Proceedings of the 2014 Winter Simulation Conference
A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, eds.

EA-BASED EVACUATION PLANNING USING AGENT-BASED CROWD SIMULATION

Jinghui Zhong

School of Computer Engineering
Nanyang Technological University
Nanyang Avenue, 639789, SINGAPORE

Linbo Luo

School of Computer Science and Technology
Xidian University
Xi'an, 710071, CHINA

Wentong Cai

School of Computer Engineering
Nanyang Technological University
Nanyang Avenue, 639789, SINGAPORE

Michael Lees

Section Computational Science
University of Amsterdam
Amsterdam, 1098 XH, NETHERLANDS

ABSTRACT

Safety planning for crowd evacuation is an important and active research topic nowadays. One important issue is to devise the evacuation plans of individuals in emergency situations so as to reduce the total evacuation time. This paper proposes a novel evolutionary algorithm (EA)-based methodology, together with agent-based crowd simulation, to solve the evacuation planning problem. The proposed method features a novel segmentation strategy which divides the entire evacuation region into sub-regions based on a discriminant function. Each sub-region is assigned with an exit gate, and individuals in a sub-region will run toward the corresponding exit gate for evacuation. In this way, the evacuation planning problem is converted to a symbolic regression problem. Then an evolutionary algorithm, using agent-based crowd simulation as fitness function, is developed to search for the global optimal solution. The simulation results on different scenarios demonstrate that the proposed method is effective to reduce the evacuation time.

1 INTRODUCTION

The world is getting more and more crowded with the rapid social and economic development. It has become a common phenomenon that a large number of people gather at a public place such as a theatre, a shopping mall, and an airport terminal. If an accident such as a fire happens in such crowded scenarios, people may get injured or even die if they fail to evacuate the place in time. Therefore, safety planning for crowd evacuation, which tries to reduce the evacuation time by guiding people's evacuation behaviours, has become an important and active research topic nowadays (Ferscha and Zia 2009b, Kamkarian and Hexmoor 2012).

Over the past decade, various crowd evacuation modeling approaches have been proposed, including the Cellular Automaton Model (Daoliang, Lizhong, and Jian 2006, Liu, Yang, Fang, and Li 2009), the Social Force Model (Helbing, Farkas, and Vicsek 2000), the Flow-based Model (Hughes 2003), and the Agent-Based Model (Zarboutis and Marmaras 2004, Toyama, Bazzan, and Da Silva 2006, Luo, Zhou, Cai, Lees, Low, and Sornum 2011, Zhong, Luo, Cai, and Lees 2014). Different approaches simulate the crowd evacuation dynamics from different points of view. Among these, agent-based approach has become the most active and dominant one, due to its capability to model heterogeneous individuals. It is also flexible to incorporate various behavioral factors in agent-based model, which is helpful to study the dynamics of crowd evacuation.

978-1-4799-7486-3/14/$31.00 ©2014 IEEE

Some efforts have also been made to study people's evacuation behaviours based on crowd simulation, so as to reduce the evacuation time. However, existing works mainly focus on dynamic planning, which requires special sensor devices to detect the crowd dynamics (or a central control station to guide the crowd) in real-time. These approaches become inapplicable when the required devices (or the central station) are not available. For example, Kamkarian and Hexmoor (Kamkarian and Hexmoor 2012) combined Coulomb's electrical law and graph theory to develop a tool for guiding people out of a public building. The simulation studies reported promising performance of the method. However, the method requires special sensor devices to dynamically detect environment features such as the body size and velocity of each individual, which can be a challenging issue in practical applications. Liu et.al. (Liu, Yang, Fang, and Li 2009) developed a Cellular Automaton Model to simulate an evacuation experiment conducted in a classroom with obstacles. Their simulation results showed that people choosing exits by taking into account the dynamic density around exits can lead to less evacuation time. Chen et. al. (Chen, Cheng, and Tseng 2012) developed a load-balancing framework to guide people evacuate from a building. Their method requires deploying a sensor network to identify hazardous region dynamically. Ferscha and Zia (Ferscha and Zia 2009b, Ferscha and Zia 2009a, Ferscha and Zia 2010) developed wearable device named LifeBelt to support the evacuation process of crowds from emergency situations. The LifeBelt assists individuals move towards the recommended exit gate based on sensing neighbourhood information such as the relative spatial relations (distance and orientation) of all individuals. Although the LifeBelt method has been shown very effective to reduce the evacuation time, it may not be practical in scenarios such as shopping malls and airport terminals, as it requires each individual wearing a special Lifebelt device.

