symmetry

MDPI

Article

A Fuzzy Gain-Based Dynamic Ant Colony Optimization for Path Planning in Dynamic Environments

Viswanathan Sangeetha [1], Raghunathan Krishankumar [1], Kattur Soundarapandian Ravichandran [1], Fausto Cavallaro [2,*], Samarjit Kar [3], Dragan Pamucar [4] and Abbas Mardani [5]

[1] School of Computing, SASTRA Deemed University, Thanjavur 613401, India; sangeetha@sastra.ac.in (V.S.); krishankumar@sastra.ac.in (R.K.); raviks@it.sastra.edu (K.S.R.)
[2] Department of Economics, University of Molise, Via De Sanctis, 86100 Campobasso, Italy
[3] Department of Mathematics, National Institute of Technology, Durgapur 713209, India; samarjit.kar@maths.nitdgp.ac.in
[4] Department of Logistics, Military Academy, University of Defence in Belgrade, 11000 Belgarde, Serbia; dragan.pamucar@va.mod.gov.rs or dpamucar@gmail.com
[5] Department of Marketing, Muma College of Business, University of South Florida, Tampa, FL 33813, USA; mabbas3@liveutm.onmicrosoft.com
* Correspondence: cavallaro@unimol.it

check for updates

Citation: Sangeetha, V.; Krishankumar, R.; Ravichandran, K.S.; Cavallaro, F.; Kar, S.; Pamucar, D.; Mardani, A. A Fuzzy Gain-Based Dynamic Ant Colony Optimization for Path Planning in Dynamic Environments. *Symmetry* **2021**, *13*, 280. https://doi.org/10.3390/sym13020280

Academic Editor: Theodore E. Simos
Received: 13 January 2021
Accepted: 3 February 2021
Published: 6 February 2021

Abstract: Path planning can be perceived as a combination of searching and executing the optimal path between the start and destination locations. Deliberative planning capabilities are essential for the motion of autonomous unmanned vehicles in real-world scenarios. There is a challenge in handling the uncertainty concerning the obstacles in a dynamic scenario, thus requiring an intelligent, robust algorithm, with the minimum computational overhead. In this work, a fuzzy gain-based dynamic ant colony optimization (FGDACO) for dynamic path planning is proposed to effectively plan collision-free and smooth paths, with feasible path length and the minimum time. The ant colony system's pheromone update mechanism was enhanced with a sigmoid gain function for effective exploitation during path planning. Collision avoidance was achieved through the proposed fuzzy logic control. The results were validated using occupancy grids of variable size, and the results were compared against existing methods concerning performance metrics, namely, time and length. The consistency of the algorithm was also analyzed, and the results were statistically verified.

Keywords: ant colony optimization; collision avoidance; dynamic environment; sigmoidal function; triangular membership function

1. Introduction

Modern technical and scientific advancements have propelled the proliferation of autonomous vehicle systems. Our everyday life has become almost inseparable from the need for autonomous vehicle systems. Such autonomous systems require four capabilities, namely: (1) world perception, (2) cognitive learning, (3) map building, and pathfinding (4) navigation. Pathfinding is a vital component for the robust working of autonomous vehicle systems. On the other hand, handling the complexity and uncertainty of real-time scenarios is a significant challenge to be addressed by the algorithms. Path planning can be conventionally classified, based on the environment's nature, as static or dynamic path planning. Static path planning scenarios are those where the obstacles are static and not moving. In the case of dynamic scenarios, there are moving obstacles at different speeds within a scenario. There will be no prior knowledge of the environment and position of obstacles. Real-time scenarios are dynamic. The class of algorithms for solving real-time scenarios are broadly classified as exact and approximate approaches. Exact algorithms may fail to produce good quality solutions because of the uncertainty in the scenario. Over the years, there has been tremendous research on the use of approximate algorithms for path planning in dynamic and real-time scenarios [1]. Metaheuristics belong to the approximate approaches, and are widely used in various optimization problems because of their ability to produce good quality quasi-optimal solutions [2]. A proper tradeoff between their exploration and exploitation would result in good quality solutions after a thorough iterative search of the solution space. There are many words in the existing literature on dynamic path planning. Some of the prominent investigations from the literature are given in Table 1. A new mutation operator was proposed for dynamic path planning by Adem Tuncer and Mehmet Yildirim [3] to enable faster convergence. They validated their approach by applying it to simulated scenarios and comparing it with the literature.

Table 1. Investigations from the existing literature.

Ref #	Method	Dataset	Optimality Achieved
[4]	Rule-based sequence planning algorithm with fuzzy optimization	Simulated scenarios	Flexibility and adaptability
[5]	Genetic algorithm and adaptive fuzzy logic control	RobuTER robot	Smooth collision-free path
[6]	Fuzzy wind-driven optimization	Real-time navigation using Khepera III mobile robot	Collision free path
[7]	Dynamic path planner	Simulated single and multi-lane roads	Static and dynamic safety, comfortability, appropriate acceleration and speed for the vehicle
[8]	Personalized path planner with fuzzy c-means clustering	Simulated grids, road simulation model in Changsha	Improved personalization of existing path planning
[9]	Improved rapidly exploring random tree (RRT) algorithm	Simulated and real-time implementation using MATLAB and robot operating system(ROS)	Correctness, effectiveness, and practicability
[10]	Fuzzy logic	Real-time navigation using mobile robot on long u shape, large concave, cluttered, maze-like dynamic environments	Minimum risk and global convergence
[11]	Hybrid heuristic optimization algorithm (Beetle antennae search)	Virtual map and the real map	Accelerated convergence speed
[12]	Dynamic Fuzzy logic based path planning	Wireless sensor networks in MATLAB	Localization ratio and localization accuracy
[13]	Dynamic fuzzy-logic-ant colony system	Regions of London, United Kingdom	Efficient route selection
[14]	Ant colony and fuzzy logic	Simulated maps in MATLAB	Shortest path in minimum time
[15]	Fuzzy logic ant colony optimization	Simulated road networks	Shortest path length
[16]	Cuckoo optimization algorithm	Simulated scenarios of size 20 × 20, 100 × 100 and 200 × 200	Safe, smooth, and collision-free path
[17]	A visual-inertial navigation system	Urban areas of Hong Kong	Effective mitigation of dynamic objects and improved accuracy
[18]	Fuzzy- genetic algorithm (GA) with three path concept	Simulated maps	Computationally efficient
[19]	Improved gravitational search	Real-time navigation using Khepera III mobile robot	The safe and shortest path
[20]	Genetic algorithm	Web based virtual mobile robot laboratory	Usability of remote controlled robot laboratory
[21]	Trapezoid fuzzy 2 DOF algorithm	Simulated proportional integral derivative (PID) control system	Faster response with low position tracking error
[22]	Extreme Learning Machine and Descartes	Virtual simulation in Unity3D	Reduced local error and correction error
[23]	Full Consistency method with D* Lite	Simulated occupancy maps	Consistent determination of weight factors for effective risk management during motion
[24]	Improved Ant colony optimization	Elevation data from international society for photogrammetry and remote sensing (ISPRS) and United States geological survey (USGS)	Faster convergence

Zhang et al. [25] made an extensive survey on path planning approaches for mobile robots. In their work, they emphasized the advantages of using a genetic algorithm, particle swarm optimization, ant colony optimization, and artificial potential fields in path planning with future directions. Mac et al. [26] developed a multi-objective approach based on particle swarm optimization for path planning. They proposed a new accelerated update methodology and tested it in cluttered environments. Jun-Hao Liang and Ching-Hung Lee [27] proposed a nature-inspired optimization-based algorithm for collision-free path planning of multiple mobile robots. They developed an efficient artificial bee colony algorithm, and applied it to cluttered simulated scenarios. A Markov decision process-based path planning algorithm for humanoid robots was developed by Mahdi Fakoor et al. [28]. Das et al. [29] hybridized an improved particle swarm optimization and a gravitational search algorithm to develop a path planning algorithm for multiple robots. Simulations were performed in the Khepera environment [19] to test the efficiency.

From the above literature analysis, the following challenges were identified:

1. In the case of dynamic scenarios, there tends to be more uncertainty. Meta-heuristic approaches tend to converge more slowly to avoid a collision. Faster convergence of approximate algorithms while handling dynamic scenarios is a significant challenge.
2. Maintaining the consistency of the algorithm when dealing with dynamic scenarios is another challenge to be addressed. The algorithms must be robust and stable in the unknown scenario.
3. Though there are many algorithms in the literature for finding the collision-free shortest path, there is still a need for more intelligent algorithms with clear approximate and syllogistic reasoning.

Motivated by the challenges in the literature, and to address them, the following contribution is made in this work to efficiently plan a safe and smooth path in dynamic scenarios:

1. A fuzzy gain based dynamic ant colony optimization (FGDACO) for collision-free path planning in dynamic scenarios is proposed. The improved pheromone enhancement in the ant system will curtail unwanted traversals during the search.
2. A fuzzy logic-based collision avoidance strategy based on approximate reasoning is proposed, in addition to the pheromone enhancement. This collision avoidance strategy is combined with gain enhanced ant colony optimization for safe path planning.
3. The proposed algorithm was observed to converge faster with the improved pheromone enhancement and no local optima trap.
4. The proposed algorithm was observed to be stable in all the scenarios with a lower deviation among the independent runs.

2. Preliminaries

2.1. Fuzzy Logic and Definitions

The concept of fuzziness was introduced by Zadeh et al. [30] in 1965 as a derivation from a common set. Fuzzy sets, unlike common sets, are those whose elements have certain degrees of membership. Membership refers to the probability of an element belonging to the set. In this work, the triangular membership function was used for path planning.

Definition 1 [30]: *The triangular membership function for a fuzzy set A on the universe of discourse X:[0,1], is defined by three parameters, a, b, c; where a is the lower limit, c is the upper limit, and b is a value such that $a \leq b \leq c$. a and c form the base of triangle whereas b is the peak of the triangle. The triangular membership function is thus defined by Equation (1)*

$$\mu_A(x) = \begin{cases} 0, & x \leq a \\ \left(\frac{x-a}{b-a}\right), & a < x \leq b \\ \left(\frac{c-x}{c-b}\right), & b \leq x < c \\ 0, & x \geq c \end{cases} \quad (1)$$

where x is the crisp value that needs to be fuzzified, and a, b, c are the triangular membership function values with $a \leq b \leq c$. Figure 1 *shows a pictorial representation of Equation (1). In* Figure 1, the *x-axis is the universe of discourse, and the y-axis is the corresponding membership values.*

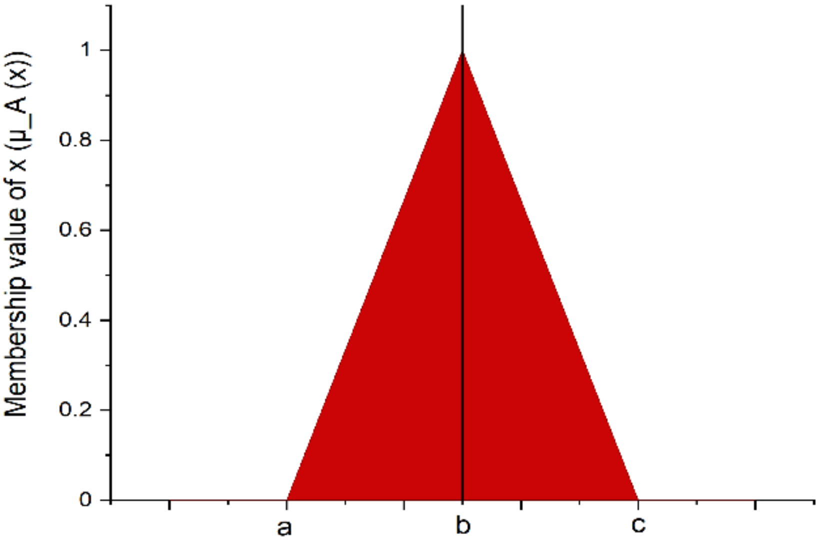


Figure 1. Pictorial representation of the triangular membership function.

