Article

An Agent-Based Simulation of Deep Foundation Pit Emergency Evacuation Modeling in the Presence of Collapse Disaster

Weilong Yang, Yue Hu, Cong Hu and Mei Yang *

College of System Engineering, National University of Defense Technology, Changsha 410073, China; yangweilong09@nudt.edu.cn (W.Y.); huyue11@nudt.edu.cn (Y.H.); hccz95@163.com (C.H.)

* Correspondence: yangmei@nudt.edu.cn; Tel.: +86-0731-8457-6272

Received: 26 September 2018; Accepted: 17 October 2018; Published: 2 November 2018

Abstract: With the gradual expansion of high buildings and underground spaces, deep foundation pits have been widely used in these engineering projects, but if they are not well-designed, safety problems occur. Proper deep foundation pit design requires proper exit distribution. However, calculating an adequate number of exit distributions for evaluation is difficult due to the numerous influential factors existing in the deep foundation pit environment. To this end, this paper presents a prototype of a decision-making system that uses agent-based modeling to simulate deep foundation pit evacuation in the presence of collapse disaster. By modeling the collapse occurrence process and agent escape process, an agent-based evacuation model is built, and a modified simulation-based particle swarm optimization algorithm is used to solve the optimization problem of exit distribution. Extensive experiments are conducted to verify the system, and the results show that the system provides a feasible framework for deep foundation pit evacuation.

Keywords: emergency evacuation; deep foundation pit; agent-based simulation; collapse disaster; particle swarm optimization; weighted Voronoi diagram

1. Introduction

With the development of high buildings and the spreading use of underground space, deep foundation pit projects have been built to complete the groundwork. A deep foundation pit usually refers to a pit that is deeper than five meters and with large scale. Once certain disaster occurs in the pit, it will spread rapidly, result in an emergency, and cause serious injury to the workers inside. For example, in 2008, a collapse disaster happened at the Hangzhou metro building site, which caused 21 worker deaths and four worker injuries [1]. A proper exit distribution plan designed for early escape is an important safety measure to protect workers from deep foundation pit collapse.

Unlike traditional engineering, a deep foundation pit is a semi-closed environment, which means the disaster occurrence process and evacuation process are different. As one of the most frequent disasters in a deep foundation pit, collapse is caused by changes of soil structure. The deeper the pit is dug, the higher the soil pressure that is put on the supporting structure. When the soil structure changes, the supporting structure cannot support the sidewall, and collapse occurs. Since the environments and soil situations are different in each pit, it is difficult to predict evacuation performances for deep foundation pits with complex layouts [2]. Traditional optimization methods [3–5] such as linear or nonlinear planning, graph theory, network optimization, multi-objective planning, and game theory can be used for solving optimization problems. However, since the disaster propagation and agent escaping are difficult to describe, traditional analytical methods cannot solve this kind of problem, which is dynamic and changing. In order to prevent injuries caused by collapse disaster, a proper

evacuation model is needed to simulate the disaster process and produce accurate results for occupants in a deep foundation pit.

For emergency evacuation, since the cost of holding realistic experiments cannot be easily afforded and the experimental data are difficult to capture, the computer simulation method has become popular to solve safety problems [6]. Some evacuation models [7–9] have been successfully established to solve crowd evacuation in high buildings and fireworks, which makes it possible to calculate relevant quantitative solutions in the field without practical experiments. Such systems use a bottom-up modeling approach in which system control is decentralized and governed only by the behavior of agents.

The safe exit in a deep foundation pit is usually a ladder set on the sidewall. The steel pipe of the ladder is driven into the soil, and handrails are arranged on both sides of the ladder. Since the setting of such escape exits must be stable, it will be tightly adhered to the sidewall. Therefore, it will affect the supporting structure of the sidewall. So, it is not suitable to set too many. How to set up a reasonable number of ladders to ensure sufficient escape routes and properly allocate them is an important issue to be solved. At present, researches on the safety of deep foundation pits mainly focus on the design of the supporting structure [10,11], but lack designs for exit distribution and setting.

This paper presents a prototype of an agent-based simulation for evacuation in deep foundation pits, named DPE. First, a collapse disaster model and agent-based escape model are designed. Intelligent technology is included to represent the self-motivation, response, and decision-making ability of agents in the escape process. Then, an effective optimization algorithm that can be adopted to simulate occupant evacuation is presented. According to the given ratio of sidewall collapse probability, a large sample of disaster events is generated. With the generated initial exit distribution, we use the parallel simulation method to simulate the escape rate. Through the simulation-based optimization algorithm, the iterative updating strategy is used to optimize the exit distribution. The contribution of this study is twofold:

1. It provides an agent-based system that is specifically designed for a deep foundation pit evacuation simulation of a collapse disaster, and a novel collapse model and agent escape model is built.
2. The system is built for customization, and provides the user with the ability to seek optimal exit distribution. A simulation-based optimization algorithm is applied to optimize the distribution, the particle swarm optimization (PSO), and generalized Voronoi diagram (GVD) algorithm is mixed in the simulation-based algorithm.

The remainder of this paper is organized as follows. Related work is presented in Section 2. The framework of system is introduced in Section 3. The mathematical method is presented in Section 4. The particle swarm optimization and weighted Voronoi algorithm are presented in Section 5. Experimental results are presented in Section 6. Finally, we conclude the paper and present an indication of future work in Section 7.

2. Related Work

Deep foundation pit emergency evacuation is a typical problem for crowd evacuation. Since the disaster in a deep foundation pit occurs suddenly, people have little time to escape. Designing proper exit distribution becomes one important approach to improve the survive rate. If the proper exit distribution is implemented beforehand, harmful effects can be significantly mitigated. Agent-based simulation is an effective approach to solve the evacuation optimization problem [12].

Several recent studies involving agent-based models for crowd evacuation simulation exist in the current literature. Simulations have been proposed to solve the crowd evacuation problem. For the evacuation process simulation, Levin et al. [13] used a model of pedestrian behavior to investigate the mechanisms of panic, and jamming by uncoordinated motions in crowds is used to prevent dangerous crowd pressures. Kirchner et al. [14] presented the simulation of evacuation processes by using a

recently introduced cellular automaton model for pedestrian dynamics. Joo et al. [8] abstracted the evacuation progress as a progress that the agent objects are driven by various factors with time flow. Boje and Li [15] investigated the level of integration between digital building models and crowd simulation. Weiss et al. proposed [16] a position-based dynamics for real-time crowd simulation.

