Proceedings of the 2015 Winter Simulation Conference
L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, eds.

CROWD EVACUATION PLANNING USING CARTESIAN GENETIC PROGRAMMING AND AGENT-BASED CROWD MODELING

Jinghui Zhong

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

Wentong Cai

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

Linbo Luo

School of Cyber Engineering
Xidian University
Xi'an, 710071, CHINA

ABSTRACT

This paper proposes a new evolutionary algorithm-based methodology for optimal crowd evacuation planning. In the proposed methodology, a heuristic-based evacuation scheme is firstly introduced. The key idea is to divide the region into a set of sub-regions and use a heuristic rule to dynamically recommend an exit to agents in each sub-region. Then, an evolutionary framework based on the Cartesian Genetic Programming algorithm and an agent-based crowd simulation model is developed to search for the optimal heuristic rule. By considering dynamic environment features to construct the heuristic rule and using multiple scenarios for training, the proposed methodology aims to find generic and efficient heuristic rules that perform well on different scenarios. The proposed methodology is applied to guide people's evacuation behaviors in six different scenarios. The simulation results demonstrate that the heuristic rule offered by the proposed method is effective to reduce the crowd evacuation time on different scenarios.

1 INTRODUCTION

In recent years, crowd evacuation planning has become an important research field that has drawn increasing attention from both academic researchers and governments (Ferscha and Zia 2009, McGrattan et al. 2010, Zhong et al. 2014). The objective of crowd evacuation planning is to develop strategies to direct people during evacuation so as to reduce the evacuation time. With the world getting more and more crowded, the number of accidents (e.g., a fire) happened in crowded scenarios such as theater, airport and shopping mall is increasing. If people fail to evacuate these places in time when an accident happens, they may get injured or even loss their lives. Therefore, designing an efficient crowd evacuation planning strategy is of great importance to the public safety.

Over the past decade, a number of crowd evacuation modeling tools have been developed to study the crowd evacuation behaviors (Zheng et al. 2009). The commonly used models include the Cellular Automaton (CA) Model (Zhao et al. 2006, Liu et al. 2009, Ferscha and Zia 2010), the Social Force Model (Helbing et al. 2000), and the Agent-Based Model (ABM) (Zarboutis and Marmaras 2004, Toyama et al. 2006, Luo et al. 2008, McGrattan et al. 2010, Zhong et al. 2014). Among them, the ABM is perhaps the most popular and effective one. This is owning to that the ABM is capable of flexibly modeling realistic individuals, by considering various behavioral factors such as social factors and psychological factors (Zhou et al. 2010). Furthermore, the significant increase of computing power makes it possible to efficiently simulate large scale crowd dynamics using ABM.

With the aids of the crowd evacuation modeling tools, efforts have also been made to optimize people's evacuation behaviours so as to reduce the evacuation time. For example, Liu et al. (2009) studied the evacuation behaviours in a classroom with obstacles by using a CA model. Their simulation results show that the density around exits is an important factor that has influences on people's evacuation behaviors. Chen et al. (2012) developed a load-balancing framework to guide people to evacuate a building. Their method requires deploying a sensor network to identify hazardous regions dynamically. Kamkarian and Hexmoor (2012) developed an evacuation tool for guiding people out of a public building based on the combined Coulomb's electrical law and graph theory. Ferscha and Zia (2009, 2010) developed a wearable device named LifeBelt to help people evacuate from emergency situations. The LifeBelt device can recommend exits to individuals based on the dynamically sensed environment situations. In the above methods, evacuation rules are manually defined based on domain knowledge and may lead to local optimal. To explore better strategies for crowd evacuation, evolutionary algorithms (EAs) have also been used recently. Abdelghany et al. (2014) proposed a simulation-optimization modeling framework for the evacuation of large-scale pedestrian facilities with multiple exit gates. They used a genetic algorithm (GA) to search for the optimal evacuation plan, and a Cellular Automata (CA) for fitness evaluation. Similarly, Zhong et al. (2014) have developed an EA-based methodology (denoted as "Static" method in this paper). The key idea is to use a gene expression programming to find a discriminant function which is used to segment the region into several sub-regions. People in the same sub-region will move towards the same exit. However, the solutions provided by the above two methods are not generic and only suitable for the specific scenario. When a new scenario is given or some changes take place in the existing scenario, they must be re-run again to get a new solution. This can be time-comsuing and impractical for real applications.

This paper follows the research direction of the "Static" method and develops a new EA-based methodology that can provide generic evacuation strategies. Specifically, there are three major drawbacks of the "Static" method. First, the evacuation strategies provided by the "Static" method are not generic enough. Second, the "Static" method assumes that people in the region are evenly distributed. This is often not true in real situations. Thirdly, the "Static" method does not consider the dynamic environment features in the evacuation planning process. This makes its solution less efficient. This paper proposes a new EA-based methodology by addressing all the above three drawbacks.