A fuzzy inference system (FIS) is a vital part of fuzzy logic control. Mamdani and Assilian [31] proposed that complex processes could be controlled using fuzzy logic rather than the conventional strict models. A fuzzy logic control uses conditional sentences, called inferences, to illustrate the logic characterized by membership functions. Figure 2 shows the fuzzy logic control system. Initially, the crisp values are taken as inputs. The three essential components of FIS are (i) fuzzification, (ii) fuzzy inference engine, and (iii) defuzzification.

(i). Fuzzification: the process of transforming a crisp value to a fuzzy value is called fuzzification. This transformation is realized using the membership function. As shown in Figure 1, triangular membership functions are used in this work. Each linguistic variable will have its fuzzy variable values defined in its universe of discourse. The fuzzy variables are characterized by the membership function, with their values in (0,1). In this work, the linguistic variables are relative distance to the target, angle towards the target, and distance towards the nearest obstacle. The fuzzy variable set for each linguistic variable is (low, medium, and high).

(ii). Fuzzy Inference Engine: this is the critical unit of a fuzzy logic controller. The role of the fuzzy inference engine is to make decisions using the IF … THEN rules. The rules are represented as given below:

$$\begin{gathered} RULE\ I: \ if\ D\ is\ a_1\ and\ E\ is\ b_1,\ then\ F\ is\ c_1 \\ RULE\ II: \ if\ D\ is\ a_2\ and\ E\ is\ b_2,\ then\ F\ is\ c_2 \\ \vdots \\ RULE\ N: \ if\ D\ is\ a_i\ and\ E\ is\ b_i,\ then\ F\ is\ c_i \end{gathered}$$

Here D and E are conditional variables, F is the response variable, and a_i, b_i, c_i are the fuzzy variables defined by triangular membership functions. Once the decisions are made, a fuzzy inference system is used to evaluate the decisions made with the rules. In this work, the Mamdani fuzzy inference system is used for rule evaluation, as given by

$$\begin{aligned} RULE\ I\ &:\ \mu_1 = \mu_{a_1}(d) \wedge\ \mu_{b_1}(e) \\ RULE\ II\ &:\ \mu_2 = \mu_{a_2}(d) \wedge\ \mu_{b_2}(e) \\ &\ \vdots \\ RULE\ N\ &:\ \mu_i = \mu_{a_i}(d) \wedge\ \mu_{b_i}(e) \end{aligned}$$

(iii). Defuzzification: the fuzzy variables are converted to crisp outputs using the defuzzification phase. In this work, the centroid method is used for defuzzification. The defuzzified output x^* obtained from the centroid method can be represented as the Equation (2)

$$x^* = \frac{\int\ \mu_A\ (x)\ xdx}{\int\ \mu_A\ (x)dx} \tag{2}$$

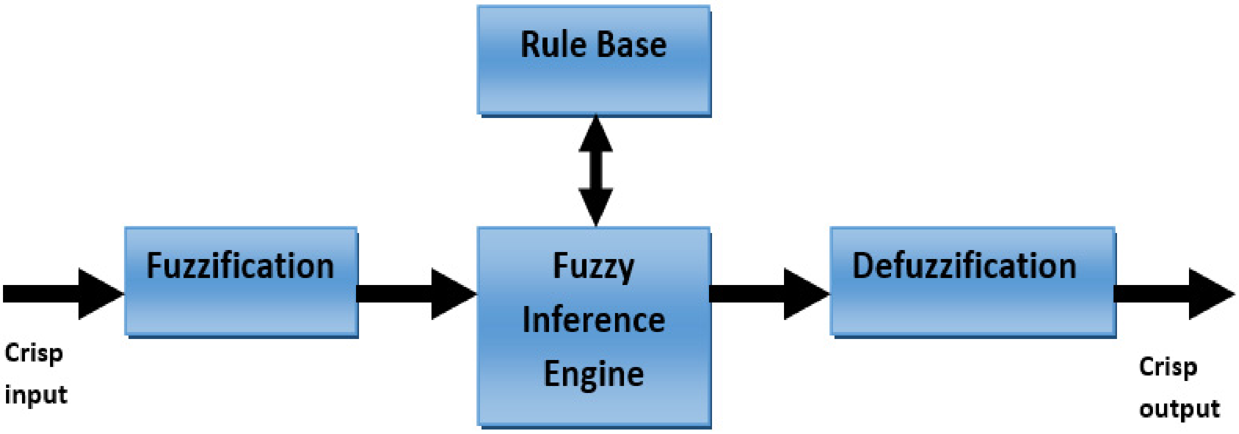


Figure 2. Fuzzy logic controller.

2.2. Ant Colony Algorithm

The ant colony optimization proposed by Marco Dorigo et al. [32] is a nature-inspired population-based swarm intelligence algorithm, inspired by the foraging behavior of ants in obtaining their food. Ants' natural tendency is to search out an optimal path between the food source and the ant nest. Ants deposit a chemical, named pheromone, as they move in search of food. A frequently used path will have more pheromone on it, while the amount of pheromone on a less frequently used path will be less. Figure 3 shows the foraging behavior of ants. Ants exhibit stigmergy by depositing pheromone on their trails. The process of adding up the amount of pheromone on a trail is called pheromone reinforcement. This reinforcement is the positive feedback indicating to the other ants to follow the trail. Artificial ants are the counterparts of natural ants. Unlike natural ants, artificial ants possess memory. In addition, to mimic natural ants' stigmergy, pheromone on the less frequently used trails decreases using the pheromone evaporation process. This evaporation is the negative feedback of the system preventing the other ants from following the trial. Similar to other meta-heuristics, ant colony optimization exhibits both exploration and exploitation. Pheromone evaporation realizes the exploration process, thus avoiding convergence to local optima. Without evaporation, all ants will follow the path left by the first ant without exploring the solution space. Performing a local search of the ants' current neighborhood window will lead to exploitation, thus working on a current solution to exploit the goodness. In this work, a pheromone enhancement mechanism to determine the current best path was adopted from [33] and combined with fuzzy logic for safe path planning. There are many ant colony optimization (ACO) variants, like the ant colony system, elitist ants, ant density model, and so on [34]. In this work, the ant system was used because of its simplicity and robustness.

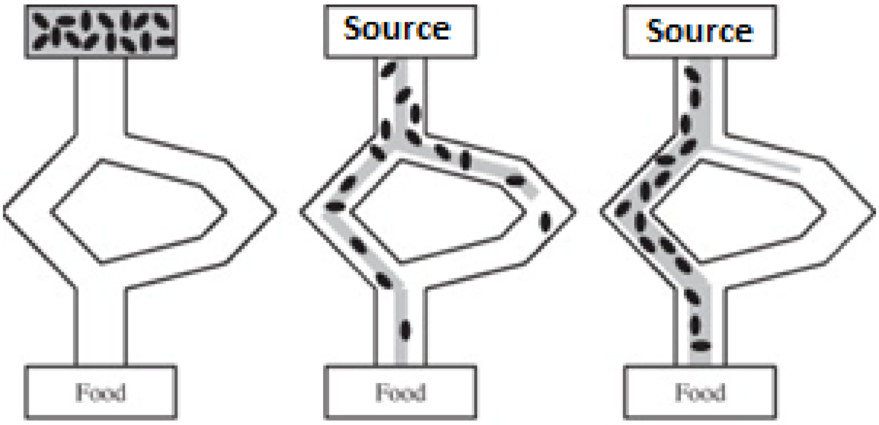


Figure 3. Foraging behavior of ants.

The notations and parameters used in ant colony optimization are as given in Table 2.

Table 2. Parameters of the ant colony algorithm.

Parameter	Description
N	Number of ants
τ_o	Initial pheromone
τ_{ij}	Quantity of pheromone deposited while traversing from *i* to *j*
η_{ij}	Heuristic function indicating the visibility of route between *i* and *j*;
d_{ij}^{k}	Cost of the route (*i*,*j*) obtained by *k*th ant
α	Impact of pheromone on the choice of next node
β	Impact of heuristic function on the selection of next node
ρ	Rate of pheromone evaporation; $0 < \rho < 1$
$visit_k$	A table containing nodes that are feasible to be visited by *k*th ant
Q	Constant related to the pheromone increment

3. Materials and Methods

3.1. Problem Definition and Formulation

Given a start, S, and destination, D, the problem of path planning in dynamic scenarios can be formulated as a minimization problem given by Equation (3) as

$$Minimize\ h(x) = W(f(x)) - (1 - W)(g(x)) \tag{3}$$

$$Minimize\ f(x) = \sum_{i=1}^{r} \sum_{j=1}^{r} x_{ij} d_{ij} \tag{4}$$

$$Maximize\ g(x) = \sum_{i=1}^{r} \sum_{j=1}^{r} x_{ij} d_{io} \tag{5}$$

subject to

$$\sum_{i=1}^{r} \sum_{j>i,i\neq j}^{r} x_{ij} = 1, \quad i,\ j = 1 \ldots ..r \tag{6}$$

