Robotica: (2021) volume 39, pp. 2033–2060. © The Author(s), 2021. Published by Cambridge University Press.
doi:10.1017/S0263574721000114

Hybrid IWD-GA: An Approach for Path Optimization and Control of Multiple Mobile Robot in Obscure Static and Dynamic Environments

Saroj Kumar†*, Dayal Ramakrushna Parhi†, Krishna Kant Pandey‡, Manoj Kumar Muni¶

†*Robotics Laboratory, Department of Mechanical Engineering, National Institute of Technology, Rourkela, Odisha, India 769008*
‡*Department of Mechanical Engineering, G.H. Raisoni Institute of Engineering and Technology, Pune, Maharashtra, India 412207*
¶*Department of Mechanical Engineering, Indira Gandhi Institute of Technology, Sarang, Odisha, India 759146*

(Accepted January 16, 2021. First published online: February 24, 2021)

SUMMARY
In this article, hybridization of IWD (intelligent water drop) and GA (genetic algorithm) technique is developed and executed in order to obtain global optimal path by replacing local optimal points. Sensors of mobile robots are used for mapping and detecting the environment and obstacles present. The developed technique is tested in MATLAB simulation platform, and experimental analysis is performed in real-time environments to observe the effectiveness of IWD-GA technique. Furthermore, statistical analysis of obtained results is also performed for testing their linearity and normality. A significant improvement of about 13.14% in terms of path length is reported when the proposed technique is tested against other existing techniques.

KEYWORDS: Path optimization; Navigation; IWD-GA; Mobile robot; Artificial intelligence.

1. Introduction

Since the development of mobile robots, the navigational control and optimal path planning of these mobile robots has remained a challenging task. However, robots have become an integral part of human life. Mobile robots are quite relatable with household activities, industries, hospitals, etc. To optimize the challenges involved in path planning, many researchers have worked using different approaches. Still, it is necessary to find best path and smooth navigation with more robust techniques. A robust navigational controller requires careful design considerations. The design should be accomplished with a variety of sensors in order to provide all the required information to the controller, so that the function of the mobile robot may be collision-free and smooth. Intelligent water drops (IWDs) and genetic algorithm (GA) are inspired by nature. The IWD technique is derived from the phenomena happening in the riverbed.[1] Here the natural phenomenon consists of the reactions and actions of water drops in the rivers. This algorithm can be used as combinatorial optimization. In this article, a newly developed approach is modified and introduced to control the navigation of the robot. Pour et al.[1] used a grid-based concept of IWD for path optimization of mobile robots. Using IWDs, path optimization has been done in two layers. Duan et al.[2] have used a refined IWD algorithm for the

* Corresponding author. E-mail: saroj4sks@gmail.com

path optimization of an unmanned combat aerial vehicle (UCAV). Using IWD and K-method, UCAV can find a smooth trajectory. Shah[3] has proposed an IWD algorithm with continuous optimization (IWD-CO). This analysis shows the successful simultaneous optimization of six different problems. Alijla et al.[4,5] have used an hybrid MRMC (master river multiple creak)-IWD model. This is a type of modified IWD algorithm. Kaur et al.[6] have used an hybrid approach of cuckoo search and IWD algorithm to identify the natural terrain feature in a specific area of land. Hosseini[7,8] has used the IWD algorithm to extract a solution for an n-queen puzzle and TSP problem. Salmanpour et al.[9,10] used a generalized IWD algorithm with fuzzy interface for path tracking problem of robots. Bansal et al.[11] have used an IWD algorithm to decode the graph-based TSP problem. Rao et al.[12] emphasized IWD with differential evolution algorithm for multi-robot navigational control in a hazy environment. Xue et al.[13] used the basics of ACO technique on the TSP problem to find solutions. Chen et al.[14] have explained the ACO algorithm for multi-robot patrolling solutions. Ismkhan et al.[15] have proposed ACO with pheromone information feedback approach for finding the solution. Priyambado et al.[16] used an ACO algorithm to plan the track of a robot in moving obstacles environment. Yang et al.[17] have used a multilayer ACO algorithm for autonomous robot navigation. Parhi et al.[18] have presented hybridized invasive weed optimization (IWO) technique and ANFIS (adaptive neuro-fuzzy interface system) for the navigation of robots. Patle et al.[19] have proposed a probability fuzzy logic (PFL) path-planning algorithm to navigate the robot. Pandey et al.[20] explained the design of a mobile robot for path planning and control with the use of fuzzy logic controller. Parhi et al.[21] have used a DAYANI arc contour intelligent method for obstacle-free path planning and navigation of two-wheel self-balanced robot in a cluttered scenario. Pandey et al.[22] have used an ANFIS controller for the navigation of a mobile robot and obstacle avoidance. Ultrasonic sensors, infrared sensors, etc. are connected for the detection of obstacles. Tan et al.[23] have used a design of mobile robot control architecture for the movement of robots in unknown environments. Li et al.[24] used a self-adaptive learning particle swarm optimization (SLPSO) technique for track planning of mobile robots in a messy environment. Patle et al.[25] have presented GA for the movement of mobile robots. Kumar et al.[26] used an hybridized approach for the navigational control of humanoid robots in a cluttered environment. Nazarahari et al.[27] have used artificial potential field (APF) and enhanced GA to get the optimal path for a mobile robot. Bakdi et al.[28] have used an hybrid GA and adaptive fuzzy logic control to plan the collision-free path using Kinect camera. Lamini et al.[29] have used GA for the movement of mobile robots. Liang and Lee[30] have used an efficient artificial bee colony (ABC) algorithm for navigational control of multi-robots. Pandey and Parhi[31] have used an hybridized fuzzy wind-driven optimization approach for path planning of mobile robots in static and dynamic obstacles environment. Qu et al.[32] have used an improved GA for multi-robot track planning. Turcer and Yildirim[33] have used an improved GA for dynamic track planning for a mobile robot. Mandava et al.[34] have presented an hybrid algorithm of potential field method and PSO for the shortest path planning of mobile robots in static and dynamic environments. Mohanta and Keshari et al.[35] have proposed an hybrid fuzzy model for the navigation of mobile robots in an unknown environment. Rath et al.[36,37] have presented path optimization techniques for an humanoid robot in an unknown environment. Pandey and Parhi[38] have made a behavior-based neural network for the navigation of mobile robots. Panahandeh et al.[39] have proposed an auto gain update neural network for the navigation of mobile robots. Li et al.[40] have proposed an hybrid algorithm of numerical integration and iterative method for the navigation of mobile robots. Chen et al.[41] have proposed an hybrid algorithm of artificial potential field and ant colony optimization for path planning of mobile robots. Ajeil et al.[42] have proposed an hybrid optimization algorithm of PSO-MFB (particle swarm optimization–modified frequency bat algorithm) for multiobjective path planning of mobile robots. Holland et al.[43] have explained the analysis and application of biological control with artificial intelligence technique. Goldberg et al.[44] have proposed genetic and evolutionary algorithm for the navigation of mobile robots. Mohnata and Keshri[45] have proposed an hybrid fuzzy-probabilistic model for path planning of mobile robots. Low et al.[46] have proposed optimal path planning of mobile robots using improved Q-learning. Ni et al.[47] have proposed an improved shuffled frog leaping algorithm for motion control of mobile robots. Kumar et al.[48] have proposed an hybrid SCA-ACO algorithm for path optimization of mobile robots. Rosas et al.[49] have used an hybrid algorithm for path planning of mobile robots. Montiel et al.[50,51] have proposed a bacterial potential field and ant colony approach in two different scenarios of path planning of mobile robots. Gracia et al.[52] have presented an hybrid approach for autonomous navigation of mobile robots. Castillo et al.[53] have presented path optimization for mobile robots using

https://doi.org/10.1017/S0263574721000114 Published online by Cambridge University Press

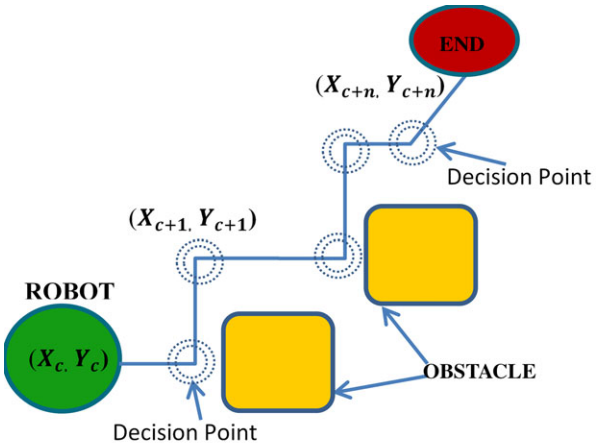

Fig. 1. Path and their turning points.

GA. Pandey et al.[54,55] have proposed a V-REP-based analysis of navigational control for mobile robots using PSO-tuned FNN and ANFIS controller in two different scenarios. Kumar et al.[56] have poposed a hybridized technique for navigational control over multiple mobile robots. Muni et al.[57] have explained prim's algorithm for path optimization of humanoid robots.