In the proposed methodology, a heuristic-based evacuation scheme is introduced at first. The key idea is to evenly divide the entire region into a set of sub-regions. For each sub-region, a heuristic rule is utilized to determine the optimal exit for agents in the sub-region. The heuristic rule is capable of flexibly guiding people's evacuation behaviours based on the latest situations, as it considers the dynamic environment features such as the number of people surrounding the exits to recommend exits. The heuristic rule can work effectively without assuming that the people are evenly distributed in the region. Based on the heuristic-based evacuation scheme, the crowd evacuation planning problem is then converted to finding the optimal heuristic rule that minimizes the total evacuation time. To solve this problem, a new evolutionary framework is developed. The proposed evolutionary framework adopts the Cartesian Genetic Programming (CGP) (Miller 2011) as the base algorithm to search for the optimal solution. The CGP is an evolution computation technique which has been proven to be effective for automating the design of computer programs (e.g., mathematical formulas) that solve user defined task. The distinct feature of CGP is that it represents a program by a directed graph, which enables it to reuse subgraphs that previously exist. This feature is useful for improving the search efficiency. Due to its fast search efficiency and ease of implementation, CGP has attracted increasing attentions over the years and has recently become a popular choice for automatic programming. Hence, this paper adopts the CGP to search for promising heuristic rules. To evaluate the fitness values of solutions given by the CGP, an agent-based crowd model is further developed. In addition, multiple scenarios with different features are used for training so as to gain general heuristic rule that can work effectively in different new scenarios. To test its effectiveness, the proposed methodology is applied to guide people's evacuation behaviors in six different scenarios. The simulation results show that the proposed method is effective to provide generic and promising heuristic rules that work better than the "Static" method and the other common evacuation strategies.

The outline of the paper is as follows. Section 2 describes the problem definition. Section 3 presents the proposed methodology, and the simulation studies are given in Section 4. Finally, Section 5 draws the conclusion.

2 PROBLEM FORMULATION

In this section, a heuristic-based evacuation scheme is proposed. Then based on the heuristic-based evacuation scheme, the crowd evacuation planning problem is formulated as an optimization problem. In the heuristic-based evacuation scheme, the entire region is divided into a set of sub-regions. During the evacuation process, a heuristic rule (denoted as H) is used to periodically recommend exits to each sub-region according to the real situations. Agents in a sub-region will move towards the corresponding exit assigned to the sub-region. Note that the real situations are changing during the evacuation process. Hence, the exit assigned to each sub-region may change dynamically. Figure 1 shows an example evacuation planning result at a specific time step. In this example, the entire region is divided into 35 sub-regions. The number of sub-regions assigned to the 1st, 2nd, 3rd, and 4th exits are 8, 8, 12, and 7 respectively.

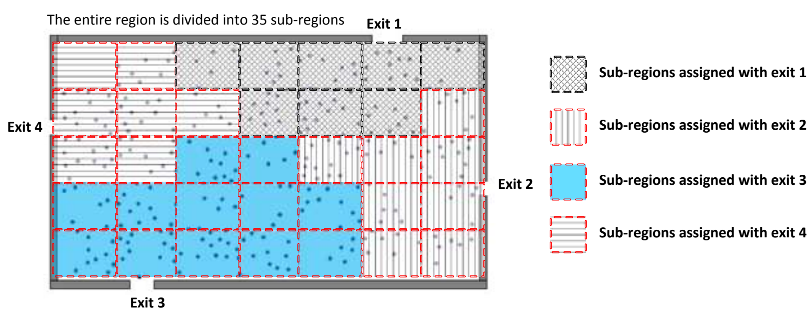

Figure 1: An example evacuation planning result at a specific time step.

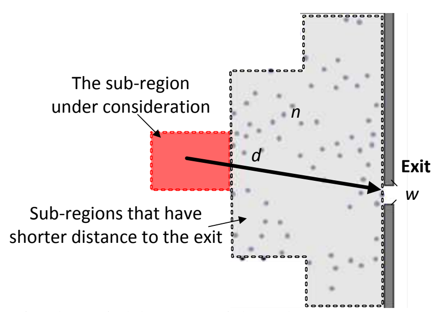

Figure 2: Terminal variables considered to construct heuristic rule.

The heuristic rule H is a mathematical formula that consists of functions (e.g., $+,-,*,/$) and terminal variables (i.e., environment features such as the distance to the exit and the width of the exit). The heuristic rule is used to calculate the score values of exits for each sub-region. The exit with the smallest score value is selected as the recommended exit for the sub-region. The function set and terminal set are manually defined in advance. Note that if problem-specific knowledge is available, we can define specific functions and terminals to reduce the search space and improve the search efficiency. In this study, we empirically choose four basic functions (i.e., $+,-,*,/$) and three terminal variables as building blocks to construct H. As shown in Figure 2, the three terminal variables are d,w, and n. The variable d is the distance to the exit, w is the width of the exit, and n is the number of individuals in those sub-regions that have shorter distance to the exit than the sub-region under considered. For example, based on these function set and terminal set, if we want to guide agents to select the nearest exit, we can define the heuristic rule as : $H(d,w,n) = d$. Based on the above definitions, the problem of optimizing the crowd evacuation strategy is converted to finding the optimal H that minimizes the evacuation time:

$$H^* = \arg\min_{H} ET(H) \tag{1}$$

where $ET(H)$ is the total evacuation time and H is the heuristic rule used to recommend exits to sub-regions.

