International Journal of Artificial Intelligence,
ISSN 0974-0635; Int. J. Artif. Intell.
2025, Volume 23, Issue Number 1

An Intuitionistic Fuzzy-Based Routing Algorithm for Autonomous Mobile Robot Navigation in Complex Environments with Obstacles

Mahshid Hosseinpoor[1] and **Mohammad Hossein Moattar**[2] and **Seyyed Javad Seyyed Mahdavi Chabok**[3] and **Alireza Jahandoost**[4]

[1]Mahshid Hosseinpoor, Department of Computer Engineering, Mashhad Branch, Islamic Azad University, Mashhad, IRAN
Email : hosseinpoor.mahshid@yahoo.com

[2] Mohammad Hossein Moattar ,Department of Computer Engineering, Mashhad Branch, Islamic Azad University, Mashhad, IRAN
Email : moattar@mshdiau.ac.ir

[3]Seyyed Javad Seyyed Mahdavi Chabok . Department Electrical Engineering, Mashhad Branch, Islamic Azad University, Mashhad,IRAN
Email : mahdavi@mshdiau.ac.ir

[4]Alireza Jahandoost, Department of Computer Engineering, Mashhad Branch, Islamic Azad University, Mashhad,IRAN
Email : Alireza.jahandoost@outlook.com

ABSTRACT

The issue of route planning by robots is very important So far, the problem has been solved using a variety of techniques, each of which has some drawbacks. The proposed approach employs intuitionistic fuzzy reasoning for more precise decisions. In this approach, the space is divided into a Voronoi diagram, and the best route is then determined using optimized A algorithm. The intuitionistic fuzzy set (IFS) is used to smooth out the path where there are fractures around obstacles. Intuitive fuzzy has inputs including the distances between the robot and obstacles, the amount of friction the robot (for the first time) has with the surface it is moving over, and the angle of the robot's head. The output of the system is the speed of the right and left wheels and the next angle of the robot's head. Although fuzzy systems have previously been used to solve the problem but IFS and paying attention to the type of coating (friction coefficient) increase the accuracy compared to traditional methods.*

Keywords: Autonomous mobile robot, Path planning, Intuitionistic fuzzy set , Voronoi diagram, A*

Computing Classification System: : I.4

1. INTRODUCTION

In Due to the importance of the robot routing issue, this topic has been researched and studied for many years. Finding the shortest path, an unobstructed path (or at least one with the fewest obstacles), getting to the destination, and doing so with the least amount of energy and time are some of the challenging aspects of guiding a mobile robot. Numerous routing algorithms have been used in recent years to find the best and shortest routes, and a variety of methods, including evolutionary algorithms, artificial intelligence methods, fuzzy systems, game theory, neural networks, soft computing methods, and combined methods, have been used to navigate the designated path. Each of the methods offered has some benefits and drawbacks. Even though evolutionary algorithms [1], [2] are used to solve many issues today, these techniques are not very beneficial in solving some problems, such as finding the robot's target. These algorithms use numerous loops, which consume a significant amount of time and resources. Also in neural networks [3], [4], [5], a significant flaw is that they take a long time to solve problems because of their multi-layer architecture, which is inappropriate given the importance of time in the robot routing problem.

The reason why time is important in the robot path planning problem is the application of robots. Robots are now used in some extremely sensitive and important environments and locations. For instance, robots are used in places where it is impossible for people to commute, like earth quake affected areas. In these situations, time is unquestionably important. Additionally, accuracy and timing have an impact on other robot applications in factories and other industrial areas, like carrying heavy objects.
Although using fuzzy systems [6], [7] to solve these problems takes less time than using evolutionary algorithms, the proposed fuzzy systems still have many flaws. For instance, when a fuzzy system is used, it may result in a large number of fuzzy rules, the fuzzy system may have some errors, like not finding or reaching the target, or all the important factors may not be taken into account as fuzzy inputs. The method described in this paper takes into account all of the fuzzy systems' influencing factors, the number of rules is appropriate, and finally, intuitionistic fuzzy set (IFS) is used in place of a fuzzy system, increasing accuracy and correspondingly reducing error. It is due to the fact that IFS calculates the degrees of non-membership and doubt in addition to the degree of membership that is calculated in conventional fuzzy systems. As a result, a thorough method that takes little time and has high accuracy is accomplished.

Along with all of the previously mentioned issues, many of the previously presented methods did not take obstacles into account, making them only useful in certain environments. Nevertheless, regardless of how many obstacles there are, this paper offers a suitable strategy for getting around them. The Voronoi diagram is used in this study to create some paths from the robot's starting point to its finishing point. The best and shortest path among those that have been created is then chosen using the A* algorithm. Although earlier methods used routing algorithms, they connect all the nodes with each other, which made the resulting graph quite complicated. This method does not connect all of the nodes; instead, only the neighboring nodes are connected. Because of this, the graph obtained is much simpler. There will be some fractures in the final route when the robot needs to get around obstacles. The robot uses IFS to overcome these fractures more quickly and smoothly as it approaches the final target. The proposed method also considers the type of environment the robot navigates in because each environment has different characteristics, and evidently, the friction between the robot's wheels and the environment affects its velocity, which increases the likelihood of an error.
Since IFS is typically a more sophisticated approach than the conventional fuzzy system, it is used to resolve complex and ambiguous issues. In addition to having higher accuracy than fuzzy systems, IFS is also more adaptable to changing environmental conditions. IFS is therefore used here to address the complex problem of robot path planning because there is always a chance that the environment or the conditions could change. Therefore, to adapt to these changes, we propose the use of IFS instead of fuzzy systems in this paper The rest of this paper is organized as follows. Previously proposed approaches for the mentioned problem are briefly explained in Section 2. The problem is generally described in Section 3, and the proposed solution besides the proposed flowchart is thoroughly explained in Section 4.