A deep survey of existing research indicates that navigational analysis and path planning are tedious work in case of a mobile robot. In the IWD technique, few research articles are available. However, in a limited way the research on GA is available, but the hybridized IWD-GA technique for mobile robot movement and its control is not done till date. Apart from these, GA is always considered to be smart due to its selection process of chromosomes and their problem-solving ability, whereas IWD is one of the best optimization techniques in the field of artificial intelligence due to its path searching capability and decision-making process. While adding IWD and GA, its efficiency will increase as the tabulated results might reflect. These facts are the motivation for implementing the proposed approach in mobile robots. The other sections are structured as follows:

- Section 2: Locomotion of the mobile robot
- Section 3: Proposed IWD-GA technique
- Section 4: Navigational analysis using IWD-GA technique
- Section 5: Comparative analysis
- Section 6: Conclusions

2. Locomotion of Mobile Robot

The stand of a mobile robot in the experimental platform is defined as X position, Y position and orientation angle (θ); in short, the stand is denoted by (X_c, Y_c, θ). 'c' is the symbol of the current point. During locomotion, a path is generated that may have many turning points as depicted in Fig. 1.

To calculate the length of the path travelled by the robot, Equation 1 is used, where $'c'$ indicates the current position and $c+1$ indicates the next stand of the robot.

$$L(\text{path}) = \sum_{i=0}^{n} \sqrt{(X_{c+1} - X_c)^2 + (Y_{c+1} - Y_c)^2} \tag{1}$$

Equation 2 is used for the calculation of the orientation of the robot, where G_y and G_x are the coordinates of the target point, and Y_{c+1} and X_{c+1} are the next stand of the robot and can be measured using equations 3 and 4. The orientation of robot for the movement and their axis position is shown in Figure 2.

$$\theta = tan^{-1}\left[\frac{G_y - Y_{c+1}}{G_x - X_{c+1}}\right] \tag{2}$$

$$X_{c+1} = X_c + cos\theta \tag{3}$$

$$Y_{c+1} = Y_c + sin\theta \tag{4}$$

https://doi.org/10.1017/S0263574721000114 Published online by Cambridge University Press

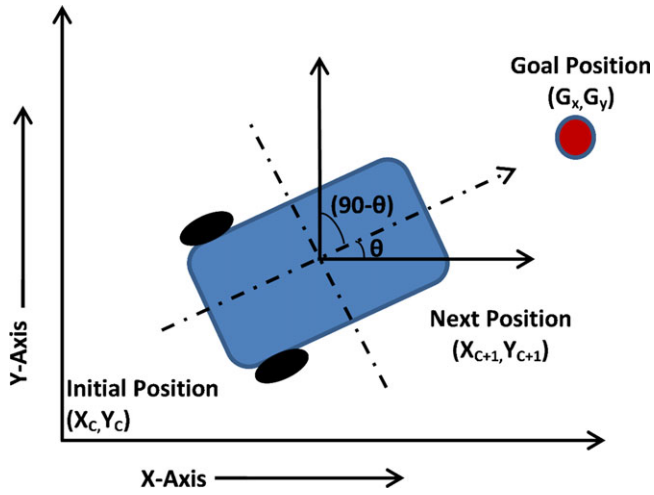

Fig. 2. Axis position and movement of the robot.

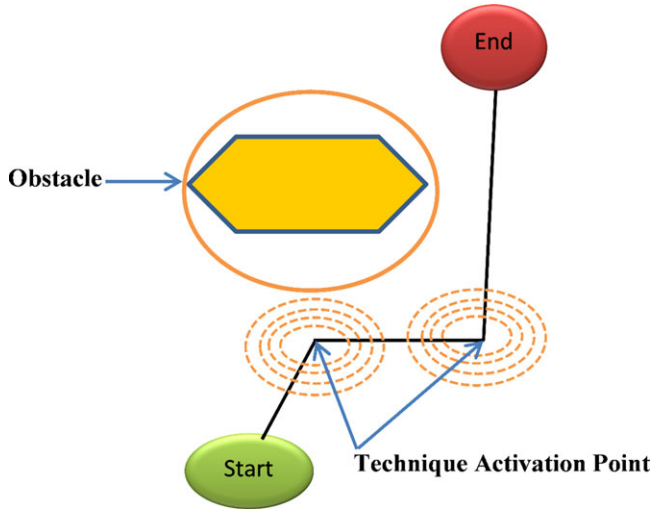

Fig. 3. Path and technique activation points.

The turning point is also known as the decision point. At the decision point, the technique activates and optimizes the path for the next movement of the robot. The activation scenario is depicted in Fig. 3.

After the decision point, it is important to describe the obstacle avoidance strategy. According to the position, the robot calculates the heading angle and moves forward to the target with avoidance of the obstacle. This strategy encourages the robot to move, but the imposed technique optimizes the path to move and the robot follows the instructions. A pictorial representation of the obstacle avoidance strategy is shown in Fig. 4. G1, G2 and G3 are the goal points where the robot reaches from the initial point. The robot is shown in green circle and obstacle in yellow box. Whenever a robot finds the obstacle within the sensor range, it is encouraged to move with an angle, which instruction is given by the optimization technique. After optimization, it is required to move forward with an optimized path.

3. Proposed Hybrid IWD-GA Technique

3.1. Architecture of IWD Technique

The IWD algorithm is explained as a population-based intelligence technique. It is encouraged by the dynamics of water drops in the river.[1] Each drop of water in the river selects a forward path; paths that have less soil and obstacles are often chosen as the forward path to achieve the goal. However,

https://doi.org/10.1017/S0263574721000114 Published online by Cambridge University Press

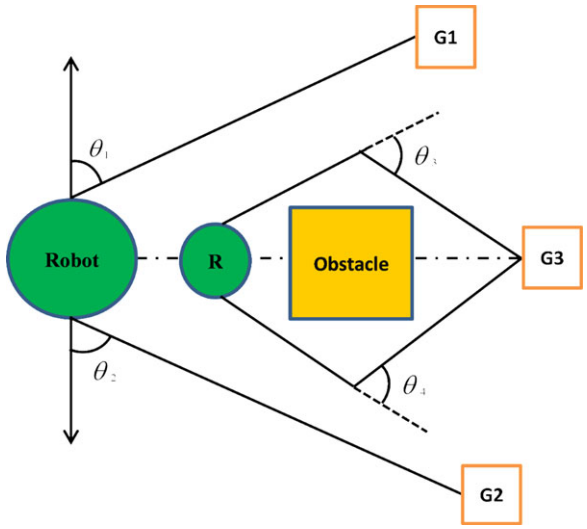

Fig. 4. Obstacle avoidance strategy.

the water drop changes its speed and direction as per the environmental conditions. The speed and direction changing behavior is similar to that of an intelligent robot. IWDs almost follow the behavior of ant colony technique; whereas the ant colony searches the path by pheromone update, the IWD finds the path by soil update in the river bed.

To travel from one point to another, each water drop takes finite steps. The velocity of water drops either increases or decreases along the path, and it depends upon the soil on the path. If the path has much amount of soil, then the velocity decreases, and if the path has less soil, then the velocity increases. However, the drops either remove or gain some amount of soil during the journey. The removal of soil from the path also depends upon the velocity of IWDs. Soil removal from the path is more when the velocity is low and vice versa. The time taken by the IWDs to travel from one point to another also depends upon the amount of soil on the path. Thus, it is clear that soil and velocity are inversely proportional to each other. The law of motion is used to calculate the time taken by the IWDs. Every node in this algorithm is directly or indirectly connected. The IWD chooses the next stop point (edge) which is connected to the next stop point. The journey of an IWD starts from the first node and finishes on the last node. In this paper, node representation is taken as ′i′ and i+1. Further, the change in velocity and amount of soil are updated for the betterment of next node or stop point search. The details of velocity and soil update are stated in the next section.

3.2. Soil and Velocity Update

In this section, the velocity and soil update method is shown in mathematical form with their static and dynamic parameters. While the water moves on a river bed, the velocity of water drops changes with change in the amount of soil. The velocity of a water drop is inversely proportional to the amount of soil.

