A Novel Swarm Intelligence Algorithm for the Evacuation Routing Optimization Problem

Jin-long Zhu[1], Wenhui Li[2], Huiying Li[2], Qiong Wu[2], and Liang Zhang[2]
[1]Department of Computer Science and Technology, ChangChun Normal University, China
[2]Department of Computer Science and Technology, Jilin University, China

Abstract: *This paper presents a novel swam intelligence optimization algorithm that combines the evolutionary method of Particle Swarm Optimization (PSO) with the filled function method in order to solve the evacuation routing optimization problem. In the proposed algorithm, the whole process is divided into three stages. In the first stage, we make use of global optimization of filled function to obtain optimal solution to set destination of all particles. In the second stage, we make use of the randomicity and rapidity of PSO to simulate the crowd evacuation. In the third stage, we propose three methods to manage the competitive behaviors among the particles. This algorithm makes an evacuation plan using the dynamic way finding of particles from both a macroscopic and a microscopic perspective simultaneously. There are three types of experimental scenes to verify the effectiveness and efficiency of the proposed algorithm: a single room, a 4-room/1-corridor layout, and a multi-room multi-floor building layout. The simulation examples demonstrate that the proposed algorithm can greatly improve upon evacuation clear and congestion times. The experimental results demonstrate that this method takes full advantage of multiple exits to maximize the evacuation efficiency.*

Keywords: *PSO, filled function, global optimum, local optimum*.

Received November 17, 2014; accepted September 10, 2015

1. Introduction

Emergency evacuation plans are developed to ensure the safest and most efficient evacuation time of all expected residents of a structure or region [4, 10, 14]. With the increasing complexity of buildings and frequency of disasters, the evacuation routing optimization problem has become a hot topic in the area of emergency evacuation planning. The problem can be divided into the microscopic and the macroscopic perspectives. To better characterize crowd behaviors for egress analysis, microscopic pedestrian models have been developed during recent decades where an evacuee's behavioral/psychological status can be modeled and simulated. Parlak *et al.* [9] considered motivational force, psychological repulsive tendencies, compression, viscous damping, personal force and sliding friction in the simulation of specific emergency evacuations. The motion of individuals was governed by the social-force model to investigate the effect of crowd evacuation. Manley and Kim [8] considered an agent-based approach to estimate formation of bottlenecks during urgent evacuation. The work of Zheng *et al.* [18, 19, 20] focused on evacuees' cooperative and competitive behaviors by using a close analogy to the Chicken-type game Tanimoto *et al.* [15] proposed a deductive approach to analyze the bottleneck problems of pedestrian evacuation by using a close analogy to the saint&temptation reciprocity game. Shi and Wang [13] proposed a microscopic framework to research crowd dynamics based on the modified lattice gas model by using snowdrift game theory. Ha and Lykotrafitis [2] proposed an Agent-based modeling of a multi-room multi-floor building emergency evacuation. Particle Swarm Optimization (PSO) [5, 11, 21] is a multi-agent based simulation method that can simulate complex behaviors of individuals in an urgent evacuation. However, such microscopic models only take the evacuees' local behavior into account, omitting other factors which may be of equal importance for them. In addition, microscopic simulations are computationally complex, making it difficult to be used directly for optimizing evacuation strategies. On the other hand, some researchers have studied evacuation planning from a macroscopic perspective. Chooramun *et al.* [1] developed an evacuation model utilizing hybrid space discretization, which uses a mixture of three basic techniques for space discretization, namely coarse networks, fine networks, continuous networks.. However, these methods only considered the global evacuation plan and ignored the influence of the behavior of individuals while simulating the evacuation behavior. Modeling the dynamic way finding of evacuees with respect to both the macroscopic and microscopic perspectives simultaneously is rare. Therefore, the present study proposes a novel PSO algorithm to optimize evacuation routing from these two perspectives simultaneously.

This study addresses two issues. The first is how to optimize particles' competitive behaviors. The second issue is how to determine optimal evacuation routes for particles. We present the Global/Local Particle Swarm Optimization (GLPSO) algorithm for multi-exit evacuation, intended to plan an optimal egress route based on global and local optimum nodes. The main contribution of this method provides a new perspective to understand the optimal control of emergency evacuation. The proposed method can improve evacuation clearance time and decrease crowd density in determining emergency evacuation strategies. In addition, it can take full advantage of multiple exits in order to obtain the safest evacuation route.

The reminder of this paper is organized as follows. Section 2 introduces the state of the art in evacuation planning. Section 3 presents the result of the optimization. Section 4 concludes this paper and describes the outlook on future studies.

2. The Description of PSO Algorithm

The PSO algorithm is based on swarm intelligence [2]. The movements of the particles are determined by their own best known position in the search-space as well as the entire swarm's best known position. An improved PSO algorithm is then:

Algorithm 1: The improved PSO algorithm.

1. *Initialization*
   Initialize the particles' positions randomly with a uniform probability. Initialize the particles' best known positions to their initial positions. Initialize the particles' velocities.
2. *Get destination node*
   Compute destination node of each particle by a filled function (described in Section 2.1).
3. *Global optimum node and Local optimum node*
   Compute the global optimum node of each particle using the destination node. Compute the local optimum node of each particle according to the global optimum node.
4. *Update velocity*
   Update the velocity of each particle:
   $v_{id}^{k+1} = \omega v_{id}^{k} + c_1\xi\left(l_{id}^{k} - x_{id}^{k}\right) + c_2\eta\left(g_{id}^{k} - x_{id}^{k}\right)$
   If $v_{id}^{k+1} > v_{id}^{\max}, v_{id}^{k+1} = v_{id}^{\max}$
   If $v_{id}^{k+1} < 0, v_{id}^{k+1} = v_{id}^{\max}$
5. *Conflict detection*
   Predict the particle's position: $x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1}$
   Calculate the distance d between two particles. Determine whether the positions conflict. If d<2r, then go to Step 6, else go to Step 7.
6. *Conflict resolution*
   According to distance d, the GLPSO model optimizes conflicts by locating each particle's position using three methods: ConflictMethod1, ConflictMethod2 or ConflictMethod3 (described in Section 2.3).
7. *Update position*
   Update all particles' position.
8. *Terminal condition*
   If a particle moves to the next neighbor node, then go to Step 3, if a particle passes through an exit, then the termination criterion of that particle is met. Else go to Step 4.