3 THE PROPOSED METHOD

To solve the optimization problem defined in (1), this section proposes an evolutionary methodology with the Cartesian Genetic Programming (CGP) and an Agent-Based crowd model. The proposed method is developed based on our early work in (Zhong et al. 2014), where a genetic programming variant was adopted to evolve exit selection rules for agent-based crowd modeling. The main differences between this work and our early work are two-fold. First, the motivation and objective are different. Our early work focuses on identifying a specific exit selection rule for each agent so that the simulation can match a desired objective behavior, while this work focuses on finding a generic behavior rule for agents in each sub-region to minimize the crowd evacuation time. Second, the crowd evacuation models developed are different. In our early work, the exit selection rule is used directly by agents to determine their destinations, while in this work, the rule is used to determine the destinations assigned to sub-regions. By assigning rule to sub-regions, we can deploy marshals to sub-regions to control the crowd. This does not require each agent equipped with a device to receive commands (or to calculate the rule) for evacuation. (Often, the number of sub-regions are much smaller than the number of agents.) Thus, the method proposed in this paper could be more convenient for practical applications. In the following parts, the general search mechanism of the proposed method is presented first. Then an agent-based simulation model for fitness evaluation is given, followed by the implementation of the CGP for evolving the optimal heuristic rule.

3.1 The General Search Mechanism

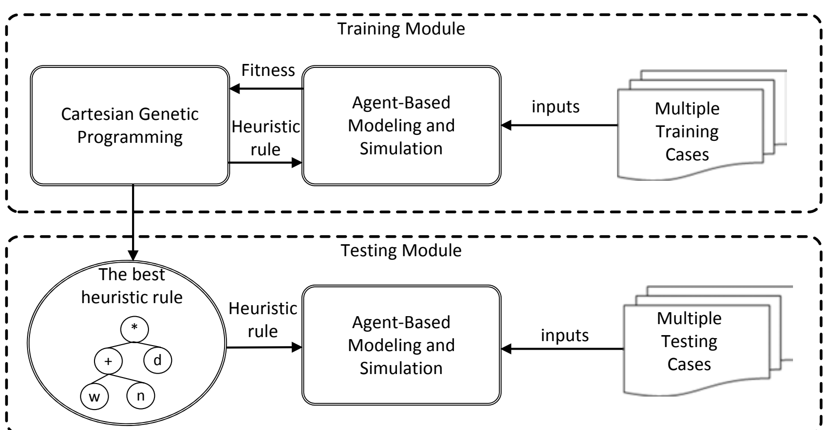

Figure 3: The general search mechanism.

The general search mechanism is illustrated in Figure 3. In the proposed framework, multiple training cases that have different features (e.g., the number, positions and widths of gates) are used for training, so as to obtain a general heuristic rule that performs well in different scenarios. Given a heuristic rule, an agent-based crowd model is proposed to evaluate its fitness. The proposed agent-based crowd model contains two layers. The top layer determines the destinations of agents based on the set of sub-regions and the heuristic rule, and the second layer determines the microscopic collision avoidance behavior and drives the agents move towards their destinations. To evaluate the fitness of a given heuristic rule, the agent-based crowd model configured with the given heuristic rule is performed to simulate the crowd behaviors of all training cases. The results of all training cases (as will be described in (3)) are used to evaluate the fitness of the heuristic rule. Based on this fitness evaluation method, the CGP is used to search for an optimal heuristic rule. In the CGP, a population of random heuristic rules are generated as the initial population. Then genetic operators such as mutation and selection are used to iteratively evolve the heuristic rules. During the evolutionary process, the proposed agent-based crowd model is performed dynamically to evaluate the fitness values of the heuristic rules given by the CGP. The CGP will output the best-so-far heuristic rule when the training termination condition is met. Then the performance of the best-so-far heuristic rule is further tested based on multiple testing cases.

3.2 The Agent-based Crowd Model for Fitness Evaluation

To evaluate the fitness of a heuristic rule, we need to develop a crowd simulation model which is used to estimate the total evacuation time of the crowd. In this sub-section, we develop an agent-based crowd simulation model based on the commonly used social-force model and the proposed heuristic-based crowd evacuation scheme. In general, as shown in Figure 4, the proposed agent-based crowd model consists of two layers. The top layer determines an exit for evacuation. In this layer, each sub-region is dynamically assigned with an exit by using the heuristic rule. Then the evacuation exit for an agent can be set to be the one assigned to the sub-region that contains the agent.

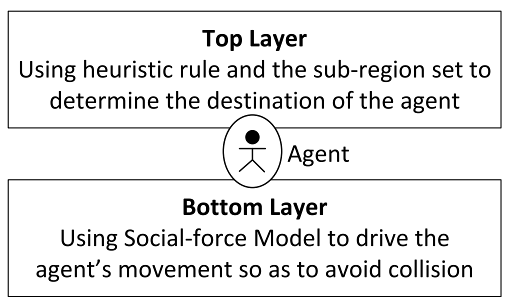

Figure 4: The proposed ABM crowd simulation model.

