcement learning model (Sutton

and Barto 2018).

$$SARSA: Q(s_t, a_t) = Q(s_t, a_t) + \alpha(r_{t+1}, +$$
$$\gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \tag{5}$$

The SARSA (State-Action-Reward-State-Action) algorithm (Helbing et al. 2000)

is similar to the Q-Learning algorithm except it is an on-policy algorithm for Temporal

Difference (TD) Learning. This algorithm varies slightly from Q-Learning, when

updating, the maximum reward is not always chosen and updates Q-values. Rather the

SARSA algorithm utilizes the same policy that was generated from the previous action

for the next upcoming action. This is accomplished by completing the Q function updates

in equation 5. The discount factor, γ, balances the immediate reward (exploration) and

future reward (exploitation), similar to the Q-Learning approach.

Research Approach

Environment

The Optimally Guided Path Planning model environment for reinforcement learning created is a 10 x 10 grid world (500 sq. in by 500 sq in.), with eleven obstacles and a terminal state (exit) (100 sq. in. opening) shown in Figure 6. The pedestrian icon represents the starting location and the exit signs represent the terminal states. Currently, the reward functions value is determined by the states

Figure 3

A Pedestrian Density Map generated from our evacuation scenario produced by AnyLogic. The simulation is of 50 pedestrians navigating throughout the environment to the exit location. The red shaded areas pictured show the most populated areas of the environment. The blue shaded areas show the areas that are not highly dense. The upper limit of the density range is set by default to 1.5 m/p² by AnyLogic software. The pedestrian density map enables us to determine the most densely populated areas and incorporate into our optimal path guidance model for path planning.
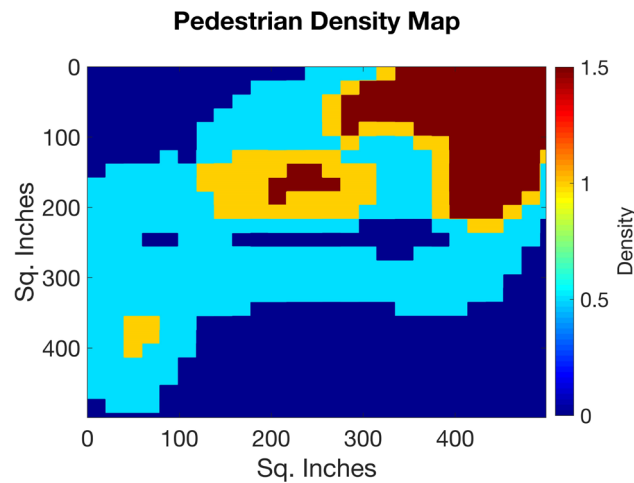
pedestrian density value and whether the goal of reaching the terminal state is met. Highly dense areas have higher a reward value and lower dense areas have a lower reward value. The reward for reaching the terminal state is +100. The social force pedestrian movement provides an input to the magnitude of the reward values of the optimally guided path planning model. A pedestrian density map (Figure 3) was provided to identify the highly traversed and congested areas. To generate the pedestrian density map, the simulation was ran following social force dynamics using AnyLogic, and the results were recorded after the entire simulation was complete. A density value was associated with each square inch. AnyLogic has a built-in function "maximumDensity" that will display and record the maximum observed density values. This value refers to the maximum total density of that location (per square inch) for the entire simulation. Following the completion of the simulation, the results were stored in a csv file via AnyLogic. MATLAB software was used to generate a density map based on the maximum density values recorded (as seen in Figure 3). The default and recommended upper limit of density is set to $1.5p/m^2$ by AnyLogic. Figures 4 and 5 have increased upper limit values of $3p/m^2$ and $5p/m^2$ to illustrate analyzing the density maps of a larger density value. The highly traversed areas represent the areas shown in red on the pedestrian density map. The most densely populated areas are recorded and converted to the optimally guided path planning model's environment. The highly traversed areas have a reward of -10. The highly negative rewards are to discourage navigation through highly populated areas and avoid congestion as much as possible. The orange squares represent the obstacles present. The obstacles are similar to walls and these states cannot be traversed through. The remaining state locations not mentioned previously are all set to a reward of -1. This is to avoid taking any unnecessary steps, further reducing the time it takes to exit. It is important to note that in an airport evacuation scenario, police officers and security guards would be placed along the path to guide pedestrians. This is needed to ensure pedestrians will follow the specified optimal guided path.
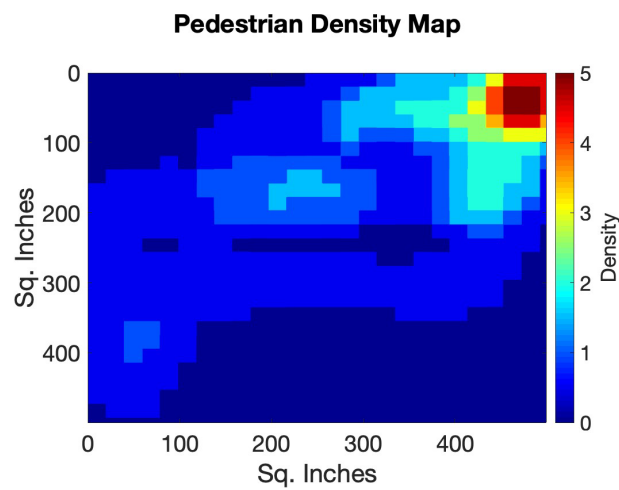
Figure 4

A Pedestrian Density Map generated from our evacuation scenario produced by AnyLogic with an upper limit density range set to 3.0 p/m². The simulation is of 50 pedestrians navigating throughout the environment to the exit location. The red shaded areas pictured show the most populated areas of the environment. The blue shaded areas show the areas that are not highly dense. The pedestrian density map enables us to determine the most densely populated areas and incorporate into our optimal path guidance model for path planning.

Pedestrian Density Map
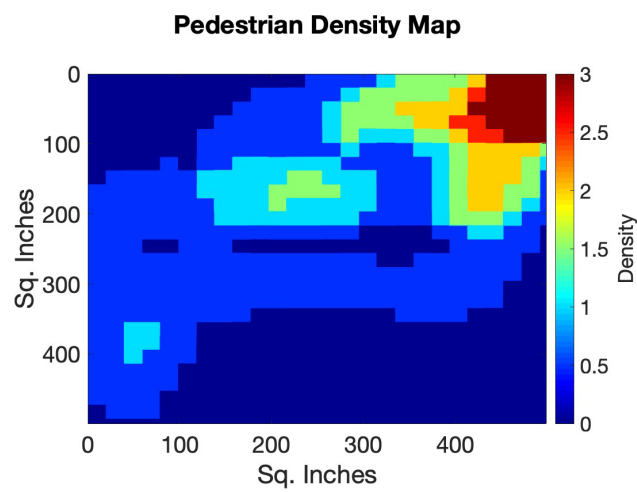
Figure 5

A Pedestrian Density Map generated from our evacuation scenario produced by AnyLogic with an upper limit density range set to 5.0 p/m². The simulation is of 50 pedestrians navigating throughout the environment to the exit location. The red shaded areas pictured show the most populated areas of the environment. The blue shaded areas show the areas that are not highly dense. The pedestrian density map enables us to determine the most densely populated areas and incorporate into our optimal path guidance model for path planning.
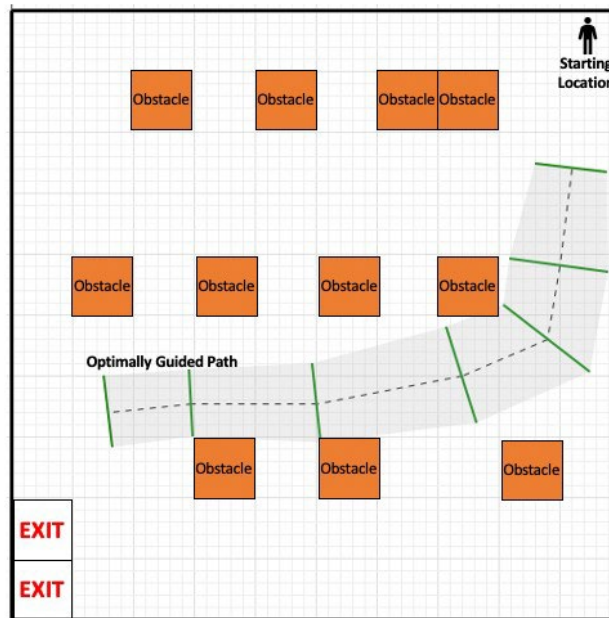
Figure 6

The Emergency Evacuation Environment, eleven objects are placed throughout the environment. The obstacle states (orange squares) are not able to be accessed by the pedestrians. All of the pedestrian's origin is the top right of the environment, labeled Starting Location. The destination (exit) is the bottom left of the environment. The optimally guided path is the shaded gray area. The pedestrians that are not following the social force model will use this optimally guided path to travel throughout the environment.

Agent Based Evacuation Simulation Anylogic

A baseline example of 50 pedestrians, exiting the environment, is created and simulated following social force model dynamics (Figure 7). The parameters for social force models have been validated by comparison with empirical data in a number of previous publications (Helbing et al. 1995). We utilize these parameters in our simulation models. The path planning model changes the destination but the social force parameters are not altered. Multiple scenarios (5) are investigated regarding average evacuation time by the number of pedestrians following each model. The initial average evacuation timing of 50 pedestrians following the social force model only was 84.95 seconds. The

goal is to incorporate both modeling techniques together to reduce the overall time it takes to evacuate the environment. The benefit of the MDP decision making is to provide an additional optimally guided route that avoids obstacles, and highly congested areas, while reducing the overall time it will take for all pedestrians to exit the room. The pedestrian walking speed is uniformly distributed between 0.5 and 1.0 m/s. The additional five scenarios involve dividing the number of pedestrians to follow different modeling techniques. Each of the five scenarios will increase the number of pedestrians following the optimally guided path planning model by five until there is an even split of 25 pedestrians following each modeling technique's path. The first scenario consists of five pedestrians following the optimally guided planned path and the remaining 45 pedestrians will follow the social force model dynamics. The pedestrians are divided into the following groups for the five scenarios: (45/5, 40/10, 35/15, 30/20, 25/25).

Results

The baseline example has an average evacuation time was 84.95 second. In the remaining scenarios we plan to improve performance by reducing the evacuation times. The evacuation times of the remaining examples are shown in Table 1. In Scenario 1, the incorporation of the five optimally guided path pedestrians is shown to reduce the overall evacuation time to 80.64 seconds, this is a 5.1% decrease of total evacuation time. In Scenario 2, five more pedestrians (ten total) are following the optimally guided path planning model. This further reduces the evacuation time to an average of 79.71 seconds. Scenario 3, 30% of the pedestrians are now using the optimally guided path. This increases the percentage decrease of the average evacuation time to 9.2%, and overall average evacuation time to 77.16 seconds. In the final scenario, 50% (maximum capacity) of the pedestrians are chosen to follow the optimally guided path. We do not overload this path by creating any additional bottlenecks or creating anymore delay. The final scenario was able to reduce the average evacuation time to 75.9 seconds. By being able to reduce the average evacuation time to 75.9 seconds, the percentage decrease of evacuation time was 10.6%. The results indicate the addition of the optimally guided path planning model was successful in decreasing the average evacuation time to evacuate.

Figure 7

Pedestrian Evacuation Simulation developed in Anylogic software. The yellow agents are the pedestrians following the social force model and the blue agents are the pedestrians following the optimally guided path planning model. Notice the social force model pedestrians are able to freely travel throughout the environment. The remaining pedestrians are guided using the optimally guided path to navigate throughout the environment.
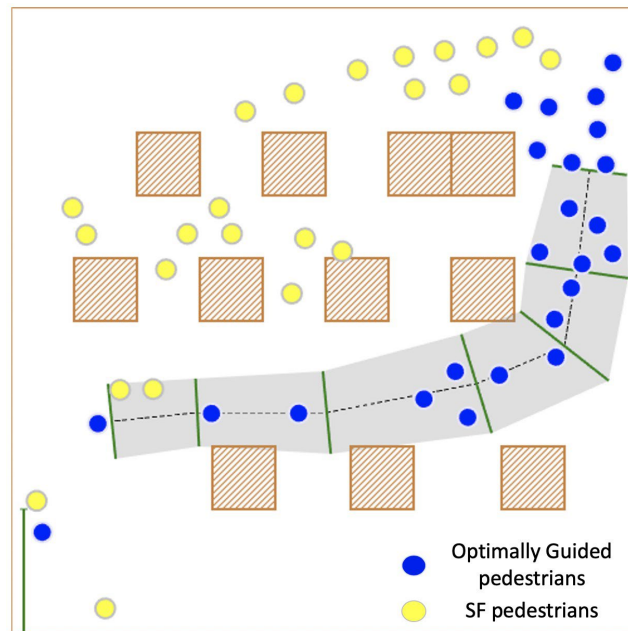
Table 1

Reduced Evacuation Time Results

Scenario	SF Path Pedestrians	Alt. Path Pedestrians	Average Evacuation Time	Percentage Decrease of Evacuation Time
Baseline Example (SF only)	50	0	84.95 sec.	—
Scenario 1	45	5	80.64 sec.	5.1%
Scenario 2	40	10	79.71 sec.	6.2%
Scenario 3	35	15	77.16 sec.	9.2%
Scenario 4	30	20	76.47 sec.	10.0%
Scenario 5	25	25	75.90 sec.	10.6%

Discussion and Conclusion

This project successfully incorporates a social force model into an optimally guided path planning model, for the evacuation of a group of pedestrians. Pedestrian navigation safety is critical to saving lives and preventing injury in a multitude of emergencies such as active shooter incidents and fires. By integrating the two models, we can further optimize mobility in emergency evacuation situations. Our optimally guided path planning model incorporates the social force model's density map and assigns highly congested areas to negative reward states. The negative reward states are used to avoid congested areas and pathways and advise pedestrians to follow the optimally guided path. The baseline example evacuation time was 84.95 seconds. Utilizing the two integrated models, we were able to show a reduction in the average evacuation time of at least 4.31 seconds or a 5.1% evacuation time decrease. The total evacuation time was able to be reduced to a total of 75.9 seconds and a 10.6% decrease in average evacuation time compared to the baseline example.

In the future, we will expand the current MDP framework to a Time-Dependent MDP (TDMP). The time-dependent MDP will incorporate dynamic reward changing based on the social force models' movement in current time. Using a TDMP may reduce the overall evacuation time by providing optimal routing choices in real-time with rewards being updated at each time step. Expanding our current modeling framework will allow for incorporating a more robust and expansive reward function. The reward function will be able to be updated based on the current states pedestrian's density map, and the agent's distance from the goal. We plan to apply this integrated modeling

technique to a simulated airport emergency evacuation. The belief is that applying this modeling technique will reduce the total overall evacuation time of pedestrians in an emergency airport situation. The findings of this project outcome will lead to a multidisciplinary computational framework for understanding and modeling the human decision-making process and resulting actions in emergency evacuations.

References

Alexander D.E.: 'Emergency, Definition of.', in K. B. Penuel, M. Statler & R. Hagen (Eds.), Encyclopedia of crisis management, Thousand Oaks, USA, 2013, pp. 324–325

'Annual disaster statistical review 2015: The numbers and trends', https://www.cred.be/annual-disaster-statistical-review-2015, accessed May 1 2019

Neria, Y., Nandi, A. Galea, S.: 'Post-traumatic stress disorder following disasters: A systematic review'. Psychological Medicine, 2008, 38, pp. 467—480

Heliovaara, S., Kuusine, J., Rinne, T., Korhonen, T., Ehtamo, H.: 'Pedestrian behavior and exit selection in evacuation of a corridor - An experimental study', Safety Science, 2012, 50, pp. 221–227

Yoon, S. W., Velasquez, J. D., Partridge, B. K., Nof., S. Y.: 'Transportation security decision support system for emergency response: A training prototype', Decision Support Systems, 2008, 46, pp. 139–148

Proulx, G.: 'A stress model for people facing a fire', Journal of Environmental Psychology, 1993, 13, pp. 137–147

Proulx, G., Sime, J.D.: 'To prevent panic in an underground emergency, why not tell people the truth?', Fire Safety Science, 1991, 3, pp. 843–852

Hasan, S., Ukkusuri, S. V.: 'A threshold model of social contagion process for evacuation decision making', Transportation Research Part B, 2011, 45, pp. 15091065

Knuth, D., Kehl, D., Hulse, L., Schmidt, S.: 'Perievent distress during fires - The impact of perceived emergency knowledge', Journal of Environmental Psychology, 2013, 34, pp. 10–11.

Vorst, H. C. M.: 'Evacuation models and disaster psychology', Procedia Engineering, 2010, 3, pp. 15–21.

