UDC 004.891
DOI: 10.15587/1729-4061.2022.252917

THE DEVELOPMENT OF FIREFLY ALGORITHM WITH FUZZY LOGIC INTEGRATION FOR PRIORITY SEARCH SIMULATION OF FLOOD EVACUATION ROUTES

T. Brenda Chandrawati
Corresponding author
PhD Student*
E-mail: t.brenda@ui.ac.id

Anak Agung Putri Ratna
Professor*

Riri Fitri Sari
Professor*

*Department of Electrical Engineering
Universitas Indonesia
Jl. Margonda Raya, Pondok Cina,
Kota Depok, Indonesia, 16424

Heavy rain in a particular area can cause flooding in both the primary area and the surrounding area. A flood is an event where water is inundated in an area due to increased water volume. Due to high level of water and other hazards arising from flooding, flood victims need to move to a location prepared for evacuation. To get to that location prepared, the victims must get through a safe route. Searching for safe evacuation routes is important to save flood victims and bring them to the evacuation centre safely. Search for evacuation routes related to obstacles on the road to get through. Slippery roads, high puddles of water on the roads, rivers that are located close to the roads that flood victims will have to get through, drainage of waterways and the vulnerability of victims are taken into consideration in choosing a route to get to the evacuation location. There are several problems in choosing a safe route: (1) how to take into account the obstacles on the road to be passed (2) how to choose the priority of the route to be passed with the obstacles encountered. The proposed solution to deal with the problems encountered are (1) to take into account road obstacles by giving the obstacle weights. Fuzzy logic is used to calculate the value of obstacle weights (2) the problem of selecting route priorities will be solved using the firefly algorithm. The firefly algorithm is an algorithm inspired by the social life of fireflies. The priority route for evacuation of flood victims is sought using the method proposed in this study which is the optimal route. The optimal route referred to in this study is the route that has the smallest obstacle weight value. The simulation results show that the fuzzy logic integrated into the firefly algorithm (FuFA) provides a safe route priority, indicated by the smallest obstacle weight value

Keywords: flood, evacuation routes, weight, obstacle, fuzzy logic, Firefly algorithm, priority route, optimal route, safe route, FuFA

Received date 03.01.2022
Accepted date 18.02.2022
Published date 25.02.2022

***How to Cite:** Chandrawati, T. B., Ratna, A. A. P., Sari,R. F.(2022). The development of Firefly algorithm with fuzzy logic integration for priority search simulation of flood evacuation routes. Eastern-European Journal of Enterprise Technologies, 1 (4 (115)), 66–76. doi: https://doi.org/10.15587/1729-4061.2022.252917*

1. Introduction

Flood disaster is one of the disasters that often hit many developed and developing countries. A flood is a natural disaster that has the potential to cause extensive damage to life and property, both physical damage, economy, and loss of life. Flood events encourage the creation of effective evacuation strategies to prevent the loss of human life. One of the effective evacuation strategies to prevent the loss of human life during a flood is to find the right route when evacuating.

Optimization of flood evacuation routes is an essential component of flood mitigation. Previous studies have stated that route optimization is a top priority in reducing flood impacts [1]. The simulation model for selecting the optimal route can be used to increase public awareness of safety in the event of a flood disaster. Several attempts have been made to develop optimizations to search for the shortest route, using the Dijkstra algorithm [2] and the ant algorithm [3].

By imitating the social behaviour of fireflies, the simulation model can imitate the important aspects of fireflies to find route priorities during flooding. Therefore, studies are devoted to developing a route priority search method based on the movement of fireflies in the search for fireflies that have a brighter light.

2. Literature Review and Problem Statement

The work [2] suggests a safe route by considering several safety factors during an earthquake. Factors considered in the path selection are street vulnerability, street blockages probability, crowding conditions along paths, presence of mortal dangers, visibility conditions. The algorithm used for path selection is Dijkstra's algorithm. This path selection algorithm is a conventional algorithm.

The work [4] proposes a new mathematical model for efficient and effective evacuation planning during flood disasters. This mathematical model is used to decide the optimal number of evacuation locations and shelters, assigning people to shelters, assigning helicopters to shelters, determining helicopter routes to rescue people and the capacity of people who can be rescued to shelters. The weakness of this study is that it does not take into account the limited number of helicopters for evacuation. The limited number of helicopters results in longer waiting times for rescue.

The work [5] proposes a safe and less congested route priority setting. Unpredictable traffic conditions have considered the selection of evacuation routes because the proposed rescue efforts for refugees use vehicles. Traffic density during the evacuation process will impact refugee anxiety and actions that ignore the evacuation protocol. Motorized vehicles are an obstacle if the disaster location is in an area with certain limitations.

In addition, in the last few decades, nature-inspired algorithms have been developed which are used as algorithms to find routes. The work [6] used the firefly algorithm to approach mobile robot navigation based on FA with the static obstacle. The firefly algorithm aims to find the shortest path safe from obstacles and has smoothness in achieving the goal. The work [6] has the advantage of using a realistic environmental map.

The work [7] used the firefly algorithm to find a collision-free path with the shortest distance for navigation of a single mobile robot. The disadvantage of work [7] is the environment used for the simulation is only rectangular and the barrier is presented as a black area in the form of a straight line.

The work [8] used the firefly algorithm for underwater robot navigation to schedule robotic swarm communications. Phase synchronization inspired by the flickering of fireflies is used as reference signals in communication. The frog call anti-phase synchronization method is used to allocate time slots according to the reference signal used for communication. Scheduling for the allocation of time slots is intended to avoid interference with communication and robot sensing.

The original firefly algorithm and the improved firefly algorithm can also plot UCAV paths. UCAV path planning aims to calculate optimal flight routes and to plan paths in complex environments, to breakthrough a hostile environments, avoid dangerous areas, and minimize fuel to complete the mission perfectly [9].

Work [10] developed a modified firefly algorithm for robot navigation, namely finding the optimal path and avoiding obstacles. Obstacle avoidance is performed by a robot based on a concentric ball-based geometric technique. The robot moves in an unknown and uncertain environment. The environment used for simulation has constraints and does not have obstacles.

The work [11] introduced compact firefly algorithms (cFA). The development of cFA, apart from being a new compact computing technique based on the firefly algorithm, also has other goals, namely reducing the complexity of the attraction model of firefly algorithms, reducing the computational capacity of classic firefly algorithms, reducing memory storage for swarms, integrating opposition-based learning into compact optimization, developing compact herd intelligence. and uniform distribution for implementation of compact evolutionary algorithms (cEA).

The work [12] presents the application and implementation of the Firefly Algorithm (FA) for Mobile Robot Navigation (MRN) in uncertain surroundings. An uncertain environment is defined as a change from static to dynamic environmental conditions.

The work [13] introduced a new variant of the vehicle routing problem (VRP), namely the Environmental Prize-Collecting Vehicle Routing Problem (E-PCVRP). E-PCVRP maximizes the aggregate reward value collected from visited nodes by minimizing fixed and variable costs. Variable costs are CO^2 emissions produced by the vehicles used.

The work [14] developed a local route plan for an unmanned vessel in a cluttered environment based on the Improved Firefly Algorithm. This aims to optimize the local route, which is an important guarantee for unmanned vessels so that maritime safety is maintained.

Analysis of work [7–14] shows general deficiencies, namely:

– the environment used for simulation is an environment that does not describe the real situation;

– the obstacle presented is an object that must be avoided;

– optimization is done more on the shortest route search.