To address the above issue, this paper proposes a safety planning methodology which generates evacuation plan before the occurrence of actual evacuation process. The proposed method does not require complex devices or a central control station. The dynamic safety planning methods can provide more robust guidelines to people than static planning strategy, because they can detect the dynamic environment features and unexpected events in real time. The objective of the study is not to develop a method that outperforms the existing dynamic safety planning methods. Instead, we aim to propose a general and low-cost methodology to guide the evacuation behaviours of individuals where existing dynamic safety planning methods are not applicable (e.g., in scenarios without sensor devices and the central control station).

The key idea of our methodology is to divide the evacuation area into sub-regions using a specific segmentation strategy defined by a discriminant function. Each sub-region is assigned with an exit gate. Once an evacuation occurs, individuals situated in a sub-region will be guided to move towards the corresponding exit gate. Since the planning strategy is generated in the design state, we can simply use sighboards to guide people to the recommended exits. In this way, safety planning is converted to a symbolic regression problem, i.e., to find the best discriminant function to divide the evacuation area. This paper develops an evolutionary algorithm to solve the symbolic regression problem. Agent-based crowd simulation is used to evaluate the fitness of each individual in a population. Our methodology is general, which can be applied to find safety planning strategies in different scenarios. To test the effectiveness of our approach, we apply our method to two evacuation scenarios. The simulation results show that our method is effective to reduce the evacuation time of the crowds, in comparison with several commonly used evacuation strategies.

The outline of the paper is as follows. Section II describes the problem definition. Section III presents the proposed methodology, and the simulation studies are given in Section IV. Finally, Section V draws the conclusion.

2 PROBLEM DEFINITION

We aim to divide the evacuation area into sub-regions and assign each sub-region with an exit gate (EG). Agents in a sub-region will choose the corresponding assigned EG when they start to evacuate. This is actually a classification problem, i.e., to classify the points in the evacuation area into different categories where each EG represents a category. One of the commonly used method in the classification research

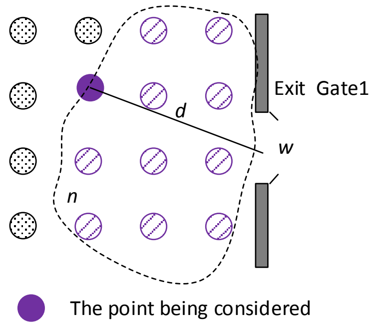

Figure 1: The three environment features considered to construct the discriminant function.

field is to construct a symbolic discriminant function that combines features of data as the classification rule (Espejo, Ventura, and Herrera 2010). Inspired by these works, this paper proposes to use a discriminant function (denoted as $\varphi(p)$), which returns a score value of each EG for any given position p, to accomplish the classification task. After calculating the score of each EG, the point p will be classified to the EG that has the highest score. Clearly, the score of each EG for p is dependent on some environment features, such as the distance from p to the EG, the number of people surrounding the EG, and the width of the EG. In this way, finding the optimal segmentation strategy is equivalent to finding the optimal discriminant function ($\varphi(p)$) which leads to the least evacuation time. The latter is a symbolic regression problem which can be solved using genetic programming (GP) (Zhou, Xiao, Tirpak, and Nelson 2003, Schmidt and Lipson 2009, Espejo, Ventura, and Herrera 2010).

Generally, the optimal segmentation problem can be described as: *Given some components such as the environment features and numerical functions, construct the optimal discriminant function by combining the components to divide the evacuation area, so that the total evacuation time of all agents using the partition strategy can be minimized.*

In this paper, we consider three environment features (as illustrated in Fig. 1) and four basic numerical operators to construct $\varphi(p)$:

$$\begin{cases} \text{feature set} = \{d, w, n\} \\ \text{function set} = \{+, -, *, /\} \end{cases} . \tag{1}$$

The first factor (d) is the distance to the EG. Intuitively agents are more likely to choose an EG with a shorter distance. The second factor (w) is the width of the EG. A wider EG would lead to a larger evacuation rate and a shorter evacuation time. The third factor (n) is the number of agents that have shorter distance to the EG than the agent at p. This factor is related to the waiting time for evacuation. A larger n will result in a longer evacuation time. We assume that agents are uniformly distributed in the evacuation area, so that we can estimate n if the total number of individuals is known in advance. Note that the feature set and function set adopted in this paper are defined empirically, based on the intuitive understanding of the evacuation behaviours in our daily life. These factors are effective to demonstrate the effectiveness of our methodology in our experiments, and it is possible that incorporating different features and functions into our methodology might generate better performance.

3 THE PROPOSED METHODOLOGY

In Section 2, we have converted the safety planning problem into an optimization problem. That is, to construct an optimal symbolic discriminant function to divide the evacuation area, so that evacuation time of agents using the partition strategy can be minimized. This optimization problem is known as a symbolic regression problem. Gene expression programming (GEP), first proposed by Ferreira (Ferreira 2001), is one of the notable evolutionary algorithm that has been shown very effective to generate accurate and

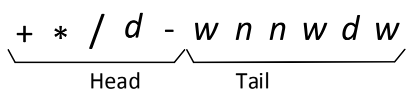

Figure 2: The structure of gene expression chromosome.