The simulations along with the implementation of the method in different environments are detailed and discussed in Section 5 and the conclusions and the directions for future works are mentioned in Section

In general, a variety of approaches have been used in recent years to address the robot problem. The target-reaching of mobile robots is a very old and challenging issue. The remainder of this section explains some of the methods that have been presented. Until now, a variety of approaches, including fuzzy systems, evolutionary algorithms, game theory, and neural networks, have been used to solve the explicated problem. For instance, type-1 fuzzy systems were employed for this purpose in [6] in 2019. In this study, a few arbitrary nodes are generated in the path. The Dijkstra algorithm is used to determine the shortest path after all nodes have been connected, and type 1 fuzzy systems are then used when the robot reaches obstacles.

A fuzzy PID is used in [8] and [9] to solve the issue. More specifically, it is used in [8] for collaborative robot movement and [9] for industrial robots. Numerous parameters have been used as fuzzy system inputs in a wealth of prior research. For instance, in [6], the robot's sensors calculate the angle to the target and the distance to the obstacle as inputs, and then, the wheels' velocity is achieved as the fuzzy output. A fuzzy decision function is used in [10] to solve the issue. This decision function is fuzzy and is analyzed as a matrix. The distance classification probability in [11] is calculated based on the robot's distance to the obstacle from the front, right, and left sides, using a combination of fuzzy systems and probability functions.

A type of type-1 fuzzy controller that can avoid collisions with obstacles and is appropriate for static environments is designed in [12]. Paper [13] was created based on the neuro-fuzzy system. The main challenges of this paper are the accuracy of route tracking and stable performance with heavy loads. Disorders like internal friction of the robot and external interaction forces - of the robot and environment - are taken into account to ensure the tracking function in an unknown environment. An adaptive decision-making method with Bayesian-fuzzy reinforcement learning for football-playing robots is presented in [14]. A fuzzy model is proposed in [15] for wheeled mobile robots with a visual odometer. This paper focuses on the control design for wheeled mobile robots with a visual odometer based on the Takagi-Sugeno fuzzy model.

The fuzzy system has been used for the collective movement of robots as well. For instance, in [16], a group of robots swarm and receive information from their sensors, which they then analyze using fuzzy logical rules. Nevertheless, the capacity for avoiding obstacles is also taken into account. Fuzzy is also used in concentrated systems. For instance, in [7], vision sensors have been used to calculate the distance to the target, and the pixels of the target the robot can see are the fuzzy system inputs to select the winning robot and send the information related to the target to other robots. Fuzzy method and neural networks have been combined in some previously published works. For instance, in [3], the control of the wheeled mobile robot is performed based on taking into account the effects of uncertainty via a neuro-fuzzy cognitive map, and [4] bases the control on taking into account GPS and ANFIS. In other words, it combines GPS and a neuro fuzzy network. As mentioned, evolutionary algorithms are frequently used to address the robot issue. For example, the PSO [1]algorithm is used in [1] and [2] to smooth the robot's path in multi-robot systems so that the motion of these mobile robots is controlled; or [17] uses the Bat algorithm in a multi-robot system. This algorithm has been optimized, and it can get around obstacles as well. The method discussed in that paper is used to address the issue of finding targets. In [18], the robot builds an accurate model using image processing techniques before using fuzzy systems and the genetic algorithm to navigate to the target without running into any obstacles. The combination of the genetic algorithm and the neuro-fuzzy system is also used for this purpose [19], and other techniques like the differential game have also been employed [20]. This method, which uses a differential game formula to create collision-free paths, is used in that paper for wheeled mobile robots. Finally, this approach guarantees that no robots collide as they approach their targets. Different routing algorithms are applied in each of the methods that have been presented. For example, [7] uses the A* algorithm, and [6] uses the Dijkstra algorithm. [21] describes three methods for segmenting the space and setting up nodes to find the best path. Fuzzy techniques are still frequently used to address the robot problem. For instance, in 2022, the problem of robots in [22] and [5] was solved using a combination of neuro-fuzzy networks and controllers. A neuro fuzzy network with some inputs and outputs was designed in [5], but this system does not take into account all of the influential factors. On the other hand, because neural networks are used, there are function design problems that are time-consuming, and in addition, the accuracy is lower than IFS. In recent years, robots have been used for things like diagnosis and treatment of corona disease [23] , it is hoped that intuitive fuzzy systems will be used in these methods to increase the performance of these types of robots