Therefore, this research will discuss the firefly algorithm development, which is used to simulate the search for priority routes in the evacuation of flood victims. The priority of the selected route is the safest route.

3. The Aim and Objective of the study

This study aims to develop the firefly algorithm with fuzzy logic integration to have simulation in finding a safe route for flood victims to the evacuation site.

To achieve this aim, the following objectives are accomplished:

– to design a modelling environment used for simulation;

– to develop obstacles and their weights on each route;

– to develop fuzzy logic that is used to calculate the weight of the route obstacle;

– to develop an improved algorithm to implement this case.

4. Material and Method

The proposed research seeks to develop a firefly algorithm with fuzzy logic integration that can be used to ensure a safe route in the simulation of finding a priority route for flood victims. A safe route is assumed to have obstacles that flood victims can face in the process of evacuation.

This algorithm will look for a route that has a low obstacle weight so that flood victims can safely go to the evacuation location. A weight value expresses the obstacle of each alternative route. The calculation of obstacle weight for each route is calculated using fuzzy logic.

This study uses several methods to solve the problem in finding a safe route priority for flood victims. Fuzzy logic is used to give an obstacle weight value on each path. At the same time, the obstacle weight of a route is calculated by adding up all the path weight values on the route. The development of the firefly algorithm is used to search for route priorities based on the low weight value of the route obstacle. Firefly algorithm performance simulation in route priority search using Matlab 2016a software.

The firefly algorithm is a search algorithm based on the social behaviour of the firefly population. The work [15] developed a firefly algorithm based on the social behaviour of fireflies. Fireflies live in warm environments and become more active on hot summer nights. This insect has a flash of light. This flash of light is the result of a biochemical process called bioluminescence. The flashing light on fireflies serves to attract other fireflies to mate. Usually, the male is the first signaler to attract the flightless female. The female will blink continuously in response to the signal given by the male. This pair of mating fireflies will have different flash signal patterns that are timed precisely. This set of flash signal patterns is intended to provide information about species identity and gender. Usually, female fireflies favour the male firefly's brighter flashes of light. On the other hand, a firefly's blinking light serves to warn potential predators nearby. The firefly's flashes of light are very striking to drive predators away [16].

The classic firefly algorithm uses three main rules that simulate the insect perpetrator:

1. All fireflies are unisex, and all can attract one another regardless of their gender.

2. The attractiveness of a firefly is directly proportional to its brightness. If two fireflies are blinking, the less bright firefly will move towards the lighter one. The attractiveness and brightness decrease as the distance between the two fireflies increases. If none of the fireflies is brighter than others, the last firefly will move randomly.

3. The brightness of the firefly is influenced or determined by the landscape of the objective function.

The search pattern in the firefly algorithm is determined by the intensity of the light emitted by each firefly. The less bright firefly will move towards the lighter firefly. Thus, each firefly in the population represents a candidate solution in the search space. Therefore, Firefly is on the move to look for potential solution candidates.

There are two significant variables in the firefly algorithm, namely light intensity and attractiveness. The formula can calculate their light intensity:

$$I(r) = I_0 e^{-\gamma r^2}, \tag{1}$$

where I_0 is the initial light intensity, γ is the light absorption coefficient, and r is the distance between the fireflies i and j.

While the attraction of fireflies will be proportional to the intensity of light seen by nearby fireflies, the attraction of fireflies' β can be defined by the formula:

$$\beta = \beta_0 e^{-\gamma r^2}, \tag{2}$$

where β_0 is the attractiveness of fireflies when $r=0$.

The distance between two fireflies i and j can be defined by the Cartesian space:

$$r_{ij} = x_i - x_j = \sqrt{\sum_{k=1}^{d} \left(x_{i,k} - x_{j,k}\right)^2}. \tag{3}$$

The movement of firefly i, which is attracted to lighter firefly j, determined the formula:

$$x_i = x_i + \beta_0 e^{-\gamma r^2_{ij}} \left(x_j - x_i\right) + \alpha\left(\text{rand} - \frac{1}{2}\right). \tag{4}$$

The second term in equation (4) is the part that shows interest. The third term is randomization. α is the randomization parameter, and the rand in the equation is a random value distributed from 0 and 1. If the rand value is considerable, it will search for distant places, and if it is small, it tends to search for local searches.

There are two key things in the firefly algorithm, namely light intensity and attractiveness. These are closely related to the distance between fireflies, r, as shown in formulas (1) to (4). The light seen by one firefly will be brighter if the two fireflies are close to each other. On the other hand, if the two fireflies are farther apart, the light received by each firefly will be dimmer. This is because the firefly will approach another firefly that has excellent traction. However, according to Yang, the specified r is not only limited to the Euclidean distance but r can be defined in n-dimensional hyperspace, depending on the problem at hand. r can be defined as a time lag or time interval when the problem is related to job scheduling. The 'distance' r can describe any measure that effectively characterizes the optimization problem's interest [17]. In finding a safe route for flood victims, r is defined as the weight of the obstacle calculated using fuzzy logic.

5. Results of experimental result development of fuzzy logic integration in firefly algorithm for priority search of flood evacuation routes

5. 1. Designing an environment model for simulation

Searching for safe evacuation routes for flood victims requires a visual environment used as a simulation: road model and obstacle design. Fig. 1 shows the evacuation route environment used to simulate the safest route search.

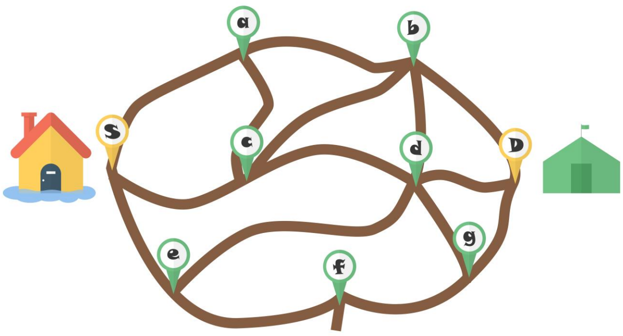


Fig. 1. Route model that flood victims can pass [18]

From Fig. 1, there are 24 alternative routes that flood victims may pass. The starting point is indicated by S. This point S is assumed to be a flood-affected location. The destination point is indicated by point D. This D point represents the evacuation location determined. a, b, c, g indicate the end and intersection of the road. Each route consists of several paths. Every path that flood victims will pass has an obstacle that prevents flood victims from passing that route to get to the evacuation location determined. Flood victims will go through a route that can be passed safely to safely arrive at the evacuation location that has been specified.

5. 2. Developing obstacle model

Each path in Fig. 1 has obstacles, namely slippery roads, high puddles, the road's distance from the river, the speed of water flow in the road channel, the vulnerability of victims affected by flooding. The level of slippery roads and the height of the puddle on each path will be different. The rivers around the road that flood victims will pass are also an obstacle to the safety of flood victims who will cross the street. Likewise, with the speed of water flow in the channel on the side of the road. Clogged drains by garbage or sedimentation can cause the non-smooth flow of water in this channel. The vulnerability of victims is also a consideration in the effort to choose a safe route. Obstacles on each road will hinder traffic and slow down flood victims to reach the evacuation site:

1. Slippery road. Slippery roads are caused by dust, sand, soil, mud, water, and garbage scattered on the road surface. When it rains, the sand and dust are lifted to the asphalt surface by the water. Mud, soil, and garbage will also be carried away by running water and covering the road surface.