For agent modeling in evacuation, Yang et al. [17] proposed an agent-based fire and human interaction model, and analyzed the grid resolution to determine the appropriate grid that will optimize the solution accuracy and time. Wang et al. [18] employed an ant colony evacuation model that included avoidance and preferential path selection behaviors. Hong et al. [19] proposed a self-evacuation modeling and simulation of passengers in metro stations to provide a quantitative analytical basis for developing evacuation strategies. Multiple agent-based models at differing resolutions have been integrated to simulate building evacuation dynamics [20]. Ha et al. [21] proposed a system of self-moving particles whose motion is governed by the social force model to investigate the effect of complex building architecture during urgent evacuation. Anh et al. [22] provided a hybrid agent-based model for roadway evacuation simulation that combined macro and micro-level simulations to increase the overall simulation efficiency while capturing necessary low-level simulation details. Pan et al. [23] used a multi-agent model to simulate behavior during evacuation that exhibits competitive, queuing, and herding behaviors. Yuksel [24] studied the pedestrian dynamics and learning process and applied the NeuroEvolution of Augmenting Topologies in evacuation simulation. Trivedi [25] proposed an agent-based evacuation model considering human panic behavior.

Researches on deep foundation pits have focused on the supporting structure design and emergency planning. Zhou et al. [26] proposed a numerical simulation of deep foundation pit dewatering and the optimization of controlling land subsidence, and presented a case study on Shanghai Metro station. Zhang et al. [27] studied the effect of foundation pit excavation on the buried pipeline. A three-dimensional model of a pipeline and a foundation pit was established, and the variation regulations of a pipeline's deformation under the foundation pit excavation were investigated. Luo et al. [28] proposed a finite element numerical simulation of three-dimensional seepage control for deep foundation pit dewatering. By analyzing the supporting structure for the deep foundation pit of the Hangzhou metro, Yang et al. [29] discussed the factors of supporting structure and soils. Zhou et al. [30] proposed a fuzzy comprehensive evaluation method based on Bayesian networks to apply in the risk precautions for deep foundation pit constructions. LiZheng Company developed a system named "LIZHENG deep foundation pit design software" [31] to simulate the building process for deep foundation pits, which can be used to design supporting structure settings.

The latest research on optimization algorithms has focused on using neural networks and fuzzy controllers to improve the searching precision. Saadat et al. [32] proposed the harmony search algorithm for training the echo state network in an online manner. Vrkalovic et al. [33] presented model-free sliding mode controllers and Takagi–Sugeno fuzzy controllers for the flux and conductivity control of reverse osmosis desalination plants. Hosen et al. [34] proposed a neural networks ensemble procedure to construct quality prediction intervals. Precup et al. [35] proposed the tuning of a class of fuzzy control systems to ensure a reduced parametric sensitivity on the basis of a new gravitational search algorithm. Due to a lack of sufficient real evacuation data, we use traditional evolutionary algorithms to solve this problem, and we will consider using the learning method in future work. Genetic algorithms (GAs) and particle swarm optimization (PSO) [36–38] have been found to be very robust and general for solving engineering design problems. They require the use of large population size, and may suffer from slow convergence. Both of these lead to large number of function evaluations, which can significantly increase the computational cost. Since the parameter in the PSO algorithm is relatively less, we apply the PSO algorithm in the system.

Agent-based simulation is a mature approach to solve evacuation, but has not been used for deep foundation pit evacuation. In this paper, taking advantage of the agent-based approach, we propose an agent-based simulation that has been used for deep foundation pit evacuation safety research. Based

on the modeling of the collapse disaster process and agent escape process, a simulation-based PSO algorithm is proposed to calculate the optimal exit distribution.

3. Simulation Framework of DPE System

In this section, we will introduce the framework of the proposed DPE system and explain the working flow between different modules. The DPE system focuses on the simulation of the deep foundation pit collapse disaster process, and can be used to calculate the optimal exit distribution for different exit numbers. First, we put user input parameters such as the pit size and related environmental information into the DPE system. Then, the DPE system will simulate the disaster process, and optimize the exit distribution and output result. During the optimization process, the simulation-based optimization method is used to obtain the result of different solutions. The modular method is applied to reduce the coupling between modules, which is convenient for users to remove or extend modules.

The framework of the DPE system is shown in Figure 1, which consists of three modules: input module, simulation engine module, and output module.

Figure 1. The framework of our prototype for an agent-based simulation for evacuation in deep foundation pits (DPE) system.

The input module refers to the User Interface (UI), which allows users to enter parameters and control the simulation process. Specifically, it is used to start, pause, continue, and stop the simulation and set the simulation speed. The simulation engine module includes a simulation engine, mathematical models, optimization, and data collection. The mathematical models refer to the mathematical representations for the environments, collapse disaster, and agent behaviors. The optimization module refers to the particle swarm optimization algorithm and the adaptive weighted Voronoi diagram algorithm. The data collection module can record the data generated during the simulation. The output module contains the display module and result generation module. The system provides several different display modes, and a tree structure is constructed to display the agents' information during simulation. The result generation module generates a report about the simulation result, which includes the following parts: description of environmental information, an optimal plan, and agents' escape tracks. In each simulation, users input the parameters of deep foundation pits into the system. After receiving the input parameters, the simulation begins to run for the optimal results of each solution using the optimization algorithm, and the mathematical model is used to support the modeling in the simulation. In the following section, the mathematical model and optimization algorithm will be introduced.

4. Mathematical Model for the DPE System

In this section, the mathematical model of our DPE system is proposed. In order to simulate the whole process of deep foundation pit evacuation, two models are designed: a collapse model and an agent-based escape model. The collapse model uses the discrete element method to simulate the gradual process of rock slopes. The agent-based escape model uses an artificial potential field to simulate the agent behavior and uses a single-serve queuing model to simulate the escape process in exit.

4.1. Collapse Model

The collapse model describes the process of collapse that happened on the sidewall of a deep foundation pit. As one of the most frequent disasters in deep foundation pits, collapse is characterized by rapid occurrence and strong destructive force, accompanied by physical and mechanical phenomena such as a discontinuity of the displacement field, large displacement, and impact collision. To simplify the process, the discrete element model (DEM) [39,40] is used to model the collapse process. Using the discrete element model, the collapsed region can be decomposed into subunits, which could be calculated for simulation, and the whole collapse process can be simulated by analyzing each subunit's movement.

The DEM is designed to simulate the gradual process of rock slopes, and has been applied to study the mechanical behavior of discontinuous media such as rock [41]. It divides the research object into a large number of granular discrete units according to the structure. By calculating the stress of each unit, the collapse process is modeled.