Once the destination (i.e., the exit) is determined, the bottom layer adopts the social-force model to drive the agent towards the destination so that agents will not collide with other agents and obstacles. The social-force model is a famous crowd simulation model that was proposed by Helbing et al. (1995). In the social-force model, the motions of agents are guided by virtual forces that can be expressed as:

$$f_i = f_{i_0} + \sum_{j(j \neq i)} f_{ij} + \sum_{w} f_{iw} \tag{2}$$

where f_{i_o} is the attractive force from the goal, f_{ij} is the repulsive force from other agents, and f_{iw} is the repulsive force from the static obstacles such as walls. The readers are referred to (Helbing et al. 2000) for detailed implementation of the social-force model.

In the simulation, the bottom layer is performed every simulation tick to update the positions and velocities of agents, while the top layer is performed every ω simulation ticks to periodically update the destinations of agents. In general, ω needs to be larger than the time required to get real-time update about the environment. However, its value cannot be too large; otherwise the changes of environment may not be included in determining agent's behavior and thus reducing the evacuation efficiency. In order to reduce the computational time of fitness evaluation, we fix the maximum simulation ticks to be T_{max}, and use the following equation to evaluate the fitness of a heuristic rule:

$$F(H) = \sum_{i=1}^{K} [max\{T_i(H), U * I(N_i(H) < N_i^*)\} - N_i(H)] \tag{3}$$

where K is the total number of training cases; $T_i(H)$ is the total evacuation time of the ith training case under the heuristic rule H; $U = 10^8$ is the big value to be used if some people were not evacuated in the ith training case; N_i^* is the total number of people to be evacuated in the ith training case; $N_i(H)$ is the total number of people actually evacuated in the ith training case under the heuristic rule H; $I(N_i(H) < N_i^*)$ is the indicator function for the event $N_i(H) < N_i^*$ so that $I(N_i(H) < N_i^*) = 1$ if the event occurrent (i.e., some people were not evacuated in ith training case) and 0 otherwise. Note that the smaller the value, the better the fitness. The fitness function is defined in this way since if all the people escaped before end of simulation, smaller T_i means more efficient evacuation strategy. Otherwise, if not all the people managed to escape before the end of simulation, a larger N_i indicates a better evacuation strategy.

3.3 The CGP for Evolving Heuristic Rules

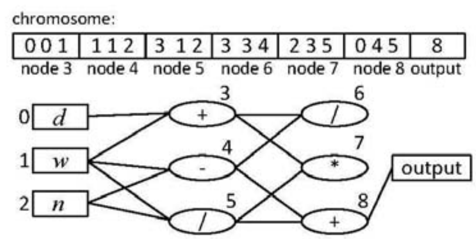

Figure 5: Traditional chromosome representation in CGP.

The fitness function defined in (3) is integrated with the CGP to search for an optimal heuristic rule. In the CGP, the chromosome is comprised of two parts: function nodes and output nodes. Each function node represents a particular function and it is encoded by a number of genes. The first gene encodes the function type (e.g., "+"), and the remaining genes encode the input sources of the function. The input sources of a function node can be a previous node or a terminal variable (e.g., d, w, and n). The number of input sources is determined by the function type of the node. For example, "+" has two input sources. In the CGP, the terminal variables and the function nodes are labelled by integers. Specifically, 0 to $C-1$ represent the C terminal variables, while C to $C+L-1$ represent the L function nodes in the chromosome. The output genes are integers that represent the sources of the corresponding outputs. In this paper, we consider that each chromosome contains a single output. Hence, each chromosome can be represented as:

$$[\varphi_1, t_{1,1}, ..., t_{1,M}, ..., \varphi_L, t_{L,1}, ..., t_{L,M}, o] \tag{4}$$

where L is the number of function nodes in each chromosome, M is the maximum number of input sources among all functions, φ_i represents the function type of the ith function node, $t_{j,k}$ represents the kth input source of the jth function node, and o represents the source index of the output. Figure 5 shows an example of the CGP chromosome which contains three program inputs, six function nodes, and one output node. Denote the input variables and functions as : $d \to 0, w \to 1, n \to 2, + \to 0, - \to 1, * \to 2, / \to 3$. The first three nodes encode the three terminals d, w, and n, respectively. The values of these three nodes are fixed and are omitted in the chromosome. The first node in the chromosome (i.e., node 3) contains three genes: $\{0,0,1\}$. The first gene "0" encodes a function type of "+". The second and third genes encode two inputs of d and w. Hence node 3 encodes a sub-function of $d+w$. The output of node 3 can be an input of anther node (e.g., node 7). In this way, the entire chromosome can be decoded as $H = (w-n) + \frac{w}{n}$.