X_a, Y_a, Z_a are the static parameters for the velocity of IWD. It is used to establish the non-linear relationship between the velocity of drops and the soil update on the river bed. The relationship between soil (i, j) and velocity of IWD is defined as

$$\Delta\text{Velocity}(IWD) \propto (\text{soil}(i, j))^{-1} \tag{5}$$

At time $'t'$, the change in velocity shows the accumulation of soil in the water drop:

$$\Delta\text{Velocity}_{WD}(t) = (X_a) \times (\text{soil}(i, j))^{-1} \tag{6}$$

During the movement of drops, the sub-paths are generated and the soil of the sub-path is concentrated on the water drops.

$$\Delta\text{Velocity}_{\boldsymbol{WD}}(\boldsymbol{t}+\mathbf{1}) = \frac{\boldsymbol{Z_a}}{(\boldsymbol{X_a} + \boldsymbol{Y_a} \times \boldsymbol{Soil}(\boldsymbol{i}+\mathbf{1}, \boldsymbol{j}+\mathbf{1}))} \tag{7}$$

https://doi.org/10.1017/S0263574721000114 Published online by Cambridge University Press

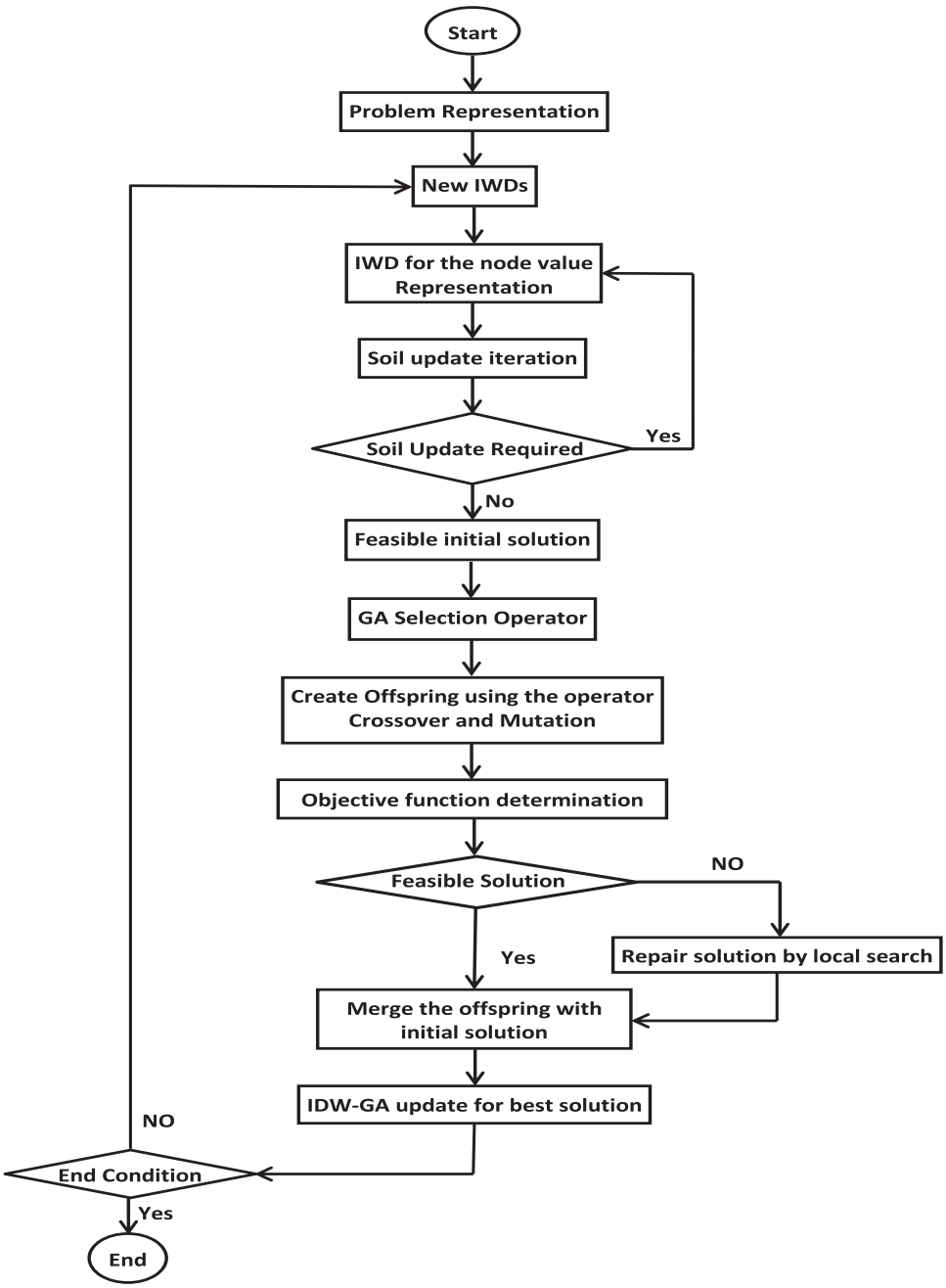

Fig. 5. Flowchart of IWD-GA algorithm.

Each drop of water moves from one node to another, which connects the soil update of each drop. The updates are expressed as:

$$Soil(i,\ j) = (1 - s_n) \cdot soil(i,\ j) - s_n \cdot \Delta soil(i,\ j) \tag{8}$$

$$Soil(\boldsymbol{i}) = soil^{\mathrm{I}}(\boldsymbol{i},\boldsymbol{j}) + \Delta soil(\boldsymbol{i},\boldsymbol{j}) \tag{9}$$

where $'s_n'$ is a constant which is selected from the soil deposited on the IWD. The range of constant is (0, 1). $Soil^{I}()$ denotes the accumulated soil in the drops, and $\triangle Soil()$ denotes the change in soil amount during motion. The soil update is the sum of the initial and the change in the initial amount of soil. Further, the velocity of water drops on the sub-path and path is updated, and the update is expressed as:

$$\Delta \mathrm{Velocity}^{\boldsymbol{n}}_{\boldsymbol{WD}}(\boldsymbol{i},\boldsymbol{j}) = soil(\boldsymbol{i},\boldsymbol{j})^{\boldsymbol{n}} + \Delta \mathrm{Soil}(\mathrm{dynamic}\boldsymbol{P}^{\boldsymbol{n}} - \mathrm{static}\boldsymbol{P}^{\boldsymbol{n}}) \tag{10}$$

https://doi.org/10.1017/S0263574721000114 Published online by Cambridge University Press

Processor	Motorola 68331, 25MHz
RAM	512 kb
Motion	2 DC Brushed servo motors with encoders
Sensors	8 Infrared proximity
Sensors	30 mm (height from ground)
Communication	Standard serial port, upto 115 kb/s
Weight	80g
Height	30mm
Diameter	70 mm
Payload	Max 250g
Speed	Maximum 500 mm/s
	Minimum 20 mm/s

Fig. 6. Description of Khepera-II mobile robot.

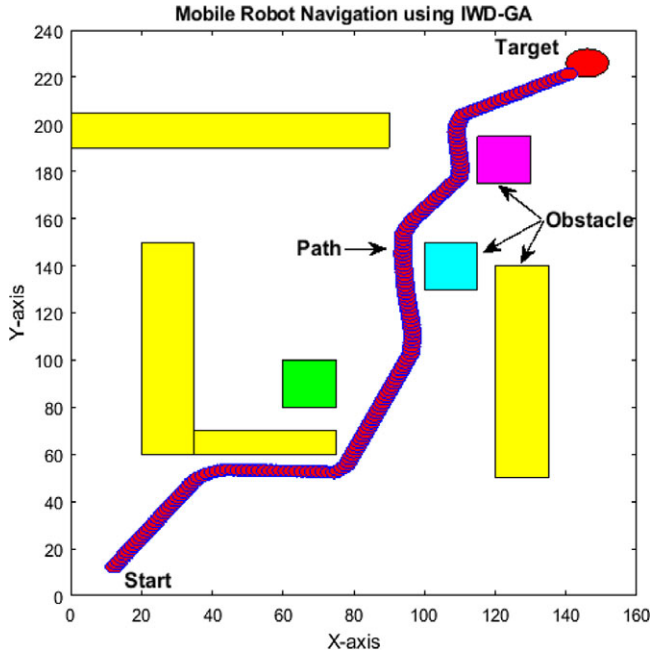

Fig. 7. Simulation analysis of the navigation of the mobile robot in scene 1.

where Z_a=1, X_a= 0.02 and Y_a = 1. These parameters are used to update the velocity and to avoid the division by zero. Equation 10 represents the final change in velocity of IWDs with the consideration of static and dynamic parameters. Here, the obstacles in the path and sub-path are considered as static parameters, while the velocity and other movable particles are considered as dynamic parameters. However, the changes due to dynamic and static parameters are minor, but it gives a better update in soil and velocity.

3.3. Node representation (next stop point)

The primary objective of the IWD is to choose the next stop point. The stop point is chosen on the basis of the concentration of soil. Less soiled paths are often preferred. Let an intelligent drop at node 'i' selects another stop point $e_{i.i+1}$ (k) to call on the next node $i+1$. The probability of the selection of node is expressed as:

$$P^{\mathrm{NS}}(e_{i,i+1}(k)) = \frac{\text{Node soil update in local search}}{\text{Soil update in global search}} \tag{11}$$

where node soil update in local search is defined as $f(soil(e_{i,i+1}(k)))$.