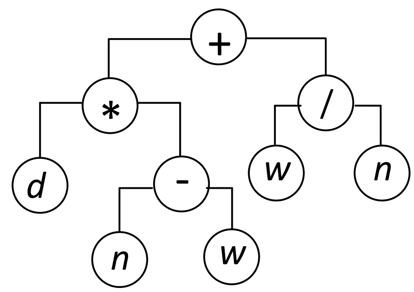

Figure 3: Example of an expression tree.

readable solutions to symbolic regression problems (Zhou, Xiao, Tirpak, and Nelson 2003, Ferreira 2006). Therefore, this paper adopts the GEP to search for the optimal discriminant function.

3.1 Chromosome Representation

In the GEP, each chromosome is represented by a vector of symbols with fixed length. The vector consists of two parts, namely the *"Head"* and the *"Tail"*, as shown in Fig. 2. Each element of the *"Head"* can be a function or a terminal, while each element of the *"Tail"* can only be a terminal. In this paper, the elements in the feature set are used as terminals. For example, given a function set $\mathbf{F} = \{+,-,*,/\}$ and a feature set $\mathbf{T} = \{d,w,n\}$, a typical GEP chromosome with length of 11 can be:

$$X = [+,*,/,d,-,w,n,n,w,d,w]. \tag{2}$$

Each chromosome can be converted to an expression tree (ET) using a breadth first traversal scheme. For example, the chromosome expressed in (2) can be converted to an ET as shown in Fig. 3, which can be further expressed in a mathematical formula as

$$d*(n-w)+\frac{w}{n}. \tag{3}$$

The length of the "Head" (H) and that of the "Tail" (L) are fixed. In order to ensure that any chromosome can result in a valid ET, H and L should have the following relation:

$$L = H \cdot (u-1)+1, \tag{4}$$

where u is the number of arguments of the function with the most arguments. For example, the number of arguments of "+", "-", "*", or "/" is equal to two. Hence, in this paper, we consider $u = 2$, and we have $L = H + 1$. Note that there may exist some redundant elements which are not useful for building the ET (e.g., the last two elements in (2)). But these redundant elements may become useful to build ETs in future evolution.

3.2 Algorithm Framework

Based on the above chromosome representation, the GEP adopts several GA-based operations (e.g., Crossover, Mutation and Transposition) to evolve a population of chromosomes, so as to search for the best discriminant function. The traditional GEP contains a number of controlling parameters. Tuning these parameters is time consuming. To solve this problem, this paper develops a modified GEP to evolve the chromosomes, by replacing the original GA-based operations with differential evolution operations. This is because that DE operations are easy to implement and contains only two controlling parameters (i.e., F and CR). Besides, DE has been shown more effective than several other EAs such as GA on various complex optimization problems (Price, Storn, and Lampinen 1997, Das and Suganthan 2011, Zhong and Zhang 2012, Zhong, Shen, Zhang, Chung, hui Shi, and Li 2013). Specifically, the procedures of the proposed algorithm are as follows.

3.2.1 Initialization

The first step is to generate a population of random chromosomes. Denote each chromosome X_i as:

$$X_i = [x_{i,1}, x_{i,2}, ..., x_{i,D}], \tag{5}$$

where $x_{i,j}$ is the j-th element of X_i, D is the length of the chromosome. The value of $x_{i,j}$ is initialized by:

$$x_{i,j} = \begin{cases} rand(\mathbf{F+T}), \text{if } x_{i,j} \text{ is belong to Head} \\ rand(\mathbf{T}), \text{otherwise} \end{cases}, \tag{6}$$

where $\mathbf{F}$ and $\mathbf{T}$ are the function set and terminal set respectively, and $rand(\mathbf{A})$ returns a random element in $\mathbf{A}$. Here we use integers to represent the function set and terminal set. Suppose we have a functions and b terminals. Then we use $1,2,...,a$ to represent the a functions, and $a+1, a+2, ..., a+b$ to represent the b terminals.

After generating the initial individuals, their fitness are evaluated. To evaluate the fitness of a chromosome, we first divide the evacuation area using the segmentation strategy represented by the chromosome. Then we evenly deploy a number of agents in the evacuation area and run the simulation. Agent belonging to each sub-region will always choose the EG assigned to the sub-region for evacuation. The fitness value of the chromosome is equal to the total evacuation time of all agents.