Cocking, C., Drury, J., Reicher, S.: 'The psychology of crowd behaviour in emergency evacuations: Results from two interview studies and implications for the fire and rescue services', The Irish Journal of Psychology, 2009, 30, pp. 59—73.

Abolghasemzadeh, P.: 'A comprehensive method for environmentally sensitive and behavioral microscopic egress analysis in case of fire in buildings', Safety Science, 2013, 59, pp. 1–9

Cassidy, T.: 'Problem-solving style, achievement motivation, psychological distress and response to a simulated emergency', Counselling Psychology Quarterly, 2002, 15, pp. 325–332

Mu, H. L. , Wang, J. H., Mao, Z. L., Sun, J. H., Lo, S. M., Wang, Q.

S.: 'Pre-evacuation human reactions in fires: An attribution analysis considering psychological process', Procedia Engineering, 2013, 52, pp. 290—296

Purser, D. A., Bensilum, M.: 'Quantification of behaviour for engineering design standards and escape time calculations', Safety Science, 2001, 38, pp. 157–182

Kobes, M., Helsloot, I., de Vries, B., Post, J. G., Oberije, N., Groenewegen, K.: 'Way finding during fire evacuation; an analysis of unannounced fire drills in a hotel at night', Building and Environment, 2009, 45 pp. 537-–548

Miyoshi, T., Nakayasu, H., Ueno, Y., Patterson P.: 'An emergency aircraft evacuation simulation considering passenger emotions', Computers & Industrial Engineering, 2012, 62, pp. 746–754

Shi, J., Ren, A., and Chen, C.: 'Agent-based evacuation model of large public buildings under fire conditions', Automation In Construction, 2009, 18, pp. 338–347

Helbing, D. and Molnar, P.: 'Social force model for pedestrian dynamics', Physical
Review E, 1995, 51, pp. 4282–4286.

Namilae, S., Srinivasan, A., Mubayi, A., Scotch, M. Pahle, R.: 'Self-propelled pedestrian
dynamics model: Application to passenger movement and infection propagation in
airplanes', Physica A: Statistical Mechanics and its Applications, 2017, 465, pp.
248–260

Kirby, A. M., Dietz, J. E., Matson, E. T., Pekny, J. F. Wojtalewicz, C.: 'Major city
evacuation planning using simulation modeling', International Journal of Disaster

Resilience in the Built Environment, 2015, 6, pp. 397-–408

Chen, J., Liu, D., Cheng, Y., Namilae, S., Thropp, J., Seong, Y.: 'Effects of exits and
number of passengers on airport evacuation efficiency using simulation', Proc. of
IISE (Institute of Industrial and Systems Engineers) Annual Research Conference,
Orlando, Florida, USA, 2018.

Namilae, S., Derjany, P., Mubayi, A., Scotch M., Srinivasan A.: 'Multiscale model for
pedestrian and infection dynamics during air travel', Physical review E, 2017, 95(5):
052320.

Chunduri, S., Ghaffari, M., Lahijani, M.S., Srinivasan A., Namilae, S.: 'Parallel Low
Discrepancy Parameter Sweep for Public Health Policy.', in Cluster, Cloud and Grid
Computing (CCGrid)', Proc. in 18th IEEE/ACM International Symposium, 2018.

Derjany, P., Namilae, S., Mubayi, A. and Srinivasan, A.: 'Computational model for
pedestrian movement and infectious diseases spread during air travel', AIAA
Scitech., 2018.

Namilae, S., Mubayi, A., Srinivasan, A., Derjany, P.: 'Model Based Policy Analysis for
Infection Spread during Air Transportation', Journal of Transport & Health, 2018, 9:
S12.

Namilae, S., Derjany, P., Liu, D., Mubayi, A., Srinivasan, A.: 'Multiscale pedestrian
dynamics and infection spread model for policy analysis', 9th Pedestrian and
Evacuation Dynamics, 2018.

Zhou, J., Guo, Y., Dong, S., Zhang, M., Mao, T.: 'Simulation of pedestrian evacuation route choice using social force model in large-scale public space: Comparison of five evacuation strategies.', PLoSone, 14(9), e022187, 2019.

Achour, N., Braikia, K.: 'An MDP-based approach oriented optimal policy for path planning', in 2010 International Conference on Machine and Web Intelligence, 2010, pp. 205–210.

Sidrane, C. Kochenderfer, M. J.: 'Closed-Loop Planning for Disaster Evacuation with Stochastic Arrivals', Proc. in 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2018, pp. 2544–2549.

Ferguson, D., Stentz, A.: 'Focussed processing of MDPs for path planning', Proc. in 16th IEEE International Conference on Tools with Artificial Intelligence, 2004, pp. 310–317.

Burlet, J., Aycard, O., Fraichard, T.: 'Robust motion planning using Markov decision processes and quadtree decomposition', IEEE International Conference on Robotics and Automation, 2004, 3 , pp. 2820–2825.

Nardi, L., Stachniss, C.: 'Uncertainty-Aware Path Planning for Navigation on Road Networks Using Augmented MDPs', IEEE International Conference on Robotics& Automation (ICRA), 2019.

lterovitz, R., Siméon, T., Goldberg, K.: 'The Stochastic Motion Roadmap: A Sampling Framework for Planning with Markov Motion Uncertainty

Robotics', Science and Systems III, Georgia Institute of Technology, Atlanta, Georgia, USA, 2007.

Science and Systems III, Georgia Institute of Technology, Atlanta, Georgia, USA, 2007.

Allamraju, R., Kingravi, H., Axelrod, A., Chowdhary, G., Grande, R., Crick, C., Sheng, W.: 'Human Aware UAS Path Planning in Urban Environments using Nonstationary

MDPs', Proc. in International Conference on Robotics and Automation, doi:10.1109/ICRA.2014.6907000, 2014.

Ruiz-Loza, S., Hernandez, B.: 'Real Time Markov Decision Processes for Crowd Simulation', Proc. in GPU Zen Advanced Rendering Techniques, Edition: 1 Chapter 17, 2017.

Keneally, S. K., Robbins, M. J., Lunday, B. J.: 'A markov decision process model for the optimal dispatch of military medical evacuation assets', Health Care Management Science, 2016, 19(2), pp. 111–129.

Ulmer, M., Goodson, J., Mattfeld, D. & Thomas, B.: 'Dynamic Vehicle Routing: Literature Review and Modeling Framework', 2017.

Dong, S., Zhou, J., Zhang, S.: 'Determining E-bike drivers' decision-making mechanisms during signal change interval using the hidden Markov driving model', Journal of advanced transportation, 2019.

Jenkins, P., Robbins, M. and Lunday, B. (2018). 'Examining military medical evacuation dispatching policies utilizing a Markov decision process model of a controlled queueing system', 2018.

Childers, A., Taaffe, K. (2008).: 'Evacuating Patients Based on Priority Modeling', Proc. of The IIE Research Conference, Vancouver, BC, Canada, 2008.

Zebala, J., Ciepka, P., RezA, A.: "Pedestrian acceleration and speeds", Problems of Forensic Sciences, 2012, 91, pp. 227–234.

Fang, Z., Wang, P., Chen, D. H., Chen, D. H., Duan, J. X., Hu, Z. R.: 'The development of evaluation software of safety evacuation for high buildings', Fire Science and Technology, 2004, 23(5), pp. 439–442.

Martinez-Gil, F., Lozano, M., Fernández, F.: 'Multi-agent reinforcement learning for simulating pedestrian navigation', ALA 2012, 2012, 113, pp. 54-–69. Sutton, R. S., and Barto, A.G.: 'Reinforcement learning: an introduction', The MIT Press, Cambridge, MA, 2018.

Helbing, D., Farkas, I., Vicsek, T.: 'Simulating dynamical features of escape panic.', Nature, 407(6803), 2000, pp. 487–490.

Lakoba, T. I., Kaup, D. J., Finkelstein, N. M.: 'Modifications of the HelbingMolnar-Farkas-Vicsek social force model for pedestrian evolution.', Simulation, 81(5), 2005, pp. 339–352.

Part 2. Multi-robot Collaboration for Airport Emergency Evacuation

Introduction

Background

In the past decades, the air transportation industry has been expanding rapidly along with the flourishing trajectory of the global economy (Airport Emergency Plan, 2011). The evolution of modern aviation has increased the demand for airport security services and the necessity for more efficient evacuation strategies in the emergencies, such as natural disaster like fire or terrorist attack.

An emergency usually occurs instantaneously, requiring immediate decision-making ability (Wang et al., 2016). This is especially true in the case of environments like airports. According to Title 14 Code of Federal Regulations (CFR), Part 139, airport certificate holders must develop and maintain an Airport Emergency Plan (AEP) to minimize the damage, whether to human or property in an emergency, as a part of their airport certification manual (Airport Emergency Plan, 2011). More detailed suggestions are given by the Federal Aviation Administration (FAA) in the Advisory Circulars (AC) 150/5200-31C (FAA, 2009). Therefore, it is imperative to continue pushing the technological capabilities and improve current emergency evacuation strategies.

The Current State of Evacuation Plans

The current emergency drill models use the existing case studies and experiments to prevent future accidents, which can be inefficient and resource-consuming (Arumugam et al., 2016). However, evacuation plans simulated through 'mock drills' sometimes cannot truly simulate the real situations thus are not able to react to sudden hazards in time effectively. For instance, airport fire evacuation drills cannot simulate the situation

of falling objects blocking the exits on the evacuation routes. In reality, when similar unexpected scenarios occur, the crowd is likely to panic, people at the scene are prone to cause crowd trampling, stampede, which might result in loss of life (Balachandar et al., 2019). Thus, a method that can intelligently generate evacuation routes according to the discrepancies in each specific emergency circumstance can be lifesaving.

Conventional approaches to generating an evacuation path plan concentrate on computer modeling techniques (Bi et al., 2019; Bunea et al., 2016). Computer simulation technology protects volunteers' safety, which may occur dangerous in 'mock drills' in reality and provides valuable evacuation information (Carino & Garciano, 2019). The modeling methods, such as the social force model (SFM) (FAA, 2009) and the hybrid model (Feng & Wang, 2019), are based on creating fundamental rules to control each agent's movement. It is prone to ignore the interaction with other agents and the environment.

Like many public transportation environments, airports are rather complex environments with numerous social interactions (Gelada & Buckman, 2019). Therefore, for traditional non-learning-based algorithms, simulating the entire airport environment and generating evacuation routes for each agent is relatively time-consuming and often far from optimal.

Agent-Based Learning

The hazardous environments with unidentified threats expose first responders and evacuees to significant risks. Robot agents or drones exploring the dangerous environment before the human evacuation can decrease the risk for the people involved. With the assistance of robot agents or drones, the evacuees can find the optimal evacuation routes without wandering around in distress situations and exposed to unknown dangers. While the agents are exploring the environment, the updated routes can be shown to human evacuees as a digital decision aid on devices such as mobile apps. The real-time revised map becomes more straightforward as the agent learns more about the changing circumstances. First responders and evacuees can add holistic views and their relative locations to facilitate the evacuation process. Additionally, robot agents can

be integrated into the AEP or other emergency procedures to improve efficiency in dealing with emergencies planning and policy making.

Emergencies are unpredictable, and people need to be evacuated from dangerous places to secure shelters as soon as possible. Researchers from different disciplines have discovered different paths to enhance the efficiency of emergency evacuations, and they have provided valuable suggestions (Arumugam et al., 2016; Bunea et al., 2016; Liu et al., 2016; Makinoshima et al., 2016; Shen et al., 2015; Wang et al., 2016). More and more researchers in this field, as of late, have shown interest in agent-based modeling for emergency evacuation, especially for complex scenarios that can transpire in unsecure environments.

Within these simulation models, reinforcement Learning (RL) has become more popular in evacuation primarily among all the machine-learning models because of its capability of "learning through exploration." The agent learns through trial-and-error while interacting with the environment that helps it make the right decision (Bunea et al., 2016). RL does not require specific control over the environment or models about the environment but instead uses positive and negative rewards to determine the best future decision. This feature is critical for emergencies, as the threats and environments are usually unidentified to decision-makers, without explicit solutions for such evacuation tasks. RL also allows agents to learn from themselves and others through the rewards and punishment mechanism, making RL an ideal method for exploring and learning an unknown environment with unknown threats.

In improving evacuation efficiency, learning-based RL variation algorithms have drawn interest from many researchers. In a large-scale and convoluted environment, multi-agent reinforcement learning (MARL) can be applied (Isele et al., 2018) to allow agents to interact with the dynamic environment and perceive it through reward and punishment signals to maximize the reward by trial and error. MARL is considered the closest algorithm related to the knowledge accumulation path as humans in reality (de Witt et al., 2018). Some researchers have already adopted those characteristics to the path planning problem (Xu et al., 2020; Fatima et al., 2017). However, the adaption of MARL to the airport evacuation simulation is not straightforward. That is part of the reason that this study was intended to present a systematic study including environment simulation,

model design, and analysis of evacuation behavior in macro and micro ways. Macroscopically speaking, the agents can learn the optimal evacuation path universally. The agents can avoid dynamic obstacles and correct the route according to the challenges that block the original course in the micro aspect. Agent rewards, success rate, and evacuation time were used to evaluate the learning efficiency.

Furthermore, this study also compared the performance of single-agent exploration and multi-agent collaboration during emergency evacuation using RL algorithms. Two different environments were utilized in the study: the simple environment with one threat and the complex environment with three threats. The benefits of MARL was further verified using a more dynamic and realistic airport diagram.

It is believed that agent-based modeling effectively simulates emergency evacuation as the agent represents the actual human/robot that makes stochastic decisions and movements. The overall rational for using multiple agents is that a single robot agent is usually used to explore different environments, but it can take a long time to learn and complete the exploration. Multiple agents exploring can lessen the time consumed, especially when agents communicate and collaborate.

Summary

In this study we intended to investigate the difference in the performance between single-agent exploration and multi-agent exploration regarding time, death counts, and rewards. The interaction between the complexity of the environment and the numbers agents were also investigated.

To summarize, the main contributions of this study are as follows:

1. Complex environments were used to simulate the airport like situations. Not only static obstacles were used to train the model, but this project also simulated the situation that the obstacles could move randomly or move in a specified trajectory.

2. Agents' communications were investigated in the study. This project takes a new sharing Q-table mechanism to enhance the learning convergence rate among

agents. Multi-agents share their Q-table as a 'communication' mechanism. At each step, agents update their own Q-table and record their update value on the summary table. Then agents take actions that will get the most reward based on the summary table. In this way, the success rate, evacuation time, and convergence rate will be improved effectively. As a result, this method is more suitable for solving the evacuation problems.

Review of the Relevant Literature

Evacuation Safety

Emergency evacuation is a complex and dangerous situation where critical decisions must be made under extreme time pressures (Wang et al., 2016), and lives could be saved if correct decisions were made promptly. During an emergency, the public may not be able to make the optimal decisions with limited time and information due to the complex nature of the emergent situation. Furthermore, an emergency can often cause chaos and congestion, especially in public transportation where the pedestrian density is high (Feng & Wang, 2019). The efficiency and safety of the emergency evacuation would be greatly compromised if no proper training or assistance was provided before or during the emergency evacuation (Purser, 2015).

For example, in an investigation of two care-home fires, evacuees were unable to identify the fire location and nearby doors due to training inadequacies. Several lives were lost in minutes because of the poor evacuation response and strategy (Purser, 2015). It may seem intuitive to find the optimal evacuation route based on the memory and prior experience, but the situation is quite different under emergency evacuation. People may panic, and under the high time pressure, they could lose the ability to think rationally and make reasonable decisions. Researchers have found several ways to improve this situation by advance assessment and evacuation assistance (Cariño & Garciano, 2019; Zhang et al., 2016; Yamamoto et al., 2018; Zhang & Yi, 2015). A little help during the evacuation may seem negligible, but the help may save the time for thinking and offer a way to survive.

Many models were developed to assess the risk under emergency. For example, Zhang et al. (2016) proposed a steel-temperature rise model to assess the risk of evacuation safety in the collapse of a large steel-structured building. This model can be used to make emergency plans for steel-structured buildings when encountering disasters like earthquakes. It can also be used to improve people's awareness of safe evacuation routes during a crisis. Cariño and Garciano (2019) developed a seismic evacuation safety index to help assess evacuation safety for schools. The evacuation safety index can help

schools evaluate the safety of the classroom and campus building to ensure the solidity of the structure during the evacuation.

There are also various models developed to investigate different evacuation situations, such as evacuation from buildings and classrooms, tsunami evacuation, toxic environment evacuation, and post-disaster evacuation (Liu et al., 2016; Wang et al., 2016). Modeling specific types of emergencies can help people better understand the nature of the events, what to expect, and how to react. Detailed emergency evacuation plans can thus be designed to accommodate different facilities layouts and for different purposes.