The flowchart of the solution process is presented in Figure 1.

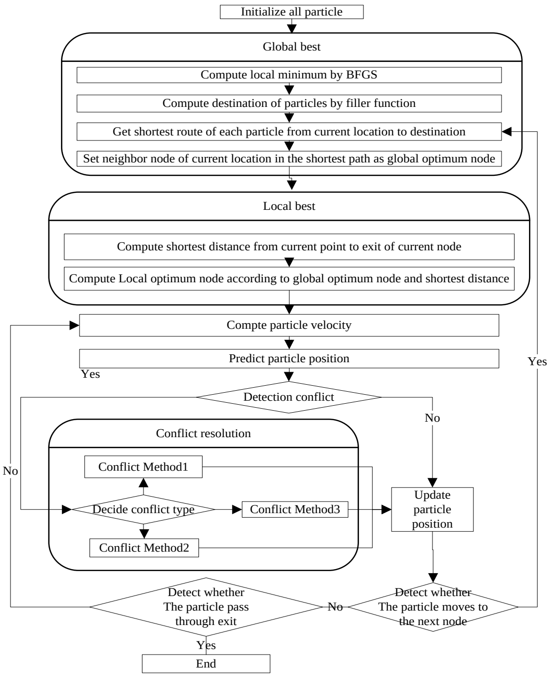


Figure 1. The flow chart of improved PSO model.

2.1. Particle Destination Nodes

In this paper, the concept of a filled function [12, 17] is introduced. This method is designed to obtain the particles' destination node. First, we treat all the evacuees of a node (e.g., a room) as a "whole entity" and assign them to each exit by formulas 1. According to the evacuee distribution, the length of the escape routes and the maximum flow rate, an estimate of evacuation clearance time must be taken into account with an expectation of the reaction of other evacuees at the same exits. We obtain the destination of particles with the following formula:

$$\begin{aligned} &\min T(x_1, x_2, \cdots, x_n) = \sum_{i=1}^{n}(T_{jam}^{i} + x_i / r_i + d_i * x_i / \sum S_n) * (1 - e^{-x_i}) \\ &s.t. 0 \le x_i \le P, i \in n; \sum x_i = P \end{aligned} \quad (1)$$

min represents the minimizing function value and "subject to" is abbreviated by "s.t.".

x_i is the number of particles in the i^{th} exit.

r_i is the maximum flow rate of the i^{th} exit.

d_i is the distance from current node to the i^{th} exit.

$x_i / \sum S_n$ is average speed.

T_{jam}^{i} is congestion time.

Equation (2) is the transformation formulation of the minimum evacuation clearance time to reduce the computational complexity.

$$\begin{aligned} T(x_1, x_2, \cdots, x_n) &= \sum_{i=1}^{n-1}(T_{jam}^{i} + x_i / r_i + d_i * x_i / \sum_{m=1}^{P} S_m) * (1 - e^{-x_i}) + \\ &(T_{jam}^{i} + (P - \sum_{i=1}^{n-1} x_i) / r_n + d_i * (P - \sum_{i=1}^{n-1} x_i) / \sum_{m=1}^{p} S_m) * (1 - e^{-(P - \sum_{i=1}^{n-1} x_i)}) \end{aligned} \quad (2)$$

We obtain the optimal evacuation plan using this filled function method, which is an approach to solve unconstrained global minimization problems. The filled function handles formulas 1 to arrive at the global minimum by breaking the algorithm into a two-step process.

We obtain the optimal evacuation plan using this filled function method, which is an approach to solve unconstrained global minimization problems. The filled function handles formulas 1 to arrive at the global minimum by breaking the algorithm into a two-step process.

- *Step1*. According to Wolfe-Powell conditions, we get the local minimum x1 of the function $T(x_1, x_2, ..., x_n)$ using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) [6, 16] algorithm, which approximates Newton's method in multiple dimensions efficiently. The BFGS method is a standard iterative method for solving unconstrained nonlinear optimization problems using the following procedure.

Algorithm 2: The BFGS method

1. *Initialization*
   Set the initial point x_0, define a positive definite matrix H_0, the tolerance $\mathcal{E}>0$, and an iterator k=0.
2. *Terminal condition*
   If $||gk||\leqslant\mathcal{E}$, then x_k is the optimal solution.
3. *Linear search*
   a. *Compute the search direction $d_k= -H_kg_k$.*
   b. *Calculate the factor a_k of the step length using a linear search, and set $x_{k+1}=x_k+a_kd_k$.*
   c. *Correct H_k to get H_{k+1}, set k=k+1, and go to Step 2.*

The iteration solution formula of H is given by

$$H_{k+1}=\left(I-\frac{\delta_k\gamma_k^T}{\delta_k^T\gamma_k}\right)H_k\left(I-\frac{\gamma_k\delta_k^T}{\delta_k^T\gamma_k}\right)+\frac{\delta_k\delta_k^T}{\delta_k^T\gamma_k} \tag{3}$$

Where, $\delta_k=x_{k+1}-x_k, \gamma_k=g_{k+1}-g_k$.