2. Stagnant water on the road. The level of stagnant water on the road is influenced by rainfall and the existing drainage system. Therefore, flood victims will have difficulty crossing the road if the stagnant water increases.

3. The river nearby the road. The flow and water level of the river are also affected by the level of rainfall. The higher the rain, the higher the water level in the river and the faster the river currents. This condition is correlated with the level of danger in the location around the river. If the river near the road reaches a particular water level, the road will be too risky to pass. The distance between the river and the road considers whether the road can be crossed or not by flood victims.

4. Drainage system. Drainage can be defined as removing water masses naturally or artificially by draining, draining, disposing, and diverting water. In spatial planning, the drainage system has a vital role in regulating water supply to prevent puddles that cause flooding. The better the road drainage system, the more stagnant water will recede.

5. Vulnerability. Vulnerability is a set of conditions determined by physical, socio-economic, and environmental factors that increase a community's propensity to be affected by a hazard. Vulnerability emphasizes the human aspect at the community level, which is directly faced with the threat of danger. Vulnerability in a social order is a major factor at a higher risk if the community has capabilities such as lack of education and knowledge, poverty, low social conditions, and vulnerable groups, including the elderly, toddlers, pregnant women, and physical or mental disabilities.

A code is given for each obstacle to simplify the calculation, namely C1, C2, ..., C5. Table 1 shows the obstacle code for each path.

Table 1

Obstacle code

No.	Code	Explanation
1	C1	Road slippery level
2	C2	The height of the puddle on the road
3	C3	The road's distance from the river
4	C4	The rate of discharge of water in the road channel
5	C5	Vulnerability of flood victims

This code holds certain information and makes it easier to remember the obstacle. Each path has obstacles. Obstacles that have been determined on each path will be given weight. The path with a greater weight means that the path is relatively more dangerous than other paths. The objective of weighting obstacles in this simulation is to state the level of danger of obstacles faced by flood victims against other obstacles.

The obstacle weight of each path is calculated using fuzzy logic. Fuzzy logic is a logical approach that contains uncertainty and can model non-linear functions, has tolerance for inaccurate data and is based on natural language.

5. 3. Development of Fuzzy Logic to calculate route obstacle weights

The design of the obstacle weight calculation with the Mamdani-type of fuzzy logic, in general, can be seen in Fig. 2.

Three steps in determining the obstacle weight of each path with Mamdani type fuzzy logic are as follows:

1. Fuzzification. The fuzzification process, in this case, serves to change the predetermined analogue quantities into fuzzy inputs. The fuzzy system will take information and determine the degree of membership in all fuzzy sets by using the membership function of each fuzzy set. In the fuzzification process, non-fuzzy variables (numeric variables) are converted into fuzzy variables (linguistic variables).

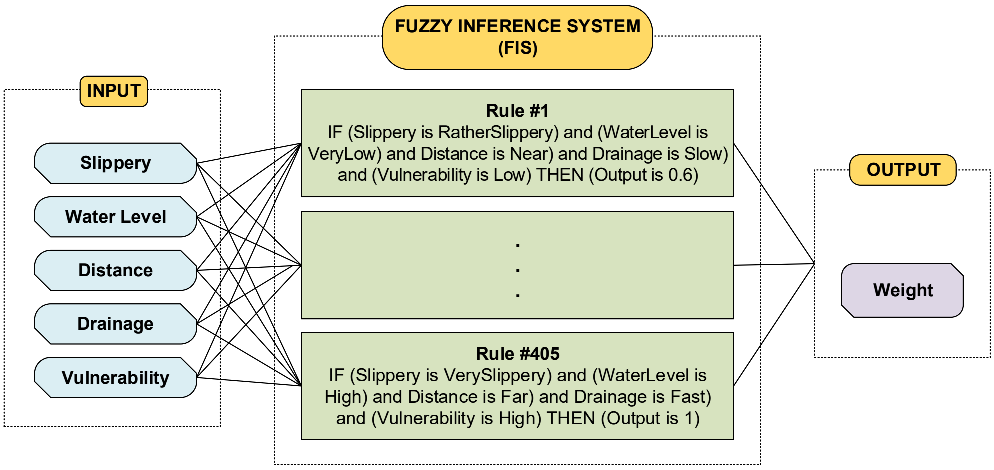
The fuzzy system in this research was designed with 5 (five) inputs, namely: the level of road slippery, the level of stagnant water, the distance to the nearest river, the speed of water disposal in the drainage channels on the road, and the vulnerability of flood victims:

a) the road slippery criterion. The value of the input linguistic variable on the road slippery level criterion is divided into 3 (three) membership functions, namely RatherSlippery, Slippery and VerySlippery. In Fig. 3 it is possible to see the membership functions of the road slippery level criteria.

The following is the membership function equation shown in Fig. 3:

$$y_{RatherSlippery}=\begin{cases}\dfrac{x-0}{0.2-0}, \text{if } 0\le x\le 0.2,\\ \dfrac{0.4-x}{0.4-0.2}, \text{if } 0.2<x\le 0.4,\\ 0, \text{if } x>0.4,\end{cases} \qquad (5)$$

$$y_{Slippery}=\begin{cases}\dfrac{x-0}{0.5-0}, \text{if } 0.3\le x\le 0.5,\\ \dfrac{0.7-x}{0.7-0.5}, \text{if } 0.5<x\le 0.7,\\ 0, \text{if } x\langle 0.3 \text{ or } x\rangle 0.7,\end{cases} \qquad (6)$$



Rule #1
IF (Slippery is RatherSlippery) and (WaterLevel is VeryLow) and Distance is Near) and Drainage is Slow) and (Vulnerability is Low) THEN (Output is 0.6)

Rule #405
IF (Slippery is VerySlippery) and (WaterLevel is High) and Distance is Far) and Drainage is Fast) and (Vulnerability is High) THEN (Output is 1)

Fig. 2. Fuzzy logic structure

$$y_{VerySlippery} = \begin{cases} \dfrac{x-0.6}{0.8-0.6}, \text{ if } 0.6 \le x \le 0.8, \\ \dfrac{1-x}{1-0.8}, \text{ if } 0.8 < x \le 1, \\ 0, \text{ if } x < 0.6; \end{cases} \tag{7}$$