Soil update in global search is formulated as $\sum_{l=0}^{n} f(soil(e_{i,j}(l)))$, $j = i + 1$

https://doi.org/10.1017/S0263574721000114 Published online by Cambridge University Press

Table I. Comparison of path lengths in simulation and corresponding experiments in scene 1

	Path length (cm)		
No. of runs	**Simulation**	**Experiment**	**% Deviation**
1	281.91	299.78	5.96
2	279.96	292.51	4.29
3	289.52	301.22	3.88
4	**278.26**	**290.23**	4.12
5	282.63	302.87	6.68
6	280.55	295.12	4.94
7	283.74	297.71	4.69
8	280.87	295.73	5.02
9	286.97	297.09	3.41
10	284.12	294.17	3.42
11	283.23	297.36	4.75
12	284.41	300.21	5.26
13	281.26	294.76	4.58
14	278.72	290.84	4.17
15	278.93	288.61	3.35
16	280.23	291.23	3.78
17	282.56	296.74	4.78
18	278.64	291.23	4.32
19	286.92	301.87	4.95
20	286.51	299.56	4.36
21	284.42	297.63	4.44
22	279.57	290.89	3.89
23	283.48	298.01	4.88
24	281.57	296.36	4.99
25	279.49	290.72	3.86
26	278.93	291.63	4.35
27	278.76	291.79	4.47
28	279.61	293.67	4.79
29	280.32	290.91	3.64
30	279.09	291.74	4.34
Average	**281.84**	**295.07**	**4.48**

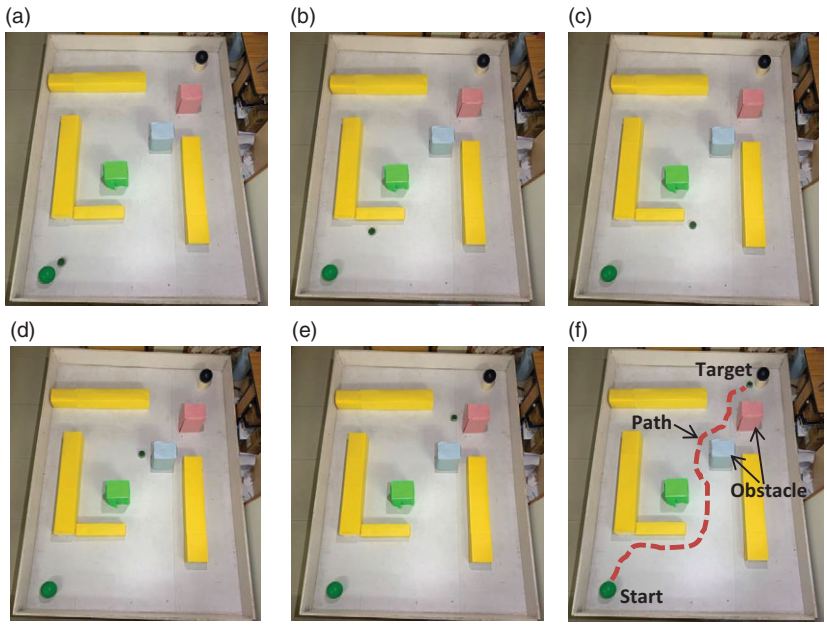

Fig. 8. Experimental analysis of the navigation of the mobile robot in scene 1.

https://doi.org/10.1017/S0263574721000114 Published online by Cambridge University Press

Table II. Comparison of execution time in simulation and corresponding experiments in scene 1

	Execution time (s)		
No. of runs	**Simulation**	**Experiment**	**% Deviation**
1	19.44	20.76	6.36
2	20.11	21.36	5.85
3	19.62	20.99	6.53
4	**19.32**	**20.52**	5.85
5	19.58	21.14	7.38
6	19.77	20.57	3.89
7	19.68	20.54	4.19
8	19.87	20.63	3.68
9	19.73	20.71	4.73
10	19.68	21.08	6.64
11	19.62	20.62	4.85
12	19.56	20.59	5.00
13	19.71	20.56	4.13
14	19.42	20.80	6.63
15	19.39	20.61	5.92
16	19.46	20.51	5.12
17	19.63	20.53	4.38
18	19.40	20.53	5.50
19	19.70	20.57	4.23
20	19.61	20.60	4.81
21	19.52	20.67	5.56
22	19.49	20.58	5.30
23	19.41	20.53	5.46
24	19.56	20.57	4.91
25	19.41	20.56	5.59
26	19.38	20.54	5.65
27	19.46	20.56	5.35
28	19.39	20.58	5.78
29	19.52	20.61	5.29
30	19.41	20.57	5.64
Average	**19.56**	**20.67**	**5.34**

where

$$f(soil\{e_{i,i+1}(k)\}) = (\varepsilon + g(soil\{e_{i,j}(k)\}))^{-1} \tag{12}$$

$$g\left(soil\left(e_{i,j}\left(k\right)\right)\right) = \begin{cases} soil(e_{i,j}(k)) if min(\ soil(e_{i,j}(l)) = 0 \\ soil(e_{i,j}(k)) - min(\ soil(e_{i,j}(l)), else \end{cases} \tag{13}$$

where P^{NS} is the probability function for the next stop point search, $'\varepsilon'$ is the small constant quantity which is used to avert division with zero. g (soil) is the function of soil accumulation on IWD. f (soil) is the function of stop point search coordinate.

3.4. Repairing of infeasible solution

To optimize the path, many considerations have been taken into account, in which obstacle avoidance is taken as one of the prime considerations. In such cases, the feasible solution is reduced with increase in the iteration. The repairing of solution mechanism is used to find the best solution in context with the global update. If [R ($PI_{currentiteration}$) $\neq$ R ($PI_{twopreviousiterations}$)], then the local constraints will update. Moreover, from global search, the new solution will be selected and considered as a repaired solution.

https://doi.org/10.1017/S0263574721000114 Published online by Cambridge University Press

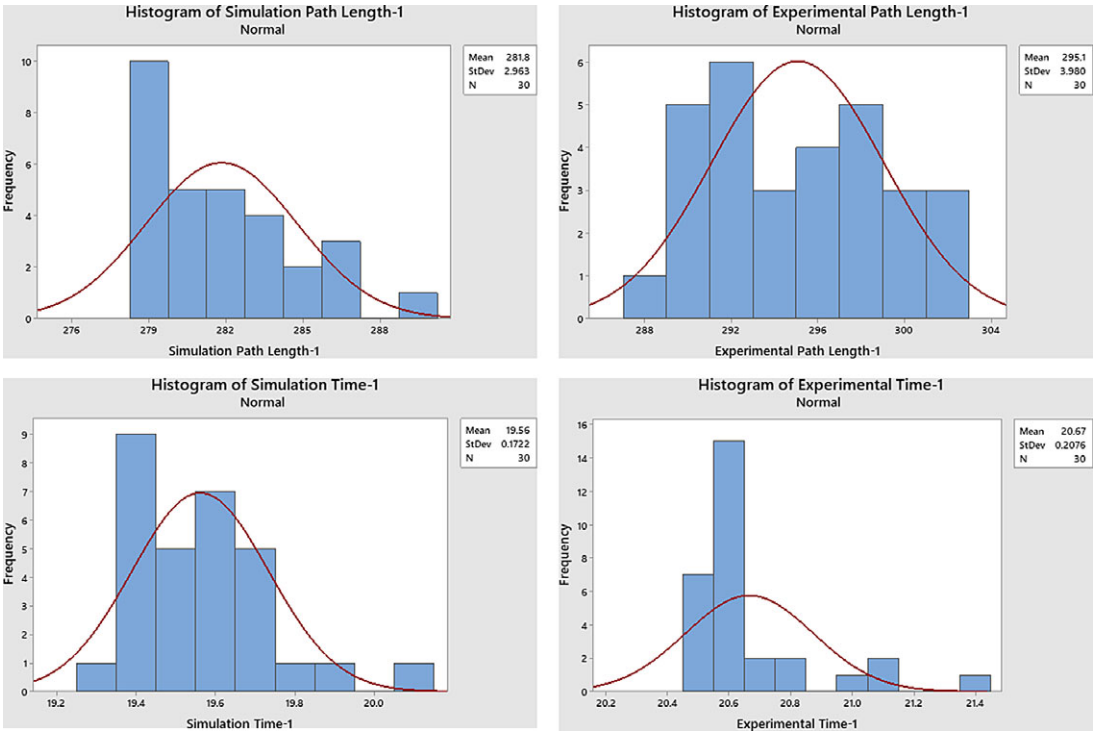

Fig. 9. Histogram of results obtained in scene 1.

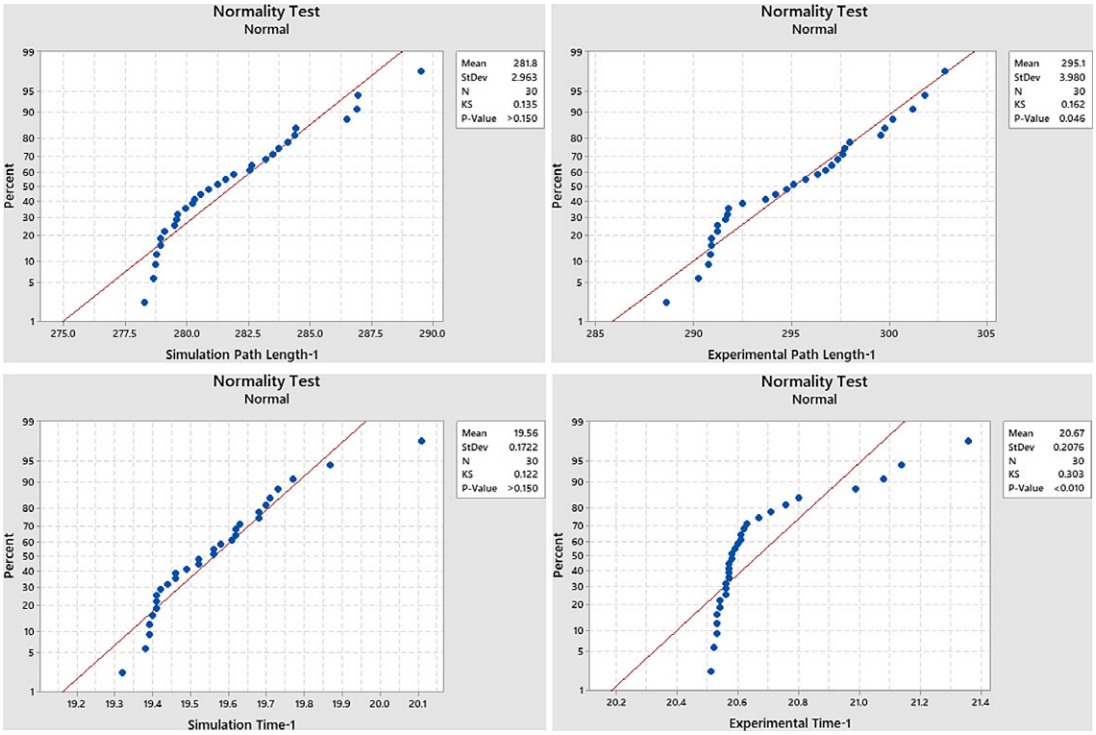

Fig. 10. Normality test of results obtained in scene 1.