- *Step 2*. Here, x1 is the outcome of Step 1, which yields a local minimum of $T(x_1, x_2, ..., x_n)$. We construct a filling function fill(x) on the local minimum x_1, and take random a point as the initial point in the neighborhood of x_1. If there is a field below x_1, x_m is a point in the field found by minimizing fill(x). Then go to Step1, using the x_m which minimizes $T(x_1, x_2, ..., x_n)$ as the initial point to get a new local minimum value x_2, and iterate, each time using a different local minimum as the initial point.

Algorithm 3: The global minimum method

1. *Initialization*
   Read parameter values, $\mathcal{E}>0$, $0<\delta<1$, $a>0$, $S\subset R^n$. $\mathcal{E}$ is allowable error, δ is a point with an offset, a is the parameter of the filled function, and S is a region containing all the minima of $T(x_1,x_2,\cdots,x_n)$.
2. *Local minimum point*
   Set $x'\in S$ as the initial point, then use Algorithm 1 to arrive at a local minimum point x1 of $T(x_1, x_2, ..., x_n)$.
3. *Construct filling function*
   Set fill(x) as the filled function of $T(x_1,x_2,\cdots,x_n)$ near the local minimum x_1.
   Set initial point using the rule $x_{i0}=x_1+(-1)^i\delta e_{[\frac{i+1}{2}]}$, i=1,2,...,2n. Where, $e_i(i=1, 2, ... ,n)$ is the i^{th} unit vector.
4. *Terminal condition*
   If minimizations of the filled function traverse all of the iteration points for any given initial point in the region S, then the algorithm terminates, and x1 is the global optimum of $T(x_1,x_2,\cdots,x_n)$.
   We assume that y_k is a randomly point in the lower area, when the iteration point y_k meet any of following condition. Set $x'=y_k$, and go to Step 5.
   a. $d_{k-1}^T\nabla v(y_k)\geq 0$, d_{k-1}^T *is the search direction of y_{k-1}.*
   b. $(y_k-x_1)^T\nabla fill(y_k)\geq 0$
   c. $\|\nabla v(y_k)\|<\varepsilon$
   d. $f(y_k)<f(x_1)$
5. *Minimization function*
   Set x' as initial point to get a new local minimum value x_2. If $f(x_2) <= f(x_1)$, then set $x_1=x_2$ and go to Step 3, or else set a=10a and go to Step 4.
   GLPSO sets the destination of each particle in the node by the result of function $T(x_1, x_2, ..., x_n)$.

2.2. Global Optimum Node and Local Optimum Node

Each particle's movement is influenced by both its local best position and global best position. The global best position is equal to the local best position in a global optimum node. The Global optimum node is the neighbor node of the current location in the shortest path to the destination. The local best position is guided toward the best positions (such as exit and door) with the global optimum node in the search-space of the current node.

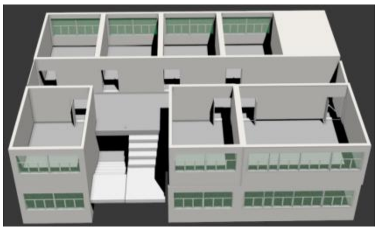
Lee and Kwan [7] proposed a Node Relation Structure (NRS) to represent buildings' internal structure. NRS abstracts the complex topological relationships among 3D features to a logical network structure. However, this model can't describe rooms or corridors containing exits. Since the presence of these exits can affect particles' behavior in their emergency response, obtaining optimal evacuation routes with such a 3D network analysis is difficult. We improved NRS to add exit and door nodes for obtaining the global optimum node and the local optimum node. We represent the architectural structure with a hierarchical undirected graph G = (V, E), consisting of a finite set V of particles (nodes) and a finite set E of edges.

There can be emergency evacuations in multi-room or multi-floor buildings. We present a hierarchical undirected graph model to extend the nodes into a subgraph, subsequently abstracting them into a super graph. This building has two levels of graphs (see Figure 3).

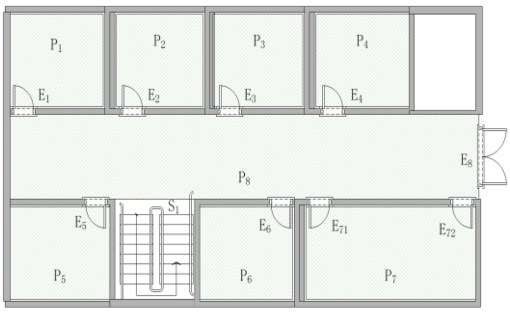
For example, consider a two-story building in which each story has seven rooms, one hallway, and one stairway (see Figure 2-a). Suppose there is an exit in the first floor (see Figure 2-b). This building has 18 enclosures (nine per story), which we label P1 to P16. E1 to E17 represent doors, and E8 is the Exit. S1 and S2 are stairways. We transform the 3D spatial units to 2D polygons. Figure 2-b shows all spatial units on the first story.

We construct the node relationships from this abstracted layout. The node relationships take the center point of each spatial unit as the location of the corresponding node. The edges connecting the nodes represent the connectivity among the spatial units. The extended subgraph presents the first story's extended node relations (see Figure 2-c). If we set hallway P8 as a node, the Local optimal node can't accurately calculate the path distances according to the route length. Thus, we find the intersection point of the door centerline and the corridor centerline as an extended node to get an accurate distance (such as P81).

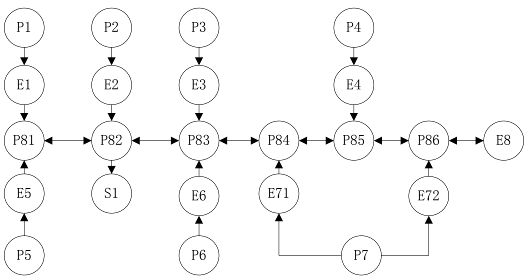
The supergraph comprises two levels of subgraph and the stairway represents the connectivity among the subgraphs (see Figure 3). A boundary node stands for each story's stairway node or an exit node.



a) A 3D building model.



b) A plane graph of one story.



c) The node-relation structure model of the simple node relationship with extended nodes.