Using the DEM to decompose the collapsing sidewall into subunits, the stress of these subunits is modeled and calculated. Each decomposed subunit cell is gridded into a size of $1 \times 1 \times 1$. The mass of each cell is put on centroid. The stress between adjacent cells produces a particle internal force. Cells will move under the resultant forces of gravity force, internal force, and damping force. When the moving displacement is more than the deformation threshold of the supporting equipment, the supporting force becomes weaker, and the sidewall will fall into the deep foundation pit, causing its collapse. The stress characteristic is shown in Figure 2.

Figure 2. Stress characteristic of cells.

We use the classical earth pressure model in Rankine's theory to calculate the stress force [42]. As shown below, the stress can be calculated with the stress in the semi-infinite elastic body combining and the limit equilibrium conditions:

$$F_{pressure} = \sigma_{ajk} K_{ai} - 2\sqrt{K_{ai}} \quad (1)$$

K_{ai} is the coefficient of soil pressure, which can be inferred from the standard table. σ_{ajk} is the weight of soil above the cell. The supporting force is decided by the supporting material and structure:

$$F_s = ref(structure, material) \quad (2)$$

structure refers to the support structure, which includes row piles, continuous walls, etc. *material* refers to the support materials, such as cement, steel and so on.

On the basis of the computational force, the deformation of each element is calculated using the discrete element displacement model:

$$M_a \ddot{d}_a = F_a^{ext} \quad (3)$$

M_a is the mass of cell a , $\ddot{d}_a$ is the acceleration vector of a , and F_a^{ext} is the external force on a .

The calculation is performed using the explicit integration method, and the acceleration of the particle is described using a central difference formula. By calculating the stress and historical displacement, the displacement at next moment can be calculated:

$$d_{n+1} = \frac{\Delta t^2}{M_a} F_a^{ext} + 2d_n - d_{n-1} \quad (4)$$

d_{n+1} , d_n , and d_{n-1} are the displacement in time $n + 1$, n , and $n - 1$. Δt is the time step.

The supporting structure has a fixed threshold for holding the wall. When the total displacement is close to the threshold, an early warning is issued. When the displacement exceeds the threshold, the supporting structure will be destroyed, and the sidewall will fall horizontally into the pit, which can be regarded as parabolic movement. The initial velocity v_0 is generated at a short duration of action under acceleration:

$$v_0 = \ddot{d}_0 \times \Delta t \quad (5)$$

$\ddot{d}_0$ is the initial acceleration, and Δt is the time the initial velocity is generated. The falling time t is calculated using the free fall formula:

$$t = \sqrt{h/g} \quad (6)$$

h is the falling height of collapse central, and g is the acceleration of gravity. The location of the dropped cell $p(x, y, 0)$ is calculated according to a parabolic formula:

$$p(x, y, 0) = p(x_0, y_0, z_0) + \Delta(f(v_0, t)) \quad (7)$$

$p(x_0, y_0, z_0)$ is the initial position of the cell and $\Delta(f(\vec{p}_0, s))$ is the position displacement of the cell.

4.2. Agent-Based Escape Model

An agent-based escape model is used to simulate the agent escape process. When collapse occurs, agents need to escape from the dangerous area, and approach a safety exit. In order to simply the problem, we only focus on the agent's maneuver behavior.

We base our model on the steering model developed by Thunderhead Engineering [43], and modify it by modeling the disaster's influence on agent evacuation. The artificial potential field method [44] is used to model the maneuver model. The maneuver model is used to plan agents' paths to their target exit, and can dynamically adjust the path while moving. By analyzing the factors that affected agents' movement, a comprehensive potential field in the deep foundation pit is formed. The obstacles generate a repulsive force, and the target provides an attraction force, integrating into a resultant force to drive the agent's movement. Under this resultant force, agents can move in the shortest path according to the gradient descent, avoiding obstacles and moving toward the target exit.

The agent decision model is single target decision model that is used to look for the nearest exit and select it as the maneuvering target T :

$$T = (e_s | S(p_a, p_{e_s}) \geq S(p_a, p_{e_i}), e_i \in \{e\})_a \quad (8)$$

In the artificial potential field method, the gravitational potential energy is defined as follows:

$$U_a = w_a (p_a - p_g)^2 \quad (9)$$

w_a refers to attraction constant, $p_a = (x_a, y_a)$ refers to the position of an agent, and $p_g = (x_g, y_g)$ refers to the position of the target. The repulsive force U_{re}^i created by obstacles is as follows:

$$U_{re}^i = \frac{1}{2}\eta \left(\frac{1}{P_a - P_{ob}^i} \right) \quad (10)$$

η refers to the repulsive force coefficient, P_a is the position of attraction, P_{ob}^i is the position of obstacle i , and the resultant force U_t is as follows:

$$U_t = U_a - \sum U_{re}^i \quad (11)$$

U_a is the force of attraction, and U_{re}^i is the force of obstacle i . According to the above formulas, an artificial potential field can be constructed in the grid map through the following steps:

1. Initializing the grid potential energy field, gravitational constant, and repulsion constant η ;
2. For each tile, calculating the distance to the target point and the gravitational potential energy;
3. Extracting the threats in a map, calculating the repulsion potential, and superimposing the force;
4. Adding the attracting potential force and the repulsion potential force to obtain the total potential force and assign it to tile $square_{(i,j)}$;
5. If it is the last tile, the algorithm comes to an end. If not, goes to step 2.

Figure 3 shows the artificial potential field calculating process. After the value is assigned by the artificial potential field method, the image after three-dimensional display is performed. In the figure, the agent moves according to the method of selecting the minimum potential energy value around it, and can reach the target point.

Figure 3. Artificial potential field method.

When collapse happens, the falling wall will cover continual area. Agents need to approach to the safety exit and get out of the pit. Unlike a traditional block-out emergency evacuation, agents are still in danger until they completely get out from the exit and arrive on the ground. Agents would not change their target exit unless the target exit is destroyed. If so, the decision-making model is used to select a new target. The escape process at an exit should represent this process.

An exit starts to have attraction for agents when collapse occurs. Since the number of exits is generally less than the number of agents, agents need to wait to use the exit and climb out to the ground. This process is similar to the queuing system, so we use the server model to model the escape process at an exit.