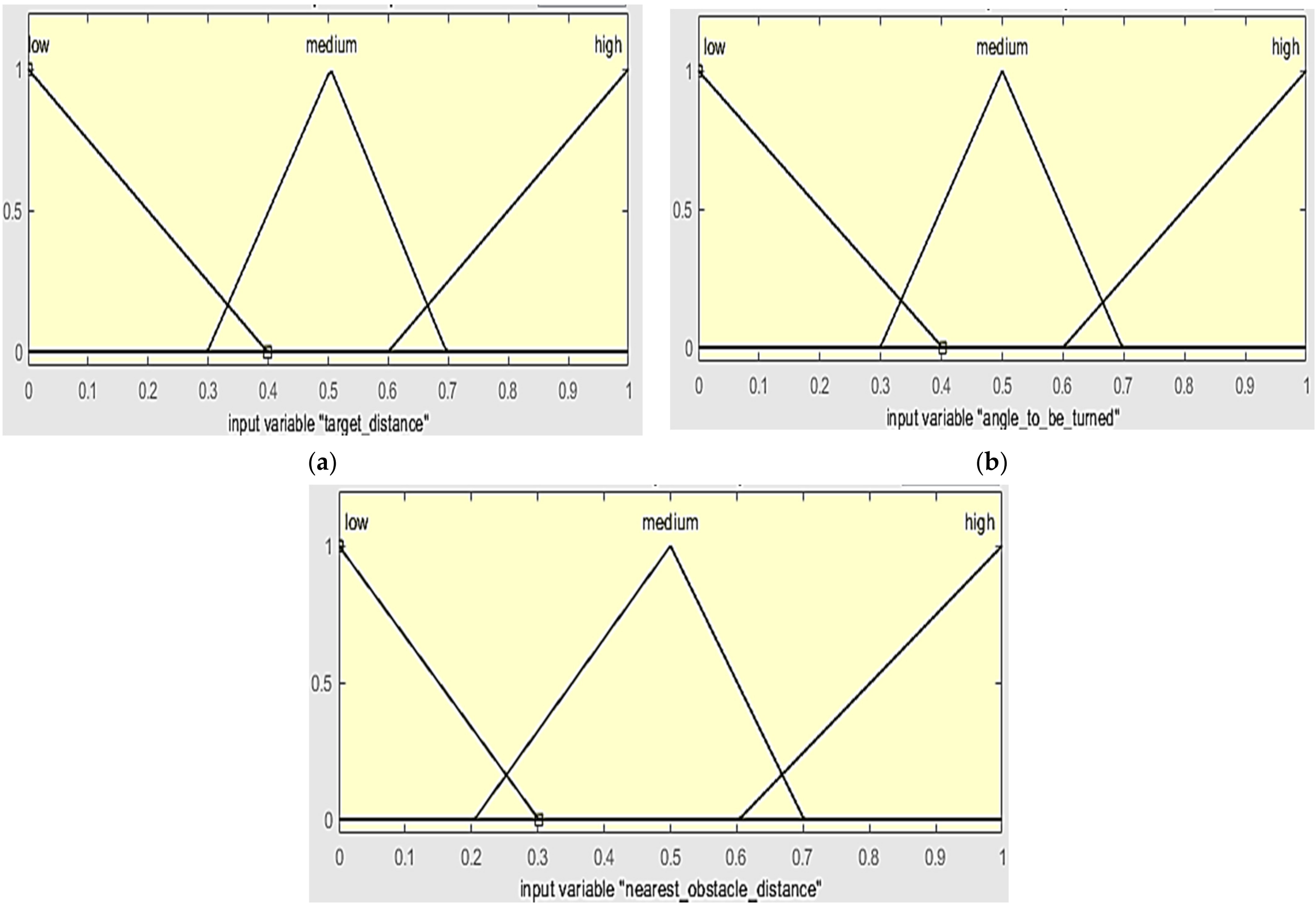
$$x_{ij} = \begin{cases} 1 & for\ a\ path\ between\ i\ and\ j \\ 0 & otherwise \end{cases} \tag{7}$$

where *r* is the total size of the scenario, $d_{ij} = \sqrt{(v_i - v_j)^2}$, $d_{io} = \sqrt{(v_i - v_o)^2}$.

In the above formulation, Equation (4) indicates the minimization of the total distance to the target during the path search indicating the target seeking behavior. Equation (5) indicates the maximization of distance concerning the obstacles indicating the obstacle avoidance behavior. Equation (6) indicates that at a time, only one node can be visited. Equation (7) indicates that a path can be included or discarded based on its existence. Equation (3) is obtained from Equations (4) and (5) with a weightage factor. Balancing the weighting between target seeking and obstacle avoidance can solve path planning. In this work, both Equations (4) and (5) are given equal weightage. The value of w is considered as 0.5.

3.2. Fuzzy Logic-Based Obstacle Avoidance

Designing a fuzzy logic control for obstacle avoidance includes creating a rule base for collision avoidance. Multiple rules can be fed into the rule base. In this work, three input parameters were considered: relative distance to the target, angle towards the target, and distance towards the nearest obstacle. The corresponding fuzzy values for the linguistic input variables were considered high, medium, and low, respectively, and characterized using a triangular membership function, with the range of [0,1]. Figure 4 shows the fuzzy variables characterized by the triangular membership function.



(**a**) (**b**)

(**c**)

Figure 4. Triangular membership function of (**a**) distance to target, (**b**) angle to be turned, and (**c**) nearest obstacle distance. X axis is the value of fuzzy variable, and Y axis is the corresponding membership.

The values are normalized to [0, 1] to reduce the computational complexity, since the final defuzzified value is given priority to the forward phase of ant colony optimization. The priority to be given to the node during the next node selection in ant colony optimization is considered an output parameter. The universe of discourse for the fuzzy variables is provided in Table 3. The values were considered intuitively, with expert opinion.

Table 3. The universe of discourse for fuzzy variables.

Linguistic Variable	U_{low}	U_{medium}	U_{high}
relative distance to the target	(0,0.4)	(0.3,0.7)	(0.6,1)
angle towards the target	(0,0.4)	(0.3,0.7)	(0.6,1)
distance towards the nearest obstacle	(0,0.3)	(0.2,0.7)	(0.6,1)

Note: Values are normalized for ethical reasons and to maintain data integrity.

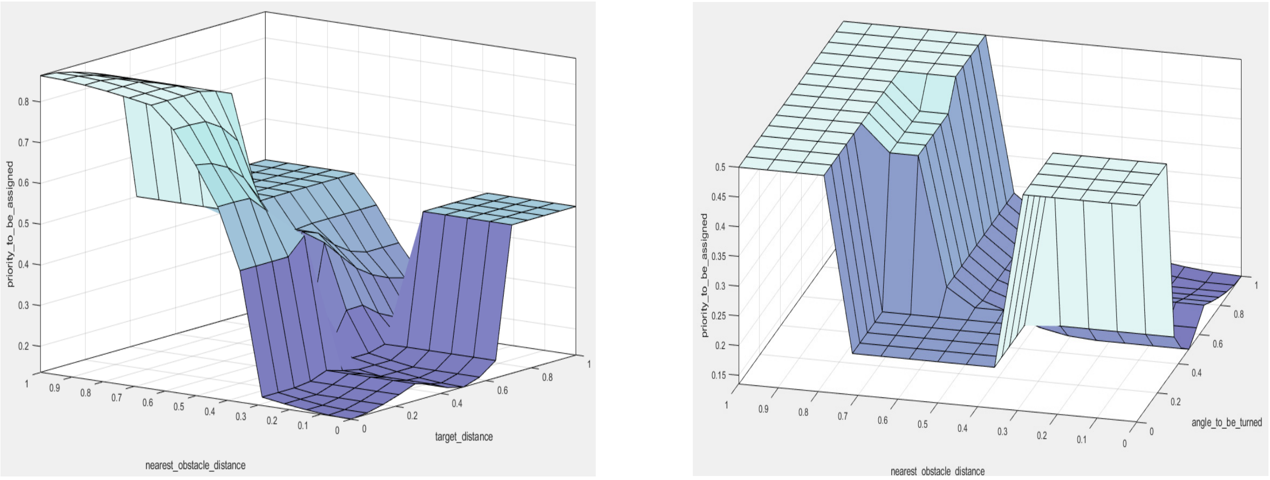
The centroid method, as given in Equation (2), is the defuzzification method. The IF-THEN rules' surface plots are shown in Figure 5a–c. Figure 5d shows the fuzzy rules characterized by the triangular membership function. Table 4 shows some of the rules considered for the fuzzy logic controller.

The fuzzy logic controller used in FGDACO for collision avoidance is given in Figure 6. This fuzzy logic controller was adopted during the process of path planning.

Table 4. Sample rules used in fuzzy control for obstacle collision avoidance.

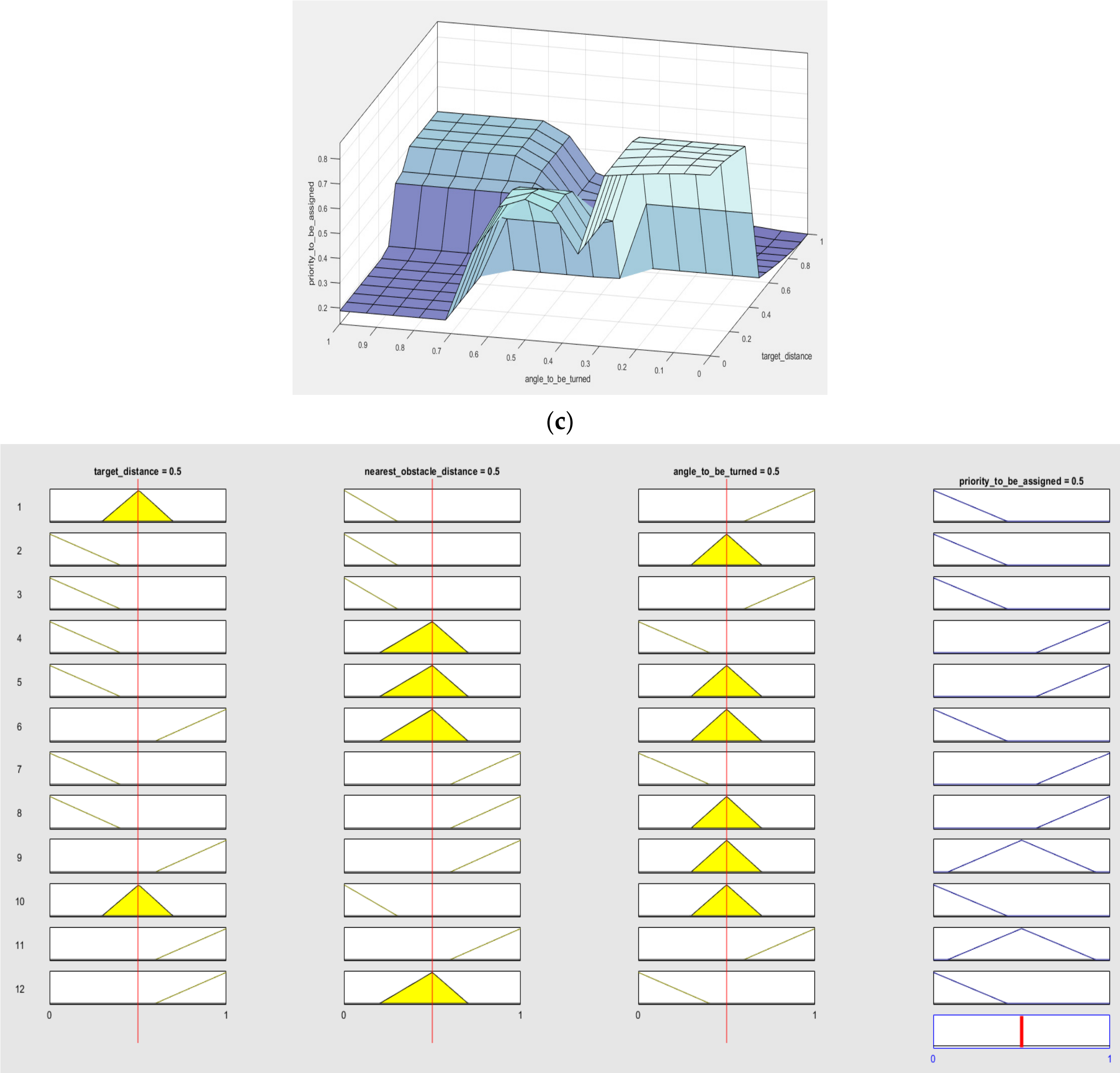
IF	IF	IF	Then
Relative distance to the target	Angle to be turned	Distance to the nearest obstacle	The priority of the node during next node selection in ant colony
Medium	High	low	Low
Low	Medium	Low	Low
Low	High	Low	Low
Low	Low	Medium	High
Low	Medium	Medium	Medium
High	Medium	Medium	Low
Low	Low	High	High
Low	Medium	High	High
High	Medium	High	Medium
Medium	Medium	Low	Low
High	High	High	Medium
High	Low	Medium	Low

Note: All rules pose an AND relation, as all factors are essential to be satisfied.