Figure 2. A 3D building model and its hierarchical relation structure.

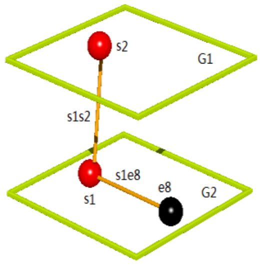


Figure 3. The abstracted supergraph, with each story abstracted to its boundary nodes.

Here are the relationships:
SG = (Gc, G)
G = (G1, G2)
G1 = (V1, E1), V1 = {s2}, E1 = {○}
G2 = (V2, E2), V2 = {s1, e8}, E2 = {s1e8}
Gc = (Vc, Ec), Vc = {s1, s2}, Ec = {s1s2},
Where
SG is supergraph set, Gc is a supergraph, G is a subgraph set, and G1 and G2 are subgraphs.
s1, s2 and e8 are boundary nodes.
s1s2 is the connectivity between the supergraph's boundary nodes.

We seek an optimum route whose total distance is the minimum to the destination node. The global optimum node is the neighborhood node of the current location in the optimum route.

2.3. Conflict Problem Optimization

In the process of evacuation, there is competitive behavior between the particles. The perception of hazards can stress people in crowds, evoke their competitive response, and trigger blocking as they attempt to pass through narrow passages [3] (e.g., a small exit) simultaneously. We propose a mechanism to manage competitive behavior in three manners as shown in Figure 4.

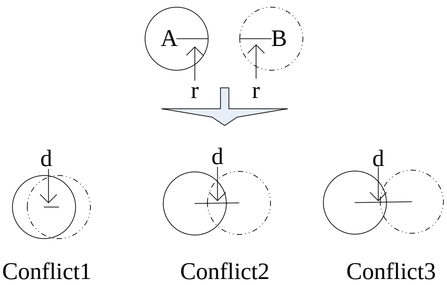


Figure 4. Three competitive behaviors between the particles.

We determine three types of conflict with the distance between the particles. r is radius of particle and d is the distance between the center of a circle of two particles. That is

$$\begin{cases} if \quad 0 < d < r/2, & then \quad \mathrm{Conflict}Method1 \\ if \quad r/2 \le d < 5r/4, & then \quad \mathrm{Conflict}Method2 \\ if \quad 5r/4 \le d < 2r, & then \quad \mathrm{Conflict}Method3 \end{cases} \tag{4}$$

a. *ConflictMethod1*: There are five particles to compete a position in the first type of conflict problem (see Figure 5). The red balls are evacuees and the yellow circle is an empty location. We put forward six strategies to solve the conflict between the particles as shown in Figures 5-a, 5-b, 5-c, 5-d, 5-e, and 5-f).

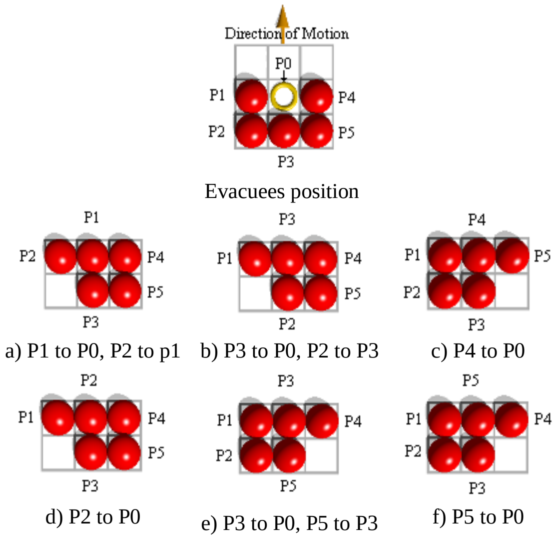


Figure 5. Study of the competitive behavior of ConflictMethod1.

*P*1…, *P*5 are five conflict particles in Figure 5, and their movement strategies comply with the following combination rules in Table 1.

Table 1. Constraint pattern for conflict method1

Constraint	Description
When P2 or P5 move to position empty location.	P1, P3 and P4 stay in the same place
When P1 or P4 move to position empty location.	The optimal position for P2 and P5 are P1' position and P4' position, respectively.
When P3 move to position empty location.	P2 or P5 move to P3'position.

We calculate the probability of each particle and form a probability table for each movement strategy. We tabulate the probability according to particle velocity s and particle number d behind the current particle. The probability function can be written as:

$$p(s_i, d_i) = P(s_i) * P(d_i) \tag{5}$$

$$P(s_i) = \frac{s_i}{\sum_{j=1, i \neq j}^{n} s_j} \tag{6}$$

$$P(d_i) = d_i / \sum_{j=1}^{n} d_j \tag{7}$$

According to the probability table, we adopt the minimum entropy principle to decide which strategy to utilize. Conflict Method1 sets $S=\{S_1, S_2, \ldots, S_n\}$ as a collection of mutually exclusive natural states. The collection of subjective probability distributions is defined as $\Delta S = \{P_{i.} = (p_{i1}, \cdots, p_{in}) \mid \Pr_i(s_j) = p_{ij}, i = 1, \cdots, m;$ $j=1, \ldots, n\}$. This is a finite set, where $Pr_i(s_j)$ is the i^th^ subjective probability about strategy s_j. The modeling process consists of the following four steps.