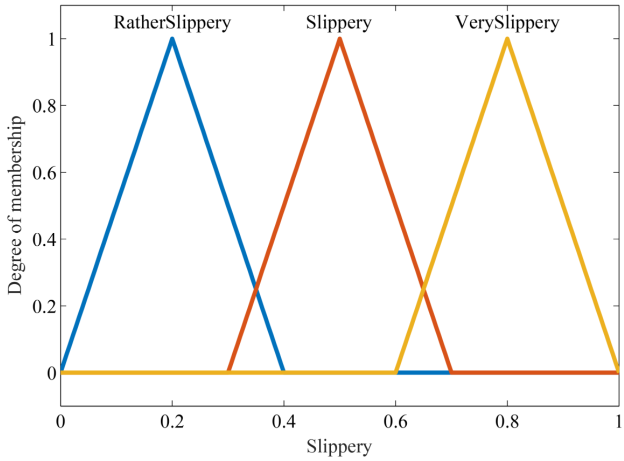


Fig. 3. Membership function of the road slippery level criteria

b) the criterion of flooded water level. The value of the input linguistic variable of the water level inundating the road is divided into 5 (five) membership functions, namely VeryLow, Low, Average, RatherHigh and High. Fig. 4 shows the membership function of the criteria for water level inundating the road. The membership function equation in Fig. 4 is as follows:

$$y_{VeryLow} = \begin{cases} \dfrac{x-(-5)}{10-(-5)}, \text{ if } -5 \le x \le 25, \\ \dfrac{25-x}{25-10}, \text{ if } 10 < x \le 25, \\ 0, \text{ if } x > 25. \end{cases} \tag{8}$$

$$y_{Low} = \begin{cases} \dfrac{x-15}{30-15}, \text{ if } 15 \le x \le 30, \\ \dfrac{45-x}{45-30}, \text{ if } 30 < x \le 45, \\ 0, \text{ if } x\langle 15 \text{ or } x\rangle 45. \end{cases} \tag{9}$$

$$y_{Average} = \begin{cases} \dfrac{x-35}{50-35}, \text{ if } 35 \le x \le 50, \\ \dfrac{65-x}{65-50}, \text{ if } 50 < x \le 65\ , \\ 0, \text{ if } x\langle 35 \text{ or } x\rangle 65. \end{cases} \tag{10}$$

$$y_{RatherHigh} = \begin{cases} \dfrac{x-55}{70-55}, \text{ if } 55 \le x \le 70, \\ \dfrac{85-x}{85-70}, \text{ if } 70 < x \le 85, \\ 0, \text{ if } x\langle 55 \text{ or } x\rangle 85. \end{cases} \tag{11}$$

$$y_{High} = \begin{cases} \dfrac{x-75}{90-75}, \text{ if } 75 \le x \le 90, \\ \dfrac{105-x}{105-90}, \text{ if } 90 < x \le 105, \\ 0, \text{ if } x < 75; \end{cases} \tag{12}$$

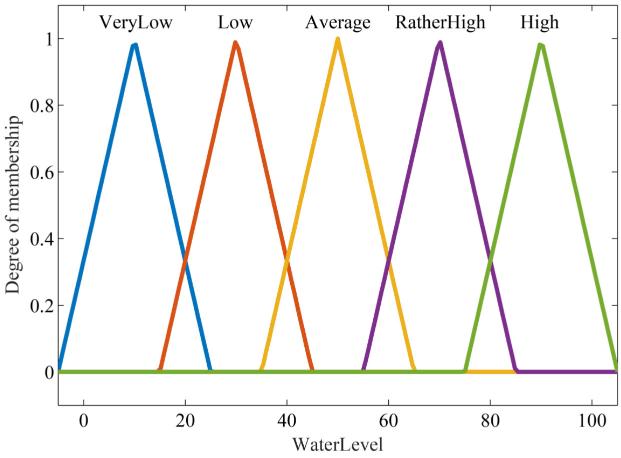


Fig. 4. Membership function of inundated water level criterion

c) the criterion of road's distance from the river. The value of the input linguistic variable on the criterion for the distance of the river to the road is divided into 3 (three) membership functions, namely Near, Average and Far, as shown in Fig. 5.

Membership function equation of the criterion of road's distance from the river in Fig. 5 are:

$$y_{Near} = \begin{cases} \dfrac{x-0}{40-0}, \text{ if } 0 \le x \le 40, \\ \dfrac{80-x}{80-40}, \text{ if } 40 < x \le 80, \\ 0, \text{ if } x > 80. \end{cases} \tag{13}$$

$$y_{Average} = \begin{cases} \dfrac{x-60}{100-60}, \text{ if } 60 \le x \le 100, \\ \dfrac{140-x}{140-100}, \text{ if } 100 < x \le 140, \\ 0, \text{if } x\langle 60 \text{ or } x\rangle 140. \end{cases} \tag{14}$$

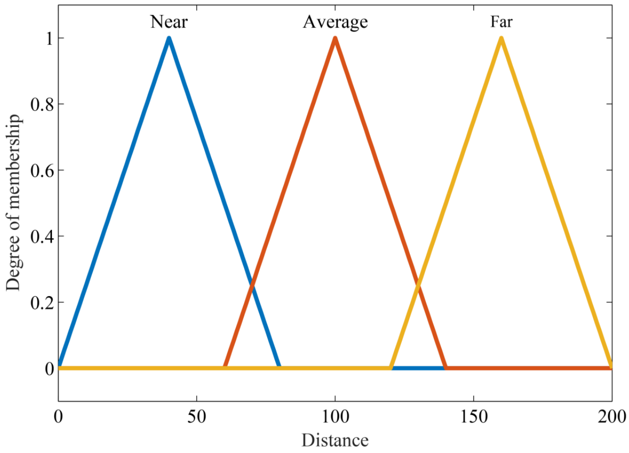


Fig. 5. Membership function of the criterion of road's distance from the river

$$y_{Far}=\begin{cases}\dfrac{x-120}{160-120}, \text{ if } 120\le x\le 160,\\ \dfrac{200-x}{200-160}, \text{ if } 160<x\le 200,\\ 0, \text{ if } x<120;\end{cases} \quad (15)$$

d) the drainage system criterion. The value of the input linguistic variable of the road drainage system criteria is divided into 3 (three) membership functions, namely Slow, Average and Fast. Fig. 6 shows the distribution of the linguistic variables of the drainage system.

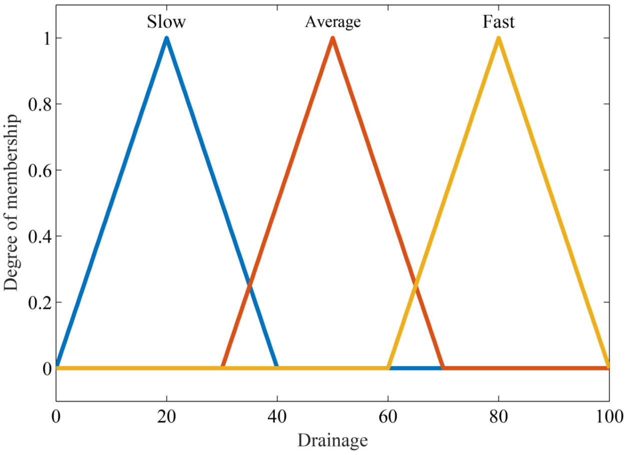


Fig. 6. Membership function of drainage system

The membership function of the drainage system in Fig. 6 has the equation:

$$y_{Slow}=\begin{cases}\dfrac{x-0}{20-0}, \text{ if } 0\le x\le 20,\\ \dfrac{40-x}{40-20}, \text{ if } 20<x\le 40,\\ 0, \text{ if } x>40.\end{cases} \quad (16)$$

$$y_{Average}=\begin{cases}\dfrac{x-30}{50-30}, \text{ if } 30\le x\le 50,\\ \dfrac{70-x}{70-50}, \text{ if } 50<x\le 70\ ,\\ 0, \text{ if } x\langle 30 \text{ or } x\rangle 70.\end{cases} \quad (17)$$

$$y_{Fast}=\begin{cases}\dfrac{x-60}{80-60}, \text{ if } 60\le x\le 80,\\ \dfrac{100-x}{100-80}, \text{ if } 80<x\le 100,\\ 0, \text{ if } x<100;\end{cases} \quad (18)$$

e) the victim vulnerability level criterion. The value of the input linguistic variable of the criterion for the victim's vulnerability level is divided into 3 (three) membership functions, namely Low, Average and High. The distribution of linguistic variables for flood victims is shown in Fig. 7.