(**a**) (**b**)

Figure 5. *Cont.*



(**c**)

(**d**)

Figure 5. Surface plots of IF...THEN rules of: (**a**) target distance, obstacle distance, and priority; (**b**) angle to be turned, obstacle distance, and priority; (**c**) angle to be turned, target distance, and priority; (**d**) fuzzy rules characterized using triangular membership function.

3.3. Gain Based Path Planning

The pheromone trail, τ_{ij}, and the heuristic, η_{ij}, are two essential aspects that guide ant colony optimization. These two aspects are related to the exploration and exploitation of ant colony optimization. Heuristics are used to perform an extensive search of the solutions space to construct solutions. Upon construction of solutions, a pheromone update is performed on the most used trails to identify good quality solutions. In this work, the process of pheromone update was enhanced using a relative distance based gain function. Gain is a sigmoid function-based local heuristic designed, based on the neighbor's relative distance to target and distance to the nearest obstacle. This function will update the pheromone on the locally best trails, thus avoiding unnecessary traversal, and enabling faster convergence of the algorithm.

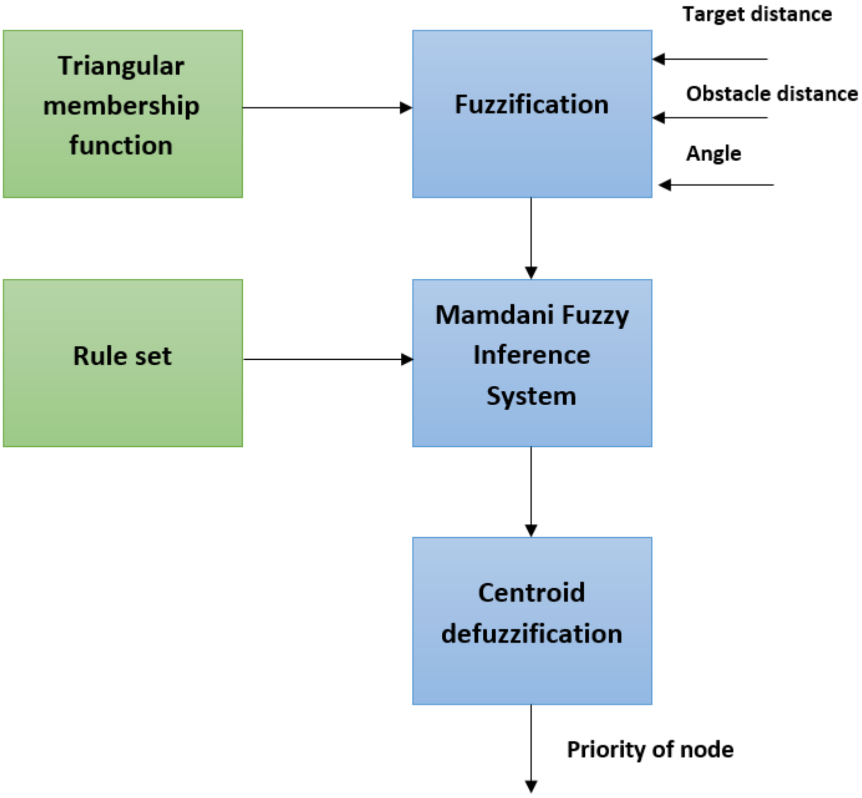


Figure 6. Fuzzy logic controller adapted for collision avoidance.

Calculating Gain

To enable quicker pathfinding, an amount of pheromone is added to the locally found best path. This enhancement is called pheromone gain. Thus the new quantity added will enable quicker pathfinding by eliminating unwanted traversals during the local search. Pheromone gain is given by (8).

$$Gain_{ij}^{k} = \frac{1}{\left(1 + e^{-\lambda * Progress_{jDest}^{k}}\right)} \tag{8}$$

$$\text{Here } Progress_{jDest}^{k} = \frac{\delta_{jDest} + d_{io}}{d_{iDest}}$$

where *Dest* = destination vertex, *j* = neighbor vertex (V_j), and *i* = current vertex (V_i). The values of learning parameter, λ, lie from 0 to 1; $\delta_{jDest} = d_{jDest}$ = euclidean distance between *j* and *dest*; for *j* = 1, 2, 3, ... 8; d_{io} is the distance between the current node and nearest obstacle. Sigmoidal function is used because of its natural smoothing character.

Consider a configuration space of 20 × 20, as shown in Figure 7.

The algorithm begins from S. To proceed further, the algorithm searches for the next node to be visited from V_i (current position). According to the gain-based sigmoid function, the neighbor node with minimum δ_{jDest} and maximum d_{io} is marked as the next node to be visited. The corresponding pheromone trail is updated. The gain based local heuristic will enable pheromone enhancement on the local best path. During the pheromone update of ant colony optimization, $Gain_{ij}$ is added to τ_{ij}^{new} of $V_{i,j3}$,(considering j3 has minimum

δ_{jDest}) and subtracted from the τ_{ij}^{new} of $V_{i,j1}$ and $V_{i,j2}$. Mathematically the procedure can be written as given in Equation (9)

$$\tau_{ij}^{new} = \begin{cases} (1-\rho)\tau_{ij}^{old} + \sum_{k=1}^{N} \Delta\tau_{ij}^{k} + Gain_{ij}^{k} & for\ min(\delta_{jDest})\text{and}max(d_{io}) \\ (1-\rho)\tau_{ij}^{old} + \sum_{k=1}^{N} \Delta\tau_{ij}^{k} - Gain_{ij}^{k} & otherwise \end{cases} \tag{9}$$

Using Equation (9), τ_{ij}^{new} is calculated and used in Equation (10).

According to Equation (8), $Gain_{ij}^{k}$ of *k*th ant is added to the node's trail with min δ_{jDest} and subtracted from other trails. This subtraction will decrease the pheromone on other trails and increase pheromone on the current best path. This pheromone update will enable the ants to settle down in the current best trail with less latency, leading to faster convergence.

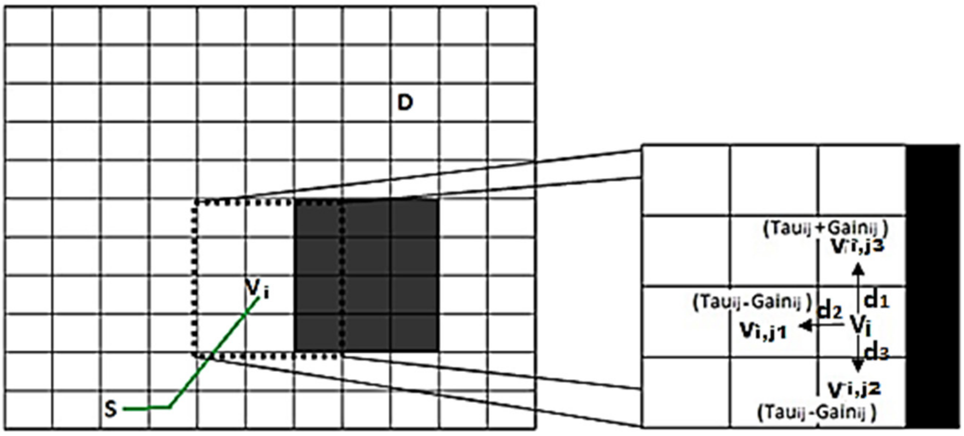


Figure 7. Illustration of Pheromone gain.

3.4. Proposed FGDACO for Target Seeking and Obstacle Avoidance

3.4.1. Environment Perception

The scenario for path planning is initially perceived and transformed into grids computationally suitable for the process. Occupancy grids were used in this work to perceive the environment for path planning. The grid resolution was one cell per meter, i.e., each cell was 1 m in size. Each grid has two probabilities, namely free space or obstacle space.

3.4.2. Ant Colony Parameters Initialization

Once the environment is perceived, the parameters of the ACO are initialized. Initially, τ_o is set to 0. α, β, ρ, start node, and destination node are initialized subsequently.

3.4.3. Node Transition and Cost Calculation

After the parameter initialization, ants start moving around the search space. This search is the onward movement of map exploration. Using the node transition probability, the next node to be visited is chosen. Equation (10) was used for the forward move. Once ants reach a node, the tabu list $visit_k$ will be checked to see if the node is allowed to be visited.

$$NTP_{ij}^{k}(t) = \begin{cases} \dfrac{\left(\tau_{ij}^{k}\right)^{\alpha}\left(\Pi_{f\epsilon parameters}\eta_{ijf}^{k}\right)^{\beta}}{\sum_{h\in visit_k}\left(\tau_{ih}^{k}\right)^{\alpha}\left(\Pi_{f\epsilon parameters}\eta_{ijf}^{k}\right)^{\beta}} & if j \in vist_k \\ 0 & otherwise \end{cases} \tag{10}$$

In Equation (10), η_{ijf} is the fuzzy cost of edge (i, j), and f is the set of parameters defined in fuzzy logic control for collision avoidance. τ_{ij} is the pheromone intensity in (i, j) and is calculated by (9) through the backward movement. η_{ijf} is defined by Equations (11)–(13).

The parameters, f, used in this work are target distance, distance to nearest obstacle, and angle to be turned.

1. Distance to target

$$\eta\ _{ijtarget\_distance} = \frac{1}{d_{idest}^{k}} \tag{11}$$

2. Nearest obstacle distance

$$\eta\ _{ijobstacle\_distance} = d_{io}^{k} \tag{12}$$

3. Angle to be turned

$$\eta\ _{ijobstacle\_distance} = \tan^{-1}\left(\frac{d_{idest}^{k}}{d_{io}^{k}}\right) \tag{13}$$

where $d_{idest}^{k} = \sqrt{\left(v_i^k - v_{dest}^k\right)^2}$; $d_{io}^{k} = \sqrt{\left(v_i^k - v_o^k\right)^2}$; v_o is the obstacle node.

η_{ijf} is the defuzzified output from the fuzzy logic control. During the calculation of τ_{ij}^{k} in Equation (9), the corresponding $Gain_{ij}^{k}$ is calculated using Equation (14).

$$Gain_{ij}^{k}\ \frac{1}{\left(1 + e^{-\lambda * progress_{jdest}^{k}}\right)} \tag{14}$$

where $progress_{jdest}^{k} = \frac{\delta_{jDest}^{k} + d_{io}^{k}}{d_{iDest}^{k}}$.

Here $\delta_{jDest}^{k} = d_{jDest}^{k}$; $d_{jDest}^{k} = \sqrt{\left(V_j^k - V_{Dest}^k\right)^2}$; $d_{iDest}^{k} = \sqrt{\left(V_i^k - V_{Dest}^k\right)^2}$.