Therefore, the goal of this paper is to present a method that considers all the influential factors, has higher accuracy than the conventional fuzzy system, and does not require the definition of a function. Examples of such influential factors include the distance of the robot from the obstacle in all directions and the amount of friction between the surface and the robot. The nodes are connected to only their neighbors under this proposed method, which significantly reduces the number of edges so that the robot is guided as effectively and precisely as possible, and as a result, robot routing becomes less complicated. In addition, the results show that the A* algorithm appears to be better suited for robots. Given its intelligence, this algorithm can guide a robot that is unable to choose the best path to its destination. In addition, compared to earlier techniques, the mentioned method has some additional benefits for moving and overcoming obstacles. Fuzzy systems, for instance, are not very accurate because only the degree of membership is calculated, and evolutionary algorithms and neural networks take a long time because of the enormous number of repetitions they contain. Therefore, IFS is used in this paper because IFS is superior to regular fuzzy for some problems that cannot be analyzed, such as the robot problem, since by taking into account the degree of non-membership and the degree of doubt for each member besides the degree of membership, it is more accurate than type-1 and type-2 fuzzy systems and is more flexible. Furthermore, the proposed method is less time-consuming compared to evolutionary algorithms due to fewer repetitions.

Literature : section 1 describes introduction and section 2 is about is about method and section 3 describes result ,discussion and section 4 is about conclusion .

2. GENERATION OF THE DATA

The realization of this work supposes the availability of a great number of repetitions of samples responding to the same known theoretical model. In practice, as the theoretical model is unknown, we use the Monte-Carlo method based on the generation of the data by computer according to a fixed theoretical model.

2. METHOD

The robot routing problem is an old problem and several methods have been proposed to solve it in the past years. The mobile robot that is the subject of this article has two wheels on the right and left sides. This robot moves in an environment with some obstacles. Figure 1 shows an example of the environment in which the robot moves. However, this approach works in any environment with any number of obstacles.

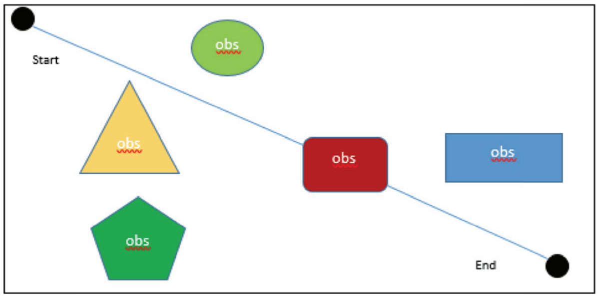

Figure 1. An example of the robot moving environment

2.1.Voronoi diagram

The area where the robot moves and reaches its goal is divided by the Voronoi diagram. At first, some points are created in random positions in the environment, which are called sites. These sites will not be located in positions occupied by obstacles. Figure 2 shows an example of this environment.

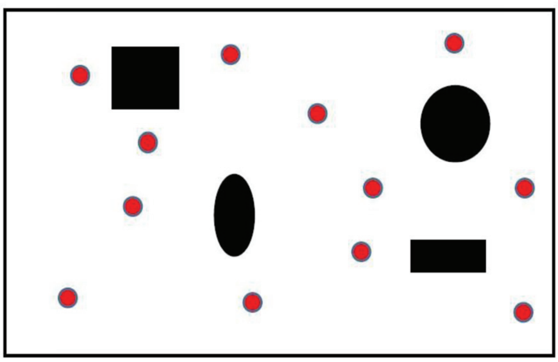

Figure 2. An example of the robot's movement environment with obstacles (denoted by black shapes) and sites (denoted by red circles)

After producing the sites, these sites are connected by lines and then the fair vertical of these lines is drawn. Therefore, an area full of sites and ridges is created. Unlike previous methods, this method only connects neighboring nodes. Therefore, there are relatively fewer edges, and thus the graph does not become crowded with edges. Therefore, this method is less complicated than the previous methods. Figure 3 shows a sample space with three locations along with its edges. As mentioned earlier, after creating the sites, two sections are drawn and a Voronoi diagram is obtained.

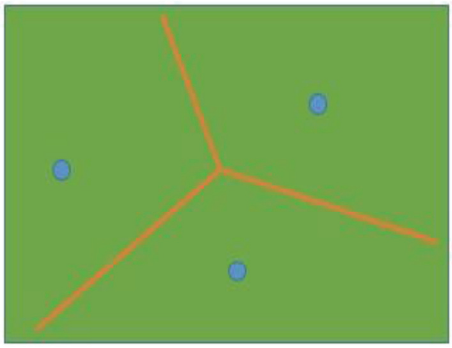

Figure 3. A moving environment for a robot with three sites and the Voronoi diagram lines

The A* algorithm is then used to find the shortest path on the graph between the starting point and the destination. After that, some fractures that are present in some parts of the path are handled by IFS, which is discussed in the next section.