3.2.2 Mutation

In the second step, a mutant string (denoted as Y_i) is generated for each X_i by:

$$Y_i = Round(X_i + F*(X_{best} - X_i) + cauchy(0,1)*(X_{r_1} - X_{r_2})), \tag{7}$$

where r_1 and r_2 are two random individual indices, and F is a scaling factor. In this paper, the scaling factor is set to be a random number which is uniformly distributed within 0 and 1:

$$F = rand(0,1). \tag{8}$$

$cauchy(\mu, \sigma)$ returns a random number with cauchy distribution. $Round(t)$ is a truncation function that returns an integer value within 1 and $a+b$:

$$Round(t) = \begin{cases} a+b, \text{if } t > a+b \\ 1, \text{if } t < 1 \\ \lfloor t \rfloor, \text{otherwise} \end{cases}. \tag{9}$$

In this way, the elements of Y_i are always feasible values that represent functions or terminals.

3.2.3 Crossover

After generating a mutant string for each individual, the mutant string will crossover with its parent string to generate a trial string (U_i):

$$u_{i,j} = \begin{cases} y_{i,j}, \text{if } rand(0,1) < CR \\ x_{i,j}, \text{otherwise}, \end{cases} \tag{10}$$

where $u_{i,j}, y_{i,j}$ and $x_{i,j}$ are the j-th element of U_i, Y_i, and X_i respectively, and CR is a parameter to control the crossover rate. The value of CR is set as:

$$CR = rand(0,1). \tag{11}$$

3.2.4 Selection

The fourth step is to evaluate all trial strings and choose a better string from each pair of trial and parent strings to form the new population:

$$X_i = \begin{cases} U_i, \text{if } f(U_i) \leq f(X_i) \\ X_i, \text{otherwise} \end{cases}, \tag{12}$$

where $f()$ is the fitness evaluation function which returns the fitness of the input chromosome. As mentioned in Section 3.2.1, the fitness of a chromosome is evaluated using the agent-based crowd simulation. The algorithm repeats the second step to the fourth step until the terminate condition (e.g., the maximum generation) is met.

4 SIMULATION STUDIES

4.1 An Agent-based Crowd Simulation Platform

An agent-based crowd simulation platform is developed to test the algorithm. In the model, a number of agents are distributed in an enclosed region. There are several exit gates (EGs) in the evacuation region. Agents are supposed to move towards an EG and evacuate from the evacuation area. We use the social force model (Helbing, Farkas, and Vicsek 2000) to guide the motions of agents. The social-force model uses virtual forces to guide the motions of agents. The force imposed on an agent is expressed as:

$$f_i = m_i \frac{dv_i}{dt} = f_{i0} + \sum_{j(j \neq i)} f_{ij} + \sum_{w} f_{iw}, \tag{13}$$

where m_i and $\frac{dv_i}{dt}$ are the mass and acceleration rate of the agent; f_{i_o}, f_{ij} and f_{iw} are attractive force towards the goal, repulsive force from other agent and repulsive force from the static obstacles (e.g., wall) respectively. The formulation of f_{i_o}, f_{ij} and f_{iw} can be found in (Helbing, Farkas, and Vicsek 2000). On top of the social force model, some high level evacuation strategies are designed to determine the goal of the agents (i.e., choosing an exit), such as the distance first strategy (agents always choose the nearest EG) and the segmentation strategy provided by the proposed method.

4.2 Simulation Settings

Based on the defined crowd evacuation model, we design two scenarios for testing. The first scenario is a 24m $\times$ 14m rectangle room with four EGs, as shown in Fig. 4. The widths of the EGs are different. Agents reaching any of the four EGs are considered successfully evacuated from the room. The second scenario has a more complicated layout and contains five EGs and multiple rooms. An agent reaching any of the EGs is considered successfully evacuated and removed from the simulation.

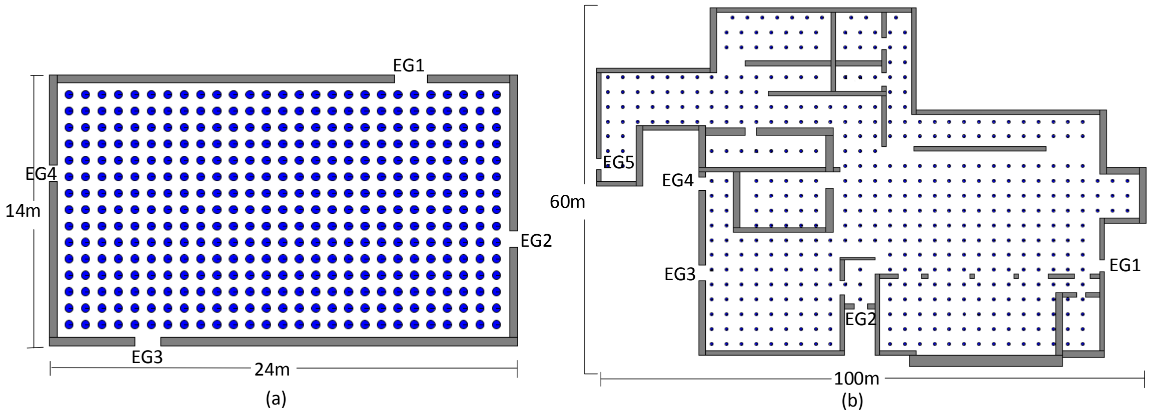

Figure 4: The two scenarios for simulation studies. (a) Scenario 1. (b) Scenario 2.