The sigmoid function is used here to determine the pheromone enhancement. Sigmoid functions have a natural 'S' shaped curve, with an interval from 0 to 1. From mechanical observations, it could be inferred that the sigmoid function curve was similar to that of a path with smooth turns [35]. An example of the sigmoid function is shown in Figure 8.

3.4.4. Path Selection

The path selection phase is the final phase. Once the forward phase is completed, the ants start retreating by tracing their path using their memory. In this stage, the pheromone update rule is used to find the current pheromone quantity. The intensity of pheromone is updated using (15). The usability of the path is either increased by pheromone reinforcement, or decreased through pheromone evaporation.

The quantity of $\Delta\tau_{ij}^{k}$ is given by (15),

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{1}{d_{ij}^{k}} + d_{io}^{k} & if\ kth\ ant\ passes\ i\ and\ j \\ 0 & otherwise \end{cases} \tag{15}$$

By Equation (15), a pheromone update is made on the trail, which has a minimum distance to the next node, and maximum distance from the nearest obstacle.

The pseudo-code for FGDACO is given in Algorithm 1.

Algorithm 1 Framework for fuzzy gain-based dynamic ant colony optimization (FGDACO)

Input: $G, N, S, D, \alpha, \beta, \rho$
Output: best_path

```
begin
Initialize $\tau_o, \alpha, \beta, \rho, S, D, N, best\_path_{cost}$ = 0, path = [], best_path = [], $path_{cost}$
while (max_number_of_iterations_not_reached) do
    for each $k \epsilon N$ do
            $Path_{ij}^{k}(t) = \begin{cases} \frac{\left(\tau_{ij}^{k}\right)^{\alpha}\left(\Pi_{f \epsilon parameters}\eta_{ijf}^{k}\right)^{\beta}}{\sum_{h \in visit_k}\left(\tau_{ih}^{k}\right)^{\alpha}\left(\Pi_{f \in parameters}\eta_{ijf}^{k}\right)^{\beta}} & if\ j \in visii_k \\ 0 & otherwise \end{cases}$
        Update $visit_k$ ;
        If ($path_{ij}^{k} < best\_path_{cost}$)
                $best\_path_{cost} \leftarrow path_{ij_{cost}}^{k}$
                best_path $\leftarrow path_{ij}^{k}$
        end if
        If(target_reached)
            Update $visit_k$ for all $N$
        break
        end if
    end for
    for each $Path_{ij}^{k}$ do
    $\tau_{ij}^{new} = \begin{cases} (1-\rho)\tau_{ij}^{old} + \sum_{k=1}^{N} + \Delta\tau_{ij}^{k} + Gain_{ij}^{k} & for\ min\left(\delta_{ij}\right) and\ max(d_{io}) \\ (1-\rho)\tau_{ij}^{old} + \sum_{k=1}^{N} + \Delta\tau_{ij}^{k} - Gain_{ij}^{k} & otherwise \end{cases}$
        // Compute $Gain_{ij}^{k}$
    $\delta_{jDest}$=$d_{jDest}$, $j = 1, 2 \ldots 8$
    for all $V_j\ of\ V_i$ do
        $progress_{jdest}^{k} = \frac{\delta_{jDest}^{k} + d_{io}}{d_{iDest}^{k}}$
        $Gain_{ij}^{k} = \frac{1}{\left(1 + e^{-\lambda * progress_{jdest}^{k}}\right)}$
    end for
    $\Delta\tau_{ij}^{k} = \begin{cases} \frac{1}{d_{ij}^{k}} + d_{io}^{k} & if\ k^{th} ant\ passes\ i\ and\ j \\ 0 & otherwise \end{cases}$
   end for
  end while
return best_ path
end
```

The algorithm continues until either of the following conditions is achieved: the shortest path is obtained, or predetermined iterations have been completed. The framework for FGDACO is as given in Figure 9.

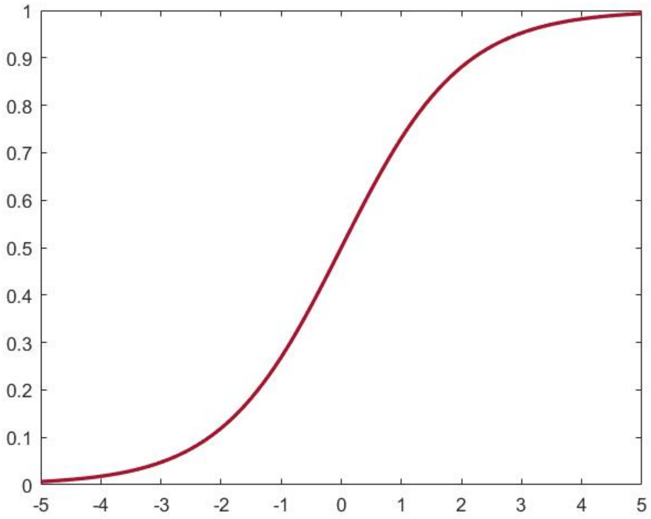


Figure 8. An example of the sigmoid function.

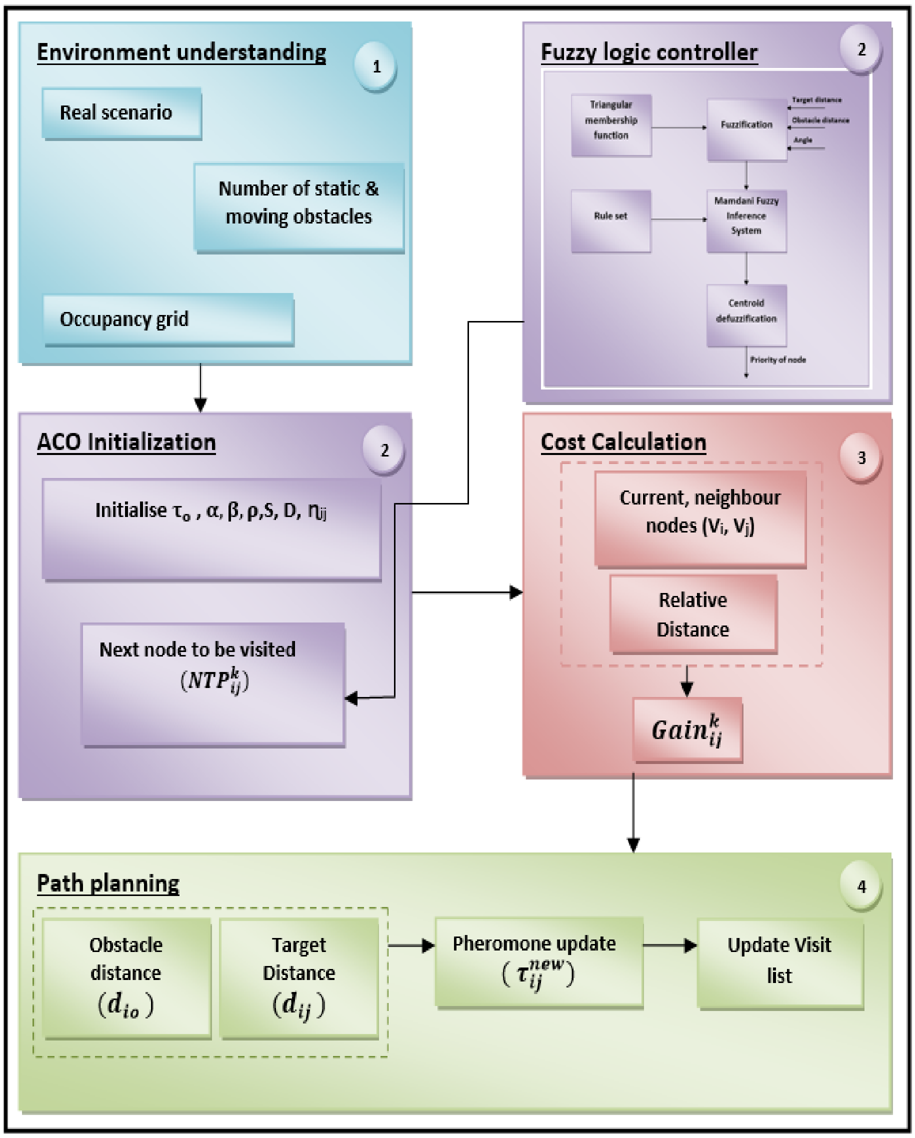


Figure 9. Framework for FGDACO.

4. Experimental Results and Discussion

4.1. Experimental Setup and Dataset Description

Simulations of the proposed algorithms were implemented in MATLAB® 2019b [36]. Three different occupancy grids of size 100 × 100, with moving and static obstacles were used for the simulations. Obstacles of different shapes were also simulated. The details of the simulations are given in Table 5. The following assumptions were considered for the simulations:

- The start and destinations were considered to be the same during the whole process of path planning, but they differed with each scenario.
- The speed of the moving obstacles was considered as random between 0.5–1.5 m/s.

Table 5. Description of obstacles in the scenarios used for the simulations.

Scenario	Static Obstacle	Moving Obstacle
1	3	3
2	4	3
3	2	3

4.2. Performance Measures:

The performance of the proposed method was evaluated using the following performance measures. Each algorithm was run 30 times, and the analysis was made using the following metrics.

- Standard Deviation: The consistency of the proposed method was verified using standard deviation. The method was stable when there was less variation in the performance between independent runs.
- Median of path length and computation time: The median of the computational time and the length of path computed for 30 independent runs were compared and analyzed.