2.2. Intuitionistic Fuzzy Systems

The theory of Intuitionistic Fuzzy Sets has many applications, such as logical planning, medical diagnoses, cognitive patterns, and clustering. This paper solves the problem of robot path planning using IFS with some accurate inputs, which increases the accuracy and decreases the time consumed, leading to increment in the robot's movement speed. In IFS, the degree of non-membership is υ_A, the degree of doubt is π_A, and the degree of membership is μ_A. An IFS *A*,from the reference set *X*, is defined as follows:

$$A = \{(\mathrm{X}, \mu\mathrm{A}(\mathrm{x}), \gamma\mathrm{A}(\mathrm{X}))|\mathrm{x}\epsilon\mathrm{X}\}$$

With the following conditions:

$$0 \leq \mu A(x) + \gamma A(x) \leq 1$$
$$\gamma A : X \rightarrow [0,1]$$

The degree of uncertainty in intuitive fuzzy is calculated with the following formula

$$\pi A(x) = 1 - \mu A(x) - \gamma A(x)$$
$$0 \leq \pi \mathrm{A}(\mathrm{x}) \leq 1$$

Therefore, any IFS member, *x*, has three real values namely, $\pi_A(x)$, $\mu_A(x)$ and $\gamma_A(x)$. So IFS, *A*, is a function that transforms set *x* into a subset of a unit cube. $\pi_A(x)$ is degree of doubt ,$\mu_A(x)$ is degree of membership γA is non membership. Note that a fuzzy set is a special case of IFS that maps the set x only to the set of points on a line on the unit square.

When the robot travels the path determined in the previous step, it reaches some points where the path has fractures. When the robot gets there, it can use IFS to cross that. Figure 4 denotes an example of it.

This fuzzy system has some inputs and outputs The system inputs are the distance to the obstacle from the right and the distance to the obstacle from the left, the distance to the obstacle from the opposite side, the angle of the robot's head and the coefficient of friction with the environment.The range taken into account for the first three inputs, which are all calculated in centimeters, is from 0 to 25 centimeters. Fuzzy diagrams typically classify distance into three categories: near, medium, and far. The next input to be taken into account is the robot's head angle, which is calculated in radians and is classified into three groups: negative, positive, and zero. Also, the friction degree is divided into three categories: low, medium, and much (0 to 100%). In IFS, the degree of non-membership (υ_A) and the degree of doubt (π_A) are taken into consideration in addition to degree of membership (μ_A). In this study, Sugeno intuitionistic fuzzy generator is used to produce an intuitionistic fuzzy complement.

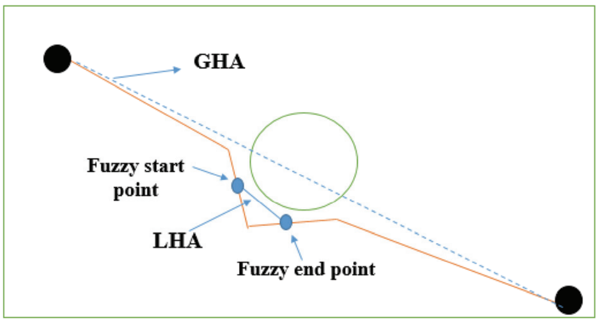

Figure 4. The robot's movement space and the start and end points of it with the presence of an obstacle and an intuitionistic fuzzy point

For an IFS, the membership degree diagram should be drawn for each of the system inputs so that the degree of non-membership is also visible. Five inputs—distance to the obstacle from the right (ROD), distance to obstacle from the left (LOD), distance to obstacle from the front (FOD), angle of the robot's head (LHA), and friction percentage (FP)—are used in the proposed method . The amount of friction coefficient makes it possible to use this method in any type of environment with any type of coating. All five inputs are of the Gaussian type. Figure 5 gives an illustration of the Gaussian degree of membership diagram. The blue and red graphs represent the degree of membership and non-membership, respectively.

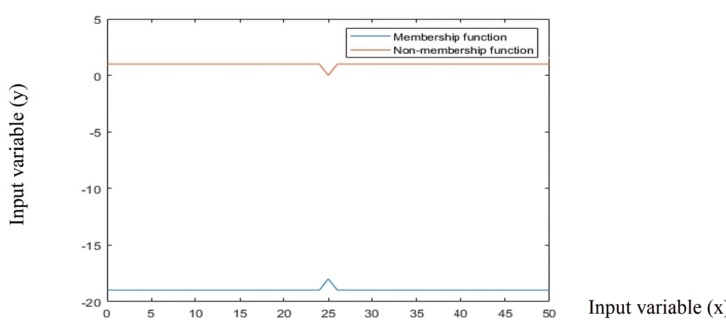

Figure 5. An example of a Gaussian fuzzy membership degree diagrams

The above IFS outputs are the speed of the right and left wheels.. These two outputs are classified into three groups in the diagram: slow, medium, and fast. Table 1 shows some of the fuzzy rules of this method related to overcoming obstacles. These rules are of the Sugeno type and are based on the system's inputs. Distances to the obstacle are classified into three groups of near, medium, and far, and the robot’s head angle is divided into three groups as well: zero, negative, and positive. In these rules, the outputs (i.e. the velocity of the robot wheels) are described as being slow, medium, and fast.