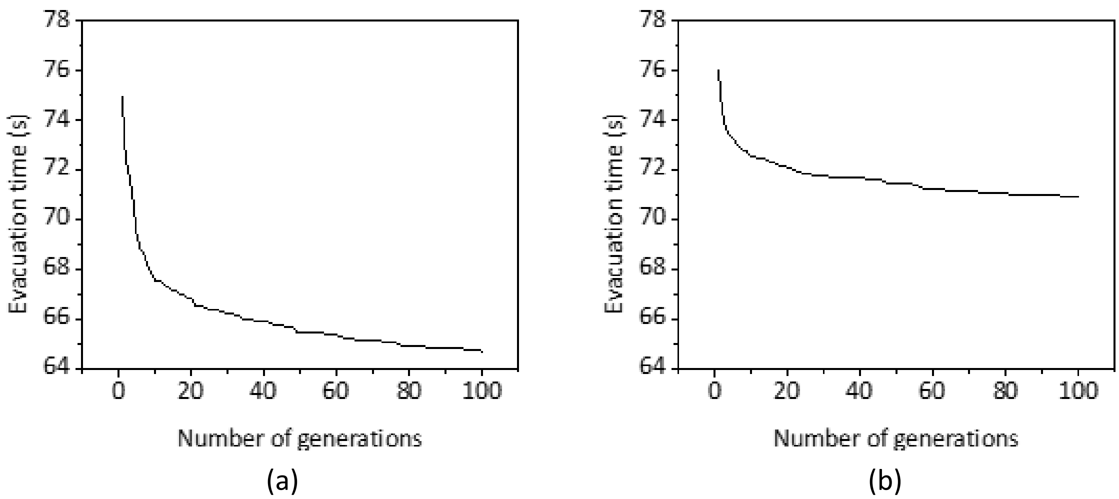

Figure 5: The evolution curves of the best fitness on the two scenarios. (a) Results of scenario 1. (b) Results of scenario 2.

To obtain the optimal segmentation strategies of the two scenarios, a number of agents are evenly deployed in the evacuation area, as shown in Fig. 4. Given a segmentation strategy, we run a simulation with the deployed agents to obtain the evacuation time which is used as the fitness of the segmentation strategy. In this way, we carry out our algorithm to search for the optimal discriminant function. Since EAs are stochastic algorithms which can provide different solutions in different runs, we perform 30 EA runs for each scenario and use the average results for analysis. The parameter setting of the proposed algorithm is as follows: population size = 20; maximum number of generations = 100; the head length of the chromosome = 5.

4.3 Simulation Results

Fig. 5 shows the evolution curves of the best fitness. The fitness values are the average results of 30 independent runs. It can be observed that the best fitness value decreases gradually in both scenarios, indicating that the best discriminant functions found by our algorithm in both scenarios are becoming better. In the first scenario, the best evacuation time found by our algorithm in the first generation is 75 seconds.