1. Compute information entropy, which is given by $H(P_i) = -\sum_{j=1}^{n} p_{ij} \ln p_{ij}, i = 1, 2, \cdots, m$
2. If $\min_{1 \le i \le m} H(P_i.) = H(P_{i_0}.)(1 \le i_0 \le m)$, then P_{i_0} is outcome

   ;

   If $\min_{1 \le i \le m} H(P_i.) = H(P_{i_1}.) = \cdots = H(P_{i_r}.) = s(r \ge 2)$. then

   compute $P_{i_0} = \sum_{k=1}^{r} Pik./r$
3. If $H(P_{i_0}) < s$, P_{i0} is outcome;

   If $s \le H(P_{i_0}) \le s + \ln r$, then randomly choose i_t and

   set P_{it} as outcome in i1， i2…ir(1≤t≤r).
4. Set movement strategy of particles with outcome and combination rules.

b. *ConflictMethod2*： PSO calculates the movement positions of the particles in the next iteration. When the predicted position of two particles meet a collision condition, ConflictMethod2 first evaluates their priority according to the fitness distance, then computes the optimal position of the particles.

In Figure 6, P_a and P_b are two particles, r is the particle radius, P_b' is the predicted position of P_b. r1 is the fitness distance, r2 equals 2r, r4 equals the velocity of particle P_b', and r3 equals the difference between r4 and r. The points (c, d) and (e, f) are the centers of the circle of two particles. S is the particle's center in non-conflict movement area. The ConflictMethod2 is the following.

Algorithm 4: The ConflictMethod2

1. *Initialization*

   The coordinates for Pa and Pb are C (c,d) and D (e,f) respectively.

   Set (a,b) as point M.

   The radius of the purple circle is r_1.

   Read parameter values, particle radius r, and particle velocity v.
2. *Optimal coordinate*

   Get point coordinates (a,b) for the shortest distance between the particle and the Exit. The point at the particle's center is a distance "r" from the boundary of the Exit.
3. *Point of intersection*

   Define A(x1, y1) and B(x2, y2) as the two points of intersection between Pa' and S. Then

$$\begin{cases} (x-c)^2 + (y-d)^2 = r_2^2 \\ (x-e)^2 + (y-f)^2 = r_3^2 \end{cases} \tag{8}$$

   A point's coordinates E are $((x_1 + x_2)/2, (y_1 + y_2)/2)$.

4. *Negotiation rounds*

   Calculate the distance $\frac{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}}{2}$ *between A and B.*

(a)Calculate the angle θ between AD and DE. If $0<\theta<\arcsin\frac{\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}}{2r_3}$, then A is the optimal point.

(b)Calculate the angle θ between CD and DB. If $\arcsin\frac{\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}}{2r_3}<\theta<2\arcsin\frac{\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}}{2r_3}$, then B is optimal point.

5. Point of tangency

If A and B are not the optimal points, then ConflictMethod2 derives the tangency point N(x_n,y_n) between S and the purple circle. The tangency point N is the optimal point as shown

$$\begin{cases}(x_n-e)^2+(y_n-f)^2=r_3^2\\ \sqrt{(x_n-a)^2+(y_n-b)^2}+r_3=\sqrt{(a-e)^2+(b-f)^2}\end{cases} \tag{9}$$

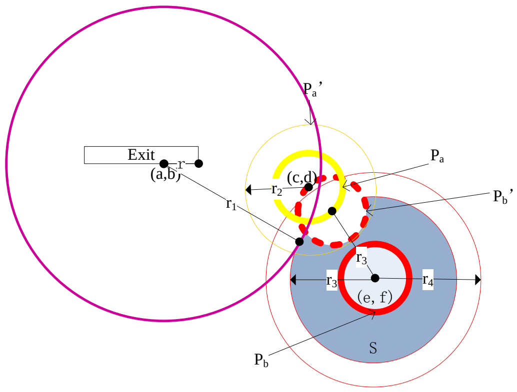


Figure 6. Study of the competitive behavior of ConflictMethod2.

c. *ConflictMethod3*: We set a yellow circle at the position of the fixed particle P_a. P_b' is the optimized location of P_b. The fitness value of P_a is less than that of P_b. The particle center coordinates for P_a and P_b are A (a,b) and B (c,d), respectively.

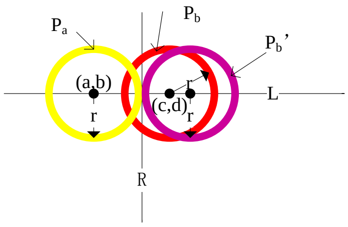


Figure 7. Study of the competitive behavior of ConflictMethod3.

As shown in Figure 7, L is a straight line joining A and B, and R is tangent to P_a and perpendicular to L. Particle P_b' is tangent to particle P_a and line R, whose center coordinate is on line L.

Algorithm 5: The ConflictMethod3

1. Initialization

Read parameter values A and B.

2. Tangent coordinates

Compute the tangent coordinates (x1, y1).

$$\begin{cases}y_1=-\dfrac{(a-c)}{(b-d)}(x_1-a)+b\\ (x_1-a)^2+(y_1-b)^2=r^2\end{cases} \tag{10}$$

3. Optimized coordinates

Compute the center coordinates $(2x_1-a,2y_1-b)$ of P_b' based on the results of Step 2.

3. Discussion and Results

Numerical testing is presented using three scenarios. The first scenario uses a single room to compare our GLPSO model with the PSO [18] and Agent [6] strategies. This experiment explained the different evacuation times estimated by three evacuation methods in microscopic simulations. The second scenario uses a larger layout and compares our optimization-based strategies with the strategy using PSO [18] based on the nearest exits and Game theory [8]. The third scenario uses a multi-room multi-floor building to compare our evacuation planning with PSO for crowed movement [18].

- *Scenario 1*. This example studies an egress scenario in which a group of pedestrians is guided to exits within a single room (as shown in Figure 8).

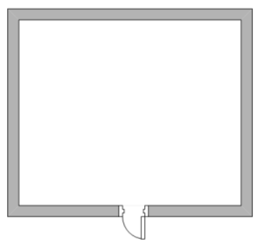


Figure 8. The single room.