Table 1: Fuzzy rules when the robot is faced with obstacles

Fuzzy Rule action	System inputs					System outputs	
	Left Distance	front distance	right distance	LHA	FP	LWV	RWV
Obstacle avoidance	Near	Near	Near	Zero	Much	Slow	Slow
Obstacle avoidance	Near	Near	Medium	Positive	Medium	Fast	Slow
Obstacle avoidance	Near	Near	Far	Positive	Medium	Fast	Slow
Obstacle avoidance	Near	Medium	Near	Zero	Much	Slow	Slow
Obstacle avoidance	Near	Medium	Medium	Positive	Medium	Fast	Slow
Obstacle avoidance	Near	Medium	Far	Positive	Medium	Fast	Slow
Obstacle avoidance	Medium	Near	Near	Positive	Medium	Fast	Slow
Obstacle avoidance	Medium	Near	Medium	Positive	Medium	Fast	Slow
Obstacle avoidance	Medium	Near	Far	Negative	Much	Slow	Medium
Obstacle avoidance	Medium	Medium	Near	Negative	Low	Medium	Fast
Obstacle avoidance	Medium	Medium	Medium	Negative	Low	Medium	Fast
Obstacle avoidance	Medium	Medium	Far	Positive	Much	Medium	Slow
Target Seeking	Far	Near	Near	Negative	Medium	Slow	Fast
Target Seeking	Far	Near	Medium	Negative	Medium	Slow	Fast
Target Seeking	Far	Near	Far	Positive	Medium	Fast	Slow
Target Seeking	Far	Medium	Near	Negative	Much	Slow	Medium
Target Seeking	Far	Medium	Medium	Negative	Much	Slow	Medium
Target Seeking	Far	Medium	Far	Positive	Much	Medium	Slow

After this, the angle of the robot's head is determined and the robot knows which direction to move to reach the target. In general, it is possible to calculate the output of fuzzy systems and get the output for different inputs.

2.3. Algorithms

Pseudo peresented in figure 6:

Algorithm: (Pseudo code of system workflow)

1. Begin
2. Get the location of the robot and target
3. Create sites and design the Voronoi diagram
4. Find shortest path to the target
5. Start the movement
6. Calculate distance from the obstacle (the right side, the left side, the front)
7. If (haven’t reached to the fuzzy point) go to step5

Else

8. Calculate LHA (angle) and FP (friction percentage)
9. While the robot reached to end fuzzy point
10. Active intuitive fuzzy system
11. Define velocity of robot
12. Move toward HD
13. If (reach to the target) stop

Else

While reach to the target

Go to the step6

End

Figure 6. ***Pseudo code of system workflow 1***

2.4.The flowchart of the proposed approach

All the steps of the proposed method are shown as a flowchart in Figure 7:

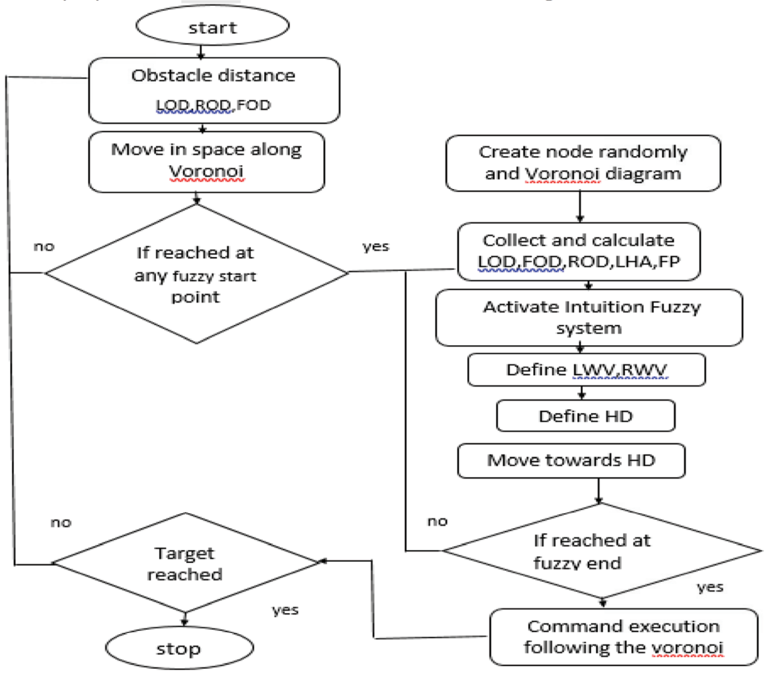

Figure 7. The flowchart of the robot's movement when it faces an obstacle

Figure 4 shows IFS in an environment with an obstacle. The start and end points of the path, as well as the robot's head angle are also illustrated. Generally, for IFS inputs, the degree of membership is illustrated, and the outputs are also calculated.. Figure 4 depicts a fuzzy system's overall structure. The presence of an obstacle causes a fracture in the middle of the path. As a result, it enters the IFS. The robot's head angle is specified along with the fuzzy start point and fuzzy end point.

3. RESULTS AND DISCUSSION

The outcomes of simulations of the proposed method are presented in this section. The experiments are performed in Matlab. Three environments with various conditions, various fuzzy points, and varying robot's head angles are used in these experiments. These environments have a common surface, so the coefficient of friction is the same for all of them, therefore, the coefficient of friction is not included in the inputs (However, if different types of environments are chosen, the friction coefficient should also be included as inputs). The left- and right-wheel velocities as well as the time consumed are calculated in these simulations. These three simulation environments, numbered from 1 to 3, are the subject of the analysis that follows.