Liu et al. (2016) simulated a classroom evacuation situation to investigate how evacuation efficiency is affected by classroom layouts. The classroom with more exits had a significantly higher efficiency in the evacuation, and students who followed the guidance can evacuate faster and safer. During the evacuation, teachers may lose control of the situation which makes training for teachers and students essential. It is suggested for school officials to conduct more fire drills regularly. Students can learn to evacuate safely and efficiently through frequent drills and avoid dangerous and chaotic evacuations in real emergency situations, as in the fatal care home fires (Purser, 2015). Liu et al. (2016) also suggested that future studies should look into specific type of emergencies, such as a bombing or earthquake, as students may show different responses for different situations. For the same reason, different emergency plans are needed for different types of situations.

On a larger scale, Wang et al. (2016) simulated a multimodal near-field tsunami evacuation. The near-field tsunamis represent the middle range of five hazard groups (Earthquake, building fire, tsunami, wildfire, and hurricane). For the middle range hazard group, there are usually 20 to 40 minutes warnings window before the actual disaster hits, so the quicker and better the decision is made, the more lives can be saved. Evacuees had options of whether to hide in the shelter or evacuate by car in the near-field tsunami evacuation model. Each option had its pros and cons regarding the safety and efficiency of this evacuation, and evacuees cannot simply tell which mode of transportation is better. It is a complex evacuation simulation involving more than just making a simple choice (Wang et al., 2016). Wang et al (2016) investigated the effect of different

individual decision-making time scales to assess the mortality rate due to immediate evacuation right after an initial earthquake or after a specified milling time. Simulation of the evacuation showed some unexpected results which gave people more insights into the emergency evacuation.

The results from the study above also gave people some insight about evacuation safety during near-field tsunamis. First of all, there is a strong correlation between the individual decision time and the mortality rate of the disaster, and the faster people make the right response, the more likely that they will survive which makes sense as they have more time to evacuate. Secondly, different structures have different resistance to the disaster, and vertical evacuation structures were more helpful than other structures, so the structure of the escape route and the shelter must be evaluated from time to time. Furthermore, depending on the actual emergency, the model for moving pedestrians should be changed accordingly. As in the emergency evacuation of the tsunami, the more people used the car, the fewer people could survive as the tsunami moves fast while people are congested on the road (Wang et al., 2016).

In addition to these results, it is advised to include partial damage to the transportation network as it is commonly seen during catastrophic events. The vulnerability of the facility, congestion on the road, and accessibility and user-friendliness of public transportation all need to be considered to better identify the bottleneck effects in the evacuation simulation models for future studies. Makinoshima et al. (2016) also presented an evacuation simulation model based on the evacuation behaviors observed during the Great East Japan Earthquake and Tsunami. Makinoshima et al. (2016) compiled the data from previous studies, surveys, and reports and developed an evacuation simulation model. In the evacuation model developed from the historical data, main roads were used for evacuation, and shelter preferences and pedestrian-car interactions were also considered. By doing so, this model was able to better estimate the actual evacuation behaviors during the catastrophic event.

For the post-disaster simulation, Bunea et al. (2016) modeled how individuals react after the disaster. Their study analyzed the scenario in the city of Iaşi, Romania. Iaşi has a high seismic vulnerability as it has a lot of old buildings. Bunea et al. (2016)

simulated an evacuation situation for the earthquake as people need to cross three bridges before entering the hospitals or shelters.

In their study, the time taken for evacuees to move across the three bridges was recorded. The vulnerability and damage of the earthquake to the bridges were taken into consideration, and different degrees of damage were included in the simulation to replicate the actual situation (Bunea et al., 2016). Their study built a good basis for a further experiment on evacuation safety in areas with many old buildings. The time for people to cross the bridge and enter the shelter can be further related to the mortality rate to show the feasibility of the structure of the city as the research done by Wang et al. (2016).

Besides the assessment of the situation before emergency, the real-time support and assistance during the disaster is also critical for the safety and efficiency of the evacuees during the evacuation. For assistance during the evacuation, Yamamoto et al. (2018) used numerical simulation to understand the effects of the new ventilation equipment on the tunnel fire. For the tunnel fire, the use of the new ventilation equipment can improve the environment greatly and make it much easier for evacuees to escape.

The Application of ML and RL in Evacuation

Machine learning is different from human learning in terms of the way systems learn. One advantage of using machine learning in evacuation is that it can protect humans from exploring dangerous situations. If the machine can learn to find the exit during the evacuation, the danger can be transferred to the machine such as drones or robot agents instead of human pedestrians.

Machine learning is based on the concept that systems learn from the dataset, identify the pattern, and make decisions. There are mainly three types of machine learning including supervised learning, unsupervised learning, and reinforcement learning.

Supervised learning is to learn how to map from the attributes describing an instance to the targeted attribute, i.e., learning from a labeled dataset with guidance (Kelleher & Tierney, 2018). Unsupervised learning is to learn without a targeted attribute, i.e., learning from an unlabeled dataset without guidance. Without the targeted

attribute, unsupervised learning becomes more difficult as the task becomes more general, such as looking for the regularities in the dataset (Kelleher & Tierney, 2018). Reinforcement learning (RL) is a branch of machine learning area based on the concept that systems learn from the dataset, identify the pattern, and make decisions. The learning-based algorithm could make the computer think (Ferber et al., 2004), primarily for the reinforcement learning algorithm.

Reinforcement Learning

Reinforcement learning encourages the agents to obtain the most rewards in the circumstances through the interaction between the agent and the environment. The algorithm is informed when the agents' behavior is incorrect, whereas the proper behavior response is not disclosed. Unlike supervised learning, agents are not granted explicit inputs and outputs. Because of these features, reinforcement learning has been widely applied in numerous fields of study (Lovas, 1998), with strong online learning and self-learning capabilities in complex environments. However, reinforcement learning has relatively poor convergence ability compared to other machine learning algorithms (Weiss & Ed, 1999). Designing the reward function and optimization function so that reinforcement learning can converge quickly in a complex environment is the key to the proper application of reinforcement learning in actual scenarios.

The fundamental reinforcement learning algorithm is based on the Markov decision processes (MDPs), which are a classical formalization of sequential decision making, where the existing action will not only influence the immediate rewards, but also long-term returns (Zhou et al., 2017). MDPs regularize the reinforcement learning by the sequence of,

$$M = (S, A, P, R, \gamma)$$

The sequence comprises a set of states, S, actions A, transition probability P, reward value from the current environment R, and the discount factor γ. Specifically, according to the distribution of time steps, MDPs have two categories: the continuous domain and the discrete domain Markov decision-making process. The agents plan to

discover the best optimization decision and get the best cumulative reward R in the current environment, which can be addressed as an MDPs problem.

Reinforcement learning based on Markov decisions can be divided into two categories, namely model-based dynamic programming learning and model-free learning. Gelada and Buckman (2019) demonstrated that model-based and model-free learning are equivalent in sampling efficiency. In contrast, the difference between the two algorithms lies in the balance between understanding the dynamic characteristics of the agent or forcing the best policy in the actual application scenario. Model-based reinforcement learning is mainly divided into value-based learning such as Q series learning (Le et al., 2017; Liu et al., 2016) and State-Action-Reward-State-Action (Sarsa), and policy-based learning such as Actor-Critic. In short, RL is quite different from the other machine learning methods because it learns by interacting with the environment without any prior datasets. An RL agent makes the most efficient decision at a given state by trial-and-error, and the only input is delayed scalar reward (Tan, 1993). For this reason, RL is extremely beneficial for evacuation simulation with unknown threats as there is no dataset provided during evacuation process.

RL has been applied in a wide range of evacuation situations. For example, researchers have used RL to adapt the complex agent-based model to a fast-linear model to solve the optimization of guidance sign for tsunami evacuation (Le et al., 2017), model situations such as sensor sensitivity of autonomous aerial vehicles (Quirion et al., 2014; Quirion et al., 2015), and get the optimal route recommendations during the emergency evacuations (Bi et al., 2019).

Yao et al. (2019) proposed a data-driven cloud evacuation framework based on the RL. This cloud evacuation framework can simulate human behaviors even when the environment is changing. The model established by Yao et al. (2019) extracted position and velocity information from videos of the real-life evacuations and then used a cloud simulation system to generate the results. The evacuation model can improve efficiency and safety greatly in advance which can be applied in real-life situations.

To give some other examples, Piyabhum et al. (2020) and Tian et al. (2018) proposed a suitable state space and reward function which results in an efficient collaborative double Q-learning RL algorithm. Arai et al. (2000) and Busoniu et al.

(2005) introduce the cut-loop routine in reinforcement learning that discards looping behavior and demonstrate its effectiveness empirically within a simplified NEO (non-combatant evacuation operation) domain. Zhou et al. (2017) and Busoniu et al. (2008) proposed a novel asynchronous reinforcement learning algorithm that can solve problems such as exponential computation complexity in a large environment. Papoudakis et al. (2020) and Cheng et al. (2014) investigated several reinforcement learning algorithms and provided detailed experimental data, analysis, and insights into each algorithm.

Zhang and Yi (2015) simulated using robots to guide people evacuating from dangerous areas. In their study, the people guided by robots had a significantly shorter evacuation time than non-robot-assisted evacuation. As in the care home fires incident, people were unable to locate the fire and closed room doors (Purser, 2015). With the help of guiding robots, people can find the exit rapidly even without prior training.

Wang et al. (2016) and Tan (1993) propose a shared-Q RL algorithm in crowd simulation, which could enhance the efficiency of crowd evacuation. Dong et al. (2020) applied a Rainbow Q- Network (DQN) to solve the multi-exit evacuation, improving data utilization and algorithm stability. Makinoshima et al. (2016) designed the RL scheme to solve the immersed tube tunnel problem of a fire evacuation.

In summary, RL is ideal for evacuation simulation because it can learn from interacting with the environment without prior knowledge, as the same criteria are needed in the emergency evacuation. During the emergency evacuation, there is no prior data that the agent can use. Agents must learn by exploring the environment themselves and find the optimal evacuation route.

Multi-agent Reinforcement Learning

A multi-agent system (Papoudakis et al., 2010) can be referred to as "societies of agents," which is a set of agents that interact jointly to coordinate their behavior and often cooperate to achieve some collective goals (Gosavi, 2004). In most cases, the researchers usually used a single agent to complete exploration for an extremely long time. Instead of using Independent Reinforcement Learning (InRL), where each agent treats its experience as part of its environment, some researchers adopted a Multi-agent RL system

(MARL) to define a range of collective situations (Lanctot et al., 2017; Shalev-Shwartz et al., 2016). Multi-agent learning is not simply adding more agents in the same environment; moreover, the interaction between each agent will significantly increase the complexity of the entire system. The behaviors between agents influence each other and accomplish the same goal through an internal cooperation mechanism. Researchers adopt this mechanism to combine the advantages of reinforcement learning to investigate a series of multi-agent reinforcement learning algorithms (Gwynne et al., 1999; Clouse, 1995).

Lanctot et al. (2017) compared agents' actions based on the observed behaviors. For the InRL, the policies learned by one agent overlaps other agents' findings, so InRL's learning policy can't be generalized to other agents. On the other hand, MARL can share learned policies with other agents. Balachandar et al. (2019) developed and evaluated different multi-agent protocols to instruct agents to collaborate to play soccer. They found that the model with communication had more promising results than the agents that didn't have it.

In a MARL system, agents could interact while sharing a common learned environment knowledge. They could either be competitive, cooperative, or a mix of the two. Martinez-Gil et al. (2011) used MARL to simulate pedestrian groups and the navigation process of the pedestrians. In their study, the researchers used two RL algorithms with multiple attributes for the simulation, including group size and speed control. However, Martinez-Gil et al. (2011) suggested that two learning algorithms can be integrated into one for future research. The agents can better collaborate with one algorithm.

Raileanu et al. (2018) implemented the MARL algorithms with imperfect information. The reward of the RL in this study depended on both agents. Each agent has its hidden state, and the agent has to make predictions on other players from what they observed during the experiment. Agents need to solve the task by the predictions they made. A unique approach called the self-other-modeling method was used in their study (Raileanu et al., 2018). In this approach, agents update the values of states and actions by incorporating the prediction of others' values of states and actions.

Many tasks, such as autonomous vehicles, multi-player games, and multi-agent navigation require multiple agents to interact and communicate. Agents need to learn to cooperate and achieve their goals more efficiently. While it may seem intuitive for human beings to cooperate and collaborate, the same cooperation process is more challenging for RL agents.

Shalev-Shwartz et al. (2016) implemented MARL into autonomous driving, where the vehicles need to form a long-term strategy that shares the same road with other traffic. The challenge of this study is the long duration of the experiment with other road users interrupting the regular operations. Shalev-Shwartz et al. (2016) used policy gradient iterations to deal with the interruptions. The interruption from other users is stochastic and unpredictable. The researchers used a hierarchical temporal abstraction in the policy gradient iteration (Shalev-Shwartz et al., 2016). To make agents easier to communicate and collaborate, it was found that having a decentralized controller may generate more desirable outcomes.

The centralized controller regards the whole system as one agent, which causes the state and action spaces to rise exponentially as more agents join the system (Balachandar et al., 2019), driving the decision-making process to become more challenging, computation becomes more complex. On the other hand, the decentralized controller could let agents communicate and collaborate individually (Balachandar et al., 2019). The decentralized controller treats every agent as an individual, eventually leading to more desirable results. De Witt et al. (2018) stated the possibility of complex decentralized coordination in the MARL when agents share some information.

Summary

In summary, there are numerous benefits for RL's application in analyzing a dangerous environment for evacuation. Multi-agent collaboration has tremendous advantages in utilizing RL across diverse fields of study. While existing studies have stressed the significance of multi-agent collaboration, few studies concentrated on the application of multi-agent collaboration using RL for navigation and exploration during an emergency evacuation, especially with unknown threats in the environment. This

study was intended to fill the gap by investigating the effect of multi-agents and environment complexity on the evacuation efficiencies, both under the static obstacles and dynamic obstacles situations, for an airport environment situation.

Methodology

This section describes the research approach and methodology, including the learning environment, algorithm, and data treatment and analysis. There was a sequence of two studies conducted for this project. Study 1 investigated the performance difference between a single-agent and a multi-agent method for a static obstacle situation. study 2 was based upon the results obtained from study 1 and extended to a more aviation specific application. In study 2, a small airport was used as the environment and the threats were modeled as moving. Study 2 investigated performance differences in a dynamic environment, when obstacles are moving in a more airport specific environment.

Research Approach for Study 1

For Study 1, there were two types of environments explored by robot agents, including single-agent exploration and multi-agent collaboration. Their goal was to investigate the environments and locate the exits as soon as possible. The performance of single-agent exploration and multi-agent collaboration during emergency evacuation using RL algorithms were compared. The complexity of the environment and the exploration method were the two independent variables, and each of them had two levels. The dependent variable for this study was the performance of the agents. The agents' performance would be compared in different situations regarding time, death counts, and rewards.

Design and Procedures

Two different environments were used as the test bed for exploration. A simple environment was a 10*10 meters space with one threat, and a complex environment was a 10*20 meters space with three threats. Agent/agents started from the left lower corner

of the map to try to locate the room's exit. There were threats with a dimension of two by three in the middle of the room placed randomly, and one unit to each direction around the threat was considered the dangerous area, as shown in Figures 1 and 2.

The figures show that the cross and circle were the beginning and the ending points, respectively; the dark blue and light blue areas were the threat and the dangerous regions. Agents would try to learn and evacuate while avoiding the hazardous area and the danger.