Based on the above chromosome representation, the standard CGP adopts the $1+\lambda$ evolution strategy (ES) (Beyer and Schwefel 2002) to evolve the chromosomes, as illustrated in Algorithm 1. First of all,

λ random chromosomes are generated as the initial population. The value of the ith gene in each initial chromosome is set by:

$$v_i = \begin{cases} rand_i(0, C+L-1), \text{if } v_i \in o \\ rand_i(0, I-1), \text{if } v_i \in \{\varphi_1, ..., \varphi_L\} \\ rand_i(0, C + \lfloor \frac{i}{(M+1)} \rfloor - 1), \text{otherwise} \end{cases} \tag{5}$$

where $rand_i(a,b)$ returns a random integer within $[a,b]$, M is the maximum number of input sources among all functions, and I is the number of functions in the function set. Then in the second step, a mutation operation is performed to mutate the best individual (denoted as p_{best}) in the population so as to generate λ new individuals. When generating a new individual, the ith gene's value is set by:

$$v_i' = \begin{cases} v_i, \text{if } rand_r(0,1) \geq pm \\ \text{set by Eq. (5)}, \ otherwise \end{cases} \tag{6}$$

where v_i is the corresponding value in the parent individual, $rand_r(a,b)$ returns a random floating-point number within (a,b), and pm is the mutation rate. In the third step, the λ new individuals together with the p_{best} are ranked and the best one is selected as the new p_{best} for the next generation. The second step and the third step are repeated until the maximum number of generations is reached.

Algorithm 1: The Evolution Procedure of CGP based on 1 + λ ES.

Begin:
for $i = 1$ **to** λ **do**
    Randomly generate the ith individual by (5)
$p_{best} \leftarrow$ the best individual in the population
$generation$ = 1
while $generation <$ *the maximum number of generations* **do**
    for $i = 1$ **to** λ **do**
        generate the ith offspring by mutating p_{best}
        using agent-based simulation and multiple training cases to evaluate the fitness of the ith offspring
    $p_{best} \leftarrow$ the best one of the λ offspring and p_{best}
    $generation = generation + 1$
End

Once the best heuristic rule is obtained by the CGP, it then can be used for evacuation planning as described in Section 2. That is, for each sub-region, the best heuristic rule is used periodically to calculate the score value of each exit, where the inputs of the heuristic rule (i.e., d, w, and n) are obtained from the real situations. Then the exit with the smallest value is selected as the recommended exit for the sub-region. Agents in the sub-region will move towards the recommended exit to evacuate the region.

4 SIMULATION STUDIES

This section designs simulations to test the effectiveness of the the proposed method. First, six scenarios with different features are designed to generate training and testing cases. Each training case or testing case is a crowd evacuation scenario with a specific crowd composition setting (i.e., the number of individuals). Then the simulation settings and the evacuation strategies considered for comparison are described. Finally, the comparison results are discussed.

4.1 Simulation Settings

We design multiple training and testing cases based on six scenarios with different features, as shown in Figure 6. The first three scenarios and the fifth scenarios are rectangular rooms with different numbers of exits. The widths and positions of the exits in the four scenarios are also different. The fourth and the sixth scenario are a bit more complicated than the other scenarios, because they contain two inner rooms, several inner gates and four exits respectively. Specifically, the first four scenarios are used to generate training cases. Each scenario is used to generate three training cases, where the numbers of individuals are set to be 100, 200, and 400 respectively. Hence there are totally twelve training cases. Similarly, we generate twenty-four testing cases based on all six scenarios. Each scenario is used to generate four testing cases, where the numbers of individuals are set to be 180, 280, 550, and 1000 respectively. Unlike our previous work, which assumes that the individuals are evenly deployed in the environment, the individuals in each training or testing case are randomly deployed in this study. It should be noted that some testing cases are generated based on the last two scenarios and we use the heuristic rule trained from the previous four scenarios to test all testing cases. By doing so, we aim to test the generality of the heuristic rules offered by the proposed method. In the simulation, as suggested in (Helbing et al. 2000), the parameters of the social-force model are set to be: $A = 1000, B = 0.08, k_1 = 120000, k_2 = 240000, v_0 = 2m/s, v_{max} = 3m/s, \tau = 0.5$. Other parameters of the simulation model are set to be: $\omega = 16, T_{max} = 300 \times \omega$, each simulation tick = 1/16 second. The parameters of the CGP are set to be: $\lambda = 10$, $pm = 0.02$, $L = 50$, $M = 2$, and the maximum number of generations is 200. Since the CGP is a stochastic optimization algorithm that can offer different solutions in different runs, we perform the CGP for 10 independent runs with different random seeds, and the average results are used for analysis.