3.1. .Simulation Environment 1

In this simulation, an environment (Figure 8) with three obstacles is considered and A* algorithm is used to obtain the shown path.

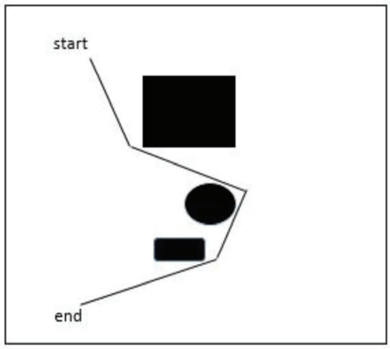

Figure 8.The first environment with three obstacles and three intuitionistic fuzzy points

Figure 9 shows the path of the robot in the mentioned environment (from top to bottom). The robot must travel through a number of nodes (or sites) in its movement environment in order to reach its destination.

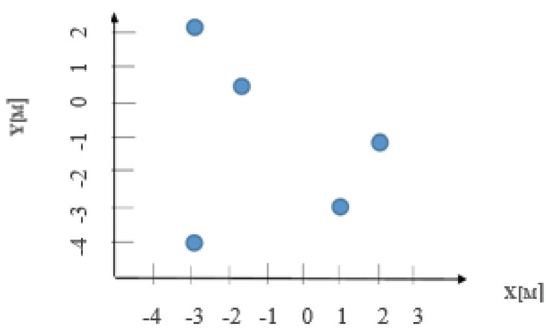

Figure 9. The set of robot movement points in the first simulation environment.

As seen in Table 2, the robot has successfully reached the target in the simulation environment.

Table2: Simulation results of the first environment

	distance to the obstacle from the left	distance to the obstacle from the front	distance to the obstacle the right	*angle*	Result
First fuzzy point	11.1	12	16.8	-19.6	Target reached
Second fuzzy point	15.3	10	7.2	26	Target reached
Third fuzzy point	13	9	8	31	Target reached

. 3.2. Simulation Environment 2

In this simulation, the environment (Figure 10) is considered with three obstacles, and then the best route is selected based on the aforementioned routing algorithm. The start and end points of the route are also displayed. Figure 11 shows the path of the robot in the mentioned environment (from top to bottom). The robot must travel through a number of nodes (or sites) in its movement environment in order to reach its destination.

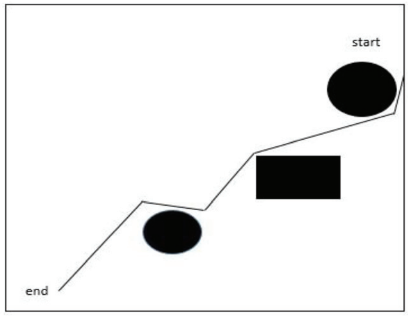

Figure 10. The second simulation environment with three obstacles and three intuitionistic fuzzy points.

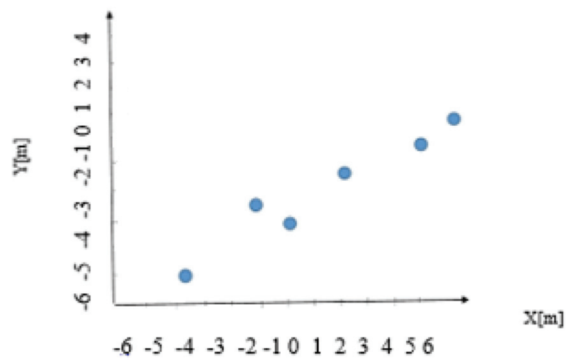

Figure 11. The set of robot movement points in the second environment

The IFS is calculated at each of the three fuzzy points in Table 3. As you can see, the robot has successfully reached the target in the simulation environment, which is an important point in the robot problem.

Table 3: Simulation results of the second environment

	distance to the obstacle from the left	distance to the obstacle from the front	distance to the obstacle from the right	*angle*	Result
First fuzzy point	17.7	19.8	9.38	9.82	Target reached
second fuzzy point	8.9	16	13.6	45	Target reached
third fuzzy point	7.4	14	17	36	Target reached
fourth fuzzy point	9.9	17	15.4	33	Target reached

3.3. Simulation Environment 3

In this simulation, an environment (Figure 12) with three obstacles is considered. Then the said routing algorithm selects the following route. The figure clearly shows that there are points where IFS should be used.

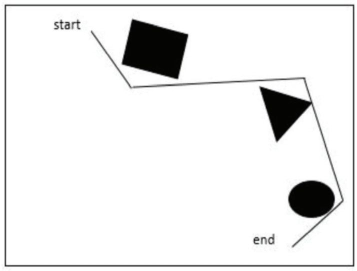

Figure 12.The third simulation environment with three obstacles and three intuitionistic fuzzy points

Figure 13 shows the points that the robot travels in this environment to reach the target. This path and this set of points are determined based on the routing algorithm.