Figure 1

Simple Environment
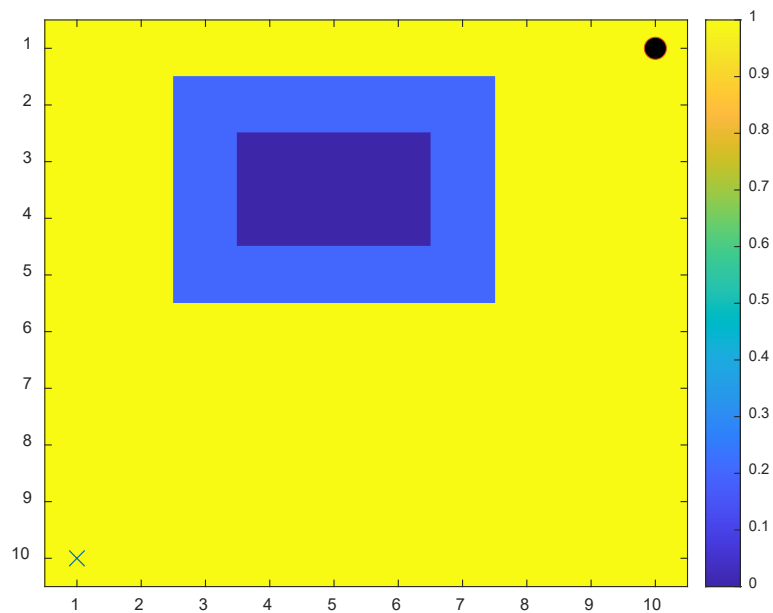
Figure 2

Complex Environment
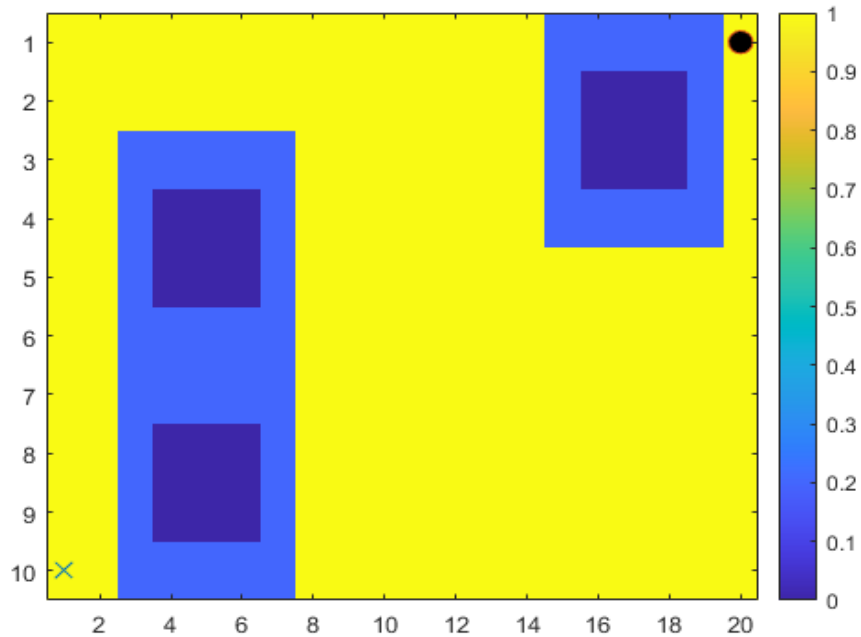
A single Q-table was employed in the single-agent simulation to record the rewards and actions. The number of maximum episodes was 5000. For multi-agent simulation, two agents both started at the origin point. They would communicate and collaborate by sharing the Q-learning table. Each agent would update its own Q table, and the final Q-table would be the average of the two individual Q-learning tables. The research operated each scenario 50 times for statistical analysis to see the differences between single-agent exploration and multi-agent collaboration methods. Each trail would end after 5000 episodes, and then agents would start over to explore the environment.

The agents could move in eight directions freely. Thus, there were eight potential action choices for each position, except at the corners or edges.

Research Approach for Study 2

In a dynamic environment, the obstacle is simulated to move continuously. To implement the Q-learning method for study 2, we used a similar layout as Study 1, and assume there are two agents starting at the entrance of a room, the room has the size of

10*10 meters, the obstacle is moving within a known trajectory which is a 9*9 square trajectory. The agents are trying to find the exit without colliding with the obstacle. Each agent uses the Q-table to record the state status and the action taken. The state status is agent location, measured by the distance between the agent and the target and the distance between the agent and the moving obstacle. The map is shown as the following Figure 3.

Figure 3

Moving threats dynamic environment
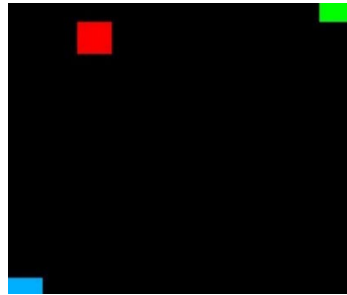
As shown in Figure 3, the blue square is the beginning point, while the green square is the target point. The red square is the obstacle area and once the moving agent collides the red area, the agent will go back to the beginning point. In this study's dynamic environment, the obstacle was moving with a square trajectory, while we set the agent moving to eight different directions as default, which means there are eight possible actions choice for every state.

Furthermore, a real airport environment was used as the environment to demonstrate the benefits of the MARL in dynamic environment. The environment is shown as the following Fig.4.

Figure 4

Three airport environments
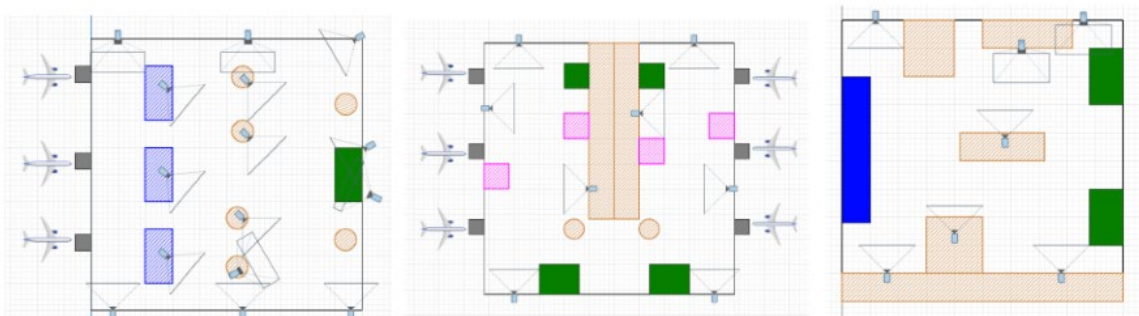
In the airport environment, a moving threat was initialized in a random position and moving at a random direction every time step. For these three maps, two agents were applied to explore and evacuate to the exit on the upper right. The reward scheme was same as indicated in study 1. Two types of agent interactions were compared, one was the two agents explored environment independently without any collaboration and the second way is have the two agents' communication and collaborate simultaneously, as described in first phase of the study 2. The average reward was compared between these two methods for all the three different airport layouts.

For both studies, a Q-learning RL algorithm was used to find the best policy to maximize the reward, and Python simulated and compiled the result, as described in the next section.

Q-Learning Algorithm

In both studies, Q-learning was utilized as an off-policy RL algorithm that seeks to find the best action for the current state. The Q-table includes states and actions, which follows the form of Q (state, action), as shown in Table 1 as a sample format. States and actions would be preserved in the Q-table, and agents would take actions based on the Q-value. Performance measures were collected including time, death counts, and rewards.

Table 1

Q-table

State/Action	A_1	A_2
S_1	Q (s_1, a_1)	Q (s_1, a_2)
S_2	Q (s_2, a_1)	Q (s_2, a_2)
S_3	Q (s_3, a_1)	Q (s_3, a_2)

Agents are required to discover the policy that maximizes the expected cumulative rewards. The bellman function is employed to find the optimal decision sequence. The current state value function V(s) could be acquired by computing the total reward of the current state's expected reward. The bellman function is shown below:

$$V\pi \ (s) = E \ (Ut \mid St = s)$$
$$= E \ [R(t+1) + \gamma \ [R(t+2) + \gamma \ [......]] \mid St = s] \qquad (1)$$
$$= E \ [(t+1) + \gamma \ (S') \mid St = s]$$

The $V^*(s)$ is the maximum cumulative expected value:

$$V^*(s)= max_\pi \, V_\pi \, (s) = max_\pi \, E \, [\textstyle\sum_{t=0}^{H} \gamma^t \, R(s_t, A_t, S_{t+1})|\pi, s_0 = s] \qquad (2)$$

The state-action function is:

$$q_\pi(s, a) = E_\pi[r_{t+1} + \gamma r_{t+2}+\ldots..|A_t a, S_t = s] = E_\pi[G_t|A_t = a, S_t = s] \qquad (3)$$

The Gt is the total discount reward value for time t and γ is the discount factor. When the discount factor is close to 1, the latter state is more critical; vice versa, when the element is close to 0, the agent only considers the current interest's influence. The best action-value function (4) was utilized to open the expectation (5):

$$Q^*(s, a) = max_\pi \, Q^* \, (s, a) \qquad (4)$$

$$Q^*(s, a) = \textstyle\sum_{s'} P(s'|s, a)(R(s, a, s') + \gamma \, max_{a'} \, Q^*(s', a')) \qquad (5)$$

The solution is thus:

$$Q_{k+1}{}^*(s, a) \leftarrow \textstyle\sum_{s'} P(s'|s, a)(R(s, a, s') + \gamma Q_k{}^*(s', a')) \qquad (6)$$

The advantage of Q-learning is employing the Temporal Difference (TD) learning method and offline learning for agents.

The TD learning incorporates the sampling method of the Monte Carlo method and the bootstrapping of the Dynamic programming method, which operates the latter state's value function to estimate the current value function. These features make TD fit in the model-free algorithm and accomplish the goal faster. The value function is presented as follows:

$$V(s) \leftarrow V(S) + \alpha(R_{t+1} + \gamma V(s') - V(s)) \qquad (7)$$

The $R_{t+1} + \gamma V(S')$ is called the TD object, and the $\delta_t = R_{t+1} + \gamma V(s') - V(s)$ is called the TD bias.

The Q value can be calculated according to the formula (8). This is the renewal process of the Q-table.

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \, max_{a'} \, Q \, (s', a') - Q(s, a)] \qquad (8)$$

The formula above is the update function of the Q-learning. The agents would select the max Q (s', a') multiplied by the discount rate and then add the real reward value as the Q-reality based on the following state: s'. The previous Q-value in the Q-table would be the Q-estimate.

$$Q(s_1, a_2)_{\text{reality}} = R + \gamma * max \, Q \, (s_2)$$

$$Q(s_1, a_2)_{\text{estimate}} = Q(s_1, a_2)$$

$$Q_{difference} = Q_{reality} - Q_{estimate}$$

$$NewQ(s_1, a_2) = Old(s_1, a_2) + \alpha * Q_{difference}$$

The loop would then be generated to calculate the final Q-table. The pseudocode of the Q-learning algorithm shown below was used:

Algorithm 1 Pseudocode of the Q-learning

Initialize *Q(s, a)* arbitrarily

Repeat (for each episode):

 Initialize *s*

 Repeat (for each step of the episode):

  Choose *a* from *s* using policy derived from Q (e-greedy)

  Take action *a*, observe *r, s'*

  $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \, max_{a'} \, Q \, (s', a') - Q(s, a)]$

  $s \leftarrow s'$

 Until s is terminal

The general algorithm for multi-agent was derived from the single-agent Q-learning. For the multi-agent algorithm, agents would examine the environment jointly and renew the same Q-table. The shared Q-table was their communication method for collaboration. The pseudocode of the multi-agent Q-learning algorithm is shown below:

Algorithm 2 Pseudocode of the Multi-agent Q-learning

Initialize *Q (s1, a)* arbitrarily

Initialize *Q (s2, a)* arbitrarily

Repeat (for each episode):

 Initialize *s*

 Repeat (for each step of the episode):

  Choose *a* from *s1* using policy derived from Q_1(ε-greedy)

  Choose *b* from *s2* using policy derived from Q_1(ε-greedy)

  Take action *a*, observe *r, s'*

  Take action *b*, observe *r2, s2'*

$$Q1(s1, a) \leftarrow Q1(s, a) + e \text{ act max}_{a'} \ Q\ 1(s'\ , a'\ ) ass, a) + e\ a$$

$$Q1(s1, b) \leftarrow Q1(s1, b) + \alpha[r + \gamma \ max_{a'} \ Q\ 1(s2'\ , b'\ ) - Q1(s2, b)]$$

$$s \leftarrow s'$$

Until s is terminal

Results

This section presents the results for both studies, including the agents' performances, including time, death counts, and reward. The results of single-agent exploration and multi-agent collaboration were presented for both studies. For study 1, a statistical analysis was conducted to compare the differences between the different conditions For study 2, graphical illustrations were used to clearly show the gap in rewards between the two agents exploration methods.

Study 1 Results

The results of agent exploration time, death counts, and reward for the simple and complex environment with/without collaboration are exhibited in Table 2. Time, death counts, and reward were the average result through 5000 episodes.

Table 2

Descriptive Statistics Between Simple and Complex Environment

Environment	N = 50	Collab. Time (s)	No Collab. Time (s)	Collab. Death	No Collab. Death	Collab. Reward	No Collab. Reward
Simple	Mean	24.099	30.116	.009	.140	-36.280	-44.273
	SD	.180	.444	.001	.006	.043	1.496
Complex	Mean	137.415	616.215	.102	2.466	-82.820	-364.295
	SD	7.063	22.348	.007	.099	.048	18.834

Note. Collab. = Collaborative method, *SD* = Standard Deviation

Hypothesis Testing

From the results above, it can be seen that the environment has some influences on the performance of agent collaboration; three two-way ANOVAs were performed to

further test the effect of collaboration and environment on the agent learning performance, in terms of time, death counts, and reward between collaboration and no-collaboration methods, and the interaction effect on these measures between the collaboration method and the complexity of environments. The assumptions of equality of variance were tested. Levene's tests of equality of variance were significant ($p < .05$), and thus unequal variances were assumed for all three ANOVAs. Three two-way between-subjects ANOVAs and interaction effects were all significant at the alpha level of .05, $p < .001$; the results were shown in Table 3. The time for agents with collaboration to find the exit was significantly lower than the agent without collaboration in both environments.

Two null hypotheses were thus rejected. There was a significant difference in the performance between single-agent exploration and multi-agent exploration regarding time, death counts, and rewards. There were interactions between the complexity of the environment and the performance, including time, death counts, and rewards.

The death counts for agents with collaboration were significantly lower than those without collaboration in both environments. The reward for collaborating agents was substantially higher than the agent without collaboration in both environments. Figures 5, 6, and 7 below show all three positive interactions. The agents' performance, including evacuation time, death counts, and reward, increased when the environment became more complicated (0 refers to simple environment and single agent without collaborations).

Table 3

Two-Way ANOVA for Environment, Collaboration or Both

DV	Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Time	Environment	6114755.287	1	6114755.287	44506.830	.000
	Collaboration	2938105.373	1	2938105.373	21385.280	.000
	Environment * Collaboration	2794053.190	1	2794053.190	20336.783	.000
Death Count	Environment	73.136	1	73.136	29626.705	.000
	Collaboration	77.785	1	77.785	31510.208	.000
	Environment * Collaboration	62.278	1	62.278	25228.536	.000
Reward	Environment	1679604.422	1	1679604.422	18821.675	.000
	Collaboration	934902.948	1	934902.948	10476.538	.000
	Environment * Collaboration	1047398.477	1	1047398.477	11737.165	.000