To investigate the effectiveness of the heuristic rules found by our method, we compare them with three other evacuation strategies. The first is the distance first strategy (denoted as DF), where individuals always choose the nearest exit. The second is the "LifeBelt" method (Ferscha and Zia 2009). In the "LifeBelt" method, individuals choose an exit based on three factors: the time to reach an exit gate (TEG), the number of individuals expected in the destination exit gate (EP), and the number of individuals that can possibly escape through that exit per unit time (EC). Based on these three factors, the evacuation time is estimated as $\tau = TEG + \frac{EP}{EC}$. As EC is in proportion to the width of the EG (w), we approximately estimate τ by $\tau \approx \frac{d}{s} + \frac{n}{w}$, where s is the speed of agents, which is set to be 2m/s in this paper. Individuals will always choose the exit with the smallest τ. The third method is the static planning strategy proposed in (Zhong et al. 2014) (denoted as "Static"). In this method, the entire region is divided into several sub-regions by using a discriminant function. The segmentation result is obtained before the evacuation and is fixed during the evacuation. In the simulation studies, the best discriminant function found by the "Static" method on the third scenario is used for comparison. The discriminant function is expressed as $\frac{(\frac{n}{w})}{(d*d)} * (w - d)$.

4.2 Simulation Results

Figure 7 shows the evolution curve of the best fitness value found by the CGP. The fitness values are the average results of 10 independent runs. As the fitness value is much smaller than 1×10^8, all individuals successfully evacuated the region by using the heuristic rules. Furthermore, the best fitness value decreases gradually as the number of generations increases. This indicates that the best heuristic rule found by the CGP becomes more and more effective to reduce the total evacuation time. The evolution trend indicates that the heuristic rule can become even better if the number of generations continues to increase.

Table 1 shows the 10 heuristic rules found by the CGP in 10 independent runs. It can be observed that some rules are easy to interpret (e.g., the 5th rule), while some others (e.g., the 10th rule) are quite complex and difficult to interpret. Take the 5th rule for example. The expression of this heuristic rule contains all three factors (i.e., d,w, and n) and is quite similar to the LifeBelt strategy. By calculating the first-order partial differential equation of the expression, we can find that using this rule the individuals are more likely to select exits with smaller distance, larger width and fewer surrounding individuals. Note that

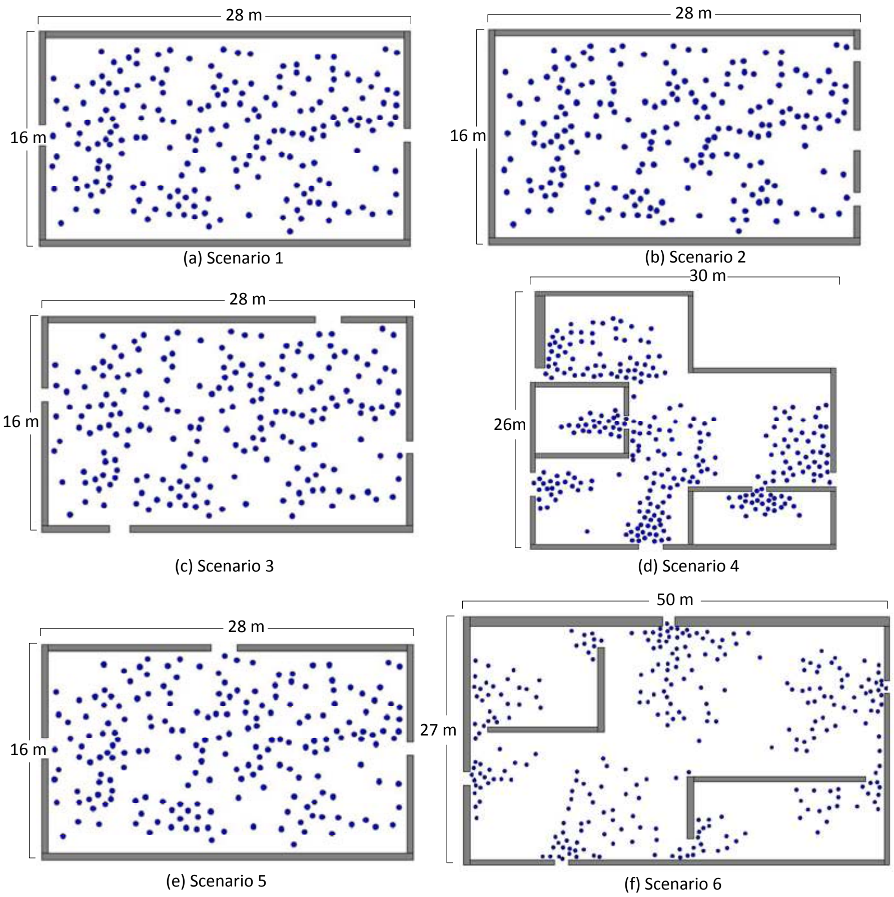

Figure 6: Six scenarios for generating training and testing cases.

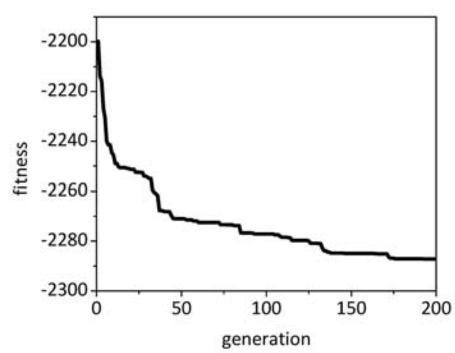

Figure 7: Evolution curve of the best fitness value.

the proposed method is a stochastic optimization algorithm that can provide different solutions in different runs, in terms of fitness and complexity. In practical applications, the decision makers can perform the proposed algorithm for multiple runs and choose the best final solution according to their preferences.