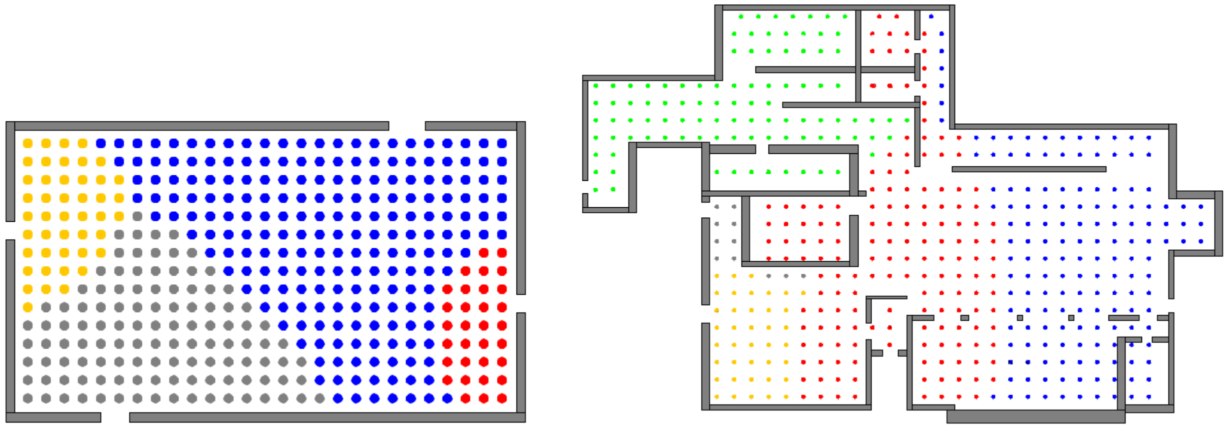

Figure 6: The best segmentation strategies found by our method on the two scenarios. (a) Results of scenario 1. (b) Results of scenario 2.

As the population is evolved over generations, the evacuation time gradually reduces to 65s. Similarly, in the second scenario, the evacuation time gradually reduces from 76 seconds to 71 seconds. These results demonstrate that the proposed algorithm is effective to reduce the evacuation time.

In the first scenario, the best discriminant function found by our algorithm is:

$$F(x) = \frac{(\frac{n}{w})}{(d*d)} * (w-d), \tag{14}$$

which leads to an evacuation time of 62 seconds. In the second scenario, the best discriminant function found by our algorithm is:

$$F(x) = (((w-d)-(d*w))-(n+d)), \tag{15}$$

which leads to an evacuation time of 69 seconds. Fig. 6 shows the segmentation results using the above two discriminant functions. It can be observed that the sub-regions assigned to each EG have irregular shapes. The sub-regions assigned to wider EGs generally have a larger size. Fig. 7 plots the execution time of two EA runs on the two scenarios respectively. It can be observed that the execution time of the two EA runs increases linearly with the number of generations, and the execution time of the EA run on the second scenario is larger than that on the first scenario. This is because that the execution time of an EA run is mainly dependent on the execution time of fitness evaluations (i.e., simulation time) and the total number of fitness evaluations (i.e., population size * number of generations), while the simulation time of the second scenario is larger than that of the first scenario. Considering that the safety planning is an off-line optimization process, the total execution time of the two EA runs (about 4500s and 12500 s respectively) are acceptable for practical application.

To test the effectiveness of the two segmentation strategies in (14) and (15), we run 30 different simulations with differing number and initial positions of agents for each scenario. The initial positions of agents follow an uniform distribution in all test instances. The 30 test instances are grouped into three categories, each containing 400, 600, and 800 agents respectively. Each category contains 10 test instances. At the beginning of the simulation, each agent first finds the nearest representative point and then uses the assigned exit gate of the representative point as its goal. We compare the simulation results using our segmentation strategy with other three evacuation strategies. The first strategy is named "Random", where agents randomly choose an EG for evacuation. The second strategy is named "Distance first" where agents always choose the nearest EG. The third strategy is the "LifeBelt" evacuation strategy (Ferscha and Zia 2009b). This method recommends EGs to agents based on three factors: the time to reach an exit gate

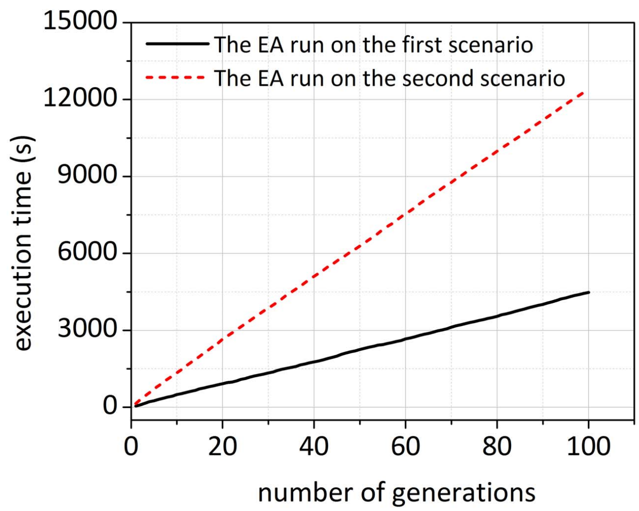

Figure 7: The execution time of two EA runs.

(TEG), the number of individuals expected in the destination EG (EP), and the number of individuals that can possibly escape through that exit per unit time (EC). The predicted evacuation time is calculated by:

$$\tau = TEG + \frac{EP}{EC}. \tag{16}$$

Since EC is in proportion to the width of the EG (w), we use the following formula to estimate τ in the simulation:

$$\tau \approx \frac{d}{s} + \frac{n}{w}. \tag{17}$$

Agents will choose the EG with the smallest τ. Table 1 shows the simulation results of the four evacuation

Table 1: Average evacuation time of different evacuation strategies.