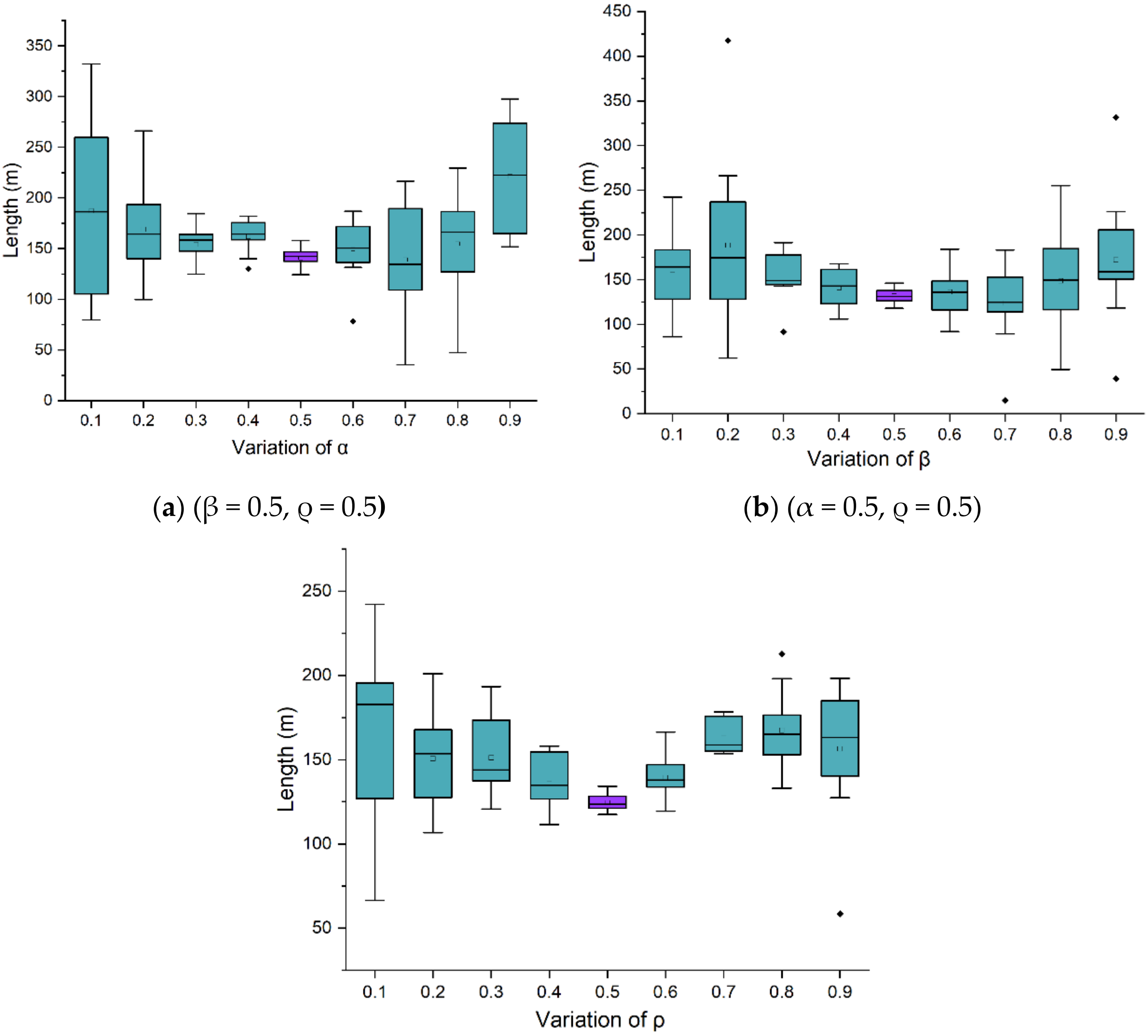
4.3. Parameter Setting:

The simulations were performed under different conditions like different iterations, varying α, β, and ρ. The values of parameters that were used for the simulations are given in Table 6.

Table 6. Parameters and their values used for simulation of FGDACO.

Parameter	Value
α	0.5
β	0.5
ρ	0.5
Time interval	3 Δt
Sampling interval	10 s
Number of iterations	100
Number of independent runs of the algorithm	30
Number of ants (N)	20

An optimal value for α, β, ρ was essential for the algorithm's adequate performance since pheromone enhancement is a significant part of FGDACO. The value of α, β, ρ was varied from 0.1 to 0.9, and the results are shown in Figure 10. Figure 10 shows that the variation in length values was the least when α = 0.5, β = 0.5, and ρ = 0.5. Thus, the algorithm is stable with less deviation among its independent runs, and no outliers, when α = 0.5, β = 0.5, and ρ = 0.5.



(**a**) $(\beta = 0.5, \varrho = 0.5)$ (**b**) $(\alpha = 0.5, \varrho = 0.5)$

(**c**) $(\alpha = 0.5, \beta = 0.5)$

Figure 10. (**a**) Variation of α, (**b**) variation of β, and (**c**) variation of ρ.

From the parameter setting simulation in Figure 10a–c, it can be seen that the value of length was minimum when $\alpha = 0.5$, $\beta = 0.5$, and $\rho = 0.5$. Thus $\alpha = 0.5$, $\beta = 0.5$, and $\rho = 0.5$ were set for performance evaluation.

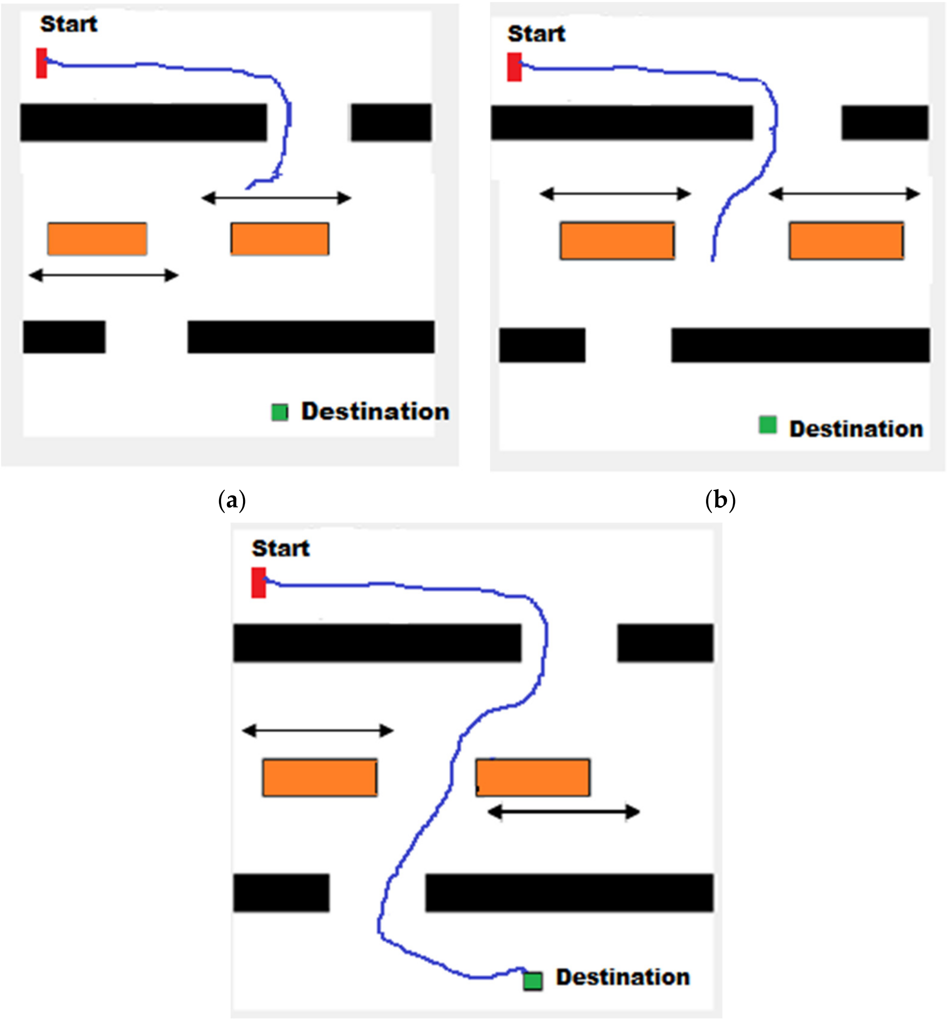
4.4. Performance Evaluation

The performance of the FGDACO was compared with existing methods, like the cuckoo optimization algorithm (COA), fuzzy-genetic algorithm (Fuzzy-GA), and fuzzy logic based ant colony optimization (FLACO), in three sample scenarios, with a varying number of obstacles. The performance results in terms of path length and computation time taken are summarized in Table 7. It is evident from Table 7 that, the proposed FGDACO computed the shortest path with minimum time when compared to the others. Moreover, FGDACO was found to be stable, with a minimum standard deviation. In dynamic path planning, one of the essential objectives is to find a collision-free safe path. The mechanism of collision avoidance is illustrated in Figure 11a–c, with sample scenarios. The direction of the moving obstacle (orange color) is shown with the help of directed arrows.

Table 7. Performance comparison of FGDACO against the other methods. A median of 30 independent runs was considered.

Scenario #	Algorithm	Time (s)		Length (m)	
		Median	**SD**	**Median**	**SD**
1	Proposed FGDACO	28.97	2.24	126.65	1.37
	COA	37.94	4.35	134.57	3.48
	Fuzzy-GA	41.79	4.58	141.24	5.69
	FLACO	31.47	3.77	129.64	2.78
2	Proposed FGDACO	38.74	1.97	135.96	2.37
	COA	51.76	4.69	158.96	4.99
	Fuzzy-GA	48.61	3.47	149.67	6.35
	FLACO	43.78	2.07	141.27	3.78
3	Proposed FGDACO	74.33	2.54	197.69	2.77
	COA	84.95	4.77	214.68	5.78
	Fuzzy-GA	81.31	3.18	219.96	6.35
	FLACO	77.12	2.11	201.43	4.69

SD: standard deviation. COA: cuckoo optimization algorithm. Fuzzy-GA: fuzzy-genetic algorithm. FLACO: fuzzy logic based ant colony optimization. The Wilcoxon signed-rank test at 95% confidence is conducted and verified to compare FGDACO with others.



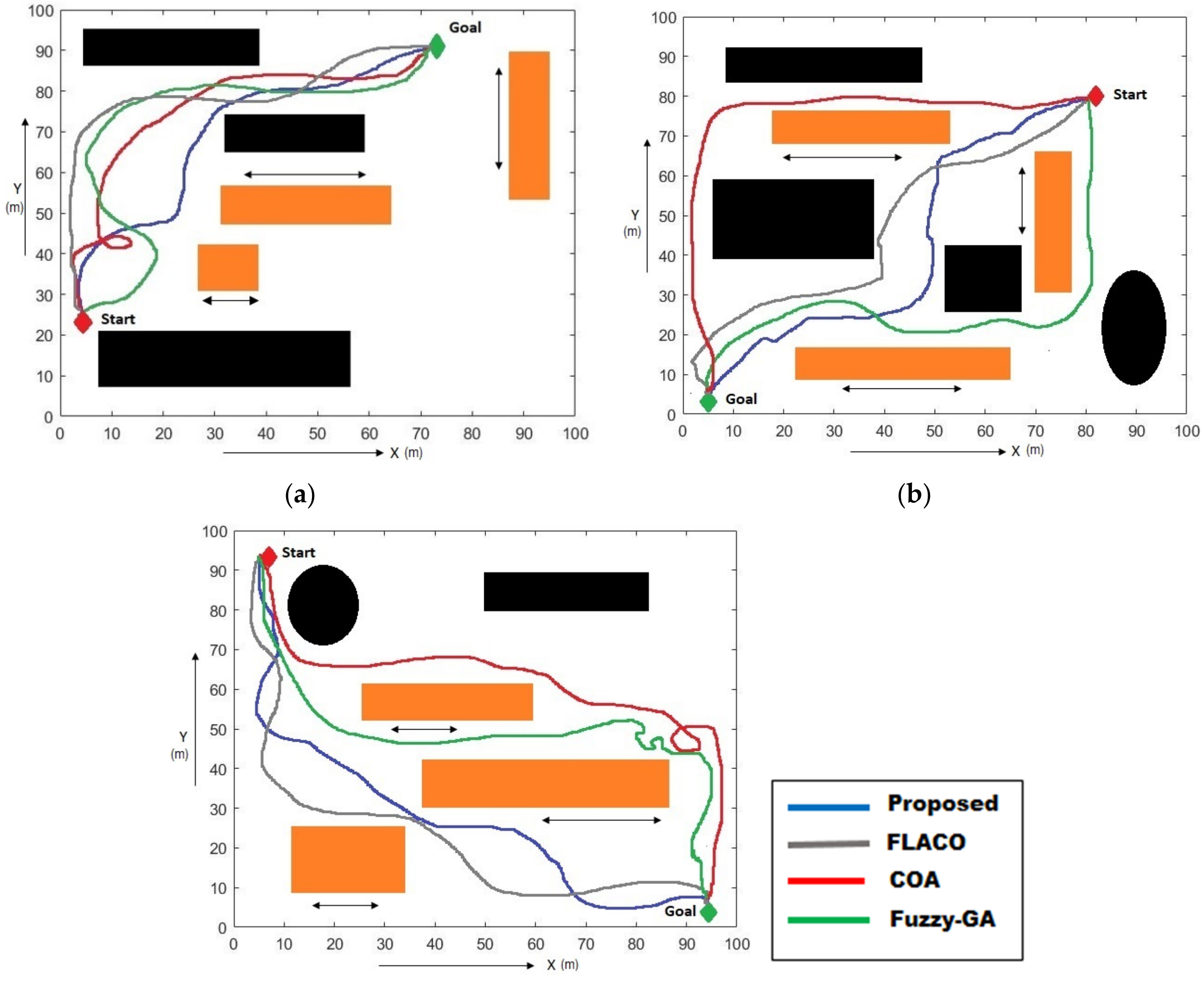
(**a**) (**b**)

(**c**)

Figure 11. (**a**–**c**) Collision avoidance behavior of FGDACO.