Figure 5
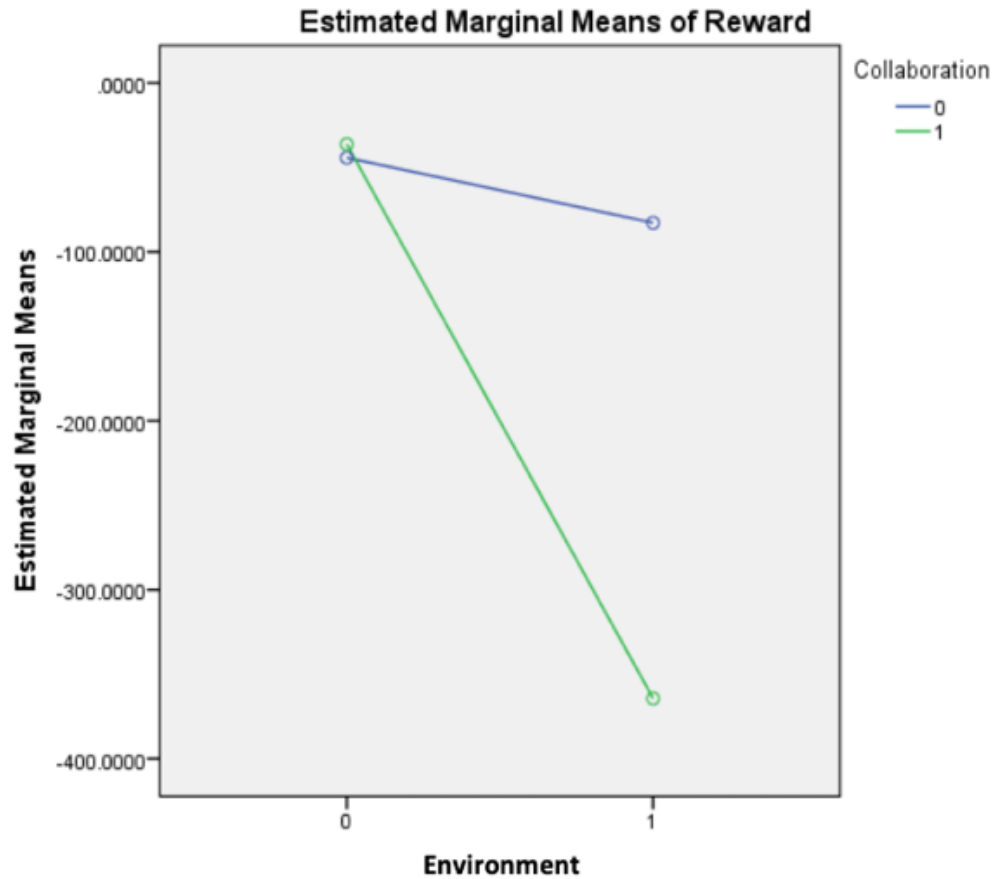
Interaction Between IVs Regarding Time
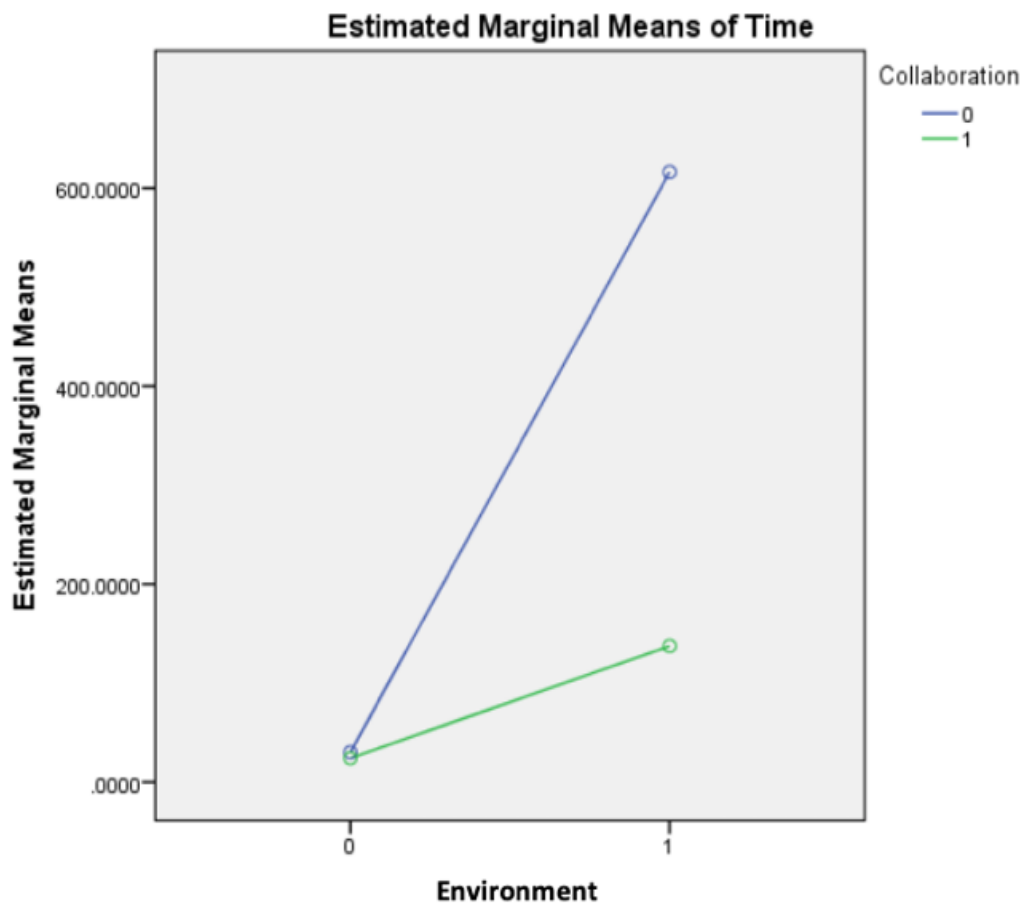
Figure 6

Interaction Between IVs Regarding Death Counts
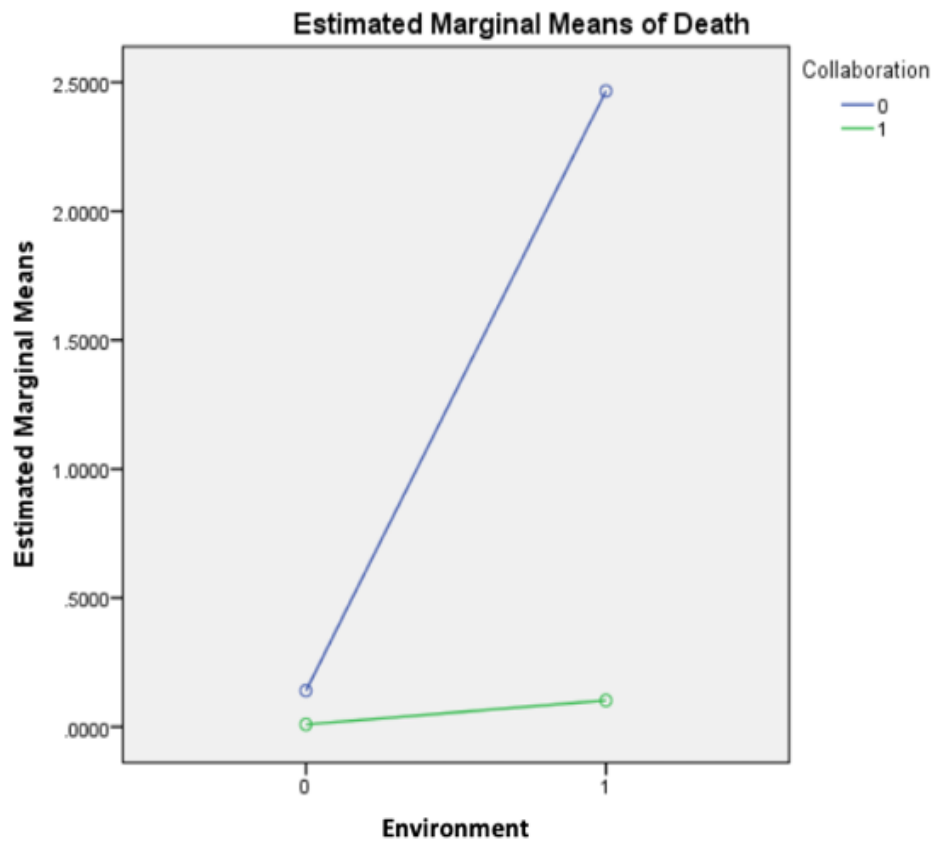
Figure 7

Interaction Between IVs Regarding Reward
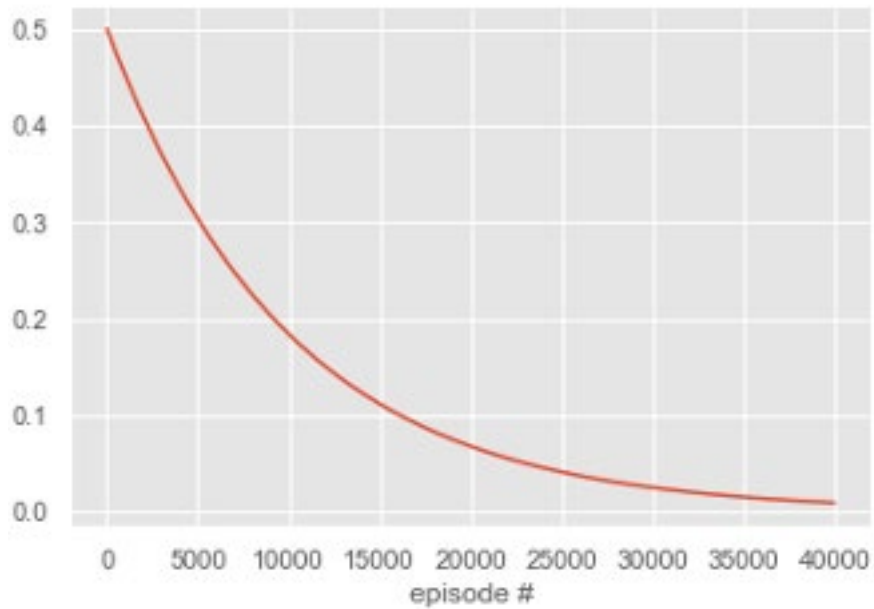

Exploration Behaviors

Agents are set to move in eight different directions as default, which suggests there are eight possible choices for every state. With the exploration and exploitation strategy, the experiment initialized the exploration rate as 0.5, while the exploration rate will be 0.9999. The formula for the exploration and exploitation will be:

$$\varepsilon = (0.9999)^{Episodes} * 0.5$$

The exploration rate based on the episodes is shown in Figure 8, with a maximum episodes of 40000:

Figure 8

Exploration Rate and Episodes

Study 1 Single Agent Exploration Behaviors

Figure 9 displayes the final actions on each position (state), and Figure 10 highlightes the learned best route (policy) for the evacuation. The agent starts with random exploration, and after several episodes, specific actions would be bypassed, and eventually, the optimal policy was learned.

Figure 9

Agents Choices for Each Position in Simple or Complex Environment
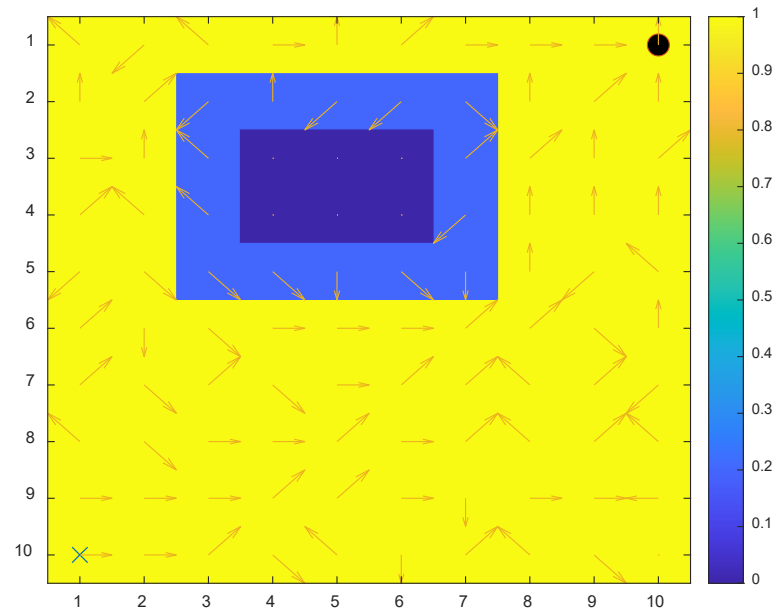
Note. Simple Environment
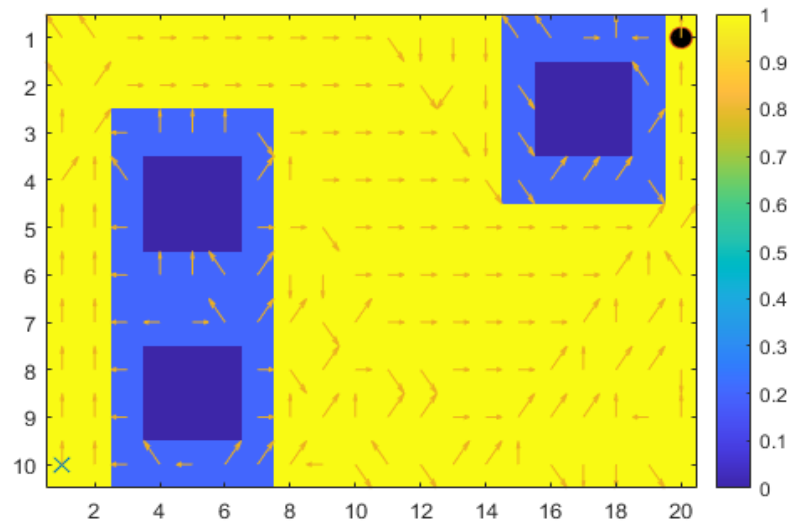
Note. Complex Environment

Figure 10

Optimal Route for Evacuation in Simple or Complex Environment
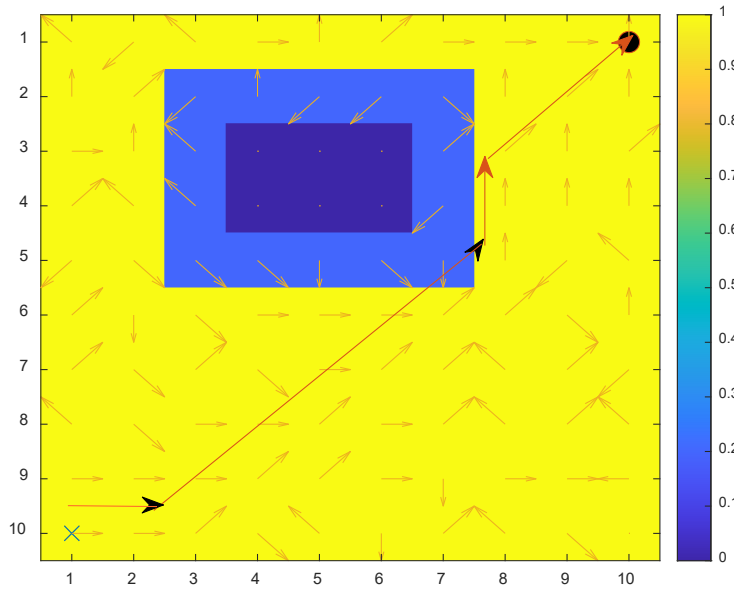
Note. Simple Environment
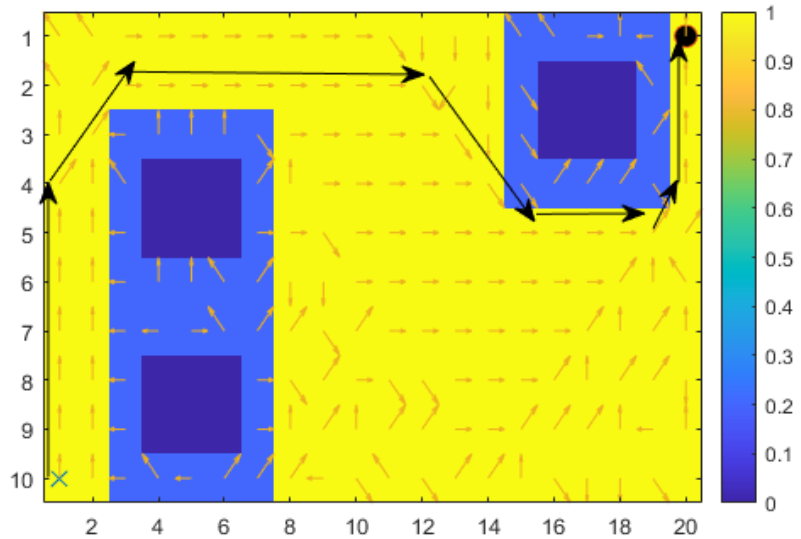
Note. Complex Environment

Figure 11 depicts the rewards and time for the simple and complex environment. From Figure 11, the learning performance constantly improved until 500 episodes, and after that, the improvement of learning diminished to a plateau steady state. The

computation time dropped sharply in the simple environment and stabilized after 1000 episodes. The time in the steady-state period was lower than 0.005 seconds per episode. In the complex environment, the agent followed a similar pattern as in the simple environment, except the agent needed a long time to learn from the environment. The reward increased continually in the first 1000 episodes and became stable around 3000 episodes. The cumulative reward was slightly lower, and the time was slightly higher than in the simple environment. The time was not as steady as those of the simple environment. The higher complexity level caused this unsteady environment and the insatiable policy's choice.