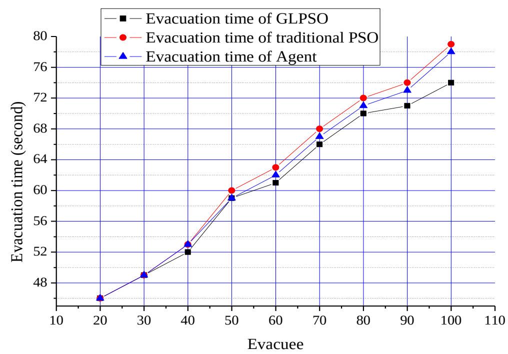


Figure 9. The evacuation time vs. evacuee.

This experiment is to test the validity of the evacuation time. Figure 9 shows the evacuation time of each method. The black curve is the average evacuation time for the GLPSO method. The red curve is the average evacuation time for the Agent method. The blue curve is the average evacuation time for the traditional PSO method. With the increase of the number of evacuees, the average evacuation time for the GLPSO model is obviously smaller than the other two algorithms.

- *Scenario 2*. This scenario studies an egress scenario in which four groups of people are guided to three exits within a small planar layout (as shown Figure 10).

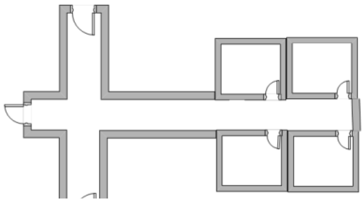


Figure 10. Exiting pattern for a 4-room/1-corridor layout.

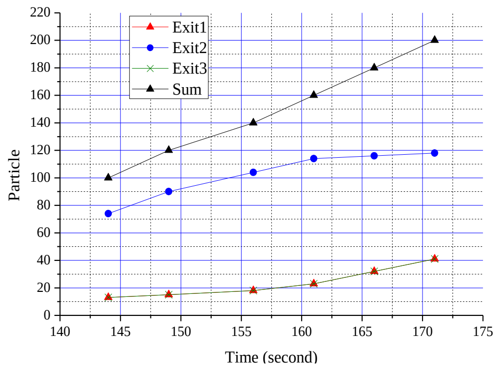


Figure 11. Optimal allocation of particle to the three exit by GLPSO.

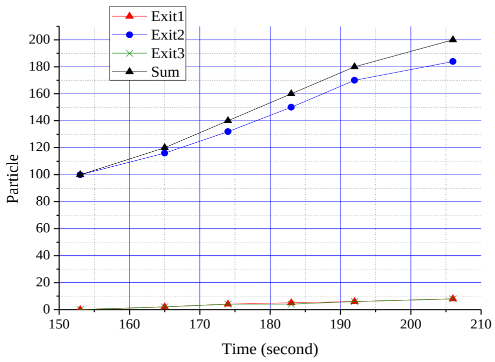


Figure 12. Optimal allocation of particle to the three exit by the nearest exits.

This experiment is to test the optimal choice for multi-exit scenarios. One of the main objectives of forecasting evacuation times is the optimization of the allocation of people and areas to the various available exits. Figures 11, 12, and 13, are the optimal allocations of each method in a plan. The evacuation times to Exits 1, 2, and 3 are shown in the red, blue, and green curves, respectively. The black curve is the total evacuation time for the GLPSO strategy.

Figures 11, 12, and 13 show evacuation planning for the three algorithms. The results show that the different of the optimal allocation of particles between Exit1 and other exits increases with increase in the number of particles. On the other hand, GLPSO and Game theory exhibit a slowly-rising trend as a function of particle number. The evacuation time estimated by GLPSO is less than Game theory. Therefore, in total, the plan made by GLPSO is better than that of the other two algorithms.

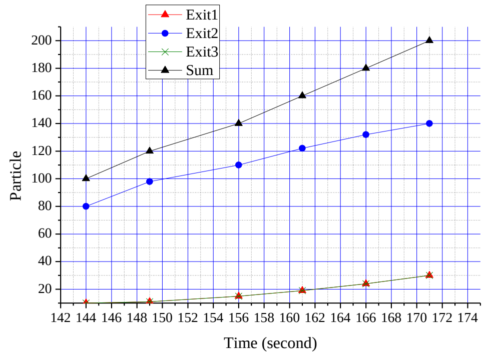


Figure 13. Optimal allocation of particle to the three exit by game theory.

- *Scenario 3.* This example studies an egress scenario in which 780 pedestrians are guided to exits within a multi-room, multi-floor building (as shown in Figure 16).

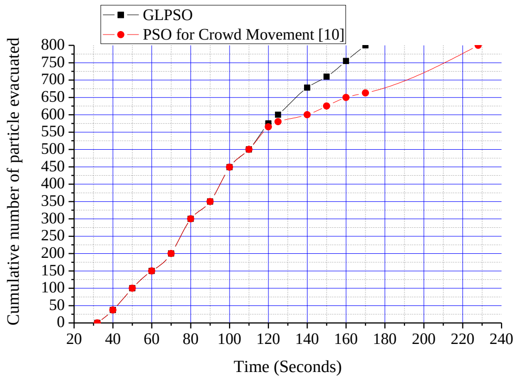
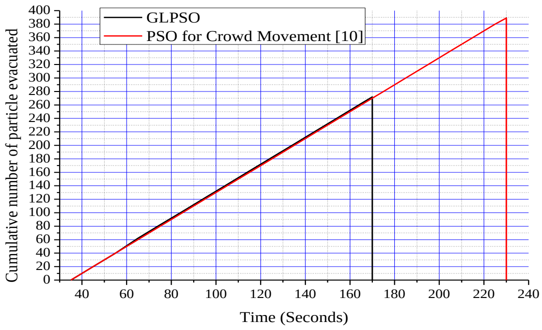


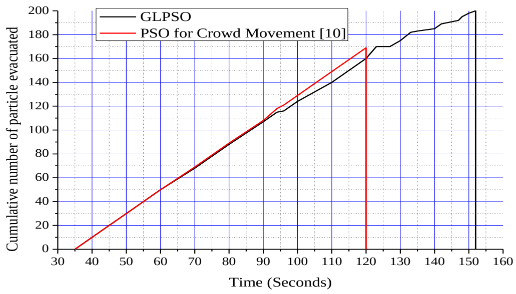
Figure 14. Evacuation curves of two algorithms.