Scenario	Number of agents	Random	Distance first	LifeBelt	The proposed method
S1	400	130.9 ± 8.0	89.7 ± 9.1	73.8 ± 3.0	**68.7** ± 3.6
	600	192.6 ± 11.4	126.6 ± 6.4	96.5 ± 3.0	**95.8** ± 2.9
	800	248.4 ± 15.8	171.2 ± 11.1	**122.8** ± 3.5	124.1 ± 5.1
S2	400	149.5 ± 20.8	81.5 ± 1.1	75.3 ± 6.7	**74.5** ± 2.9
	600	190.7 ± 22.4	85.7 ± 1.8	79.1 ± 2.6	**78.9** ± 3.3
	800	251.5 ± 26.9	104.9 ± 3.8	**85.1** ± 4.8	92.4 ± 3.5

strategies. It can be observed that the "Random" method performed the worst, and the "Distance first" method performed the second worst. Owning to the effective segmentation strategy, the evacuation time of our method is always much better than that of the "Random" and "Distance first" strategies on all test instances. Meanwhile, the proposed method offered similar performance to the "LifeBelt" method. The proposed method performed slightly better on the first two groups of instances, while the "LifeBelt" performed slightly better on the third group of instances. Note that our method is a static planning strategy; whereas the LifeBelt methods is a dynamic planning strategy which requires special devices to sensor the environment features dynamically. From this point of view, our method is effective and convenient for practical use.

5 CONCLUSION

This paper has proposed an EA-based methodology for safety planning in crowd evacuation. The key idea is to find an optimal discriminant function to divide the evacuation area into sub-regions. Each sub-region is assigned to an exit gate, and agents in a sub-region will choose the corresponding exit gate for evacuation. An EA is utilized to optimize the discriminant function so as to minimize the evacuation time of agents. The proposed method was applied to two different scenarios. The simulation results have demonstrated that the proposed method is effective to reduce the evacuation time of agents, in comparison with several other evacuation strategies.

This work is a preliminary research on safety planning in crowd evacuation using EA. A major limitation of the current work is that the the simulation results are obtained in an ideal condition that individuals are uniformly distributed in the evacuation region. Besides, the current method uses a single discriminant function to control behaviors of all agents which can reduce the flexible and robustness of the obtained solutions. Hence, an interesting future research direction is to consider the non-uniform distribution of individuals and using multiple discriminant functions to generate more robust safety planning strategy. For example, to consider the non-uniform distribution of individuals, we can first analyze some historical video data to obtain the local density of sub-regions, and then use the local density factor as a new component to construct the discriminant function. Besides, we can assign different discriminant functions to different classes of agents that have distinct features (e.g., agents in specific regions may have much slower moving speed than those in other regions). By using multi-gene chromosome representation of GEP, the proposed method is applicable to evolve multiple discriminant functions simultaneously. In this way, we can integrate the heterogeneity among agents to generate more robust and flexible safety planning strategy. In addition, by incorporating sensing devices to detect the feature values of agents, and control the behaviours of agents by calculating the discriminant function dynamically with the detected features values, the proposed methodology can be extended to provide dynamic safety planning strategy.

REFERENCES

Chen, L.-W., J.-H. Cheng, and Y.-C. Tseng. 2012. "Evacuation Time Analysis and Optimization for Distributed Emergency Guiding Based on Wireless Sensor Networks". In *2012 International Conference on Connected Vehicles and Expo (ICCVE)*, 130–135. IEEE.

Daoliang, Z., Y. Lizhong, and L. Jian. 2006. "Exit dynamics of occupant evacuation in an emergency". *Physica A: Statistical Mechanics and its Applications* 363 (2): 501–511.

Das, S., and P. N. Suganthan. 2011. "Differential evolution: A survey of the state-of-the-art". *IEEE Transactions on Evolutionary Computation* 15 (1): 4–31.

Espejo, P. G., S. Ventura, and F. Herrera. 2010. "A survey on the application of genetic programming to classification". *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 40 (2): 121–144.

Ferreira, C. 2001. "Gene expression programming: a new adaptive algorithm for solving problems". *arXiv preprint cs/0102027*.

Ferreira, C. 2006. *Gene expression programming*. Springer Berlin.

Ferscha, A., and K. Zia. 2009a. "Lifebelt: Silent directional guidance for crowd evacuation". In *International Symposium on Wearable Computers, 2009. ISWC'09.*, 19–26. IEEE.

Ferscha, A., and K. Zia. 2009b. "On the efficiency of lifebelt based crowd evacuation". In *Proceedings of the 2009 13th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications*, 13–20. IEEE Computer Society.

Ferscha, A., and K. Zia. 2010. "Lifebelt: Crowd evacuation based on vibro-tactile guidance". *Pervasive Computing, IEEE* 9 (4): 33–42.