Figure 11

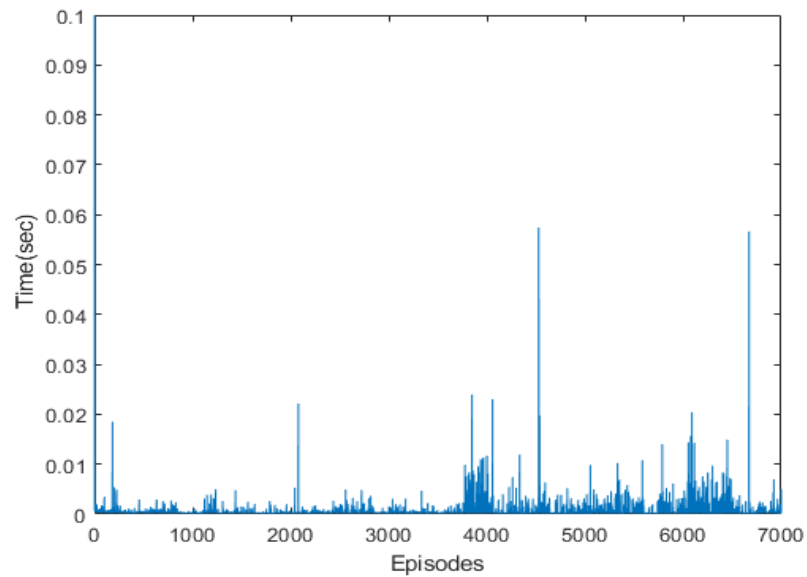*Reward and Computation Time in Simple and Complex Environment*
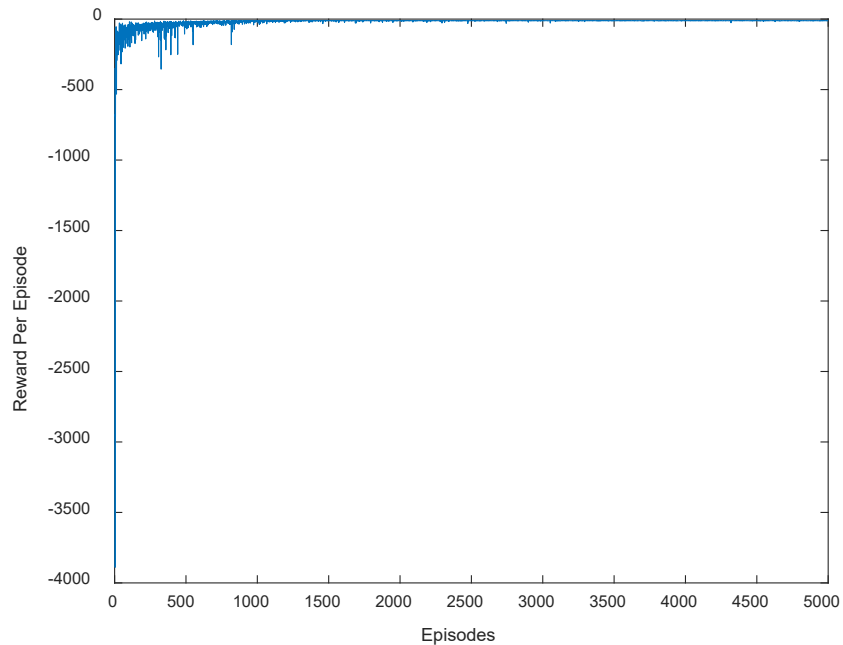
Note. Time in Simple Environment
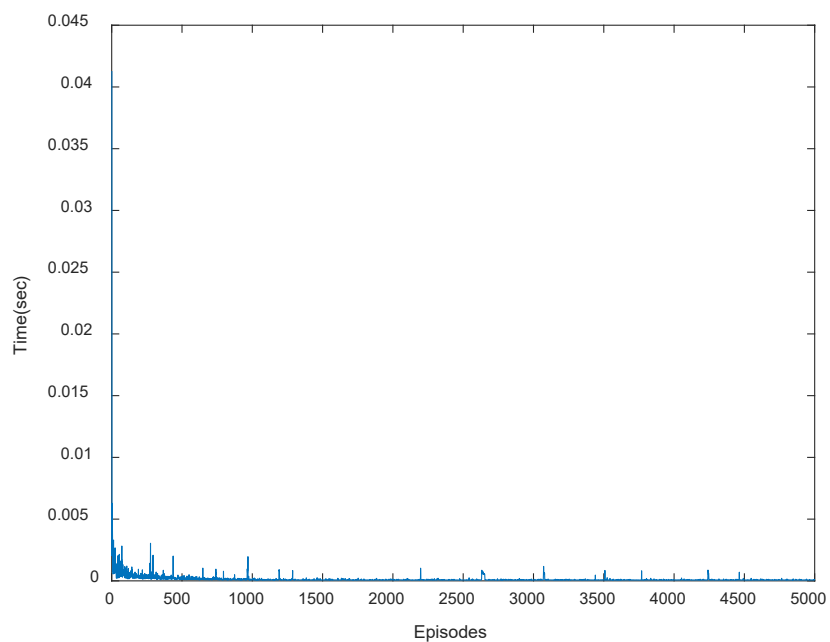
Note. Time in Simple Environment
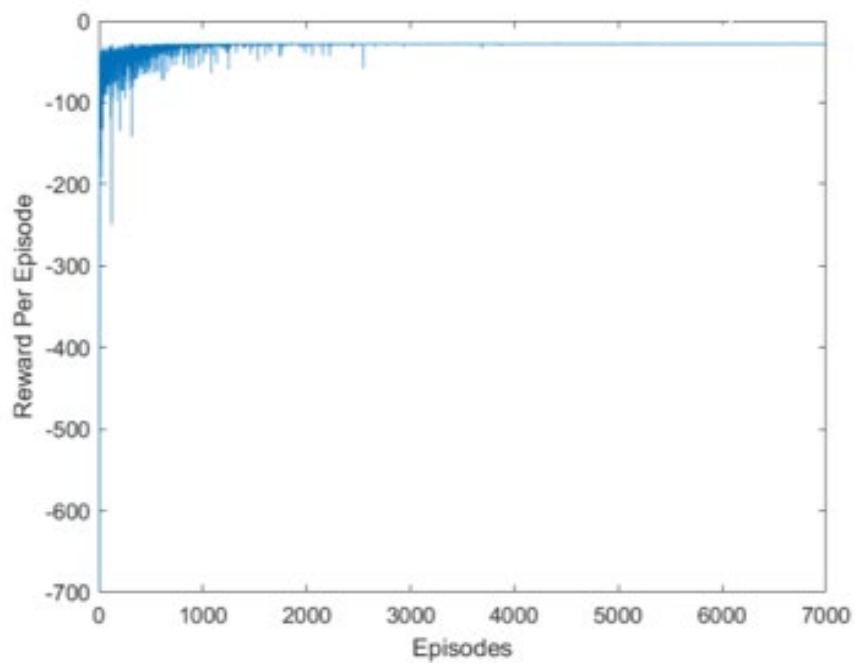
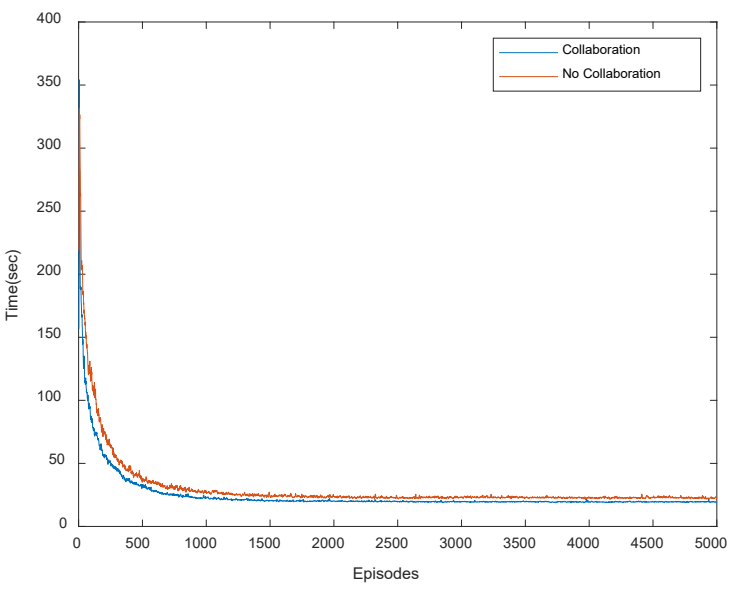*Note.* Reward in Complex Environment

Note. Time in Complex Environment

**Study 1 Multi-Agent Collaboration Behaviors**

The mean value of fifty trails and a moving average filter was utilized to demonstrate the main effects. The results are displayed in Figure 12. The red line represents the result for no collaboration exploration, and the blue line represents the collaboration exploration. The left figure shows the time, and the right figure shows the number the agent that touched the threat and died.
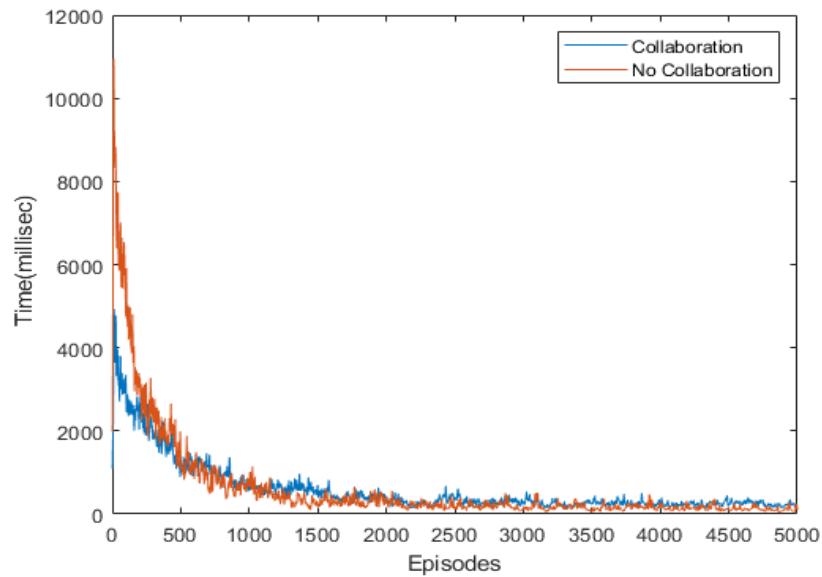
Figure 12

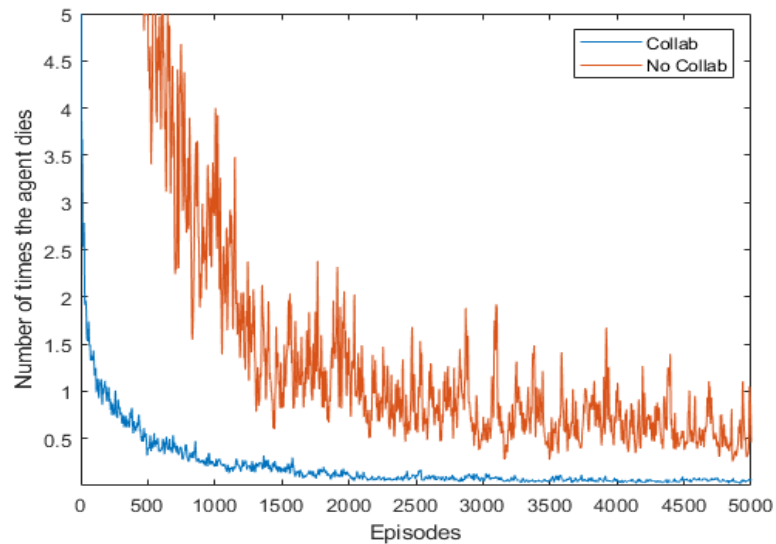*Time and Death Counts With/Without Collaboration in Simple and Complex Environment*
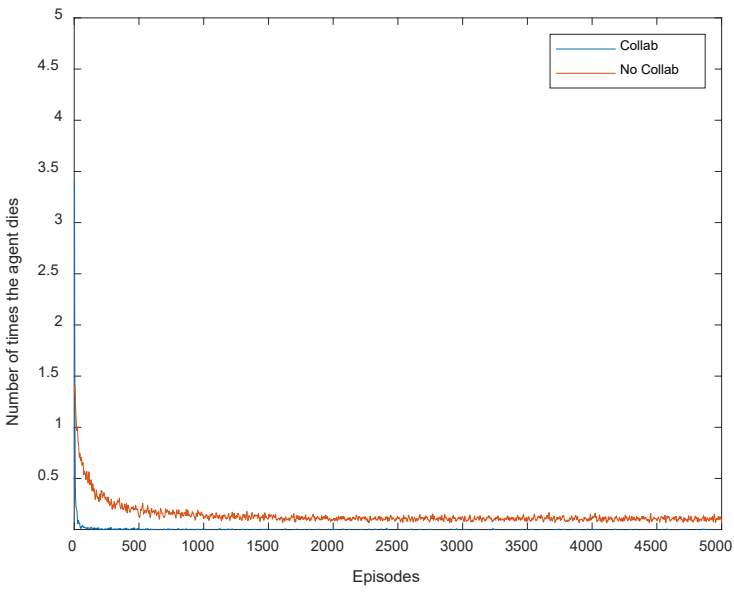
Note. Time in Simple Environment

Note. Death Counts in Simple Environment

Note. Time in Simple Environment

Note. Death Counts in Simple Environment

It is fairly straightforward to visualize for simple and complex environments, and collaboration outperformed the non-collaboration agents. More specifically, for the simple environment, the time for agents with collaboration to converge to the optimal policy and reach a stable value was much faster than non-collaboration agents. The collaboration's method needs 19.46 milliseconds agents without collaboration used 22.8 seconds, which improved by 17.6 percent. In the complex environment, agents with

collaboration had similar steady-state values, but agents with collaboration converged to the stable value faster.

In the simple environment, the tendency of touching the threat was similar between the agents with and without collaboration. However, it can be noticed that the number of threats touching agents with collaboration dropped much faster than those of the agents with no collaboration.

The advantage of the collaboration for agents to avoid the threat was much more pronounced in the complex environment than in the simple environment. From Figure 9, it can be observed that the probability of death in the collaboration model dropped much faster than in the non-collaboration model. As the environment became more complex, the benefits of the collaboration became more prominent.

Study 2 Results

The result of study 2 showed a similar benefit for multiple agent collaboration. In this study, it was found that the main benefits for multiple agents' collaboration is in the time to converge to the steady state, and the rewards profiles are similar. After examining an exploitation strategy and implementing the Q-learning algorithm into this experiment, study 2 adopted the heat-map statistic method to obtain the final route computed by the Q-learning algorithm, to take a deeper look at the agent behaviors, as shown in Figure 13.

Figure 13

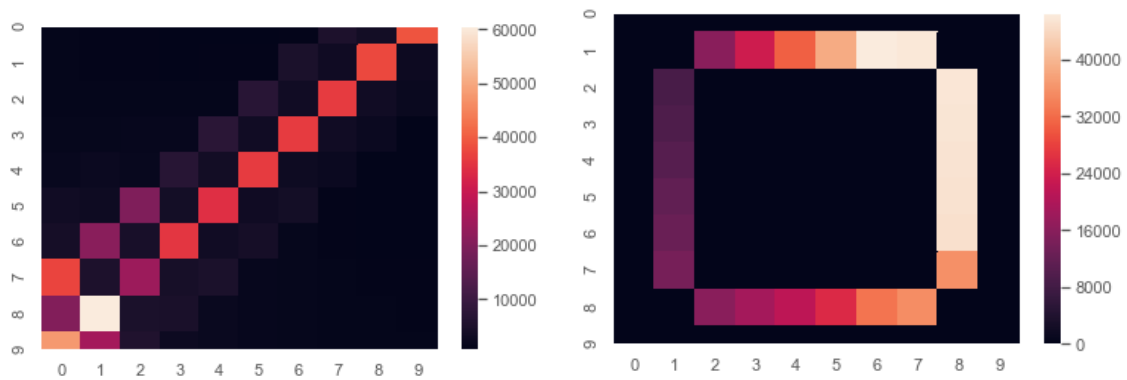*Heat-Map of Agent Movement vs. Obstacle Movement*

Figure 13 exhibits the obstacle and the agent's movement, demonstrating that the agent followed the diagonal route to evacuate. The agent finally reached everywhere on

the 10*10 map; the reason for the map concentrated on this diagonal line is that the agent spent more time on this optimal route than any other position. The common log transformation was adopted to avoid the skewed distributions of state visits. The formula for the log transform is: $= log(x + 1)$, where x is the number of times an agent passes a location. After doing the log transformation, the second heat-map is generated for the agent and the obstacle movement, as following Figure 14, which ended up with a smoother distribution:
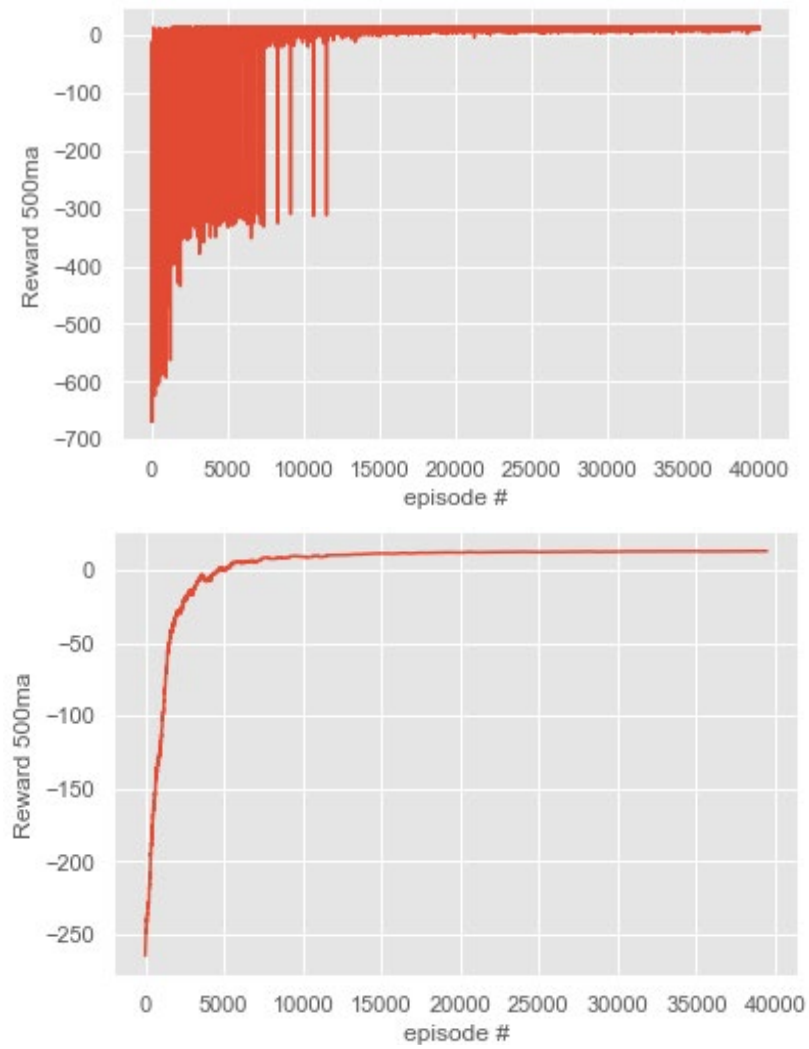
Figure 14