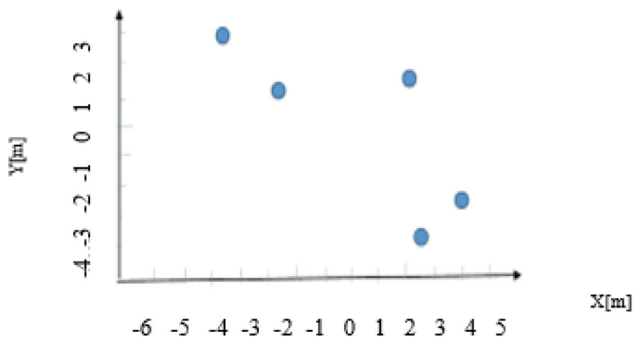

Figure 13.. The points of robot movement in the third environment

As seen in Table 4, the robot has successfully reached the target in every simulation environment, which is a crucial factor in the robot routing problem.

Table4: Simulation results of the third environment

	distance to the obstacle from the left	distance to the obstacle from the front	distance to the obstacle the right	*angle*	Result
First fuzzy point	9	19	14	33	Target reached
Second fuzzy point	15.54	14	8	25	Target reached
Third fuzzy point	15.44	16	7.43	41	Target reached

3.2. comparison of the proposed method's and others

3.2.1.. comparison of the proposed method's and the fuzzy system method's path lengths

The robot's path length is calculated in Table 5 for each of the three environments. In these calculations, the path length is based on the number of pixels. These experiments demonstrate that the path length of the proposed method is shorter than the typical fuzzy method.

Table5: A comparison of the proposed method's and the fuzzy system method's path lengths

Simulation environment	Path length in normal fuzzy method (in pixels)	Path length in the proposed method (in pixels)
Environment 1	588	574
Environment 2	1010	983
Environment 3	1210	1011

3.2.2.A* algorithm vs. Dijkstra algorithm

As we know, in many past articles, the Dijkstra algorithm was used for robot routing But we used Star algorithm in this method. Table 6 compares the amount of time spent applying the A* as the algorithm used in the proposed method and Dijkstra algorithms . The average time of 10 executions of A star and Dijkstra algorithms shows that the time of A star algorithm is less.

Table6: Comparison of the running time of the A* and Dijkstra algorithms in the mentioned problem (times are in seconds)

Run	Dijkstra	A*
1	0.080734	0.042031
2	0.083189	0.045585
3	0.078118	0.051560
4	0.126217	0.041585
5	0.103139	0.045985
6	0.075946	0.053995
7	0.088886	0.062212
8	0.089028	0.073031
9	0.089316	0.050403
10	0.100969	0.040392
Average	0.091554	0.050678

As denoted in Table 6, the average execution time of the A* and Dijkstra algorithms in the mentioned problem are 50.6779 milliseconds and 91.5542 milliseconds, respectively which demonstrates the lower running time of the A* algorithm.

Below, the presented method is compared with the methods presented in the past. The results prove the high accuracy and appropriate speed of the presented method.

3.2.3.Comparison with neuro-fuzzy approach

Since neuro-fuzzy approach has recently been used to solve the robot problem [22], Tables 7, 8, and 9 provide a comparison of the robot's velocity (measured in meters per second) and time (measured in seconds) between the proposed method and the neuro-fuzzy approach [22]. Both approaches take into account the robot's head angle and distance to the obstacle (measured in square centimeters). The same environment is used for all of the simulations. The outcomes demonstrate a rise in robot velocity and a fall in time spent on the proposed method.

Table7: The first comparison of velocity and time for intuitionistic fuzzy system and neuro-fuzzy network [22]

Method	Distance From Right	Distance From left	Distance From front	angle	Velocity	Time
IFS based proposed method	32	25	15	60.9	0.8	15
Neuro-fuzzy method [22]	32	25	15	60.9	0.2	29

In Table 7, the obstacle's distance is 32 cm from the right, 25 cm from the left, and 15 cm from the front, and the robot's head angle is 60.9 degrees. Then, both methods are used to calculate the velocity and time, which demonstrates an improvement in both time and velocity in the proposed method.

Table8: The second comparison of velocity and time for intuitionistic fuzzy system and neuro fuzzy network [22]

Method	Distance From Right	Distance From left	Distance From front	angle	Velocity	Time
IFS based proposed method	28	31	64	71	0.99	19
Neuro-fuzzy method [22]	28	31	64	71	0.4	37

In Table 8, the distances to the obstacle from the right, left, and front are 28 cm, 31 cm, and 64 cm, respectively, and the angle of the robot's head is 71 degrees. The velocity and time are then calculated using both methods, showing improvement in both time and velocity in the proposed method.

*Table 9:*The third comparison of velocity and time for intuitionistic fuzzy system and neuro-fuzzy network [22]

Method	Distance From Right	Distance From left	Distance From front	angle	Velocity	Time
IFS based proposed method	20	30	45	100	1.9	87
Neuro-fuzzy method [22]	20	30	45	100	1.01	108

In the experiments of Table 9, the distances to the obstacle from the right, left, and front are 20, 30, and 45, respectively, and the angle of the robot's head is 100 degrees. Then, using both approaches, the velocity and time are calculated, demonstrating the velocity and time improvements in the suggested method.

3.3.Accuracy and precision

In Table 10, the number of times the robot has reached the target (out of fifty executions) has been compared. The results show that the presented method has a better result due to its high accuracy and considering all factors affecting the movement of the robot, and the number of times it has reached the target is more than the previous methods.