Helbing, D., I. Farkas, and T. Vicsek. 2000. "Simulating dynamical features of escape panic". *Nature* 407 (6803): 487–490.

Hughes, R. L. 2003. "The flow of human crowds". *Annual review of fluid mechanics* 35 (1): 169–182.

Kamkarian, P., and H. Hexmoor. 2012. "Crowd evacuation for indoor public spaces using Coulomb's law". *Advances in Artificial Intelligence* 2012:4.

Liu, S., L. Yang, T. Fang, and J. Li. 2009. "Evacuation from a classroom considering the occupant density around exits". *Physica A: Statistical Mechanics and its Applications* 388 (9): 1921–1928.

Luo, L., S. Zhou, W. Cai, M. Lees, M. Low, and K. Sornum. 2011. "HumDPM: a decision process model for modeling human-like behaviors in time-critical and uncertain situations". *Transactions on computational science XII*:206–230.

Price, K. V., R. M. Storn, and J. A. Lampinen. 1997. *Differential evolution*. Springer.

Schmidt, M., and H. Lipson. 2009. "Distilling free-form natural laws from experimental data". *science* 324 (5923): 81–85.

Toyama, M. C., A. L. Bazzan, and R. Da Silva. 2006. "An agent-based simulation of pedestrian dynamics: from lane formation to auditorium evacuation". In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, 108–110. ACM.

Zarboutis, N., and N. Marmaras. 2004. "Searching efficient plans for emergency rescue through simulation: the case of a metro fire". *Cognition, Technology & Work* 6 (2): 117–126.

Zhong, J., L. Luo, W. Cai, and M. Lees. 2014. "Automatic rule identification for agent-based crowd models through gene expression programming". In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, 1125–1132. International Foundation for Autonomous Agents and Multiagent Systems.

Zhong, J., M. Shen, J. Zhang, H.-h. Chung, Y. hui Shi, and Y. Li. 2013, Aug. "A Differential Evolution Algorithm With Dual Populations for Solving Periodic Railway Timetable Scheduling Problem". *IEEE Transactions on Evolutionary Computation* 17 (4): 512–527.

Zhong, J., and J. Zhang. 2012. "SDE: a stochastic coding differential evolution for global optimization". In *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference*, 975–982. ACM.

Zhou, C., W. Xiao, T. M. Tirpak, and P. C. Nelson. 2003. "Evolving accurate and compact classification rules with gene expression programming". *IEEE Transactions on Evolutionary Computation* 7 (6): 519–531.

AUTHOR BIOGRAPHIES

JINGHUI ZHONG is a Research Fellow in the School of Computer Engineering, Nanyang Technological University. He received his B.Eng. degree, M.Eng degree, and Ph.D. degree from the School of Information Science and Technology, Sun YAT-SEN University in 2005, 2007, and 2012 respectively. His research interests include: computational intelligence, machine learning, and agent-based modeling. His email address is jinghuizhong@ntu.edu.sg.

LINBO LUO received the B.Eng.(First Class Honours) and Ph.D. degrees in computer engineering from Nanyang Technological University (NTU), Singapore, in 2005 and 2011, respectively. After his Ph.D., he was a Postdoctoral Associate in the Singapore-MIT Alliance for Research and Technology (SMART) Centre and then a Research Fellow in the Parallel and Distributed Computing Centre (PDCC) at NTU. He currently joins the School of Computer Science and Technology, Xidian University, Xi'an, China. His current research interests include: agent-based modeling, crowd simulation, serious games and large-scale complex systems. His email address is lbluo@xidian.edu.cn.

WENTONG CAI is a Professor in the School of Computer Engineering at Nanyang Technological University. He is also the Director of the Parallel and Distributed Computing Centre. He received his Ph.D. in Computer Science from University of Exeter (UK) in 1991. His expertise is mainly in the areas of Modeling and Simulation (particularly, modeling and simulation of large-scale complex systems, and system support for distributed simulation and virtual environments) and Parallel and Distributed Computing (particularly, Cloud, Grid and Cluster computing). He is an associate editor of the ACM Transactions on Modeling and Computer Simulation (TOMACS) and an editor of the Future Generation Computer Systems (FGCS). His email address is aswtcai@ntu.edu.sg.

MICHAEL LEES is an Assistant Professor in the Section Computational Science at University of Amsterdam. He completed his Ph.D. at Nottingham University, and graduated from Edinburgh Universityin July of 2001 with a joint honours Computer Science/Artificial Intelligence degree. His research interests are primarily in modeling and simulation of large scale complex systems. He is particularly interested in understanding the effect that human behavior has on such systems and the important role that individual behavioral interactions have on system level dynamics. His email address is m.h.lees@uva.nl.