Figure 14 shows the evacuation curves of two algorithms. For GLPSO, 100% of particles have been evacuated out of the building at 170 seconds. However, using PSO for Crowd movement model, 229 seconds elapsed before all particles were evacuated. The cumulative number of particles evacuated is different between the two models by 112 seconds. This is a result of GLPSO taking full advantage of multiple exits to evacuate particles.

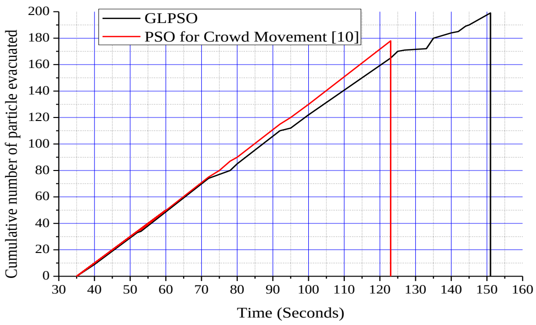
Figure 15 shows the evacuation time for each exit in a plan. In the case of the GLPSO algorithm, the simulation output shows an evacuation plan with a maximum evacuation time of 170 seconds at Exit 1, and a minimum evacuation time of 140 seconds at Exit 4. The PSO algorithm results in a maximum evacuation time of 230 seconds at Exit 1 and a minimum evacuation time of 114 seconds at Exit 4. Obviously, a reasonable planning can take full advantage of multiple exits and drastically improve the evacuation time. The simulation output of GLPSO provides a more rational result than PSO for crowd movement model because the different values and the maximum evacuation times are smaller.



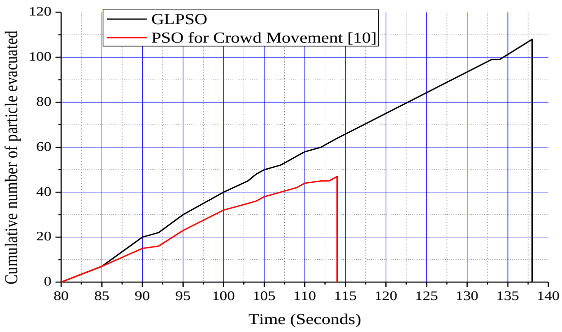
a) Graph compares the time to the evacuation number to pass exit1.



b) Graph compares the time to the evacuation number to pass exit2.



c) Graph compares the time to the evacuation number to pass exit3.



d) Graph compares the time to the evacuation number to pass exit4.

Figure 15. Cumulative evacuees passing through exit (Allocation of 780 people to four exits).

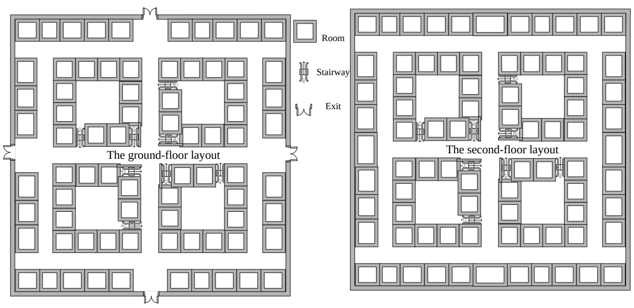


Figure 16. An egress structural layout.

4. Conclusions

In summary, we have presented a GLPSO which combines the filled function method and the PSO algorithm for the evacuation routing optimization problem in complex scenarios. This method seeks to find the global minimum of the evacuation time using filled function methods. The improved NRS method obtains a global optimum node and a local optimum node with the global optimum particle distribution. The two nodes guide particles to their own best-known positions in the search-space from a macroscopic point of view. We have demonstrated that under different scenarios, GLPSO takes full advantage of multiple exits to reduce evacuation and congestion times. In addition, we proposed three methods to manage the competition for space in GLPSO. These methods simulate particle movement and optimize competition behavior on the micro level. GLPSO has been established to examine how the rational evacuation planning of the evacuees will affect the evacuation process. Our results provide compelling evidence for a global/local optimization in emergency evacuation, which are effective in maximizing the evacuation efficiency and optimized competitive behavior. Further works will need to examine the effect of familiarity and environmental stimuli as well as accident prevention effect on multi-exit selection.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (Grant No. 61202307), Natural Science Research Foundation of Jilin Province of China (20140101181JC), Natural Science Research Foundation of Jilin Province of China (20140520071JH).

References

[1] Chooramun N., Lawrence P., and Galea E., "An Agent Based Evacuation Model Utilising Hybrid Space Discretisation," *Sagety Science*, vol. 50, no. 8, pp. 1685-1694, 2013.

[2] Ha V. and Lykotrafitis G., "Agent-Based Modeling of a Multi-Room Multi-Floor Building Emergency Evacuation," *Physica a-Statistical Mechanics and Its Applications*, vol. 391, no. 8, pp. 2740-2751, 2012.

[3] Hoogendoorn S. and Bovy P., "Simulation of Pedestrian Flows by Optimal Control and Differential Games," *Optimal Control Applications and Methods*, vol. 24, no. 3, pp. 153-172, 2003.