Table10: The number of times the robot has reached the target in 50 executions

proposed method	46
controller-fuzzy method	31
Neuro-fuzzy method	38
Classic fuzzy method(Regardless of the type of coating and friction)	41

4. DISCUSSION AND CONCLUSION

This study introduces a new robot routing technique. Intuitive fuzzy systems are more accurate than fuzzy systems and also spend less time compared to evolutionary algorithms. other hand, in none of the previous papers, the type of cover of the environment in which the robot moves has not been given importance, while the type of cover and in The result of the friction created between the robot wheels and the environment is effective in the speed of the robot. The movement speed of the robot and the results of the simulations show the high accuracy of the method.
For future work, it is worth evaluating the proposed approach in other scenarios and for robots with more sensors. Also, using machine learning approaches to select the most efficient rules is another direction for future research

ACKNOWLEDGEMENTS

Thanks to All Tri Play magazine and all those who helped us and authors.

5. REFERENCES

Vijay, M. and D. Jena, *PSO based neuro fuzzy sliding mode control for a robot manipulator.* Journal of Electrical Systems and Information Technology, 2017. **4**(1): p. 243-256.

Ajeil, F.H., et al., *Multi-objective path planning of an autonomous mobile robot using hybrid PSO-MFB optimization algorithm.* Applied Soft Computing, 2020. **89**: p. 106076.

Amirkhani, A., et al., *Controlling wheeled mobile robot considering the effects of uncertainty with neuro-fuzzy cognitive map.* ISA transactions, 2020. **100**: p. 454-468.

Gharajeh, M.S. and H.B. Jond, *Hybrid global positioning system-adaptive neuro-fuzzy inference system based autonomous mobile robot navigation.* Robotics and Autonomous Systems, 2020. **134**: p. 103669.

Gharajeh, M.S. and H.B. Jond, *An intelligent approach for autonomous mobile robots path planning based on adaptive neuro-fuzzy inference system.* Ain Shams Engineering Journal, 2022. **13**(1): p. 101491.

Mohanta, J.C. and A. Keshari, *A knowledge based fuzzy-probabilistic roadmap method for mobile robot navigation.* Applied Soft Computing, 2019. **79**: p. 391-409.

Nascimento, T.P., et al., *Intelligent state changing applied to multi-robot systems.* Robotics and Autonomous Systems, 2013. **61**(2): p. 115-124.

Kashyap, A.K. and D.R. Parhi, *Particle Swarm Optimization aided PID gait controller design for a humanoid robot.* ISA transactions, 2021. **114**: p. 306-330.

Vasu, S., *Fuzzy PID based adaptive control on industrial robot system.* Materials Today: Proceedings, 2018. **5**(5): p. 13055-13060.

Patle, B., et al., *The optimized path for a mobile robot using fuzzy decision function.* Materials Today: Proceedings, 2019. **18**: p. 3575-3581.

Patle, B., et al., *Application of probability to enhance the performance of fuzzy based mobile robot navigation.* Applied Soft Computing, 2019. **75**: p. 265-283.

Singh, N.H. and K. Thongam, *Mobile robot navigation using fuzzy logic in static environments.* Procedia Computer Science, 2018. **125**: p. 11-17.

Li, J., et al., *Neural fuzzy approximation enhanced autonomous tracking control of the wheel-legged robot under uncertain physical interaction.* Neurocomputing, 2020. **410**: p. 342-353.

Shi, H., et al., *An adaptive decision-making method with fuzzy Bayesian reinforcement learning for robot soccer.* Information Sciences, 2018. **436**: p. 268-281.

Sun, C.-H., et al., *Sequentially switched fuzzy-model-based control for wheeled mobile robot with visual odometry.* Applied Mathematical Modelling, 2017. **47**: p. 765-776.

Din, A., et al., *Behavior-based swarm robotic search and rescue using fuzzy controller.* Computers & Electrical Engineering, 2018. **70**: p. 53-65.

Tang, H., et al., *A multirobot target searching method based on bat algorithm in unknown environments.* Expert Systems with Applications, 2020. **141**: p. 112945.

Bakdi, A., et al., *Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control.* Robotics and Autonomous Systems, 2017. **89**: p. 95-109.

Refoufi, S. and K. Benmahammed, *Control of a manipulator robot by neuro-fuzzy subsets form approach control optimized by the genetic algorithms.* ISA transactions, 2018. **77**: p. 133-145.

Mylvaganam, T. and M. Sassano, *Autonomous collision avoidance for wheeled mobile robots using a differential game approach.* European Journal of Control, 2018. **40**: p. 5361.

Jogeshwar, B.K. and K. Lochan, *Algorithms for Path Planning on Mobile Robots.* IFAC-PapersOnLine, 2022. **55**(1): p. 94-100.

Haider, M.H., et al., *Robust mobile robot navigation in cluttered environments based on hybrid adaptive neuro-fuzzy inference and sensor fusion.* Journal of King Saud UniversityComputer and Information Sciences, 2022. **34**(10): p. 9060-9070

Arjon Turnip,D.Novita,Darmawan Hidayat,Endra Joelianto,Dwi Esti Kusumandari, Ubaidillah, N.Nasser,Nguyen, Le Hoa, International Journal of Artificial Intelligence.2024