From Figure 11a–c, the obstacle avoidance behavior of FGDACO can be inferred. In Figure 11a, the process of path planning begins from the start position. The search proceeds further after encountering the static obstacles. Upon encountering moving obstacles, the next node's priority is determined from the FIS engine's fuzzy rules. The next nodes' priorities are assigned based on the obstacles' distance and relative distance to the target. In Figure 11b, it can be seen that the path is planned such that it is farther from the oncoming obstacle. Once the moving obstacles are encountered and the search has passed through them, the rest of the static obstacles are encountered. Figure 11c shows the final path planned.

The paths planned by FGDACO and the existing algorithms are shown in Figure 12a–c. It could be inferred from the literature that paths with fewer sharp turns are favored for real-time traversal. From Figure 12a–c, it can be seen that in all scenarios, the path planned by FGDACO had a smaller number of sharp turns when compared with the paths planned by the other algorithms. The path planned by FGDACO was efficient in all three scenarios, irrespective of the number and shape of obstacles.



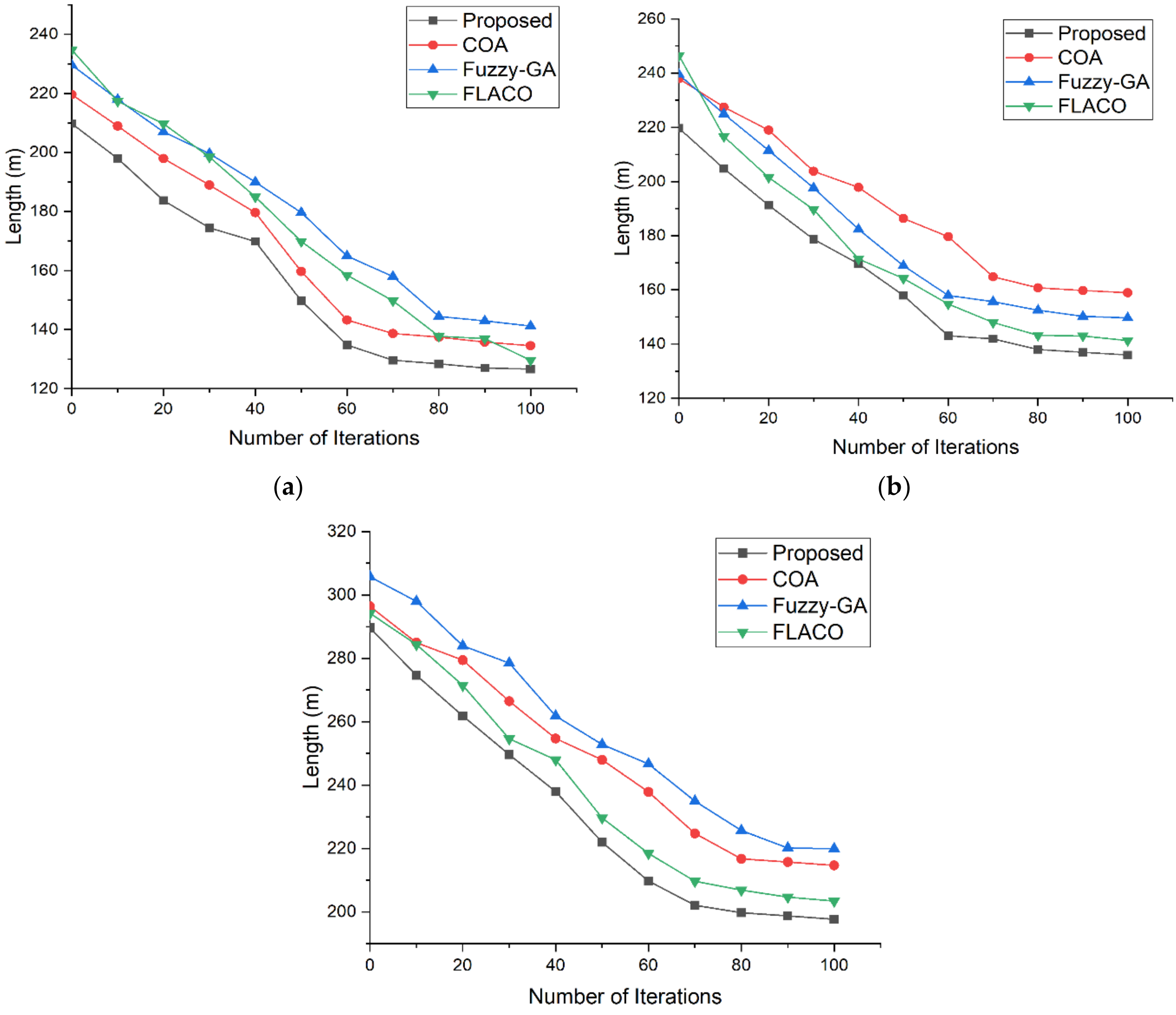
(**a**)

(**b**)

(**c**)

Figure 12. Path planned by the proposed FGDACO and other existing algorithms in (**a**) scenario 1, (**b**) scenario 2, and (**c**) scenario 3.

Convergence plots of FGDACO and other existing methods are given in Figure 13a–c. Figure 13a–c shows that the proposed FGDACO converged faster than COA, Fuzzy-GA, and FLACO with regard to length. FGDACO was also less likely to get trapped in local optima because of the enhanced pheromone update. The gain based sigmoid function improved the current best path, thereby leading to better exploitation.



(**a**)

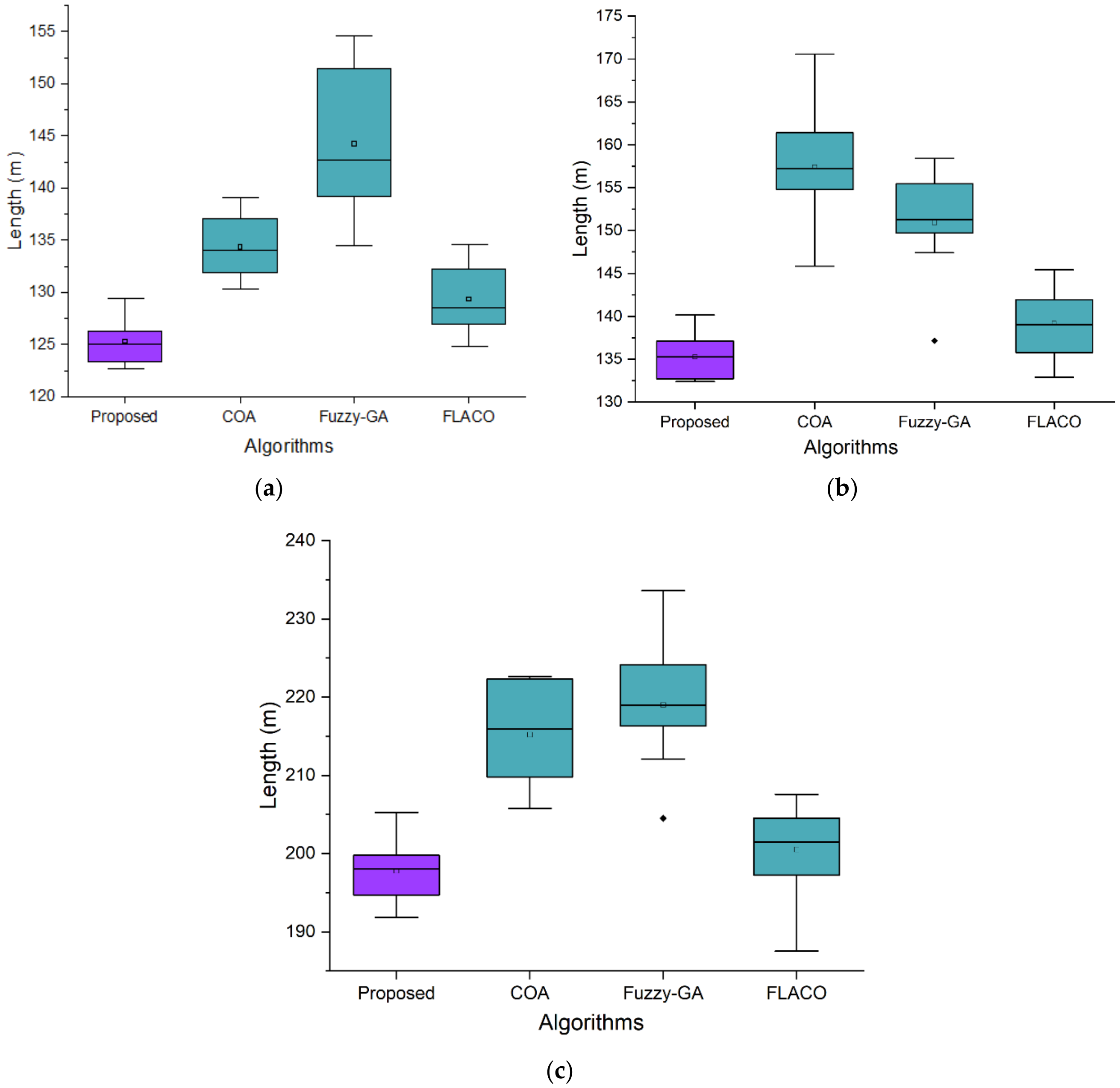
(**b**)

(**c**)

Figure 13. The convergence of the proposed FGDACO and existing methods in (**a**) scenario 1, (**b**) scenario 2, and (**c**) scenario 3.

Box plots are the standard graphical representation of the distribution of values in a range. The consistency of the proposed FGDACO was analyzed using standard deviation. The standard deviation of the proposed FGDACO and existing algorithms could be graphically analyzed using a box plot. Figure 14a–c shows the box plots of FGDACO and existing algorithms for all three scenarios. It can be seen that FGDACO exhibited the least variation in the converged length values among the independent runs. It can also be seen that in all three scenarios, the box plots of FGDACO are almost symmetric, without outliers.

A performance comparison of FGDACO with COA, Fuzzy-GA, and FLACO is summarized in Table 7.



(**a**)

(**b**)

(**c**)

Figure 14. Box plots of the proposed FGDACO and existing methods in (**a**) scenario 1, (**b**) scenario 2, and (**c**) scenario 3.