https://doi.org/10.1017/S0263574721000114 Published online by Cambridge University Press

Table III. Comparison of path length in simulation and corresponding experiments in scene 2

	Path length (cm)		
No. of runs	**Simulation**	**Experiment**	**% Deviation**
1	296.79	312.17	4.93
2	299.35	319.23	6.23
3	301.51	318.54	5.35
4	297.78	315.47	5.61
5	296.98	316.96	6.30
6	298.45	315.41	5.38
7	298.44	313.73	4.87
8	**295.65**	**310.11**	**4.66**
9	295.97	311.56	5.00
10	296.12	312.42	5.22
11	296.96	313.75	5.35
12	297.37	312.85	4.95
13	300.45	314.23	4.39
14	301.72	315.72	4.43
15	301.45	314.16	4.05
16	302.42	316.12	4.33
17	298.42	317.96	6.15
18	296.49	312.63	5.16
19	297.73	310.78	4.20
20	296.83	310.63	4.44
21	295.97	311.01	4.84
22	300.12	316.23	5.09
23	297.73	312.19	4.63
24	299.42	310.87	3.68
25	298.36	314.83	5.23
26	299.52	311.73	3.92
27	298.67	311.81	4.21
28	298.92	313.41	4.62
29	299.46	312.53	4.18
30	296.77	311.56	4.75
Average	**298.39**	**313.69**	**4.87**

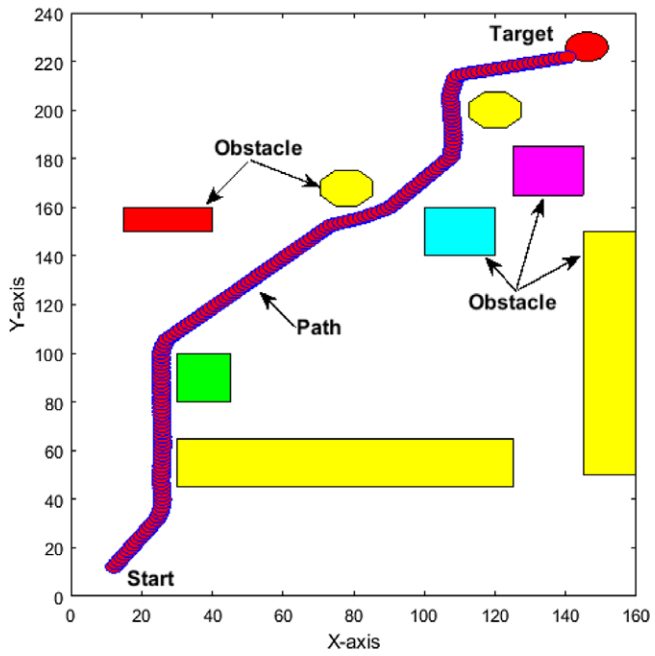

Fig. 11. Simulation analysis of the navigation of the mobile robot in scene 2.

https://doi.org/10.1017/S0263574721000114 Published online by Cambridge University Press

Table IV. Comparison of execution time in simulation and corresponding experiments in scene 2

	Execution time (s)		
No. of runs	**Simulation**	**Experiment**	**% Deviation**
1	20.74	21.56	3.80
2	21.08	22.17	4.92
3	21.41	22.52	4.93
4	20.98	22.13	5.20
5	20.77	21.91	5.20
6	20.61	21.62	4.67
7	20.78	21.71	4.28
8	**20.53**	**21.54**	**4.69**
9	20.91	21.67	3.51
10	20.73	21.7	4.47
11	20.57	21.57	4.64
12	20.61	21.59	4.54
13	21.02	22.01	4.50
14	21.12	22.11	4.48
15	21.1	22.16	4.78
16	21.32	22.27	4.27
17	20.92	21.97	4.78
18	20.72	21.59	4.03
19	20.62	21.73	5.11
20	20.7	21.76	4.87
21	20.59	21.59	4.63
22	21.09	22.24	5.17
23	20.68	21.72	4.79
24	20.87	21.96	4.96
25	20.83	22.16	6.00
26	20.9	22.18	5.77
27	20.63	21.79	5.32
28	20.74	21.63	4.11
29	20.68	21.7	4.70
30	20.57	21.57	4.64
Average	**20.83**	**21.86**	**4.73**

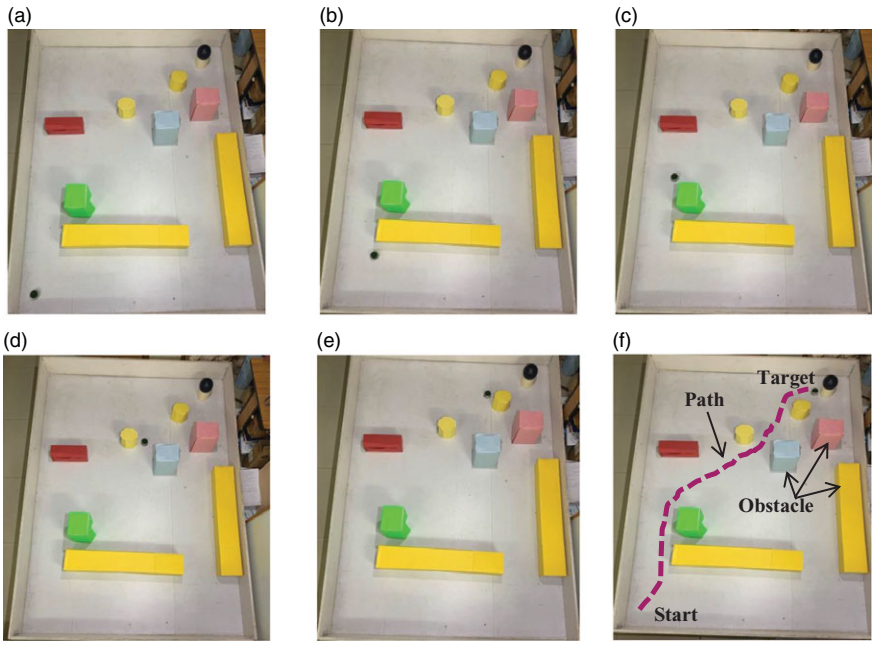

Fig. 12. Experimental analysis of the navigation of the mobile robot in scene 2.

https://doi.org/10.1017/S0263574721000114 Published online by Cambridge University Press

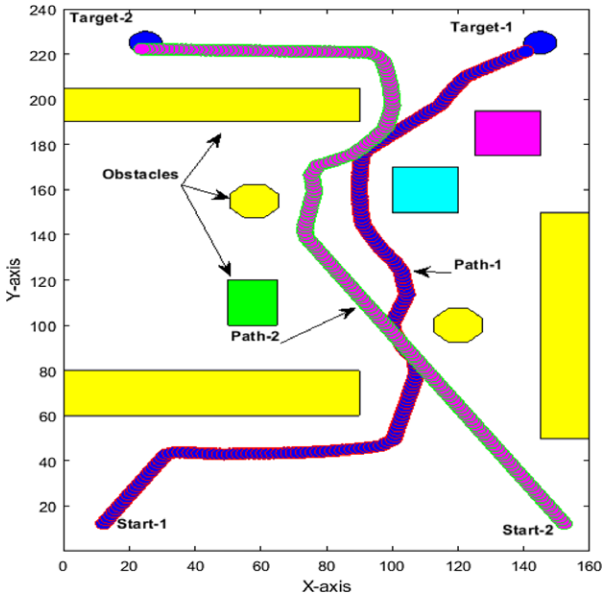

Fig. 13. Simulation analysis of the navigation of multiple mobile robots in scene 3.