The membership function equations in Fig. 7 are shown in (19)–(21):

$$y_{Slow}=\begin{cases}\dfrac{x-0}{4-0}, \text{ if } 0\le x\le 4,\\ \dfrac{8-x}{8-4}, \text{ if } 4<x\le 8,\\ 0, \text{ if } x>80.\end{cases} \quad (19)$$

$$y_{Average}=\begin{cases}\dfrac{x-6}{10-6}, \text{ if } 6\le x\le 10,\\ \dfrac{14-x}{14-10}, \text{ if } 10<x\le 14,\\ 0, \text{ if } x\langle 6 \text{ or } x\rangle 14.\end{cases} \quad (20)$$

$$y_{Fast}=\begin{cases}\dfrac{x-12}{16-12}, \text{ if } 12\le x\le 16,\\ \dfrac{20-x}{20-16}, \text{ if } 16<x\le 20,\\ 0, \text{ if } x<12.\end{cases} \quad (21)$$

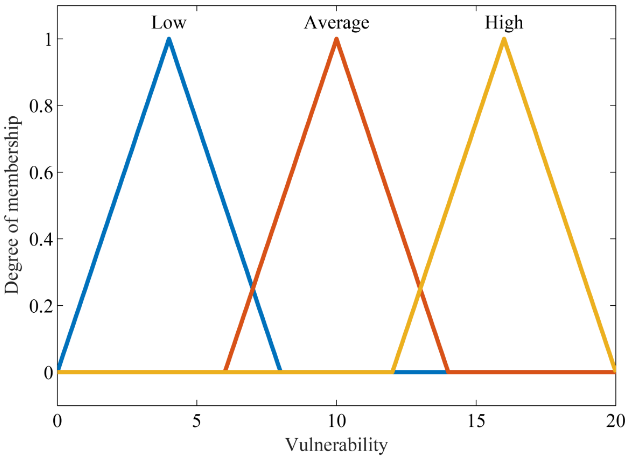


Fig. 7. Membership function of the victim vulnerability criterion

2. Determining the composition of the rules on the inference engine. After the input crips value is obtained through fuzzification, based on several rules, this inference engine performs reasoning on the input crips. The rules on fuzzy logic in the inference engine describe the expected conditions and the desired results. In this study, 405 fuzzy rules are used to obtain the obstacle weight output on each path. The rules on fuzzy logic in this study are given in (22):

$$\begin{aligned}&\text{IF } C1=x_1 \text{ AND } C2=x_2 \text{ AND } C3=x_3\\ &\text{AND } C4=x_4 \text{ AND } C5=x_5 \text{ THEN Weight}=z,\end{aligned} \quad (22)$$

where x_1, x_2, x_3, x_4, x_5 and z are membership functions of the corresponding input and output.

3. Defuzzification. After the inference engine output is aggregated to get a single crisp value, the Mamdani defuzzification model uses the Center of area or centroid method to get the output crisp. The centre of area or centroid method is a method for calculating the centre of mass of a closed curve using the formula:

$$y^*=\frac{\int y\mu_R(y)\mathrm{d}y}{\mu_R(y)\mathrm{d}y}. \quad (23)$$

The output of defuzzification in this research is a number that shows the obstacle weight of each path in the environment used for simulation. The output membership function of this research can be seen in Fig. 8.

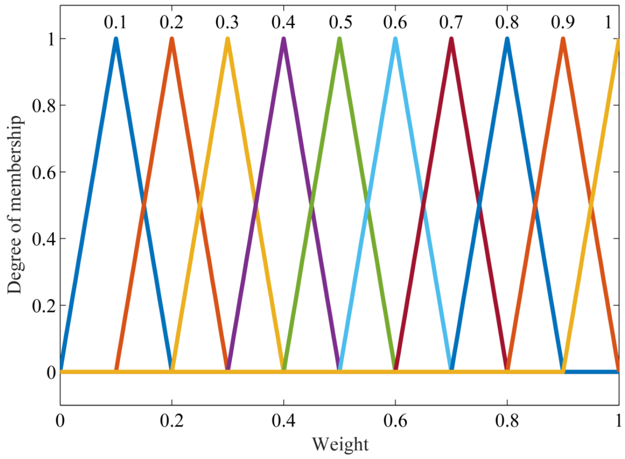


Fig. 8. The output membership function of the weight of each path

The membership function equation shown in Fig. 8 can be seen in (24)–(33):

$$y_{VVSmall}=\begin{cases}\dfrac{x-0}{0.1-0}, \text{if } 0\le x\le 0.1,\\ \dfrac{0.2-x}{0.2-0.1}, \text{if } 0.1<x\le 0.2,\\ 0, \text{if } x>0.2.\end{cases} \quad (24)$$

$$y_{VSmall}=\begin{cases}\dfrac{x-0.1}{0.2-0.1}, \text{if } 0.1\le x\le 0.2,\\ \dfrac{0.3-x}{0.3-0.2}, \text{if } 0.2<x\le 0.3\ ,\\ 0, \text{if } x\langle 0.1 \text{ or } x\rangle 0.3.\end{cases} \quad (25)$$

$$y_{Small}=\begin{cases}\dfrac{x-0.2}{0.3-0.2}, \text{if } 0.2\le x\le 0.3,\\ \dfrac{0.4-x}{0.4-0.3}, \text{if } 0.3<x\le 0.4,\\ 0, \text{if } x\langle 0.2 \text{ or } x\rangle 0.4.\end{cases} \quad (26)$$

$$y_{RatherSmall}=\begin{cases}\dfrac{x-0.3}{0.4-0.3}, \text{if } 0.3\le x\le 0.4,\\ \dfrac{0.5-x}{0.5-0.4}, \text{if } 0.4<x\le 0.5,\\ 0, \text{if } x\langle 0.3 \text{ or } x\rangle 0.5.\end{cases} \quad (27)$$

$$y_{Medium}=\begin{cases}\dfrac{x-0.4}{0{,}5-0{,}4}, \text{if } 0.4\le x\le 0.5,\\ \dfrac{0.6-x}{0.6-0.5}, \text{if } 0.5<x\le 0{,}6,\\ 0, \text{if } x\langle 0.4 \text{ or } x\rangle 0.6.\end{cases} \quad (28)$$

$$y_{RatherLarge}=\begin{cases}\dfrac{x-0.5}{0.6-0.5}, \text{if } 0.5\le x\le 0.6,\\ \dfrac{0.7-x}{0.7-0.6}, \text{if } 0.6<x\le 0.7,\\ 0, \text{if } x\langle 0.5 \text{ or } x\rangle 0.7.\end{cases} \quad (29)$$

$$y_{Large}=\begin{cases}\dfrac{x-0.6}{0.7-0.6}, \text{if } 0.6\le x\le 0.7,\\ \dfrac{0.8-x}{0.8-0.7}, \text{if } 0.7<x\le 0.8,\\ 0, \text{if } x\langle 0.6 \text{ or } x\rangle 0.8.\end{cases} \quad (30)$$

$$y_{VLarge}=\begin{cases}\dfrac{x-0.7}{0.8-0.7}, \text{if } 0.7\le x\le 0.8,\\ \dfrac{0.9-x}{0.9-0.8}, \text{if } 0.8<x\le 0.9,\\ 0, \text{if } x\langle 0.7 \text{ or } x\rangle 0.9.\end{cases} \quad (31)$$

$$y_{VVLarge}=\begin{cases}\dfrac{x-0.8}{0.9-0.8}, \text{if } 0.8\le x\le 0.9,\\ \dfrac{1-x}{1-0.9}, \text{if } 0.9<x\le 1,\\ 0, \text{if } x\langle 0.8 \text{ or } x\rangle 1.\end{cases} \quad (32)$$

$$\mathrm{y}_{VVVLarge}=\begin{cases}0, \text{if } x\le 0.9,\\ \dfrac{x-0.9}{1-0.9}, \text{if } 0.9<x\le 1,\\ 1, \text{if } x>1.\end{cases} \quad (33)$$