Heat-Map of Agent Movement vs Obstacle Movement (After Log Transform)
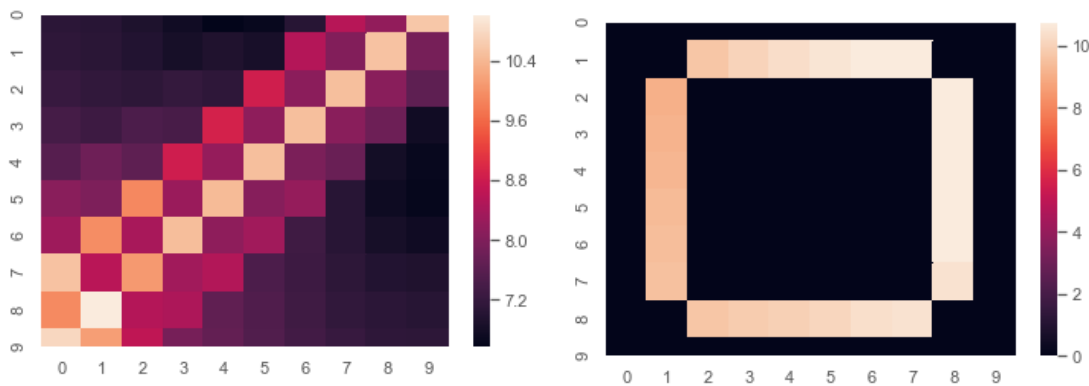
The experimental data improved particularly in the first 5000 experiments. Because the reward for each episode is zero at the start of the training, the agent will presumably move randomly so that the reward gradually changes based on every episode, sometimes these changes can be abrupt due to the different level of randomness involved. In this study, in order to observe the overall trend, the moving average algorithm was implemented by the NumPy convolution to smooth the reward curve, in order to better visualize the results. For the moving average, a time window is slid along with the input and compute the mean of the window's contents, the comparison between the reward curve and the reward curve after running means is shown in Figure 15:
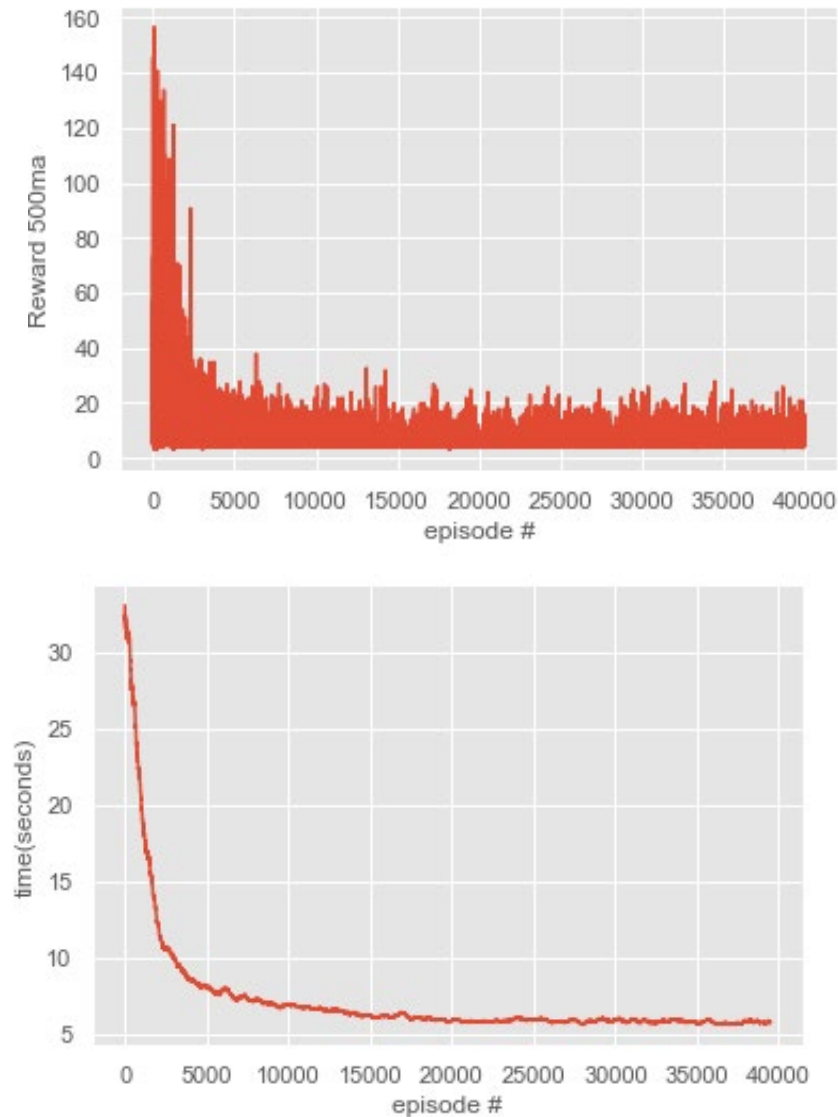
Figure 15

Original Reward Curve vs. Smoothed Reward Curve

The moving average computation time value is around five milliseconds per episode, the running time to achieve a steady-state is shown in Figure 16:

Figure 16

Computation Time Curve vs. Running Mean Computation Time

The two agents start simultaneously in the beginning, and they mostly likely went in different directions, due to the random initial actions. During the learning phase, they updated the Q-learning table simultaneously shared Q-learning table data with each other. So it is expected that final Q-table converged much more rapidly than the case without collaborations. This experiment compared the two agents moving to the target separately and the two-agent moving to the target by using the collaboration algorithm. The computation time is being compared in Figure 17.

Figure 17

*Episodes Over Time for Collaboration vs. Separate*
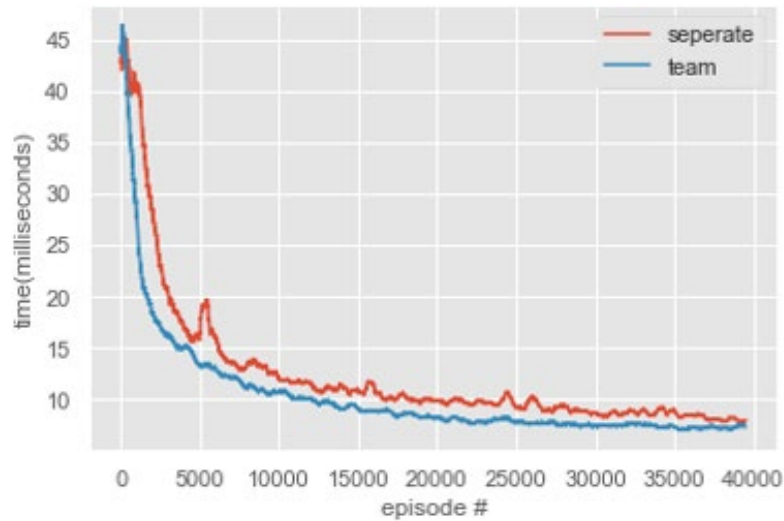
In Figure 17, the red line represents the no-collaboration case, whereas the blue line represents the collaboration case. The figure shows that the blue curve drops and reaches a stable value faster than the red curve, and the calculation time is shorter to get the steady state. The collaboration's method is cost 6.5 milliseconds per episode. However, the no-collaboration one costs 7.5 milliseconds per episode, which improves 15.4 percent. And from the curve, with the training progress, the computation time curve of the collaboration is found to be smoother than the separation one. That implies that the collaboration algorithm is a little more stable in converging to the optimal policy. And it is worth noting that in this situation, the size of the room is only 10*10 meters; with the increasing size of the map, the collaboration method could be more effective.

Figure 18-20 show the comparison results for the real airports simulations based on Figure 4.

Figure 18

Airport layout 1 reward comparison
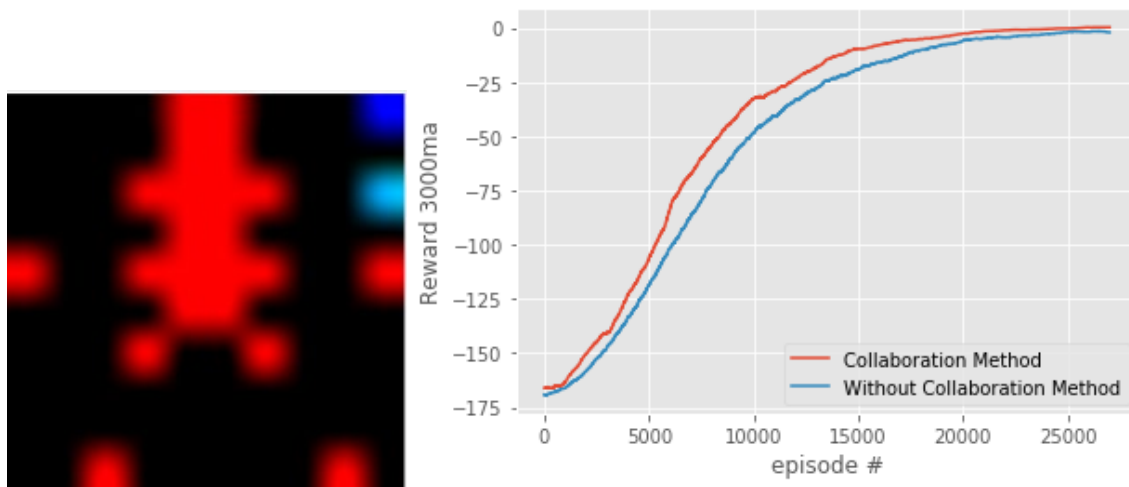
Figure 19

Airport layout 2 reward comparison
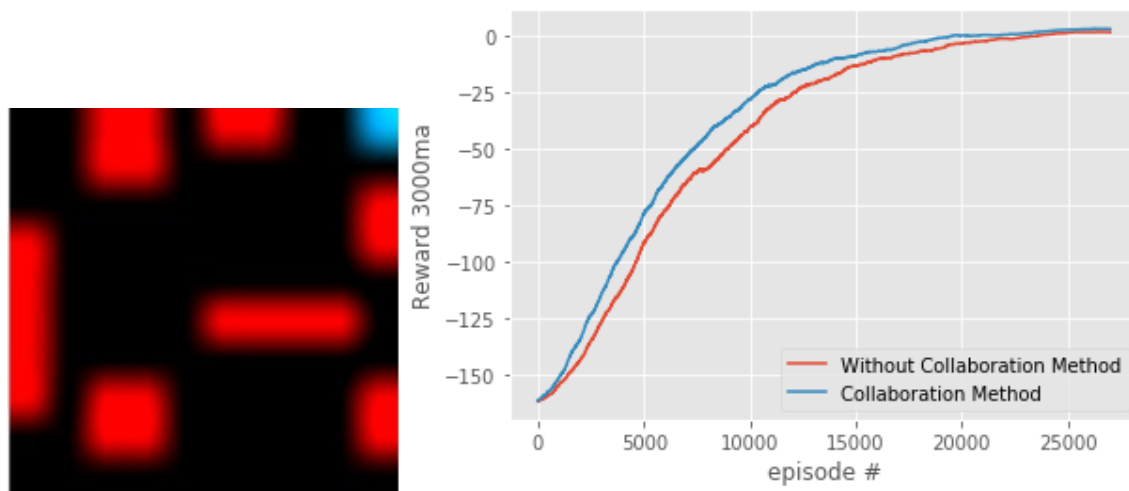
Figure 20

Airport layout 3 reward comparison
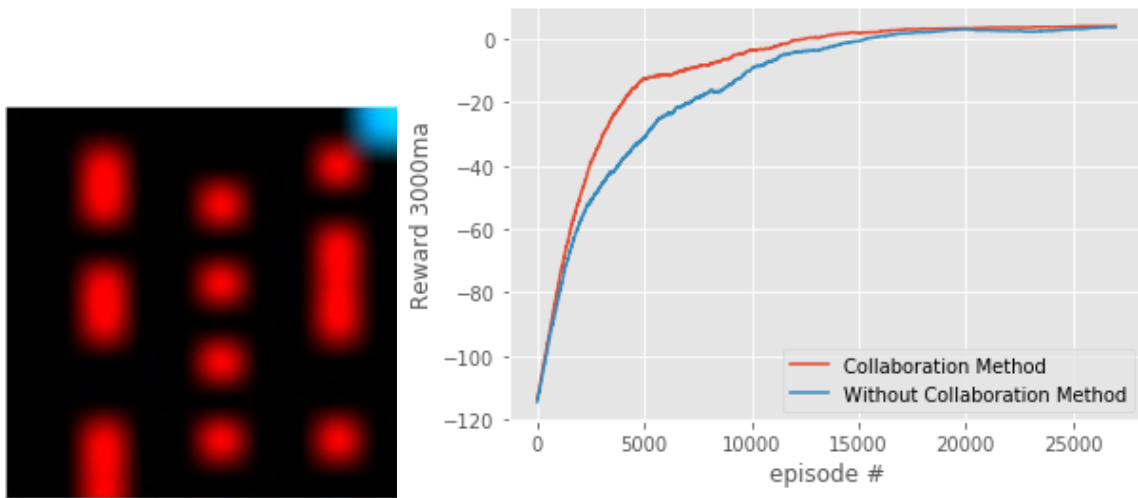
From the experiments with three airport layout maps, it can be seen that the collaboration showed a similar benefit for all three maps. Although both methods achieve same results after 25000 episodes or so, the results implied that they can both successfully converge to the optimal evacuation policy, however, from the three plots, the collaboration case converges much faster. This is same as the previous case with 10 *10 room experiment. With a more complex environment like the airport, with more obstacles involved, multi-agent collaboration demonstrated efficiency in learning the evacuation route, implying saving times to find the best evacuation route.

Discussion, Conclusions, and Recommendations

Discussions

An emergency could happen anytime at any place. Under urgent circumstances, it is hazardous and time-consuming for evacuees to identify the most proper evacuation route, especially if there are some uncertainties and life-threatening conditions in the surroundings, more so if the evacuees are not familiar with the environment. The public may panic during the evacuation and exhibit flawed decision-making capability. For the transportation system with high pedestrian density, the insufficiency of training and assistance will cause significant chaos and congestion. Training can be added and emphasized during daily operations, but different kinds of assistance should also be in place to deal with the crisis whenever necessary. The use of autonomous agents for exploration could save time and minimize the risk for human beings. The agents can quickly examine the unfamiliar environment and form the optimal evacuation route from learning, and these routes can be communicated to humans to guide the evacuation process.

Using the RL algorithm, this study used robotic agents to model the navigation process in an unknown environment. Agents can learn the environment quickly, find the optimal evacuation route efficiently, and avoid threats effectively. Two types of environments, simple and complex, and two types of agent interaction, collaboration or no-collaboration, were compared to investigate their effects on the learning performance. Results discovered that the multi-agent collaboration algorithm delivered a significantly better performance than the single-agent method regarding the time needed to locate the exit and the reward, which were calculated by total steps and punishments in both simple and complex environments. The differences were much more evident in the complex environment.

The RL agents learned the optimal policy from the environment through trials and reinforcement. Time is critical during an emergency evacuation. The efficiency of learning depends on the knowledge of the environment; the more knowledge about the environment, the an agent converges to the optimal navigation policy. By collaborating,

in this case, sharing the learning experience, the agents would have a more prompt and accurate understanding of the conditions, thus, using significantly lower time to learn the route-to-exit than the agents without collaboration. In the scenario of the simple environment, agents' performance reached a steady state at around 5000 episodes, and agent steady-state performance is better than that in the complex environment, which is expected, as the learning in the complex environment is much more difficult. The episodes needed to reach a steady state also showed the level of complexity of the environment. The number needed to learn can be used as a learning time limit for planning and resource assignments. In the real-life evacuation situation, the learning episodes could be translated to the number of agents required according to the size and complexity level of the area in the building.