In the following parts, we choose the first heuristic rule in Table 1 as an example to investigate its effectiveness. This heuristic rule is applied to simulate the evacuation behaviours in the 24 testing cases. For each testing case, we run 10 different simulations, where the initial positions of agents in the same testing case are set differently. The initial positions of individuals are randomly generated in all simulations. The average evacuation time of the 10 simulations is then used as the evacuation time as the corresponding

Table 1: The 10 heuristic rules found by the CGP in 10 independent runs.

Run index	The best heuristic rule found by the CGP
1	$((((d/w)*(n+w))/w)+(d-(n-((n/d)+((n/d)*(n/d))))))$
2	$((((n+w)/(n+w))-((d+w)/d))-(((n+w)*(((d+w)/d)/(n+w)))/(n+w)))$
3	$((((n/w)+d)+(((n/w)+d)/((((n/w)-w)+((n/w)-w))*d)))*(((n/w)+d)+(((n+d)/w)+$ $((n/w)+d))))$
4	$(((n/((d/w)+w))*((((d/w)+w)*(d/w))/w))+(d+((d/w)+w)))$
5	$(d-(w-(n/w)))$
6	$(((n-(d*d))+((w+w)+(w-(w+w))))+(d/(w/((d*d)/(w+w)))))$
7	$((n-(w-d))/(((w-d)+((d+n)*w))/(n-(w-d))))$
8	$(((n/(w/d))/((d/n)+(d-w)))+(d/(w/d)))$
9	$((((((d+n)/(d+n))/(d/n))*w)+((d/(w/((d+n)/w)))-((d+n)/(d+n))))*((d+n)/(d+n)))$
10	$(((w+(((n/w)/((d*w)/(n/w)))/(d*w)))*(d/(((n/w)/((d*w)/(n/w)))/((n/w)/(d*w)))))+$ $((d+(n/w))+((n/w)/(d*w))))$

Table 2: Average evacuation time of different evacuation strategies.

	N = 180				N = 280				N = 550				N = 1000			
	DF	LB	ST	HR	DF	LB	ST	HR	DF	LB	ST	HR	DF	LB	ST	HR
S1	47.13	40.97	48.88	**40.67**	68.09	55.58	66.75	**54.86**	127.05	98.66	119.80	**96.99**	234.10	**169.16**	210.79	177.10
S2	43.86	41.71	60.20	**41.58**	56.24	**50.09**	79.53	50.23	94.11	75.58	111.50	**75.49**	149.58	**127.50**	161.60	128.4
S3	38.97	33.62	**31.03**	32.94	50.80	44.25	**40.08**	42.5	78.43	64.80	66.21	**64.56**	127.02	**97.43**	114.79	97.72
S4	37.50	**34.30**	44.60	34.85	47.38	42.13	50.25	**42.08**	75.75	62.5	65.5	**61.85**	127.28	92.90	94.13	**92.18**
S5	33.89	30.25	31.68	**27.36**	49.67	39.46	41.74	**38.38**	85.96	59.52	71.89	**57.94**	156.83	96.01	123.08	**95.59**
S6	39.95	**40.625**	48.38	40.88	49.35	47.78	57.83	**47.13**	74.08	71.48	84.3	**69.3**	121.05	116.13	130.85	**111.13**

"Si" represents the *i*th scenario, "DF" represents the distance first strategy, "LB" represents the LifeBelt strategy, "ST" represents the "Static" method, and "HR" represents the heuristic rule offered by the proposed method. The values are in seconds.

testing case. Table 2 shows the results of the four evacuation strategies. It can be observed that the "DF" method performed the worst. The "Static" method performed the best on the first two testing cases of scenario 3. This is because that the discriminant function is specifically obtained based on the similar settings. However, as the number of individuals increases and the scenario changes, the "Static" method degrades significantly. This indicates that the solution provided by the "Static" method is not general enough. The "LifeBelt" method performed the best on six of the twenty-four testing cases, owning to the fact that it takes into account the dynamic environment features to guide people's evacuation behaviors. Meanwhile, the proposed method performed the best on sixteen out of the twenty-four cases. In addition, the performance of the proposed method on the remaining eight testing cases are also very competitive. The above results demonstrate that the proposed method is very effective to reduce the evacuation time. The heuristic rule offered by our method is general as it works well on different new scenarios.

5 CONCLUSIONS

This paper has proposed an EA-based methodology to generate optimal crowd evacuation planning strategy. In the proposed methodology, the entire region for evacuation is divided into a set of sub-regions. A heuristic rule is used to dynamically recommend an exit to agents in each sub-region. To search for an optimal heuristic rule that minimizes the evacuation time, an evolutionary framework based on the CGP and an agent-based crowd model is developed. The proposed method was tested on six scenarios with different features. The simulation results have demonstrated that the proposed method is effective to generate generic and effective heuristic rule to minimize the crowd evacuation time.