The weight of the obstacle on each path will be used to determine the weight of the obstacle on each route in Fig. 1. The obstacle weight of each route is calculated using the formula:

$$W_m=\sum_{n=1}^{k} wp_n, \quad (34)$$

where $n=1, 2,..., k$, k – the number of paths on path, m – the number of route, $m=1, 2,..., 24$.

In this route priority search research, several data sets have been used that show the value of each obstacle. This data set will be processed using fuzzy logic to get the obstacle weights for each path from environmental model. Fuzzy logic is generally used to solve problems that contain uncertainty, imprecision, noise, and so on. Given the type of obstacle, fuzzy logic is most appropriate in this case.

Table 2 shows the results of calculating the weight of each path, calculated by fuzzy logic.

The author has carried out four experiments with different obstacle values on each path. The results of the weighting of each path are used as a base for the simulation of route priority search by the firefly algorithm.

Table 2

Data of obstacle for each path

No	Name of Path	Output Weight			
		1	2	3	4
1	*S-a*	0.50000	0.55000	0.55000	0.99991
2	*S-c*	0.84422	0.30005	0.30005	0.50005
3	*S-e*	0.15008	0.25006	0.30005	0.45002
4	*a-b*	0.50006	0.30010	0.40003	0.10013
5	*a-c*	0.69993	0.40001	0.40003	0.10013
6	*b-c*	0.50005	0.35024	0.45012	0.15048
7	*b-d*	0.50005	0.99987	0.89993	0.59995
8	*b-D*	0.42619	0.79971	0.89960	0.30029
9	*c-d*	0.50005	0.10013	0.20007	0.40007
10	*d-D*	0.40007	0.30031	0.40018	0.69982
11	*d-e*	0.79990	0.10009	0.20007	0.35004
12	*d-g*	0.89993	0.30010	0.40003	0.50005
13	*e-f*	0.72422	0.57555	0.59999	0.87531
14	*f-g*	0.79995	0.51501	0.55000	0.33921
15	*g-D*	0.97526	0.84994	0.89993	0.30005

5. 4. Development of Improved Firefly Algorithm for route priority search

The firefly algorithm is used for the case of simulating a safe route priority search. The firefly algorithm is adopted in finding priority routes available for victims to get to a predetermined evacuation centre. In Algorithm 1, the firefly algorithm begins to search for the lowest obstacle weight from the route by placing the position of the firefly population at the coordinates of the predetermined starting point and at the coordinates of each end of the determined path. From the environment designed for simulation, there are 24 alternative routes available, as shown in Fig. 1. Each route has obstacle weight path. The size of the obstacle weight value determines the level of path difficulty to be passed. The greater the obstacles weight of the path, the more dangerous the path to pass. The obstacle weight of each route is the sum of the obstacle weights from several paths on the route, as shown in equation (34).

```
Algorithm 1: Fuzzy Firefly Algorithm (FuFA)
Determine the starting point, destination point and path
Obstacle weight mapping on each path
Obstacle weight mapping on each route
Determine the light absorption coefficient $\gamma$
Determine the initial attraction $\beta_0$
Determine the initial firefly population $x_i$
Determine the maximum iteration
while ($t$<MaxIteration)
  for $i$=1:$n$ (all n fireflies)
    for $j$=1:$i$ (all n fireflies)
      if ($I_j$>$I_i$), calculate route weight between two solutions
      Calculating the attraction between two solutions
      Move fireflies to brighter fireflies
      New $W$ evaluation and updated light intensity
    end for j
  end for i
  Rate the fireflies and find the smallest obstacle weight
end while
```

In the case of finding evacuation routes for flood victims, the firefly algorithm is used to find routes that have a relatively high level of danger that flood victims can face. The choice of this route is not related to the distance that must be travelled. The distance of a road taken is quite close, but the road is dangerous if it has to be passed by flood victims, then the road will not be chosen by this firefly algorithm. The firefly algorithm will select a route with all the safe path for flood victims.

Before starting the simulation, the firefly algorithm parameters need to be set, as shown in Table 3.

Table 3

Firefly Algorithm Parameters

Parameters	Value
Maximum iteration	50
Number of fireflies in the population	9
β_0	1
γ	1

The γ value represents the level of light absorption around the firefly. Firefly will move towards another firefly that has brighter light. Firefly that has a brighter light means the firefly has a relatively small path weight.

Fig. 9–12 shows the simulation results of the FuFA in searching the safest route priority with obstacle data from Table 2.

The simulation results show that the algorithm can find the route that has the lowest obstacle weight:

1. Simulation using the first weight data set.

Fig. 9 shows that the obstacle weight of the path from point 1 to point 2 is 0.5000, from point 1 to point 4 is 0.84422, and from point 1 to point 6 is 0.15008 respectively.

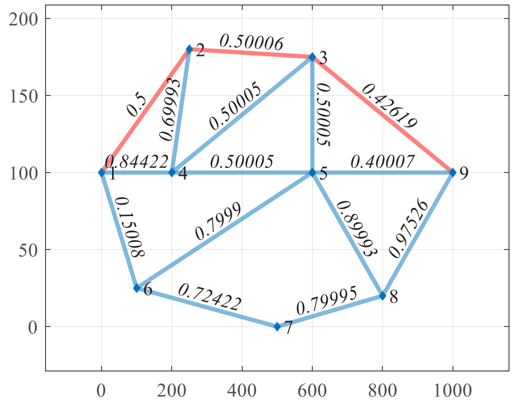


Fig. 9. The results of the route priority search simulation

This algorithm does not choose the path from point 1 to point 6, which has the lowest obstacle weight among the three. This is because the path after path from point 1 to point 6 has a reasonably large obstacle weight of 0.72422 so the path is dangerous if passed. The obstacle weight is high and the path from point 1 to point 4 likewise. At point 2, there are two possible path options, namely path from point 2 to point 3 and path from point 2 to point 4. By comparing the values between the two paths, path from point 2 to point 3 is chosen; which has the lower obstacle weights. Likewise, when in point 3, two possible paths are the path from point 3 to point 9 and the path from point 3 to point 5. This algorithm will choose the path with the lowest obstacle weight value, namely path from point 3 to point 9 with an obstacle weight value of 0.42619, and this path is also the last path to arrive at the destination point, namely, point 9. By using the formula (34) then route 1–2–3–9 has an obstacle weight of 1.42625. This is the lowest value among the obstacle weights owned by other routes. In this simulation, the choice of route priority by the firefly algorithm is shown by a red line in Fig. 9.