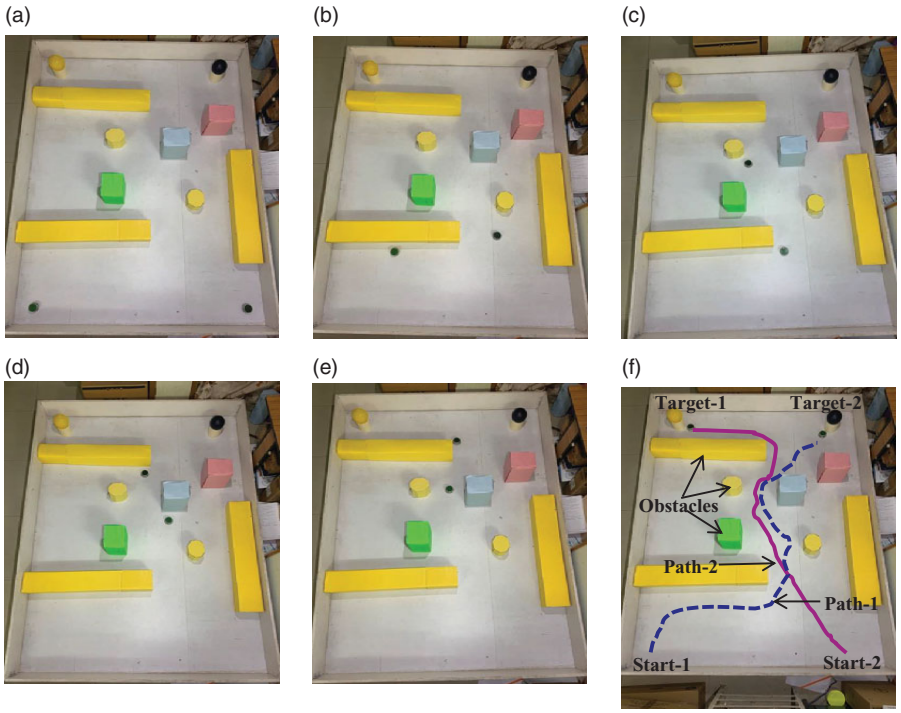

Fig. 14. Experimental analysis of the navigation of multiple mobile robots in scene 3.

3.5. Overview of hybrid IWD-GA

The GA concept was established by Holland[43] and explained by Goldberg.[44] The algorithm starts with the selection of a random population, which consists of a set of chromosomes. In random population size, the selected chromosomes may be fit or unfit. The fitness of the chromosomes is decided by their potential to choose an optimized solution. Each chromosome may be in binary code, octal code, hexadecimal code, etc. This code representation is used to improve fitness value, and it is the objective of the GA. The selection operator is used to get the fittest chromosomes from their parent

https://doi.org/10.1017/S0263574721000114 Published online by Cambridge University Press

Table V. Comparison of path length in simulation and corresponding experiments in scene 3

	Path length (cm)					
	Simulation		Experiment		% Deviation	
No. of runs	**Rob-1**	**Rob-2**	**Rob-1**	**Rob-2**	**Rob-1**	**Rob-2**
1	352.51	340.78	370.72	355.12	4.91	4.04
2	354.12	341.21	370.63	355.41	4.45	4.00
3	356.71	342.56	371.15	354.89	3.89	3.47
4	**352.12**	**340.41**	**369.87**	**354.45**	**4.80**	**3.96**
5	352.94	340.89	370.23	355.56	4.67	4.13
6	353.75	341.51	371.82	355.78	4.86	4.01
7	353.4	342.06	370.19	356.23	4.54	3.98
8	352.74	341.32	370.56	355.57	4.81	4.01
9	353.19	341.09	371.29	356.12	4.87	4.22
10	352.69	340.56	370.45	354.71	4.79	3.99
11	353.57	341.54	370.21	355.54	4.49	3.94
12	353.63	342.63	370.67	356.17	4.60	3.80
13	354.71	342.83	371.21	356.76	4.44	3.90
14	352.68	343.12	370.83	356.19	4.89	3.67
15	354.01	342.76	372.41	357.67	4.94	4.17
16	353.21	342.68	371.53	357.43	4.93	4.13
17	352.53	341.57	370.67	355.73	4.89	3.98
18	353.53	341.53	371.64	357.64	4.87	4.50
19	355.17	340.96	372.34	356.42	4.61	4.34
20	354.24	342.61	372.86	356.72	4.99	3.96
21	352.19	341.52	371.21	355.67	5.12	3.98
22	353.52	341.83	370.67	355.91	4.63	3.96
23	353.73	341.74	370.58	356.48	4.55	4.13
24	354.09	340.83	369.87	355.41	4.27	4.10
25	352.94	341.57	370.19	356.73	4.66	4.25
26	353.76	342.46	370.83	354.74	4.60	3.46
27	352.97	341.57	369.91	355.67	4.58	3.96
28	353.16	340.67	370.53	355.38	4.69	4.14
29	354.91	340.93	370.81	355.29	4.29	4.04
30	352.51	341.06	371.17	354.69	4.39	3.84
Average	**353.59**	**341.63**	**370.90**	**355.87**	**4.67**	**4.00**

chromosomes. The most appropriate chromosome selection is based on iteration or selection method (Roulette wheel selection) of GA, and the crossover and mutation operators are used to produce offspring and to set up new genes in the offspring. However, the bias towards optimal solution chromosomes is mostly selected. The GA is designed to choose the optimal solution by their iteration process, and it stops when meeting the stopping criteria given by the users. That's why GA is termed a versatile, robust optimization technique. Now, in the context of this approach, the GA is designed to find the best path in a complex environment. In view of path planning, GA is used to search the best feasible next stop for the mobile robot. In the working platform, the current position of the robot is used as the initial point, and it is considered for the calculation of heading angle and coordinates of the next stop. The inputs for GA are obstacle distance and target distance, which are fed with the assistance of sensors reading and mathematical calculations. The target distance is expressed as:

$$Tar(dis) = \sqrt{(C_{ix} - G_x)^2 + (C_{iy} - G_y)^2} \tag{14}$$

C_{ix} and C_{iy} is the current stand of the robot and G_x, G_y are the target coordinates in x and y planes. The idea of hybridization of IWD-GA is to merge the best attributes of these two algorithms to extract the most appropriate solution. The reason behind the use of GA is to make random deviation of soil

https://doi.org/10.1017/S0263574721000114 Published online by Cambridge University Press

Table VI. Comparison of execution time in simulation and corresponding experiments in scene 3

	Path length (cm)					
	Simulation		Experiment		% Deviation	
No. of runs	**Rob-1**	**Rob-2**	**Rob-1**	**Rob-2**	**Rob-1**	**Rob-2**
1	24.45	23.24	25.32	23.99	3.44	3.13
2	24.56	23.52	25.39	24.23	3.27	2.93
3	24.89	23.41	25.61	24.16	2.81	3.10
4	**24.03**	**23.12**	**25.10**	**23.83**	**4.26**	**2.98**
5	24.23	23.70	25.22	24.42	3.93	2.95
6	24.86	23.42	25.47	24.26	2.39	3.46
7	24.42	23.56	25.56	24.45	4.46	3.64
8	24.13	23.49	25.29	24.31	4.59	3.37
9	24.73	23.34	25.43	24.40	2.75	4.34
10	24.41	23.19	25.21	24.14	3.17	3.94
11	24.17	23.59	25.13	24.36	3.82	3.16
12	24.67	23.81	25.63	24.71	3.75	3.64
13	24.53	23.56	25.74	24.91	4.70	5.42
14	24.87	23.27	25.45	24.67	2.28	5.67
15	24.61	23.67	25.96	24.16	5.20	2.03
16	24.53	23.90	25.76	24.97	4.77	4.29
17	24.19	23.26	25.91	24.38	6.64	4.59
18	24.37	23.73	25.64	24.29	4.95	2.31
19	24.49	23.26	25.71	24.67	4.75	5.72
20	24.94	24.03	25.67	24.34	2.84	1.27
21	25.01	23.61	26.13	24.49	4.29	3.59
22	24.18	23.53	25.68	24.61	5.84	4.39
23	24.29	23.39	25.34	24.53	4.14	4.65
24	25.06	23.14	26.21	24.51	4.39	5.59
25	24.23	23.34	25.69	24.17	5.68	3.43
26	24.67	24.12	25.38	24.68	2.80	2.27
27	24.31	23.71	25.61	24.71	5.08	4.05
28	24.29	23.37	25.34	24.36	4.14	4.06
29	24.91	23.48	25.78	24.29	3.37	3.33
30	24.46	23.34	25.31	24.73	3.36	5.62
Average	**24.52**	**23.50**	**25.56**	**24.42**	**4.06**	**3.76**

in the paths. The variation of soils in the way guides to find the best route by IWD. In the IWD-GA approach, the results obtained from Equations 7 and 9 are given as inputs to GA; then the probability fitness function can be formalized as:

$$M_i = \sum_{j=1}^{i} S_i \tag{15}$$

where S_i is the probability of choosing each chromosome, and its function can be expressed as:

$$S_i = \frac{Sar(V_i)}{F} \tag{16}$$

where V_i represents each chromosome and $i = 1,\ 2,\ 3 \ldots N$. N is the initial population size. F is the fitness function of each chromosome that can be expressed as:

$$F = \sum_{i=1}^{N} Sar(V_i) \tag{17}$$

https://doi.org/10.1017/S0263574721000114 Published online by Cambridge University Press