One common problem using the RL algorithm is the convergence rate: the speed to converge or stabilize to a specifically targeted performance value. In this study, results obtained by adopting the multi-agent collaboration showed a higher convergence rate than the single-agent method results, as demonstrated in both studies, which is another advantage of utilizing agent collaboration. It is evident from the results that, the multi-agent collaboration method can find an explicit and steady optimal route solution faster and easier.

The use of autonomous agents for exploration could save time and minimize the risk for human beings. The agents can provide a generic outline before the exploration is finished. As the exploration process goes on, a more detailed evacuation route will be suggested. The information, including a holistic view of the map, relative location of the evacuee, and available evacuation route, can be manifested through mobile devices or available digital displays. The information detected by the robot agents can also be transmitted to first responders, like police, firefighters, etc., who need to enter the buildings of the transportation system. For airport certificate holders, this kind of assistance can be amended to the airport emergency plan as a part of their airport certification manual to comply with Title 14 CFR, Part 139.

In the scenario of the simple environment, agents' performance reached a steady state at around 2000 episodes, and agent steady-state performance is better than that in the complex environment, which is expected, as the learning in the complex environment

is much more difficult. The episodes needed to reach a steady state also showed the level of complexity of the environment. The number needed to learn can be used as a learning time limit for planning and resource assignments. In the real-life evacuation situation, the learning episodes and death counts could be translated to the number of agents required according to the size and complexity level of the area in the building.

The benefit of the multi-agent collaboration was further confirmed by the second study, where the threats were moving at either a pre-defined route or randomly, on top of the complexity of the environment, as well as the experiment from a more complicated airport environment. The communication and sharing of information helped agents to focus on the most likely optimal routes without the need to explore all the possibilities of the evacuation routes. This is certainly demonstrated by the faster convergence rate of the agent's rewards plot. Although the final rewards were similar between the no-collaboration and collaborative agents, the saving the convergence times implies that the collaborating agents can find the optimal routes faster, which implies time saving the evacuation situations. Because in an emergency every second counts, the time saved could translate into lives saved with faster evacuation.

It has been recognized that one common problem using the RL algorithm is the convergence rate: the speed to converge or stabilize to a specifically targeted performance value. In this study, results gathered by adopting the multi-agent collaboration showed a higher convergence rate than the single-agent method results, which is another benefit of using agent collaboration. It is apparent from the results that the multi-agent collaboration method can find a clear and steady optimal route solution quicker and easier.

One limitation of this study was that the experiment environment was fully observable even though unknown to the agents in the beginning. The agent can't fully observe the environment in certain circumstances, which means the holistic map is unavailable. Under this situation, it is necessary to introduce the partially observable Markov decision process (POMDP). In POMDP, it is assumed that the system dynamics have the Markov property, but the agent cannot observe the underlying state directly. In ordinary Q-learning, when the state and action space are low-dimensional and discrete, the Q-Table could be used to store the Q value of each state-action pair. When the state

and action space are high-dimensional and continuous, it is challenging to use the Q-table to track state-action values. Under this circumstance, it is more effective to transform the Q-table update into a function-fitting problem. And to be specific, in our experiment environment, the obstacle is moving in a fixed trajectory. However, in the real-life setting, the threats are probably moving in a random trajectory which is hard to predict. So, this phenomenon will significantly increase the complexity of the circumstances.

During this experiment, we observed that some agents fell into a deadlock (endless loops) due to the combined effect of exploration and exploitation. In some situations, the exploration rate decreased before the best actions in the current state had been learned. The agents may thus not take the optimal action or even fall into a deadlock loop, which will make the reward for that episode extremely low.

The size of the room is limited to 10*10 or 10*20 meters in both studies. For agents with collaboration, the possibility of agents touching the threats would be decreased to zero after specific trials, but death counts still exist after a long period of learning. Even the death counts could stabilize to a low value for the agent without collaboration. For a bigger-size environment, this could be a severe problem. As the interaction effects between the complexity and the collaboration method were significant, the collaboration method will have much better performance in a more complicated environment. In a real-life emergency evacuation, the actual area of the evacuation would be much bigger and more complex. The collaboration method will thus be much more beneficial to save resources and time for the evacuation process.

Conclusions

Agent exploration can be an excellent substitute for human exploration in a dangerous environment during evacuation. The use of agent exploration can generate a digital map with the optimal route in the mobile app, which would reduce the time for evacuees to find the exits. A multi-agent collaboration method is an approach that lets agents find exits faster with lower risks. The agents showed better performance in discovering time, death counts, and rewards. There were significant interactions between the complexity of the environments and the collaboration method on discovering time,

death counts, and rewards. It is timesaving in the emergency evacuation to use the agents to complete the tasks and use multi-agent collaboration to find the optimal evacuation route. It could further reduce the time and threat of the tasks.

Recommendations

For future research, a better exploration and exploitation strategy can produce a higher performance in evacuation efficiency. The fixed environment can also be changed to a stochastic and dynamic environment. A more complex stochastic environment can cause a much lower convergence rate for the algorithm. Deep RL and partially observable Markov decision process would be suggested for solving this problem in a future experiment. A deep neural network has a good effect on extracting complex features. Combining deep learning with reinforcement learning may be an excellent solution for better performance in a complicated environment.

References

Airport Emergency Plan, 14 CFR § 139.325. (2011). https://www.law.corn-ell.edu/cfr/text/14/139.325.

Arai, S., Sycara, K., & Payne, T.R. (2000). Experience-Based Reinforcement Learning to Acquire Effective Behavior in a Multi-agent Domain. In: Mizoguchi R., Slaney J. (eds). *PRICAI 2000 Topics in Artificial Intelligence*, *1886*. https://doi.org/10.1007/3-540-44533-1-16.

Arumugam, G. P., Augustine, J., Golin, M. J., Higashikawa, Y., Katoh, N., & Srikanthan, P. (2016). Optimal evacuation flows on dynamic paths with general edge capacities. arXiv:1606.07208.

Balachandar, N., Dieter, J., & Ramachandran, G. S. (2019). Collaboration of AI agents via cooperative multi-agent deep reinforcement learning. *arXiv preprint*. arXiv:1907.00327.

Bi, C., Pan, G., Yang, L., Lin, C., Hou, M., & Huang, Y. (2019). Evacuation route recommendation using auto-encoder and markov decision process. *Applied Soft Computing, 84,* 105741. https://doi.org/10.1016/j.asoc.2019.105741

Bunea, G., Leon, F., & Atanasiu, G. M. (2016). Post disaster evacuation scenarios using multiagent system. *Journal of Computing in Civil Engineering, 30*(6). https://doi.org/10.1061/(ASCE)CP.1943-5487.0000575

Busoniu, L., De Schutter, B., & Babuska, R. (2005). Multiagent reinforcement learning with adaptive state focus. *In Proc. 17th Belgian–Dutch Conf. Artif. Intell. (BNAIC-05),* 35–42.

Busoniu, L., Babuska, R., & De Schutter, B. (2008). A Comprehensive Survey of Multiagent Reinforcement Learning. *In IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 38*(2), 156-172. doi: 10.1109/TSMCC.2007.913919.

Cheng, L., Reddy, V., Fookes, C., & Yarlagadda, P, K, D, V. (2014). Impact of Passenger Group Dynamics on an Airport Evacuation Process Using an Agent-Based Model. *2014 International Conference on Computational Science and Computational Intelligence,* 161-167. doi: 10.1109/CSCI.2014.111.

Cariño, J. M. N., & Garciano, L. E. O. (2019). Proposed evacuation safety index (ESI) for school buildings. *International Journal of Disaster Resilience in the Built Environment, 11*(3), 309-328. https://doi.org/10.1108/IJDRBE-06-2018-0028

Clouse, J. (1995). Learning from an automated training agent. *Presented at the Workshop Agents that Learn from Other Agents, 12th Int. Conf. Mach. Learn,* 9–12

Chaysri, P., Blekas, K., & Vlachos, K. (2020). Multiple mini-robots navigation using a collaborative multiagent reinforcement learning framework. *Advanced Robotics*, *34*(13), 902-916. DOI: 10.1080/01691864.2020.1757507

De Witt, C. A. S., Foerster, J. N., Farquhar, G., Torr, P. H., Boehmer, W., & Whiteson, S. (2018). Multi-agent common knowledge reinforcement learning. *arXiv preprint arXiv:1810.11702*

Fatima, M., & Pasha, M. (2017). Survey of Machine Learning Algorithms for Disease Diagnostic. *Journal of Intelligent Learning Systems and Applications*, *9*, 1-16. doi: 10.4236/jilsa.2017.91001

Federal Aviation Administration (FAA). (2009). AC 150/5200-31C. Airport Emergency Plan. U.S. *Department of Transportation, Federal Aviation Administration.*

Feng, J., & Wang, Q. (2019). Emergency safety evacuation decision based on dynamic gaussian bayesian network. *IOP Conference Series. Materials Science and Engineering, 688*(5), 55076. https://doi.org/10.1088/1757-899X/688/5/055076

Ferber, J., Gutknecht, O., & Michel, F. (2004). From Agents to Organizations: An Organizational View of Multi-agent Systems. In: Giorgini P., Müller J.P., Odell J. (eds) Agent-Oriented Software Engineering IV. *AOSE 2003. Lecture Notes in Computer Science*, *2935*. https://doi.org/10.1007/978-3-540-24620-6-15.

Gelada, C., & Buckman, J. (2019). Three Paradigms of Reinforcement Learning, https://jacobbuckman.com/2019-10-25-three-paradigms-of-reinforcement-learning/, 2019

Gosavi, A. A. (2004). Reinforcement Learning Algorithm Based on Policy Iteration for Average Reward: Empirical Results with Yield Management and Convergence Analysis. *Machine Learning 55*, 5–29.

https://doi.org/10.1023/B:MACH.0000019802.64038.6c.

Gwynne, S. (1999). A review of the methodologies used in the computer simulation of evacuation from the built environment. *Building and Environment*, *34*(6), 741-749.

Kelleher, J. D., & Tierney, B. (2018). *Data science*. The MIT Press.

Lovas, G. G. (1998). On the importance of building evacuation system components. *In IEEE Transactions on Engineering Management*, *45*(2), 181-191. doi: 10.1109/17.669766

Lanctot, M., Zambaldi, V., & Gruslys, A. (2017). A unified game-theoretic approach to multiagent reinforcement learning. *Advances in Neural Information Processing Systems*, 4190–4203

Le, V., Vinh, H. T., & Zucker, J. (2017). Reinforcement learning approach for adapting complex agent-based model of evacuation to fast linear model. *2017 Seventh International Conference on Information Science and Technology (ICIST)*, 369-375. https://doi.org/10.1109/ICIST.2017.7926787

Liu, R., Jiang, D., & Shi, L. (2016). Agent-based simulation of alternative classroom evacuation scenarios. *Frontiers of Architectural Research, 5*(1), 111-125. https://doi.org/10.1016/j.foar.2015.12.002

Martinez-Gil, F., Lozano, M., & Fernández, F. (2014). 'MARL-Ped: A multi- agent reinforcement learning based framework to simulate pedestrian groups. *Simul. Model*, *47*, 259–275.

Makinoshima, F., Imamura, F., & Abe, Y. (2016). Behavior from tsunami recorded in the multimedia sources at kesennuma city in the 2011 tohoku tsunami and its simulation by using the evacuation model with pedestrian-car interaction. *Coastal Engineering Journal, 58*(4), 1640023-1-1640023-28. https://doi.org/10.1142/S0578563416400234

Martinez-Gil, F., Lozano, M., & Fernández, F. (2011). Multi-agent reinforcement learning for simulating pedestrian navigation. *In International Workshop on Adaptive and Learning Agents*, 54-69

Papoudakis, G., Christianos, F., Schäfer, L., & Albrecht, S. V. (2020). Comparative Evaluation of Multi-Agent Deep Reinforcement Learning Algorithms.

Purser, D. A. (2015). Fire safety and evacuation implications from behaviours and hazard development in two fatal care home incidents: FIRE SAFETY AND EVACUATION IMPLICATIONS FOR CARE HOMES. *Fire and Materials, 39*(4), 430-452. https://doi.org/10.1002/fam.2250

Quirion, N., Liu, D., Boquet, A., Lau, M. Y., & Vincenzi, D. A. (2014). Autonomous aerial vehicle sensor sensitivity modeling using signal detection theory. *IIE Annual Conference.* 2397.

Quirion, N., Liu, D., Ludu, A., & Vincenzi, D. (2015). Sensor sensitivity and reinforcement learning models on target acquisition task for autonomous vehicle. *IIE Annual Conference.* 1825.

Raileanu, R., Denton, E., Szlam, A., & Fergus, R. (2018). Modeling others using oneself in multi-agent reinforcement learning. *arXiv preprint arXiv:1802.09640.*

Shalev-Shwartz, S., Shammah, S., & Shashua, A. (2016). Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295.*

Shen, Y., Wang, Q., Yan, W., & Sun, J. (2015). An evacuation model coupling with toxic effect for chemical industrial park. *Journal of Loss Prevention in the Process Industries, 33*, 258-265. https://doi.org/10.1016/j.jlp.2015.01.002

Tan, M. (1993). Multi-agent reinforcement learning: Independent vs. cooperative agents. *In Proceedings of the Tenth International Conference on Machine Learning*, 330-337.

Tian, K., & Jiang, S. (2018). Reinforcement learning for safe evacuation time of fire in Hong Kong-Zhuhai-Macau immersed tube tunnel. *Systems Science & Control Engineering*, *6*(2), 45-56, DOI: 10.1080/21642583.2018.1509746

Tan, L., Hu, M., & Lin, H. (2015). Agent-based simulation of building evacuation: Combining human behavior with predictable spatial accessibility in a fire emergency. *Information Sciences*, *295*, 53-66. ISSN 0020-0255, doi: org/10.1016/j.ins.2014.09.029.

Tan, M. (1993). Multi-agent reinforcement learning: Independent vs. cooperative agents. *In Proc. 10th Int. Conf. Mach. Learn. (ICML-93)*, 330–337.

Weiss, G., & Ed. (1999). Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. *MA: MIT Press*

Wang, Q., Liu, H., Gao, K., & Zhang, L. (2019). Improved Multi-Agent Reinforcement Learning for Path Planning-Based Crowd Simulation. *In IEEE Access*, *7*, 73841-73855. doi: 10.1109/ACCESS.2019.2920913.

Wang, H., Mostafizi, A., Cramer, L. A., Cox, D., & Park, H. (2016). An agent-based model of a multimodal near-field tsunami evacuation: Decision-making and life safety. *Transportation Research. Part C, Emerging Technologies, 64*, 86-100. https://doi.org/10.1016/j.trc.2015.11.010

Xu, D., Huang, X., Mango, J., Li, X., & Li, Z. (2020). Simulating multi-exit evacuation using deep reinforcement learning.

Yang, Y., Zhang, K., Liu, D., & Song, H. (2020). Autonomous UAV Navigation in Dynamic Environments with Double Deep Q-Networks. *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC),* San Antonio, 1-7. doi: 10.1109/DASC50938.2020.9256455.

Yamamoto, K., Sawaguchi, Y., & Nishiki, S. (2018). Simulation of tunnel fire for evacuation safety assessment. *Safety (Basel), 4*(2), 12. https://doi.org/10.3390/safety4020012

Yao, Z., Zhang, G., Lu, D., & Liu, H. (2019). Data-driven crowd evacuation: A reinforcement learning method. *Neurocomputing (Amsterdam), 366*, 314-327. https://doi.org/10.1016/j.neucom.2019.08.021

Zhou, L., Yang, P., Chen, C., & Gao, Y. (2017).Multiagent Reinforcement Learning With Sparse Interactions by Negotiation and Knowledge Transfer. *In IEEE Transactions on Cybernetics*, *47*(5), 1238-1250. doi: 10.1109/TCYB.2016.2543238.

Zhang, Z., Zhao, D., Gao, J., Wang, D., & Dai, Y. (2017). FMRQ—A multiagent reinforcement learning algorithm for fully cooperative tasks. *IEEETrans. Cybern*, *47*(6), 1367–1379.

Zhang, G., Zhu, G., Yuan, G., & Wang, Y. (2016). Quantitative risk assessment methods of evacuation safety for collapse of large steel structure gymnasium caused by

localized fire. *Safety Science, 87*, 234-242.
https://doi.org/10.1016/j.ssci.2016.04.013

Zhang, S., & Guo, Y. (2015). Distributed Multi-Robot evacuation incorporating human behavior. *Asian Journal of Control, 17*(1), 34-44. https://doi.org/10.1002/asjc.1047