2. Simulation using the second weight data set.

Based on the second output weight in Table 2, a route priority search simulation is performed by the firefly algorithm, a route priority search simulation is performed by the firefly algorithm. Analogous to the route priority search that has been done previously, from Fig. 10, at point 1, the path with the lowest obstacle weight is chosen. The path from point 1 to 6 has the lowest weight among other paths choices, so the firefly algorithm will choose this path as the initial path to be passed.

At point 6, there are two branches which are the path choices to be passed. The firefly algorithm will choose between these two paths with the slightest obstacle weight, namely path from point 6 to point 5. Point 5 has three possible path options with obstacle weights of 0.30010, 0.30031 and 0.99987, respectively. The firefly algorithm chooses a path from point 5 to point 9, which has a weight of 0.30031 to pass.

The firefly algorithm chooses a path that does not have the lowest obstacle. This choice was taken because the path from point 5 to point 9 is the last path to get to the endpoint, while the other two paths, the path from point 5 to point 8 and the path from point 5 to point 3 are not the last path before arriving at the destination point. The total obstacle weight on each route 1–6–5–9 is 0.65046.

Route 1–6–5–3–9 has an obstacle weight of 3.14973 and has a dangerous path to pass because it has a large obstacle weight value. Route 1–6–5–8–9 has an obstacle weight of 1.50019 and has a path with a large obstacle weight. This firefly algorithm chooses route 1–6–5–9 to be the priority route because it has the lowest obstacle weight and does not have a path with dangerous obstacles, as shown in Fig. 10 in red.

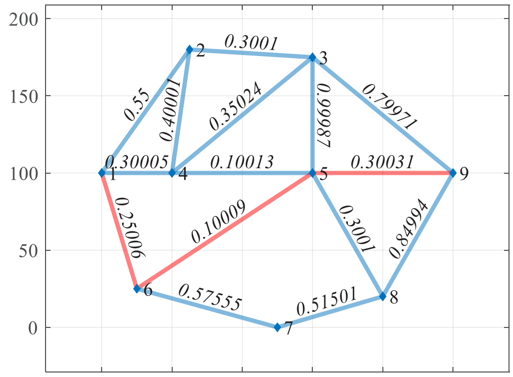


Fig. 10. The results of the route priority search simulation

3. Simulation using the third weight data set.

For the next simulation, the third output weight in Table 2 is used to obtain the obstacle weights for each path from each model route.

Analogous to the work steps as before, the firefly algorithm can find route priorities with the smallest obstacle weights. At point 1, there are three choices of paths that can be passed. The path from point 1 to point 2 has an obstacle weight of 0.55000. The path from point 1 to point 4 has an obstacle weight of 0.30005, and the path from point 1to point 6 has an obstacle weight of 0.30005. In this simulation, the algorithm chooses path point 1 toward point 4. At point 4, there are two branches, namely path from point 4 to point 3 and path from point 4 to point 5. The weight of path from point 4 to point 3 is 0.45012. Path from point 4 to point 5 weighs 0.20007. There are two branches, the path from point 4 to point 3 and the path from point 4 to point 5. The firefly algorithm will choose path from point 4 to point 5 because it has a lower obstacle weight than path from point 4 to point 3. At point 5, there are three path choices. Of the three available path options, only two paths can be selected. Path from point 5 to point 3 has a relatively large weight than the other two paths. The firefly algorithm chooses the path from point 5 to point 9 because it has a relatively small path weight and it is the last path to arrive at the destination point. The firefly algorithm selects route 1–4–5–9 with an obstacle weight of 0.90030. Fig. 11 shows the route priority search simulation result using the firefly algorithm.

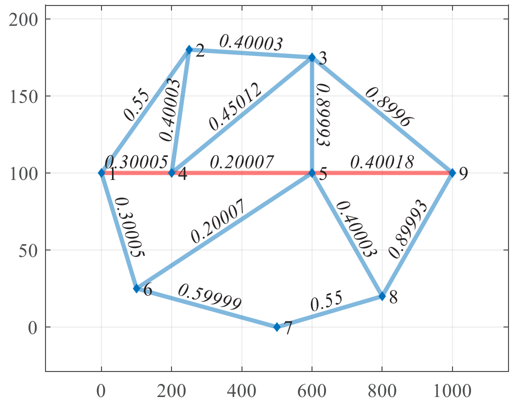


Fig. 11. The results of the route priority search simulation

4. Simulation using the fourth weight data set.

The following simulation of route priority search by the firefly algorithm is a simulation that uses the fourth weight data in Table 2.

Analogous to the previous steps of the firefly algorithm, the firefly algorithm can find routes that have relatively small weights. Fig. 12 shows the simulation results of finding a safe route priority, indicated by the slightest obstacle weight on the related route.

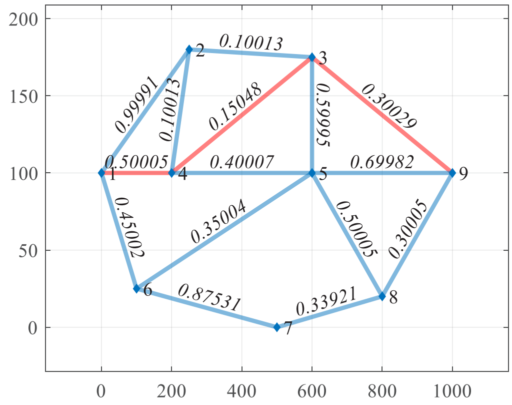


Fig. 12. The results of the route priority search simulation

At point 1, the firefly algorithm will choose path from point 1 to point 4 which weighs 0.50005. Furthermore, the firefly algorithm will choose path from point 4 to point 3 which has an obstacle weight of 0.15048 and path from point 3 to point 9 which has an obstacle weight of 0.30029. The total weight of route 1–4–3–9 is 0.95082. The algorithm does not choose path from point 1 to point 2 because these paths have a large weight. Path from point 1 to point 2 is dangerous to pass. Path from point 1 to point 6 is also not the choice of the firefly algorithm because after path from 1 to point 6 there is the path from point 6 to point 5 which has a weight of 0.35004 and the path from point 5 to point 9 which weighs 0.69962, meaning that route 1–6–5–9 with an obstacle weight of 1.49968 has a large obstacle weight when compared to 0.95082. The priority route chosen by this algorithm is route 1–4–3–9 which has an obstacle weight of 0.95082.

Table 4

The optimal and the highest obstacle weight of route priority

Simulation	Optimal Obstacle Weight	Highest Obstacle Weight
1	1.42625	5.54936
2	0.65046	4.34071
3	0.90030	4.55007
4	0.95082	3.71508

Table 4 shows the optimal and the highest obstacle weight value for the route priority from the simulation results. The simulations carried out the proposed method that provides an adequate solution to search for safe route for flood victims. The effectiveness of the proposed method in finding a safe route for flood victims by comparing the lowest value of the selected route weight with the most significant route weight, and the average result is from 14 % to 26 %.

6. Discussion of result development of firefly algorithm with fuzzy logic integration for priority search of flood evacuation routes

An improved firefly algorithm with fuzzy logic integration has been developed. The aim of this algorithm is to find a safe priority route for flood victims to get to the evacuation location. Each route in the model that has been built consists of several paths, as shown in Fig. 1. Each path has obstacles that hinder the movement of flood victims to the evacuation location. The obstacles designed for each path are the level of road slippage, the water level on the road, the road's distance to the river, the drainage system, and the vulnerability of flood victims. A weighted value expresses the obstacle on each path. The greater the value of the obstacle weight on a path, the more dangerous the path will be for flood victims to cross. The value of the obstacle weight for each path will be calculated using fuzzy logic.