A typical queuing system can be described as: in order to obtain service, customers need to wait in the queue while waiting for service, and will leave after receiving service. In a deep foundation pit evacuation, exits are the servers and agents are the customers. Since the congestion of agents will cause the escape speed in the exit to decrease, we assume that each agent needs to spend time to enter the exit. The interval time between adjacent agents' entry is λ . By importing the interval time, the influence on agents' escape speed is eliminated. After receiving service, agents begin to climb out via the exit or a safety ladder. The climb process will take time before the agents finally get to the ground. The escape process has the following features:

1. Input process. Agents arrive at the server according to the time they move to the escape route, and the arrival times are independent of each other.
2. Queuing rules. When the server is occupied, agents will wait in the queue, and the queue length is not limited. The service order follows the first-come, first-served rule. The waiting time of agents for the service is only related to the pit situation. While waiting for service, the impact of disaster is still calculated. When the disaster approaches, agents will give up the queue and reselect another escape exit.
3. Service characteristic. There are multiple exits in the deep foundation pit, which can be regarded as a parallel connection of multiple servers. Due to the long distance between each server, an agent can select a server at any time. Therefore, each exit can be seen as a single-server model.

Figure 4 shows the exit ladder for escaping. Based on above features, the escape process in exit can be modeled as a generally GI/M/1/∞/FCFS server model [45]. Each exit is modeled as a single-server model in which customers arrive at random, with constant service time, unlimited team leaders, and first-in, first-out rule. The service process can be modeled by using the Markov state transition equation, as shown in Figure 5.

Figure 4. The exit ladder.

Figure 5. The queuing process at an exit.

1. Arrival. wait queue: $s = s + 1$.
2. Accept the service: wait queue: $s = s - 1$, and the server status is set to False.
3. Service completed: exit queue: $r = r + 1$, the server status is set to True.
4. Reselect the exit: wait queue: $s = s - 1$.
5. Escape success: exit queue: $r = r - 1$.

5. Optimization Algorithm

Based on above models, an evacuation system is built. Simulation can be conducted in the system. In order to generate an optimal solution, an optimization algorithm is need. We use the particle swarm

optimization algorithm to optimize the exit distribution, and during each epoch, we use the weighted Voronoi diagram to generate a global optimal exit-agent corresponding solution, which is needed in calculating a particle's fitness.

5.1. Particle Swarm Optimization Algorithm

The goal of the DPE system is to generate an optimal exit distribution, which can be abstracted into a single-objective constraint satisfaction problem, and be solved by applying an evolution optimization algorithm [46].

The optimization problem is to maximize the total escape rate of agents, which is influenced by the exits' distribution $\{p^i\}$:

$$G(i, j) = \max f(\sum survive_j, \{p^i\}) \quad (12)$$

i is the exit ID. j is the agent ID. $survive_j$ is the state of agent j (survival or death). $\{p^i\}$ is the exits distribution. $f(\sum survive_j, \{p^i\})$ is the escape rate under the exits' distribution j . The optimization algorithm is used to optimize the exits' distribution to reach a higher escape rate. The result of the optimization algorithm is the exits' distribution.

This problem is a typical NP-hard (Non-deterministic Polynomial) planning problem, which can be solved by applying the evolutionary algorithms. A traditional genetic algorithm is complex, with many intermediate factors, which is not suitable for rapid iterative optimization. The PSO algorithm is an evolution algorithm that simulates bird crowd behavior to search for the optimal result. Starting from a random solution, the PSO algorithm iteratively finds the optimal solution, and the quality of the solution is evaluated by fitness. Compared with the genetic algorithm, the PSO algorithm does not have the "crossover" and "mutation" operations. It follows the current searched optimal value to look for global optimality. Since the PSO algorithm is easy to implement and has high precision in this paper, we use the PSO algorithm to solve the exits' distribution optimization problem, which is easy to code and has less parameters to be adjusted.

For this problem, the positions of randomly generated exits are first input into the PSO algorithm as the initial particles. In each iterative epoch, the particles stand for different exits' distribution. The algorithm process is shown in Figure 6. First, we randomly initialize the exit distribution, and calculate the average escape rate of each exit in large sample disasters, which stands for the fitness of each particle. Secondly, we update the velocity and vector of each particle's next position according to the fitness of each particle. According to the qualification, we judge whether the maximum number of iterations is reached or the global optimal value is reached. Finally, we output the optimal result and end the simulation.

The search space is a two-dimensional stand for coordinate x and coordinate y . The total number of particles is n , which we set as 30. The position of particle i is expressed as $X_i = (x_{i1}, x_{i2})$; the optimal position searched so far of particle i is $P_i = (p_{i1}, p_{i2})$. The optimal position is $P_g = (p_{g1}, p_{g2})$, and the position change rate of particle i is $V_i = (v_{i1}, v_{i2})$. The position of each particle is calculated as follows:

$$v_{id}(t+1) = w \times v_{id}(t) + c_1 \times rand() \times [p_{id}(t) - x_{id}(t)] + c_2 \times rand() \times [p_{gd}(t) - x_{id}(t)] \quad (13)$$

c_1, c_2 are acceleration factors. The learning factors c_1, c_2 are used to control the relative influence between the particle's own cognition and the social shared information. We set c_1 as 3 and c_2 as 5. $rand()$ is a random number between $[0, 1]$. It is used to randomly generate initial particles. $p_{id}(t)$ is position, $v_{id}(t)$ is speed, and $x_{id}(t+1)$ is the distance of particle i in time t . $x_{id}(t+1) = x_{id}(t) + v_{id}(t+1), 1 \leq i \leq n, 1 \leq d \leq 2$. The positions of each particle indicate the position of each exit $v_i = \{p_1, p_2, p_3, \dots, p_k\}$. k is the exit's number. w is the inertia factor. When w is larger, the global search ability of the particles is stronger. When the latter is smaller, its local search ability will be strengthened. Since there is no initial information at the beginning, the particles should fly forward at a large speed.

Through simulation, the escape rates of different exits' numbers are obtained, which stand for the fitness of particles. According to the updating rule, the exit distribution will optimize to reach a higher escape rate.

Figure 6. Particle swarm optimization (PSO) algorithm.

5.2. Adaptive Weighted Voronoi Diagram

In each generation epoch of PSO, the escape rate needs to be calculated. To search for the optimal escape rate under different exit distribution situations, an adaptive weighted Voronoi diagram is used to calculate the global optimal exit–agent corresponding solutions. Comparing with a behavior-based model, this global method can reach a higher escape rate, because it considers the agent jam in exits.