[4] Izquierdo J., Montalvo I., Pérez R., and Fuertes V., "Forecasting Pedestrian Evacuation Times by using Swarm Intelligence," *Physica A*, vol. 388, no. 7, pp. 1213-1220, 2009.

[5] Junaedi H., Hariadi M., and Purnama I., "Multi Agent with Multi Behavior Based on Particle Swarm Optimization (PSO) for Crowd Movement in Fire Evacuation," *in Proceeding of 4th International Conference on Intelligent Control and Information Processing*, Beijing, pp. 366-372, 2013.
[6] Liu D. and Nocedal J., "On the Limited Memory BFGS Method for Large Scale Optimization," *Mathematical Programming*, vol. 45, no. 3, pp. 503-528, 1989.
[7] Lee J. and Kwan M., "A Combinatorial Data Model for Representing Topological Relations among 3D Geographical Features in Micro-Spatial Environments," *International Journal of Geographical Information Science*, vol. 19, no. 10, pp. 1039-1056, 2005.
[8] Manley M. and Kim Y., "Modeling Emergency Evacuation of Individuals with Disabilities (Exitus): An Agent-Based Public Decision Support System," *Expert Systems with Applications*, vol. 39, no. 9, pp. 8300-8311, 2012.
[9] Parlak A., Lambert J., Guterbock T., and Clements J., "Population Behavioral Scenarios Influencing Radiological Disaster Preparedness and Planning," *Accident Analysis and Prevention*, vol. 48, no. 9, pp. 353-362, 2012.
[10] Pan X., Han C., Dauber K., and Law K., "Human and Social Behavior in Computational Modeling and Analysis of Egress," *Automation in Construction*, vol. 15, no. 4, pp. 448-461, 2006.
[11] Raman B. and Subramani S., "An Efficient Specific Update Search Domain based Glowworm Swarm Optimization for Test Case Prioritiz," *The International Arab journal of Information Technology*, vol. 12, no. 6A, pp.1-7, 2015.
[12] Renpu G., "A Filled Function Method for Finding a Global Minimizer of a Function of Several Variables," *Mathematical Programming*, vol. 46, no. 1-3, pp. 191-204, 1990.
[13] Shi D. and Wang B., "Evacuation of Pedestrians from a Single Room by using Snowdrift Game Theories," *Physical Review E*, vol. 87, no. 2, pp. 0228021-0228026, 2013.
[14] Sewall J., Wilkie D., and Lin M., "Interactive Hybrid Simulation of Large-Scale Traffic," *ACM Transactions on Graphics*, vol. 30, no. 6, pp. 1-18, 2012.
[15] Tanimoto J., Hagishima A., and Tanaka Y., "Study of Bottleneck Effect at an Emergency Evacuation Exit Using Cellular Automata Model, Mean Field Approximation Analysis, and Game Theory," *Physica a-Statistical Mechanics and its Applications*, vol. 389, no.24, pp. 5611-5618, 2010.
[16] Yuan Y., "A Modified BFGS Algorithm for Unconstrained Optimization," *IMA Journal of Numerical Analysis*, vol. 11, no. 3, pp. 325-332, 1991.
[17] Zhang L., Ng C., Li D., and Tian w., "A New Filled Function Method for Global Optimization," *Journal of Global Optimization*, vol. 28, no. 1, pp. 17-43, 2004.
[18] Zheng X. and Cheng Y., "Conflict Game in Evacuation Process: A Study Combining Cellular Automata Model," *Physica a-Statistical Mechanics and its Applications*, vol. 390, no. 6, pp. 1042-1050, 2011.
[19] Zheng X. and Cheng Y., "Modeling Cooperative and Competitive Behaviors in Emergency Evacuation: A Game-Theoretical Approach," *Computers and Mathematics with Applications*, vol. 62, no. 12, pp. 4627-4634, 2011.
[20] Zheng X., Tingkuan Z., and Mengting L., "Modeling Crowd Evacuation of a Building Based on Seven Methodological Approaches," *Building and Environment*, vol. 44, no. 3, pp. 437-445, 2009.
[21] Zheng Y., Chen J., Wei J., and Guo X., "Modeling of Pedestrian Evacuation Based on the Particle Swarm Optimization Algorithm," *Physical A*, vol. 391, no. 17, pp. 4225-4233, 2012.

Jin-long Zhu was born in 1984. He received the B.E. and M.S. degree in Computer Science and Technology from Jilin University, China, in 2003 and 2007 respectively. His research interest in the area of digital image processing and virtual reality technique. And his research interests include computer algorithms and simulation, evacuation planning, image processing and 3D modelling.

Wen-hui Li is a Professor of Jilin University. He received the B.E., M.S. and PhD degree in Computer Science from Jilin University, in 1984, 1987 and 1996 respectively. His research interests include computer vision, image processing and computer graphics.

Huiying Li is a Lecturer of Jilin University, China. She received the B.E., M.S. and PhD degree in Computer Science and Technology from Jilin University, China, in 2001, 2004 and 2008 respectively. She was a visiting scholar to the Department of Electrical and Computer Engineering at University of Hannover from 2007 to 2009. Her research interests include computer algorithms and simulation, image processing and 3D modelling.

Qiong Wu was born in 1990. She received the B.E. degree from Changchun University of Information and Computing Science in 2009. And he is now pursuing the M.E. degree in the Computer Application Technology, Jilin University of Computer Science and Technology from 2013. His research interests include digital image processing and virtual reality technique. And he is now studying for evacuation planning.

Liang Zhang was born in 1989. He received the B.E. degree from Changchun University of Information and Computing Science in 2009. And he is now pursuing the M.E. degree in the Computer Application Technology, Jilin University of Computer Science and Technology from 2013. His research interests include digital image processing and virtual reality technique. And his research interests include computer algorithms and simulation, evacuation planning, image processing and 3D modelling