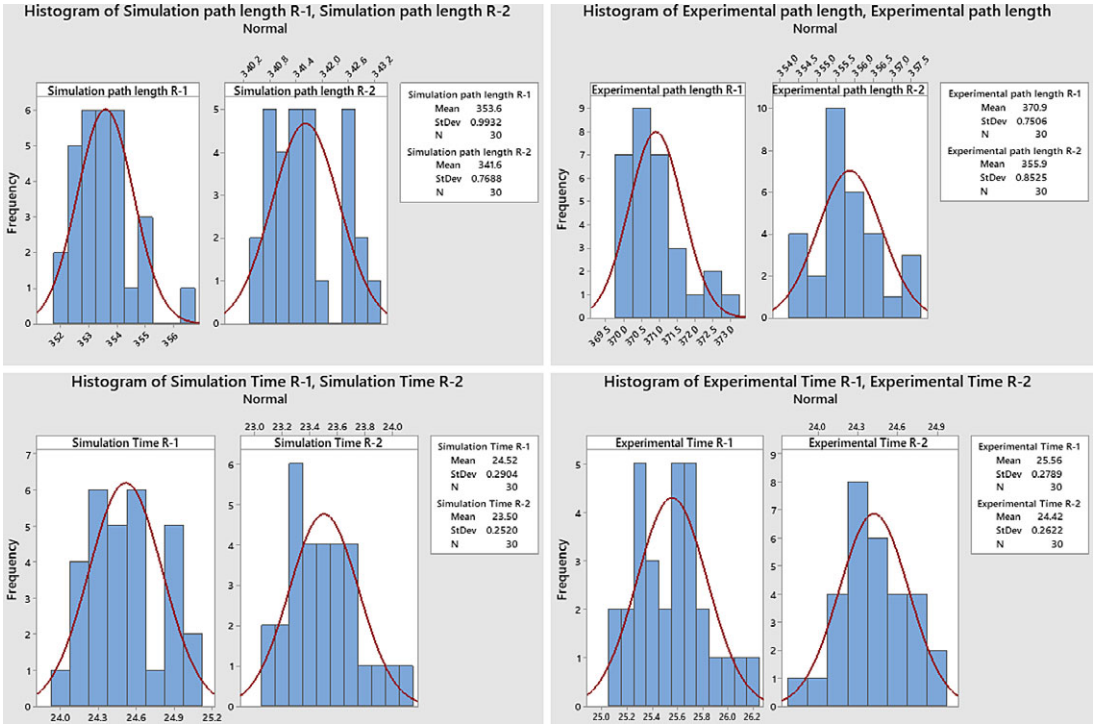

Fig. 15. Histogram of results obtained in scene 3.

Sar() is the function of the chromosome population. The best IWD-GA edges are updated according to the travelled nodes of IWD. This can be expressed as:

$$\text{Soil}(i,\ j) = (1 + s_n) \cdot \text{soil}(i,\ j) - \rho_n \cdot \frac{1}{N_{LT} - 1} \cdot \text{soil}_{LT}^{n} \tag{18}$$

$$\forall (i,\ j) \in P^{WD}$$

$$\text{Soil}(i,\ j) = \begin{cases} f(\text{soil}\{e_{i,i+1}(k)\})N = S_i \\ g\left(\text{soil}\left(e_{i,j}\ (k)\right)\right) \text{Otherwise} \end{cases} \tag{19}$$

where $soil_{LT}^{n}$ is the accumulated soil by the best IWD, and N_{LT} is the length travelled by the best IWD. s_n and P_{iwd} are constant parameters. Figure 5 describes the flowchart of IWD-GA.

4. Navigational Analysis Using IWD-GA Technique

The effectiveness of the developed technique is presented here. A simulation environment with obstacles is developed through MATLAB software. The target point, start point, generated path and obstacles are clearly indicated by the arrows in respective simulation figures. The technique is further investigated in the real environment under laboratory conditions by the deployment of KHEPERA-II robot; the description is given in Fig. 6. Eight infrared sensors and ambient sensors detect obstacles in the environment.

4.1. Navigation using a single robot in a complex environment

A complex environment of size 240 × 160 cm^2 is established for simulation and real-time experiment. The simulation analysis is shown in Fig. 7. In Figs. 8(f) and 12(f), the complete environment is clearly indicated. Tables I and II show the results generated by the robot in scene 1.

https://doi.org/10.1017/S0263574721000114 Published online by Cambridge University Press

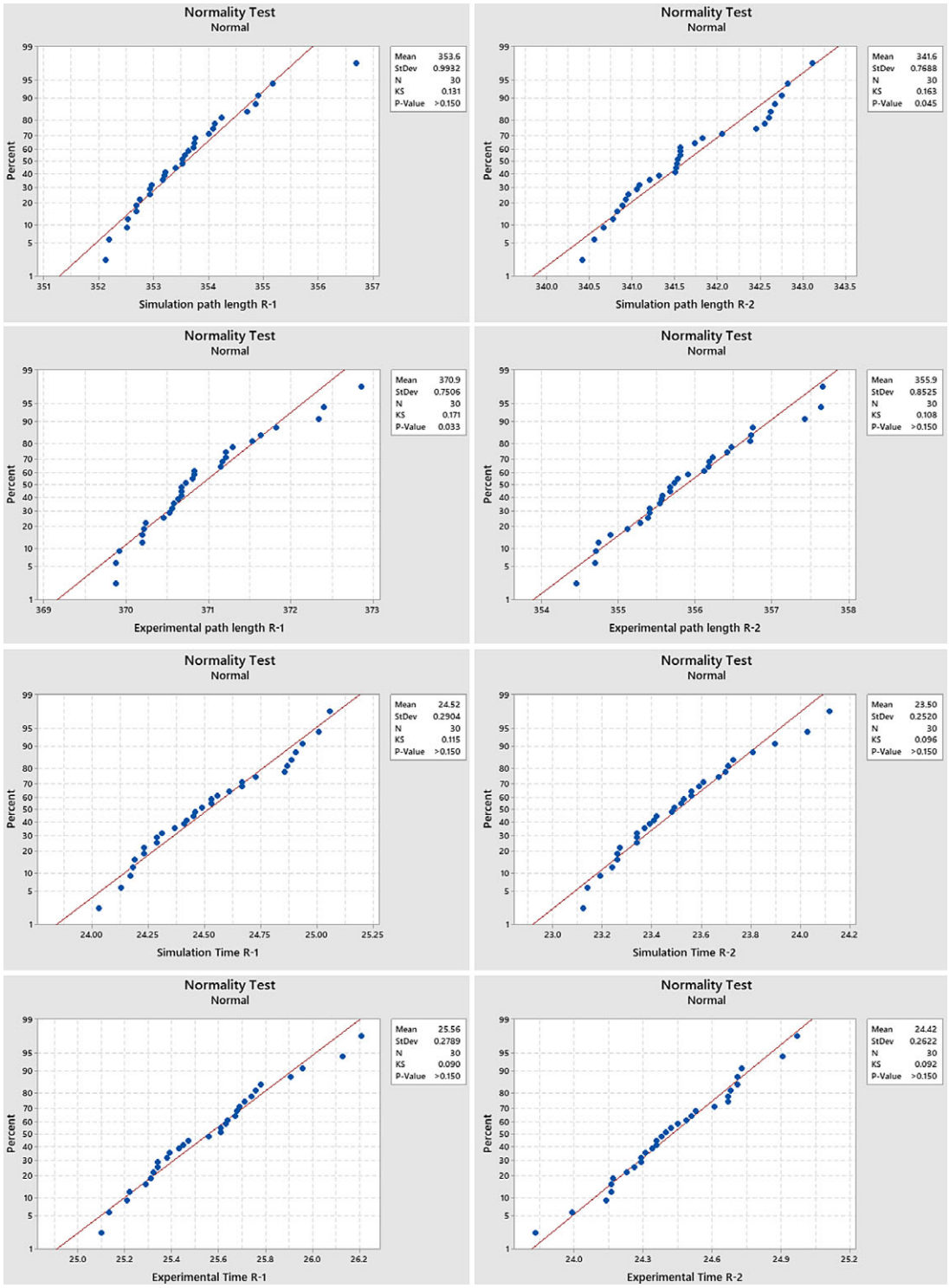

Fig. 16. Normality test of results obtained in scene 3.

4.1.1. Statistical analysis of results. A statistical analysis of results obtained during the navigation of robot in simulation and experimental platforms is performed. The normality test of results and histograms are created. For testing normality, the Kolmogorov–Smirnov test is performed, and their graphs are cited Figs. 9 and 10.

https://doi.org/10.1017/S0263574721000114 Published online by Cambridge University Press

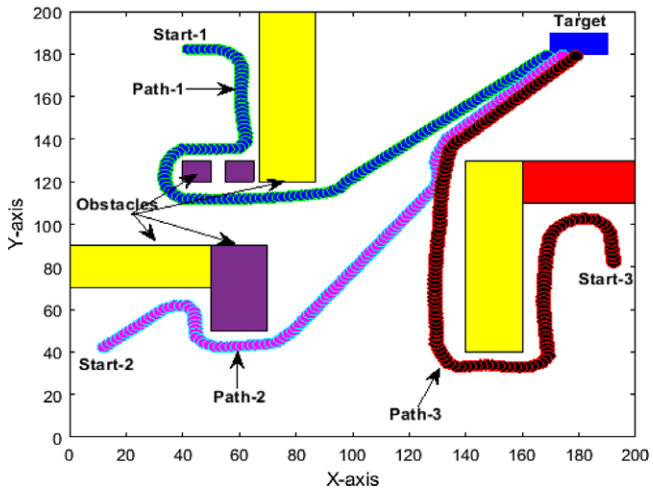

Fig. 17. Simulation analysis of the navigation of three mobile robots in scene 4.