By using a global optimization algorithm, the problem can be abstracted as a multi-agent constrained task assign problem considering following factors: the distance between agents and exits, the queuing problem at exits and the disasters' influence on agent's behavior. Planning the optimal escape route for a single agent will result in the overall benefit that is not optimal even the result for the individual agent is optimal. Therefore, in order to achieve a higher team score, we change the research perspective, not from the perspective of agent selection, but from the perspective of exit service.

The Generalized Voronoi Diagram algorithm is used to solve the pit partition problem. And the agents in the same subarea will escape from the same exit. The optimization goal is to average the total escape time in each exit. We use the Voronoi diagram method to construct a generalized Voronoi diagram (GVD) map and assign weights to dynamically adjust the diagram. By constructing the Voronoi diagram, it is possible to assign agents to different exits.

$$G((\text{sub}a_1, \text{sub}a_2, \dots, \text{sub}a_n)) = \min(\sum_{i,j} \left| \sum_{n_a} ea_i - \sum_{n_a} ea_j \right|) \quad (14)$$

$(suba_1, suba_2, \dots, suba_n)$ are the subareas. sa_i, sa_j are the numbers of escape agents in i, j . n_a is the agent in the subarea a .

Assuming the pit is a plane, $Q = \{q_1, q_2, \dots, q_n\} (3 \leq n \leq \infty)$ are the points on the plane, and the weighted Voronoi diagram [47] is defined as:

$$V(q_m, \omega_m) = \left\{ x \in V(q_m, \omega_m) \left| \frac{d(x, q_m)}{\omega_m} \leq \frac{d(x, q_l)}{\omega_l}, l = 1, 2, \dots, n, l \neq m \right. \right\} \quad (15)$$

$d(x, q_m)$ and $d(x, q_l)$ are the distances of point x to vertexes $\{q_i\}$. ω_m is the weight of the vertex q_m . By calculating the density distribution of agents, each exit's weights are obtained, so that the density of agents in each diffused area is ensured to be almost the same.

$$w = \frac{\bar{n}_a}{n_{ai}} \times \frac{1}{f_a} \quad (16)$$

$\bar{n}_a$ represents the agents included in the normal partition, n_{ai} represents the agents included in each actual partition, and f_a represents the density of agents.

Since the weight value is calculated by overall agents' density, there is still a deficiency in dealing with the details whether the distance between two exits is close. We use the adaptive weight Voronoi method to dynamically adjust the weight according to the specific conditions of each area and the arrival time of agents, so that the diffusion speed of the source in each direction is no longer fixed. Taking the waiting time at exits into consideration, it is also necessary to calculate the number of agents that arrives at the same exit in a short period of time. Therefore, a modified weighted Voronoi diagram is proposed. Figure 7 shows the GVD for exits, the broken line refers to the borderline between each exit.

Figure 7. The generalized Voronoi diagram (GVD) algorithm.

The adaptive weight w_i is calculated as follows:

$$w_i = ar_m + br_l + cr_d \quad (17)$$

r_m is the agents' density coefficient, which reflect the number of agents in the region. The agents' density coefficient is related to direction i . $r_m = \bar{\rho} / \rho_i$. ρ_i is the agents' density in the region, and $\bar{\rho}$ is the average density. r_l is the carrier rate factor of an exit. r_d is the distance coefficient. a, b, c are weighting coefficients, $a + b + c = 1$.

We assume that collapse has a binding effect on the spread of all of the exits. That means that the further an exit is to the collapse resource, the smaller the Voronoi weight. The impact is as follows:

$$w = w_0 \times \left(\frac{1}{\ln(1 + s_i)} \right) \quad (18)$$

w_0 refers to the original adaptive weight, and s_i refers to the distance of disaster to the exit i .

The adaptive weighted Voronoi diagram algorithm is shown in Figure 8. First, we partition the map into uniform grids. Second, we calculate the weights of each exit based on agent density of different regions. If agents exist that are not assigned to exits, a new iteration is started. In each iteration, an exit has its radius, and those agents that are not farther than this radius will be assigned to this exit. At the end of the iteration, the radius of each exit will increase according to its weight. When all of the agents are covered by exits, the algorithm comes to the end.

Figure 8. Adaptive weighted Voronoi diagram.

6. Experiment

6.1. Experimental Environment and Settings

In order to verify the simulation, an empirical study is carried out, and the performance of the optimization algorithms is compared to that of traditional algorithms. The experiment is carried out in three deep foundation pits, whose shapes are pentagon, rectangle, and octagon. The setting of the three deep foundation pits is shown in Table 1. The dimension represents the type and size of the deep foundation pits. The vertex coordinates represent the vertexes' positions. The agent number stands for the total agent number in the pit. The collapse probability stands for the collapse occurring probability of each sidewall, which can be used to generate a disaster set.

Table 1. The setting of deep foundation pits.

	Pit A	Pit B	Pit C
Dimension	100 m × 100 m × 10 m	160 m × 160 m × 15 m	200 m × 150 m × 20 m
Vertex coordinates	(15, 50), (30, 15), (80, 15), (85, 70), (40, 90)	(0, 0), (160, 0), (160, 120), (85, 70), (0, 120)	(20, 40), (40, 20), (160, 20), (180, 40), (180, 120), (160, 140), (40, 140), (20, 120)
Agent number	50	100	100
Collapse probability	15%, 35%, 20%, 10%, 20%	35%, 10%, 25%, 30%	5%, 30%, 5%, 10%, 0%, 20%, 15%, 15%

The three-dimensional scene of deep foundation pits is shown in Figure 9.

Figure 9. The three-dimensional scene of deep foundation pits.

6.2. Experimental Results and Analysis

Three serial experiments are conducted to verify the algorithms improvement: the exit distribution optimization experiment, the agent–exit match optimization experiment, and the warning time influence experiment. The partial algorithms and datasets are available in Github: <http://github.com/hccz95/FoundationPit.git>.

6.2.1. Exit Distribution Optimization Experiment

This experiment is carried out to validate the effect of the exit distribution optimization algorithm. Compared with the PSO algorithm, we use a randomly generated algorithm as the benchmark algorithm. The warning time is limited to 10 s. Since the size of the deep foundation pits is different, the number of exits that we calculate is also different. In each pit, we calculated twice the number of sidewalls as the maximum number of exits. That is 10 exits for pit A, eight exits for pit B, and 16 exits for pit C. The results are shown in Figures 10–12.