4.5. Discussion

From the above performance evaluation and Table 7, the following summarization can be made:

1. The proposed FGDACO outperformed COA, Fuzzy-GA, and FLACO by 6%, 11%, and 3%, respectively, in terms of length in scenario 1.
2. The proposed FGDACO outperformed COA, Fuzzy-GA, and FLACO by 15%, 10%, and 4%, respectively, in terms of length in scenario 2.
3. The proposed FGDACO outperformed COA, Fuzzy-GA, and FLACO by 8%, 10%, and 2%, respectively, in terms of length in scenario 3.
4. With regard to consistency, FGDACO exhibited higher consistency, with a deviation of 2% on average among its independent runs.

From the above discussion, it is evident that FGDACO effectively planned the shortest path in the least time in the tested scenarios. Furthermore, FGDACO was efficient, irrespective of the shape and number of obstacles in a real-time environment.

5. Conclusions

A fuzzy gain-based dynamic ant colony optimization was proposed for safe and smooth collision-free path planning. The pheromone update mechanism of ant colony optimization was improved with a relative distance based local heuristic. This improved mechanism helped in curtailing unwanted traversals during the search. To avoid collision with moving obstacles, fuzzy logic based collision avoidance was proposed. A rule base with syllogistic reasoning was created and evaluated using the Mamdani fuzzy inference system. The output of FIS was considered the node's priority during the next node selection in ant colony optimization. Simulations were performed on occupancy grids, and evaluated using path length and time taken metrics. From Table 7, it can be seen that the proposed FGDACO outperformed the existing methods, COA, Fuzzy-GA, and FLACO, in terms of length by 9.6%, 10.3%, and 3% in the tested scenarios. The proposed algorithm also converged more quickly (Figure 13a–c with high consistency with a deviation of 2% among its independent runs). There was less deviation among its independent runs (Figure 14a–c). The results were also statistically significant at a 95% level of confidence. From the discussions and investigations, it is evident that the proposed FGDACO is robust, stable, reasonable, and safe, with faster convergence. In the future, the algorithm can be applied in real-time road networks and on variants of vehicle routing problems.

Author Contributions: The contributions and responsibilities of the authors were as follows. V.S. has contributed to conceptualization, formal analysis, literature analysis, coding, validation, and writing. R.K. has contributed to conceptualization, formal analysis, literature analysis, validation, writing. F.C., D.P., A.M., K.S.R. and S.K. have contributed to overall administration and curation, resources acquisition, conceptualization, formal analysis, reviewing, and editing. All authors have read and agreed to the published version of the manuscript.

Funding: Authors thank the funding agencies viz., Defence Research & Development Organization, India; and Department of Science & Technology, India for their financial aid from grant nos. (ERIP/ER/1203080/M/01/1569; and SR/FST/ETI-349/2013) respectively.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors thank the editor and the anonymous reviewers for their valuable comments, which improved our research quality.

Conflicts of Interest: Authors hereby declare that there is no conflict of interest.

References

1. Yang, X.S. *Nature-Inspired Algorithms and Applied Optimization*; Springer: Berlin/Heidelberg, Germany, 2017; Volume 744.
2. Boussaïd, I.; Lepagnot, J.; Siarry, P. A survey on optimization metaheuristics. *Inf. Sci.* **2013**, *237*, 82–117. [CrossRef]
3. Tuncer, A.; Yildirim, M. Dynamic path planning of mobile robots with improved genetic algorithm. *Comput. Electr. Eng.* **2012**, *38*, 1564–1572. [CrossRef]
4. Son, C. Intelligent rule-based sequence planning algorithm with fuzzy optimization for robot manipulation tasks in partially dynamic environments. *Inf. Sci.* **2016**, *342*, 209–221. [CrossRef]
5. Bakdi, A.; Hentout, A.; Boutami, H.; Maoudj, A.; Hachour, O.; Bouzouia, B. Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control. *Robot. Auton. Syst.* **2017**, *89*, 95–109. [CrossRef]
6. Pandey, A.; Parhi, D.R. Optimum path planning of mobile robot in unknown static and dynamic environments using Fuzzy-Wind Driven Optimization algorithm. *Def. Technol.* **2017**, *13*, 47–58. [CrossRef]
7. Hu, X.; Chen, L.; Tang, B.; Cao, D.; He, H. Dynamic path planning for autonomous driving on various roads with avoidance of static and moving obstacles. *Mech. Syst. Signal. Process.* **2018**, *100*, 482–500. [CrossRef]
8. Chen, P.; Zhang, X.; Chen, X.; Liu, M. Path planning strategy for vehicle navigation based on user habits. *Appl. Sci.* **2018**, *8*, 407. [CrossRef]

9. Wei, K.; Ren, B. A method on dynamic path planning for robotic manipulator autonomous obstacle avoidance based on an improved RRT algorithm. *Sensors* **2018**, *18*, 571. [CrossRef] [PubMed]
10. Wang, M.; Liu, J.N. Fuzzy logic-based real-time robot navigation in unknown environment with dead ends. *Robot. Auton. Syst.* **2008**, *56*, 625–643. [CrossRef]
11. Wu, Q.; Chen, Z.; Wang, L.; Lin, H.; Jiang, Z.; Li, S.; Chen, D. Real-Time Dynamic Path Planning of Mobile Robots: A Novel Hybrid Heuristic Optimization Algorithm. *Sensors* **2020**, *20*, 188. [CrossRef]
12. Alomari, A.; Phillips, W.; Aslam, N.; Comeau, F. Dynamic fuzzy-logic based path planning for mobility-assisted localization in wireless sensor networks. *Sensors* **2017**, *17*, 1904. [CrossRef] [PubMed]
13. Salehinejad, H.; Talebi, S. Dynamic fuzzy logic-ant colony system-based route selection system. *Appl. Comput. Intell. Soft Comput.* **2010**, *2010*, 1–13. [CrossRef]
14. Purian, F.K.; Sadeghian, E. Mobile robots path planning using ant colony optimization and Fuzzy Logic algorithms in unknown dynamic environments. In Proceedings of the 2013 International Conference on Control, Automation, Robotics and Embedded Systems (CARE), Jabalpur, India, 16–18 December 2013; pp. 1–6.
15. Song, Q.; Zhao, Q.; Wang, S.; Liu, Q.; Chen, X. Dynamic Path Planning for Unmanned Vehicles Based on Fuzzy Logic and Improved Ant Colony Optimization. *IEEE Access* **2020**, *8*, 62107–62115. [CrossRef]
16. Hosseininejad, S.; Dadkhah, C. Mobile robot path planning in dynamic environment based on cuckoo optimization algorithm. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 1729881419839575. [CrossRef]
17. Bai, X.; Wen, W.; Hsu, L.T. Robust Visual-Inertial Integrated Navigation System Aided by Online Sensor Model Adaption for Autonomous Ground Vehicles in Urban Areas. *Remote Sens.* **2020**, *12*, 1686. [CrossRef]
18. Singh, S.J.; Roy, S.; Singh, K.M.; Khelchandra, T. Motion planning of mobile robot using Fuzzy-GA method along with three path concept in dynamic environment. *J. Intell. Fuzzy Syst.* **2018**, *35*, 1445–1457. [CrossRef]
19. Das, P.K.; Behera, H.S.; Jena, P.K.; Panigrahi, B.K. Multi-robot path planning in a dynamic environment using improved gravitational search algorithm. *J. Electr. Syst. Inf. Technol.* **2016**, *3*, 295–313. [CrossRef]
20. Solak, S.; Yakut, Ö.; Dogru Bolat, E. Design and Implementation of Web-Based Virtual Mobile Robot Laboratory for Engineering Education. *Symmetry* **2020**, *12*, 906. [CrossRef]
21. Bi, M. Control of Robot Arm Motion Using Trapezoid Fuzzy Two-Degree-of-Freedom PID Algorithm. *Symmetry* **2020**, *12*, 665. [CrossRef]
22. Li, S.; Xie, J.; Wang, X.; Ren, F.; Zhang, X.; Bao, Q. Path Planning of Hydraulic Support Pushing Mechanism Based on Extreme Learning Machine and Descartes Path Planning. *Symmetry* **2021**, *13*, 97. [CrossRef]
23. Zagradjanin, N.; Pamucar, D.; Jovanovic, K. Cloud-Based Multi-Robot Path Planning in Complex and Crowded Environment with Multi-Criteria Decision Making Using Full Consistency Method. *Symmetry* **2019**, *11*, 1241. [CrossRef]
24. Sangeetha, V.; Krishankumar, R.; Ravichandran, K.S.; Kar, S. Energy-efficient green ant colony optimization for path planning in dynamic 3D environments. *Soft Comput.* **2021**, *1*, 21.
25. Zhang, H.-Y.; Lin, W.-M.; Chen, A.-X. Path Planning for the Mobile Robot: A Review. *Symmetry* **2018**, *10*, 450. [CrossRef]
26. Mac, T.T.; Copot, C.; Tran, D.T.; De Keyser, R. A hierarchical global path planning approach for mobile robots based on multi-objective particle swarm optimization. *Appl. Soft Comput.* **2017**, *59*, 68–76. [CrossRef]
27. Liang, J.H.; Lee, C.H. Efficient collision-free path-planning of multiple mobile robots system using efficient artificial bee colony algorithm. *Adv. Eng. Softw.* **2015**, *79*, 47–56. [CrossRef]
28. Fakoor, M.; Kosari, A.; Jafarzadeh, M. Humanoid robot path planning with fuzzy Markov decision processes. *J. Appl. Res. Technol.* **2016**, *14*, 300–310. [CrossRef]
29. Das, P.K.; Behera, H.S.; Panigrahi, B.K. A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning. *Swarm Evol. Comput.* **2016**, *28*, 14–28. [CrossRef]
30. Zadeh, L.A. Fuzzy sets (PDF). *Inf. Control* **1965**, *8*, 338–353. [CrossRef]
31. Mamdani, E.H. Application of fuzzy algorithms for control of simple dynamic plant. *Proc. Inst. Electr. Eng.* **1974**, *121*, 1585–1588. [CrossRef]
32. Dorigo, M.; Maniezzo, V.; Colorni, A. Ant system: Optimization by a colony of cooperating agents. *IEEE Trans. Syst. Man Cybern. Part B* **1996**, *26*, 29–41. [CrossRef]
33. Sangeetha, V.; Ravichandran, K.S.; Shekhar, S.; Tapas, A.M. An Intelligent Gain-based Ant Colony Optimisation Method for Path Planning of Unmanned Ground Vehicles. *Def. Sci. J.* **2019**, *69*, 167–172.
34. Padhy, N.P. *Artificial Intelligence and Intelligent Systems*; Oxford University Press: Oxford, UK, 2005.
35. Ravankar, A.; Ravankar, A.A.; Kobayashi, Y.; Hoshino, Y.; Peng, C.C. Path smoothing techniques in robot navigation: State-of-the-art, current and future challenges. *Sensors* **2018**, *18*, 3170. [CrossRef] [PubMed]
36. MATLAB® (2019). *9.7.0.1261785 (R2019b)*; The MathWorks Inc.: Natick, MA, USA.