The proposed method has potential to be applied in real applications. The key issues are how to gain the dynamic environment features (e.g., n) and how to make agents move towards their recommended exits. A feasible approach is to utilize the LifeBelt device. The LifeBelt device can receive commands from a global center and recommend the next moving step to individuals using a special component. Therefore, the global center can use the heuristic rule to calculate the recommended exit for each sub-region and then send this information to the corresponding LifeBelt devices to guide agents' behaviors. The second possible method is to deploy a marshal to each sub-region. The marshals have special devices to receive commands from the global center, and guide agents' behaviors according to the received commands.

The proposed framework is flexible, as some of its components can be modified to get more convincing results. For example, we can use other crowd evacuation models and other EAs such as the SL-GEP (Zhong et al. 2015) to get better heuristic rules more efficiently. However, the current work still has several issues which limit its use in practical applications. The first issue is that it requires frequently assessing the real situations (i.e., using the heuristic rule to evaluate exits) to recommend exit to each sub-region. How frequently the situation is assessed may affect the evaluation results. Thus, studying the impact of situation assessment frequency on evacuation efficiency and considering how the value could be set in practical applications could be a direction for future work. The second issue is that how the environment is divided may have an effect on the rules. In addition, if we deploy marshals to control the crowd, it is desirable to divide the region into larger sub-regions so that fewer marshals are required to control the crowd. Therefore, extending the proposed method by considering the strategy to divide the environment and the deployment of marshals could be another direction for future work.

ACKNOWLEDGMENTS

The research reported in this paper is financially supported by the Tier 1 Academic Research Fund (AcRF) under project Number RG23/14. Linbo Luo is partially supported by Fundamental Research Funds for the Central Universities (Project JB150305).

REFERENCES

Abdelghany, A., K. Abdelghany, H. Mahmassani, and W. Alhalabi. 2014. "Modeling Framework for Optimal Evacuation of Large-scale Crowded Pedestrian Facilities". *European Journal of Operational Research* 237 (3): 1105–1118.

Beyer, H.-G., and H.-P. Schwefel. 2002. "Evolution Strategies–A Comprehensive Introduction". *Natural Computing* 1 (1): 3–52.

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 *IEEE International Conference on Computer Vision 2012*, 130–135. IEEE.

Ferscha, A., and K. Zia. 2009. "On the Efficiency of Lifebelt Based Crowd Evacuation". In *IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications 2009*, 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.

Helbing, D., and P. Molnar. 1995. "Social Force Model for Pedestrian Dynamics". *Physical review E* 51 (5): 4282.

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. Y. H. Low, F. Tian, Y. Wang, X. Xiao, and D. Chen. 2008. "Agent-Based Human Behavior Modeling for Crowd Simulation". *Computer Animation and Virtual Worlds* 19 (3-4): 271–281.

McGrattan, K. B., S. Hostikka, and J. E. Floyd. 2010. "Fire Dynamics Simulator (Version 5), Users Guide". *NIST Special Publication* 1019 (5): 1–186.

Miller, J. F. 2011. *Cartesian Genetic Programming*. Springer.

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 2006 International Conference on Autonomous Agents and Multi-Agent 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.

Zhao, D., Y. Lizhong, and J. Li. 2006. "Exit Dynamics of Occupant Evacuation in an Emergency". *Physica A: Statistical Mechanics and its Applications* 363 (2): 501–511.

Zheng, X., T. Zhong, and M. Liu. 2009. "Modeling Crowd Evacuation of a Building Based on Seven Methodological Approaches". *Building and Environment* 44 (3): 437–445.

Zhong, J., W. Cai, L. Luo, and M. Lees. 2014. "EA-Based Evacuation Planning Using Agent-Based Crowd Simulation". In *Proceedings of the 2014 Winter Simulation Conference*, 395–406. IEEE Press.

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., Y.-S. Ong, and W. Cai. 2015. "Self-Learning Gene Expression Programming". *IEEE Transactions on Evolutionary Computation* doi:10.1109/TEVC.2015.2424410.

Zhou, S., D. Chen, W. Cai, L. Luo, M. Y. H. Low, F. Tian, V. S.-H. Tay, D. W. S. Ong, and B. D. Hamilton. 2010. "Crowd Modeling and Simulation Technologies". *ACM Transactions on Modeling and Computer Simulation* 20 (4): 1–34.

AUTHOR BIOGRAPHIES

JINGHUI ZHONG is a Research Fellow in the School of Computer Engineering, Nanyang Technological University (NTU), Singapore. 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, China, in 2005, 2007, and 2012 respectively. His research interests include: Computational Intelligence, Machine Learning, and Agent-based Modeling and Simulation. His email address is jinghuizhong@ntu.edu.sg.

WENTONG CAI is a Professor in the School of Computer Engineering at Nanyang Technological University (NTU), Singapore. 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.

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 is currently an Associate Professor in the School of Cyber Engineering, Xidian University, 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.