It can be seen that for all three pits, the optimal distribution achieves a higher escape rate than the random distribution. Compared with the random distribution algorithm, the optimal distribution that was generated by the PSO optimization algorithm improves the escape rate by 73.7% in pit B, and 5% in pit A. For different exit numbers, the escape rate improved within a range. That is because the agent initial distribution is random, and the disaster occurrence place is not the same, resulting in the results not having a linear increase. Although more exits means that agents are easier gather to the exits, if the disaster spreads to the exits, the agents in the exits will all die. We can see that in pit B, the escape rate reaches almost 90%, which is higher than the other two pits. That is because the pit B is an inerratic shape, and the length of each sidewall is the same.

Figure 10. The escape rate of random distribution and optimal distribution in pit A.

Figure 11. The escape rate of random distribution and optimal distribution in pit B.

Figure 12. The escape rate of random distribution and optimal distribution in pit C.

6.2.2. Agent–Exit Match Optimization Experiment

This experiment compares the adaptive weighted Voronoi diagram optimization method to the naive Voronoi diagram algorithm. The warning time is 20 s.

In all three pits, the escape rate of the adaptive weighted Voronoi diagram algorithm is higher than the naive Voronoi diagram algorithm. Especially in pit B and pit C, the adaptive weighted Voronoi diagram algorithm can quickly achieve a high escape rate. The optimization algorithm has a rate of increase of 1–23% compared to the naive algorithm. With the addition of exit numbers, the escape rate will increase, and finally tend to stabilize. This is because the agent initial distribution is random, which means there are some agents who are far away from all of the exits. Therefore, the escape rate is difficult to improve. The improvement rate of the optimization algorithm after stabilization is 23% for pit A, 11% for pit B, and 11% for pit C, which is shown in Figures 13–15.

Figure 13. The escape rate of different solutions in pit A.

Figure 14. The escape rate of different solutions in pit B.

Figure 15. The escape rate of different solutions in pit C.

6.2.3. Warning Time

In addition to comparing the effects of the optimization algorithm, we also analyze the warning time factor. The warning time experiment is shown in Figures 16–18.

We compare the escape rate under different warning times and different numbers of exits with the naïve Voronoi diagram algorithm. If the warning time is below 10 s, adding more exits cannot certainly improve the escape rate, because the exits can be destroyed and the agents near them will be killed, which will reduce the escape rate. If the warning time exceeds 15 s, the escape rate tends to be stable. It will only be effective if the number of exits is small. When the warning time is moderate, about 10 s to 15 s, increasing the number of safe passages will increase the escape rate. If the exit number is more than four, the increment in warning time has less impact on the escape rate. If the exit number is less than three, regardless of the length of the warning time, adding the exit number will greatly increase the escape rate. If the exit number is between three and six, with a certain warning time, the escape rate will increase as the number increases. If the exit number is more than six, adding the exit number increases the escape rate very slowly. Similar to experiment 1, the escape rate in pit B reaches the highest among the three pits for almost all of the time series.

Figure 16. The escape rate under different warning times in pit A.

Figure 17. The escape rate under different warning times in pit B.

Figure 18. The escape rate under different warning times in pit C.

6.3. Discussion of Experimental Results

In this section, we will discuss the simulation-based method and the traditional analysis method, and we will simulate the evacuation process and compare the escape rate under two methods.

We choose the constraint satisfaction problem (CSP) method as the benchmark method. The constraint satisfaction problem refers to a set of variables with constraints; each has its own assignment domain. A feasible solution can be obtained when the assignment of each variable satisfies the constraints of all variables simultaneously. The constraint satisfaction problem usually includes three components X , D , and C :

- X is the variables set $\{X_1, X_2, \dots, X_n\}$;
- D is the range set $\{D_1, D_2, \dots, D_n\}$;
- C is the constraint set $\{C_1, C_2, \dots, C_n\}$.

The range set D is a collection of possible values of the variable set X . Each constraint is an ordered pair of $\langle \text{scope}, \text{rel} \rangle$, in which scope is the set of variables in the constraint, and rel defines the relationship that these variables should satisfy. For this problem, you can abstract the solution problem into the following functions:

$$\max f(X) = \sum e_i \begin{cases} \sum path_i \leq c(p) \\ length_{q_j} \leq c_{length} \\ t_{wait,j} \leq c_t \end{cases} \quad (19)$$

where $\max f(X)$ is the agents' escape exits set, i is the agent ID, and j is the exit ID. The goal $f(X)$ is to achieve the highest escape rate, that is, the maximum escaped agent number. Constraint conditions include the shortest weighted path for agent $path_i$, the queue length $length_{q_j}$ and the wait time $t_{wait,j}$. $c(p)$, c_{length} , and c_t are the thresholds for constraints. In our DPE system, this information can be obtained by the data collection modules, as shown in Figure 19.

Figure 19. The collected information during simulation.

Based on the description of the constraint satisfaction method, the branch and bound method is used to solve the problem. The branch and bound algorithm (B&B) was proposed by Land and Dakin to solve the optimization problem. The main principle of the B&B algorithm is to divide and decompose the problem into several sub-problems. The process of decomposing a problem into sub-problems is called a branch, and the process of estimating the target value for each sub-problem is called bounding. Branching simplifies the problem. Bounding determines the target value range of the branch, and removes the branch with a poor trend or no optimal solution. The operation of deleting a branch is called pruning, which can accelerate and optimize the algorithm. The use of the branch and bound method is a process of continuously decomposing and pruning the problem and quickly reaching the optimal solution.

In this problem, the constraint satisfaction problem is A, and the relaxation problem is B. Using the B&B algorithm, we can find the maximum value of the objective function of A. Suppose the optimal objective function of the problem is z^* . Since problem B is the relaxation problem of A, then the optimal objective function obtained by problem B must be the upper bound of A, denoted as z^- , and any feasible solution of A, The value of the objective function is a lower bound of z^* , which is noted. The branch and bound method gradually reduces z^- and increases, and finally finds, z^* .

Our agent-based method is a simulation-based optimization method. SBO is an optimization method combining simulation technology and optimization technology. The superior optimization algorithm uses the simulation model to obtain the evaluation index of the different solutions, which is used to guide the optimization process of the exits' distribution, and continuously improve the output. The non-enumeration finds the best input variable value from the possible values, so that the output result is the optimal solution or satisfaction. Since SBO is a simulation to evaluate the indicators of actual complex systems, it can better reflect the actual system's operation under the established scheme, and better observe the uncertainty and randomness than the formalized model with a lot of simplification and abstraction.