Each route has several paths. The obstacle weight value for each route is the sum of the obstacle weight values for all paths on that route. The sum of the obstacle weight values for a route uses equation (34).

An improved firefly algorithm is used to find safe priority routes for flood victims to pass. The route priority selection process by the improved firefly algorithm considers the value of the route obstacle weight and the weight value of each path. If there is a path with a significant obstacle weight value in a route, the improved firefly algorithm will not choose that route. This condition is evidenced by a simulation using the first data set. From Fig. 9, it can be seen that the route that passes through point 1, point 6, point 5, point 9 has a total obstacle weight value of 1.35005. The improved firefly algorithm did not choose this route even though this route has the smallest obstacle weight value. Improved firefly algorithm selects a route that passes through point 1, point 2, point 3, point 9. The route has an obstacle weight value of 1, 42625. In the 2nd, 3rd and 4th simulations, the improved firefly algorithm chooses a route with a weight value of the lowest obstacle. In addition to having a low total obstacle weight value, each path on the route has a common obstacle weight value. The choice of route priority by the improved firefly algorithm shows good performance. This algorithm does not consider the shortest distance traversed by flood victims but instead considers the risk of danger faced by people who will pass through it.

It should be noted that the improved firefly algorithm with fuzzy logic integration for priority route search proposed in this study could be improved further, namely:

– adding several other obstacles to increase the accuracy of the route obstacle weights, such as the level of rainfall, the duration of the rain and other hazards;

– inclusion of the distance and travel time to the evacuation location;

– development of GIS-based routing model.

7. Conclusions

1. In this research, the design of the environmental model used for the simulation is essential. An environmental model is needed to describe the interaction between the route and the obstacle that represents the level of difficulty that must be faced when crossing the route. The environmental model in this study has a total of 24 routes.4.

2. There are five obstacles developed in this study. This obstacle is sufficient to describe the level of difficulty for flood victims to cross a route to get to the evacuation location.

3. The fuzzy logic developed in this study has 405 rules designed for FIS. With these 405 rules, the developed fuzzy logic can provide information on obstacle weights from each path.

4. The development of the firefly algorithm to find route priorities for flood victims shows good performance. As evidenced by the given obstacle weights for different routes, the firefly algorithm can show routes with low obstacle weights and routes with a small number of paths. The selection of the route with the lowest weight is essential so that the flood victims can safely arrive at the evacuation location.

Acknowledgement

The author would like to thank the Ministry of Education, Culture, Research and Technology of the Republic of Indonesia for financial support for this research under the PDD Grant number NKB-22/UN2.RS/HKP.05.00/2020.

References

1. Gomes, R., Straub, J. (2017). Genetic algorithm for flood detection and evacuation route planning.Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XXIII.doi: https://doi.org/10.1117/12.2266474
2. Bernardini, G., Santarelli, S., Quagliarini, E., D'Orazio, M. (2017). Dynamic guidance tool for a safer earthquake pedestrian evacuation in urban systems. Computers, Environment and Urban Systems, 65, 150–161. doi: https://doi.org/10.1016/j.compenvurbsys.2017.07.001
3. Forcael, E., González, V., Orozco, F., Vargas, S., Pantoja, A., Moscoso, P. (2014). Ant Colony Optimization Model for Tsunamis Evacuation Routes. Computer-Aided Civil and Infrastructure Engineering, 29 (10), 723–737. doi: https://doi.org/10.1111/mice.12113
4. Khalilpourazari, S., Pasandideh, S. H. R. (2021). Designing emergency flood evacuation plans using robust optimization and artificial intelligence. Journal of Combinatorial Optimization, 41 (3), 640–677. doi: https://doi.org/10.1007/s10878-021-00699-0
5. Tamakloe, R., Hong, J., Tak, J., Park, D. (2021). Finding evacuation routes using traffic and network structure information. Transportation Research Part D: Transport and Environment, 95, 102853. doi: https://doi.org/10.1016/j.trd.2021.102853
6. Hidalgo-Paniagua, A., Vega-Rodríguez, M. A., Ferruz, J., Pavón, N. (2015). Solving the multi-objective path planning problem in mobile robotics with a firefly-based approach. Soft Computing, 21 (4), 949–964. doi: https://doi.org/10.1007/s00500-015-1825-z
7. Brand, M., Yu, X.-H. (2013). Autonomous robot path optimization using firefly algorithm.2013 International Conference on Machine Learning and Cybernetics.doi: https://doi.org/10.1109/icmlc.2013.6890747
8. Sutantyo, D., Levi, P. (2015). Decentralized underwater multi-robot communication using bio-inspired approaches. Artificial Life and Robotics, 20 (2), 152–158. doi: https://doi.org/10.1007/s10015-015-0201-5
9. Wang, G. et. al. (2012).A modified firefly algorithm for UCAV path planning. International Journal of Hybrid Information Technology, 5 (3), 123–144. Available at: http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=CAD0E8B60657571D44BEBA8E78A4299F?doi=10.1.1.643.3618&rep=rep1&type=pdf
10. Patle, B. K., Parhi, D. R., Jagadeesh, A., Kashyap, S. K. (2017). On firefly algorithm: optimization and application in mobile robot navigation. World Journal of Engineering, 14 (1), 65–76. doi: https://doi.org/10.1108/wje-11-2016-0133
11. Tighzert, L., Fonlupt, C., Mendil, B. (2018). A set of new compact firefly algorithms. Swarm and Evolutionary Computation, 40, 92–115. doi: https://doi.org/10.1016/j.swevo.2017.12.006
12. Patle, B. K., Pandey, A., Jagadeesh, A., Parhi, D. R. (2018). Path planning in uncertain environment by using firefly algorithm. Defence Technology, 14 (6), 691–701. doi: https://doi.org/10.1016/j.dt.2018.06.004
13. Trachanatzi, D., Rigakis, M., Marinaki, M., Marinakis, Y. (2020). A firefly algorithm for the environmental prize-collecting vehicle routing problem.Swarm and Evolutionary Computation, 57, 100712.doi: https://doi.org/10.1016/j.swevo.2020.100712
14. Feng, K., Wang, X., Han, J., Liu, S. (2019). Local Path Planning Method of Unmanned Ship Based on Improved Firefly Algorithm. International Journal of Engineering and Advanced Research Technology (IJEART), 5 (11). Available at: https://www.ijeart.com/download_data/IJEART0511002.pdf
15. Yang, X.-S. (2010). Nature-inspired metaheuristic algorithms.Luniver Press.
16. Fister, I., Fister, I., Yang, X.-S., Brest, J. (2013). A comprehensive review of firefly algorithms. Swarm and Evolutionary Computation, 13, 34–46. doi: https://doi.org/10.1016/j.swevo.2013.06.001
17. Yang, X.-S. (2009). Firefly Algorithms for Multimodal Optimization. Lecture Notes in Computer Science, 169–178. doi: https://doi.org/10.1007/978-3-642-04944-6_14
18. Chandrawati, T. B., Ratna, A. A. P., Sari, R. F. (2020). Path Selection using Fuzzy Weight Aggregated Sum Product Assessment. International Journal of Computers Communications & Control, 15 (5).doi: https://doi.org/10.15837/ijccc.2020.5.3978