The escape rates of two methods are shown in Figure 20. The escape rate of the SBO method is higher than the CSP method. When the exit number is 1, the escape rate is almost the same, which is because one exit cannot reflect the advantage of optimal distribution. When the number of exits increases, both methods reach a higher escape rate. In these instances, the SBO method performs better than the CSP, with an improvement of 15–19%, which reveals the superiority of our proposed method above traditional methods.

Figure 20. The escape rate under the constraint satisfaction problem (CSP) and simulation-based optimization (SBO).

7. Conclusions and Future Work

In summary, we have simulated the process of collapse disaster and the agent escape process for deep foundation pit evacuation. By using the simulation-based optimization method, an optimal exit distribution is obtained. Based on the DPE system, several key factors for the prevention of disasters in deep foundation pits are analyzed, and safety precautions are put forward. Our experiments show that the simulation can be used to calculate the escape rate under different conditions. Compared with the CSP method, the proposed SBO method has an improvement of about 19%, and the PSO algorithm and weighted GVD algorithm both contribute to the improvement of the agent escape rate.

Following the results, safety solutions can be adopted to ensure agents' safety inside the deep foundation pit. Effective monitoring and adequate warning time are of great significance for deep foundation pit disasters. It is necessary to establish a detection system that is capable of early warning. Users should rationally arrange different types of monitoring devices to detect indicators such as displacement and pressure, and dynamically adjust the monitoring frequency according to the monitoring data.

Due to the modeling granularity, our modeling only considers some key information, but some details are not considered, such as the influence of terrain changes on the agent escape speed and the randomness of disaster spread. When calculating the disaster occurrence process, it is assumed that the force change can be calculated at any time, but in fact, this process requires calculation time and deviation. In future work, we will research how to use real-time monitoring data for disaster warning and simulate other disasters that occur in deep foundation pits such as fire and water inrush disasters. We will also apply other algorithms, including learning algorithms, to search for the optimal solution more quickly.

Author Contributions: Conceptualization, W.Y. and M.Y.; Methodology, W.Y. and M.Y.; Formal analysis, Y.H.; Software, C.H.; Writing—original draft preparation, W.Y.; Writing—review and editing, Y.H.

Funding: This research was funded by the National Natural Science Foundation of China under Grant No. 61473300.

Acknowledgments: We want to thank Long Qin and Xu Xie for the discussion about agent modeling and optimization algorithm design, and Lichun Zhang for the help in software design.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hangzhou Metro Xianghu Station Collapse Accident. Available online: <http://news.163.com/11/0519/14/74E2C3S500014JB6.html> (accessed on 20 September 2018).
2. Jain, S.; McLean, C. Components of an incident management simulation and gaming framework and related developments. *Simulation* **2008**, *84*, 3–25. [CrossRef]
3. Domijan, M.; Kirkilionis, M. Graph Theory and Qualitative Analysis of Reaction Networks. *Netw. Heterog. Media* **2017**, *3*, 295–322. [CrossRef]
4. Adami, C.; Schossau, J.; Hintze, A. Evolutionary game theory using agent-based methods. *Phys. Life Rev.* **2016**, *19*, 1–26. [CrossRef] [PubMed]
5. Ghallab, M.; Nau, D.; Traverso, P. *Automated Planning: Theory and Practice*, 1st ed.; Elsevier (Singapore) Pte. Ltd.: Singapore, 2004.
6. Borshchev, A.; Filippov, A. From system dynamics and discrete event to practical agent based modeling: Reasons, techniques, tools. In Proceedings of the 22nd International Conference of the System Dynamics Society, Oxford, UK, 25–29 July 2004.
7. Miyoshi, T.; Nakayasu, H.; Ueno, Y.; Patterson, P. An emergency aircraft evacuation simulation considering passenger emotions. *Comput. Ind. Eng.* **2012**, *62*, 746–754. [CrossRef]
8. Joo, J.; Kim, N.; Wysk, R.A.; Ling, R.; Son, Y.J.; Oh, Y.G.; Lee, S. Agent-based simulation of affordance-based human behaviors in emergency evacuation. *Simul. Model. Pract. Theory* **2013**, *32*, 99–115. [CrossRef]
9. Shi, J.; Ren, A.; Chen, C. Agent-based evacuation model of large public buildings under fire conditions. *Autom. Constr.* **2009**, *18*, 338–347. [CrossRef]
10. Zhao, J.; Cao, R.; Liu, J.; Shi, X. Research on the Deep Foundation Pit Wall Skew Prediction Model Based on GA-SVM. *Value Eng.* **2018**, *9*, 147–149.
11. Lin, C. Simulation and analysis on the support method for deep foundation pit of metro station by FLAC^{3D}. *J. Hunan Univ. Arts Sci.* **2018**, *30*, 70–74.
12. Helbing, D.; Farkas, I.; Vicsek, T. Simulating dynamical features of escape panic. *Nature* **2000**, *407*, 487–490. [CrossRef] [PubMed]
13. Levin, B.C.; Braun, E.; Navarro, M.; Paabo, M. *Further Development of the n-Gas Mathematical Model—An Approach for Predicting the Toxic Potency of Complex Combustion Mixtures*; American Chemical Society: Washington, DC, USA, 1995; pp. 293–311.
14. Kirchner, A.; Schadschneider, A. Simulation of evacuation processes using a bionics-inspired cellular automaton model for pedestrian dynamics. *Phys. A Stat. Mech. Its Appl.* **2002**, *312*, 260–276. [CrossRef]
15. Boje, C.; Li, H. Crowd simulation-based knowledge mining supporting building evacuation design. *Adv. Eng. Inform.* **2018**, *37*, 103–118. [CrossRef]
16. Weiss, T.; Littenecker, A.; Jiang, C.; Terzopoulos, D. Position-based multi-agent dynamics for real-time crowd simulation. In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Los Angeles, CA, USA, 28–30 July 2017; ACM: New York, NY, USA, 2017.
17. Yang, P.; Wang, X.; Liu, T. Agent-based simulation of fire emergency evacuation with fire and human interaction model. *Saf. Sci.* **2011**, *49*, 1130–1141.
18. Wang, C.; Li, C.; Liu, Y.; Cui, J.; Zhang, T. *Behavior-Based Simulation of Real-Time Crowd Evacuation*; IEEE Computer Society: Washington, DC, USA, 2011; pp. 456–461.
19. Hong, L.; Gao, J.; Zhu, W. Self-evacuation modelling and simulation of passengers in metro stations. *Saf. Sci.* **2018**, *110*, 127–133. [CrossRef]
20. Yang, B.; Wu, Y.-G.; Ren, B. Application of multi-resolution modelling in emergency evacuation simulation. *Int. J. Simul. Process Model.* **2012**, *7*, 89–97. [CrossRef]
21. Ha, V.; Lykotrafitis, G. Agent-based modeling of a multi-room multi-floor building emergency evacuation. *Phys. A Stat. Mech. Its Appl.* **2012**, *391*, 2740–2751. [CrossRef]

22. Anh, N.T.N.; Daniel, Z.J.; Du, N.H.; Drogoul, A.; An, V.D. A hybrid macro-micro pedestrians evacuation model to speed up simulation in road networks. In Proceedings of the International Conference on Advanced Agent Technology, Taipei, Taiwan, 2–6 May 2011; Springer: Berlin/Heidelberg, Germany, 2011; Volume 7068, pp. 371–383.
23. Pan, X.; Han, C.S.; Dauber, K.; Law, K.H. A multi-agent based framework for the simulation of human and social behaviors during emergency evacuations. *Ai Soc.* **2007**, *22*, 113–132. [CrossRef]
24. Yuksel, M.E. Agent-based evacuation modeling with multiple exits using NeuroEvolution of Augmenting Topologies. *Adv. Eng. Inform.* **2018**, *35*, 30–55. [CrossRef]
25. Trivedi, A.; Rao, S. Agent-Based Modeling of Emergency Evacuations Considering Human Panic Behavior. *IEEE Trans. Comput. Soc. Syst.* **2018**, *5*, 277–288. [CrossRef]
26. Zhou, N.; Vermeer, P.A.; Lou, R.; Tang, Y.; Jiang, S. Numerical simulation of deep foundation pit dewatering and optimization of controlling land subsidence. *Eng. Geol.* **2010**, *114*, 251–260. [CrossRef]
27. Zhang, J.; Xie, R.; Zhang, H. Mechanical response analysis of the buried pipeline due to adjacent foundation pit excavation. *Tunn. Undergr. Space Technol.* **2018**, *78*, 135–145. [CrossRef]
28. Luo, Z.J.; Zhang, Y.Y.; Wu, Y.X. Finite element numerical simulation of three-dimensional seepage control for deep foundation pit dewatering. *J. Hydrodyn. Ser. B* **2008**, *20*, 596–602. [CrossRef]
29. Yang, Y.; Jinguang, W.U. Analysis of retaining structures for deep foundation pit of qiutao road station in hangzhou metro. *Chin. J. Rock Mech. Eng.* **2008**, *8*, 3386–3392.
30. Zhou, H.B. Risk assessment of deep foundation pit by using fuzzy comprehensive evaluation method based on bayesian networks. *J. Shanghai Jiaotong Univ.* **2009**, *43*, 1473–1479.
31. LiZheng Deep Foundation Pit Software. Available online: <https://www.lizheng.com.cn/index2.html> (accessed on 20 September 2018).
32. Saadat, J.; Moallem, P.; Koofgar, H. Training echo state neural network using harmony search algorithm. *Int. J. Artif. Intell.* **2017**, *15*, 163–179.
33. Vrkalovic, S.; Lunca, E.-C.; Borlea, I.-D. Model-free sliding mode and fuzzy controllers for reverse osmosis desalination plants. *Int. J. Artif. Intell.* **2018**, *16*, 208–222.
34. Hosen, M.A.; Khosravi, A.; Nahavandi, S.; Creighton, D. Improving the Quality of Prediction Intervals Through Optimal Aggregation. *IEEE Trans. Ind. Electron.* **2015**, *62*, 4420–4429. [CrossRef]
35. Precup, R.-E.; David, R.-C.; Petriu, E.M.; Preitl, S.; Paul, A.S. Gravitational search algorithm-based tuning of fuzzy control systems with a reduced parametric sensitivity. In *Soft Computing in Industrial Applications*; Springer: Berlin/Heidelberg, Germany, 2011; Volume 96, pp. 141–150.
36. Vaez, S.R.; Fallah, N. Damage Detection of Thin Plates Using GA-PSO Algorithm Based on Modal Data. *Arabian J. Sci. Eng.* **2016**, *42*, 1–13.
37. Kassa, Y.; Zhang, J.H.; Zheng, D.H.; Wei, D. A GA-BP hybrid algorithm based ANN model for wind power prediction. In Proceedings of the IEEE Smart Energy Grid Engineering, Oshawa, ON, Canada, 21–24 August 2016; pp. 158–163.
38. Li, G.; Guo, C.; Li, Y.; Deng, W. Fractional-Order PID Controller of USV Course-Keeping Using Hybrid GA-PSO Algorithm. In Proceedings of the 8th International Symposium on Computational Intelligence and Design (ISCID), Hangzhou, China, 12–13 December 2016; pp. 506–509.
39. Jihong, Y.E.; Nian, Q.I. Collapse process simulation of reticulated shells based on coupled DEM/FEM model. *J. Build. Struct.* **2017**, *38*, 52–61.
40. Ying, Y.U.; Luo, Y. Structural collapse analysis based on finite particle method II: Key problems and numerical examples. *J. Build. Struct.* **2011**, *32*, 27–35.
41. Munjiza, A. *The Combined Finite-Discrete Element Method*; John Wiley & Sons: Hoboken, NJ, USA, 2004.
42. Rankine, W.J.M. On the mathematical theory of the stability of earthwork and masonry. *J. Franklin Inst.* **1986**, *63*, 84–85. [CrossRef]
43. Thornton, C.; O’Konski, R.; Klein, B.; Hardeman, B.; Swenson, D. New wayfinding techniques in pathfinder and supporting research. In *Pedestrian and Evacuation Dynamics 2012*; Springer: Cham, The Netherlands, 2014; pp. 1315–1322.
44. Zhang, J.N.; Fan, H.J. Crowd evacuation system simulation based on artificial potential field and agent. *J. Henan Norm. Univ.* **2012**, *32*, 1753–1756. [CrossRef]
45. Tang, Y.H. A single-server M/G/1 queueing system subject to breakdowns—Some reliability and queueing problems. *Microelectron. Reliab.* **1997**, *37*, 315–321. [CrossRef]

- 46. Bin, L.I. Simulation based optimization for parameter selection in PSO. *Comput. Eng. Appl.* **2011**, *47*, 30–35.
- 47. Karavelas, M.I.; Yvinec, M. Dynamic Additively Weighted Voronoi Diagrams in 2D. In *European Symposium on Algorithms*; Springer: Berlin/Heidelberg, Germany, 2002; Volume 2461, pp. 586–598.

© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).