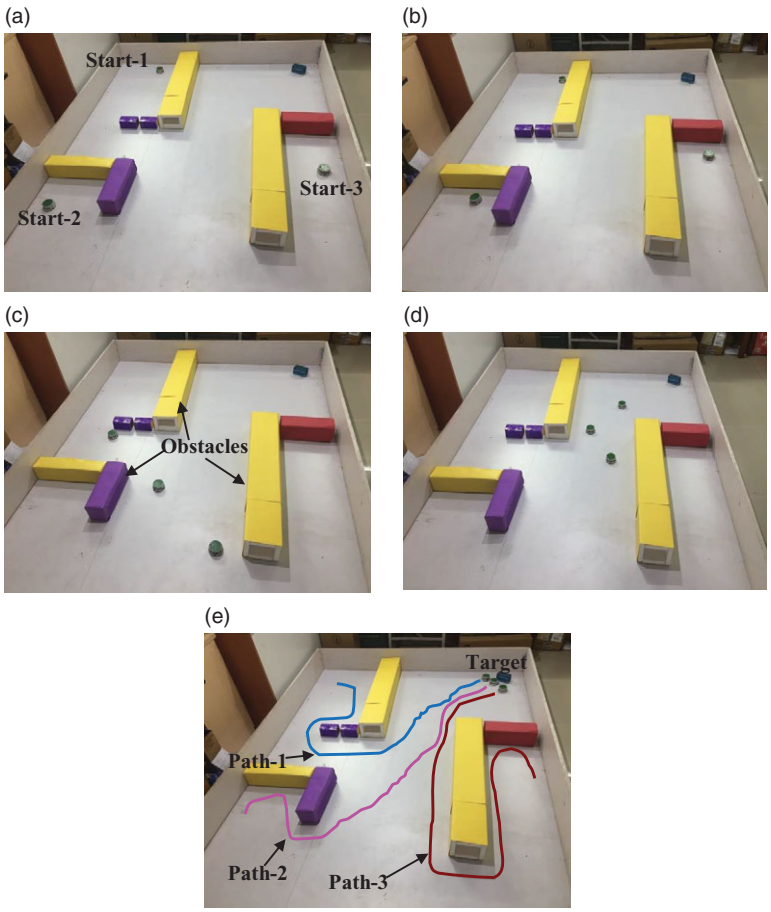

Fig. 18. Experimental analysis of the navigation of three mobile robots in scene 4.

Another simulation and real-time environment is developed with different obstacles orientation using the same arena size. In this scene, eight obstacles were deployed to create a complex environment. Figures 11 and 12(f) show the simulation and real-time experimental results. The robot successfully reached the target after avoiding the obstacles, and the results are recorded in Tables III and IV. The averages of all the runs in terms of path length, time taken and deviation by robots are

https://doi.org/10.1017/S0263574721000114 Published online by Cambridge University Press

Table VII. Comparison of path length in simulation and corresponding experiments in scene 4

	Path length (cm)								
	Simulation			Experiment			% Deviation		
No. of runs	**Rob-1**	**Rob-2**	**Rob-3**	**Rob-1**	**Rob-2**	**Rob-3**	**Rob-1**	**Rob-2**	**Rob-3**
1	266.45	256.78	302.41	275.52	267.4	314.72	3.29	3.97	3.91
2	**266.11**	**256.06**	**302.21**	**275.23**	**266.89**	**314.13**	**3.31**	**4.06**	**3.79**
3	267.01	256.12	302.56	275.46	267.12	314.23	3.07	4.12	3.71
4	266.72	256.21	302.81	276.23	267.56	314.42	3.44	4.24	3.69
5	266.32	256.41	303.02	275.71	267.23	314.56	3.41	4.05	3.67
6	266.56	257.45	302.84	275.85	267.63	314.49	3.37	3.80	3.70
7	267.11	256.56	302.36	275.63	267.41	314.56	3.09	4.06	3.88
8	267.23	256.71	302.45	276.11	267.42	314.72	3.22	4.00	3.90
9	266.72	256.62	302.73	276.23	267.66	314.9	3.44	4.12	3.86
10	266.36	256.19	302.92	275.43	267.08	314.36	3.29	4.08	3.64
11	267.37	257.16	302.65	276.42	269.64	315.63	3.27	4.63	4.11
12	267.64	257.31	303.41	277.16	268.51	315.42	3.43	4.17	3.81
13	267.19	257.14	303.49	277.61	267.91	316.24	3.75	4.02	4.03
14	268.39	256.18	303.42	278.61	268.74	316.74	3.67	4.67	4.21
15	267.69	257.81	304.52	276.61	269.17	315.63	3.22	4.22	3.52
16	267.81	257.53	303.46	279.46	267.17	316.85	4.17	3.61	4.23
17	268.24	258.13	304.19	278.94	268.17	316.47	3.84	3.74	3.88
18	267.91	257.29	303.42	279.53	269.42	315.42	4.16	4.50	3.80
19	267.28	257.49	304.91	276.16	268.61	316.19	3.22	4.14	3.57
20	268.16	257.64	303.67	277.64	267.84	314.92	3.41	3.81	3.57
21	266.64	258.91	303.93	279.83	267.83	315.46	4.71	3.33	3.65
22	266.57	258.37	304.15	280.19	266.97	316.42	4.86	3.22	3.88
23	267.15	256.93	303.64	278.67	267.94	317.86	4.13	4.11	4.47
24	267.83	257.37	304.17	276.42	268.97	316.48	3.11	4.31	3.89
25	268.16	257.46	304.26	278.61	268.83	316.73	3.75	4.23	3.94
26	267.42	257.52	303.83	280.14	268.48	317.45	4.54	4.08	4.29
27	267.49	256.83	303.58	276.94	267.41	315.42	3.41	3.96	3.75
28	266.79	256.67	304.87	275.94	268.46	315.42	3.32	4.39	3.34
29	267.48	257.51	303.52	278.69	269.17	314.95	4.02	4.33	3.63
30	267.61	257.61	302.73	277.34	267.64	315.64	3.51	3.75	4.09
Average	267.25	257.13	303.40	277.28	268.01	315.55	3.61	4.06	3.85

inscribed in bold font. It is also observed that the navigation of a single robot with the proposed technique is very good as the deviation in both platforms is found within the acceptable limit of 5.5%. This deviation authenticates the agreement in both platforms.

4.2. Navigation using multiple robots in a complex environment

It is also required to check the efficacy with multi-robots. Therefore, a highly complex environment is created. A simulation environment and experimental workspace is developed and the movement of robot is recorded. The simulation and experimental figures are depicted in Figs. 13 and 14, respectively. The results are tabulated in Tables V and VI. The obtained results are compared, and it is observed that the deviation in simulation and real-time experiments is within 5%.

4.2.1. Statistical analysis of results. For testing normality, the Kolmogorov–Smirnov test is performed and their graphs are cited in Figs. 15 and 16.

Further, an investigation is performed with three robots in a complex environment as shown in Figs. 17 and 18. In this investigation, the robots have different start positions but same target point, and the robots successfully reached the target without any collision. The results are tabulated in Tables VII and VIII. The deviations are within 5%, which is acceptable.

https://doi.org/10.1017/S0263574721000114 Published online by Cambridge University Press

Table VIII. Comparison of execution time in simulation and corresponding experiments in scene 4

	Execution time (s)								
	Simulation			Experiment			% Deviation		
No. of runs	Rob-1	Rob-2	Rob-3	Rob-1	Rob-2	Rob-3	Rob-1	Rob-2	Rob-3
1	23.71	19.54	29.03	24.65	20.3	29.92	3.81	3.74	2.97
2	**23.09**	**19.13**	**28.74**	**24.12**	**20.08**	**29.65**	**4.27**	**4.73**	**3.07**
3	23.11	19.17	29.12	24.23	20.12	30.22	4.62	4.72	3.64
4	23.32	19.74	29.13	24.39	20.21	30.11	4.39	2.33	3.25
5	23.26	19.8	28.91	24.16	20.19	30.04	3.73	1.93	3.76
6	23.42	19.62	28.87	24.32	20.21	29.82	3.70	2.92	3.19
7	23.33	19.63	28.84	24.3	20.23	29.83	3.99	2.97	3.32
8	23.36	19.43	28.96	24.17	20.3	29.91	3.35	4.29	3.18
9	23.29	19.26	28.98	24.21	20.24	29.68	3.80	4.84	2.36
10	23.17	19.24	29.01	24.18	20.17	29.76	4.18	4.61	2.52
11	23.19	19.31	29.14	24.21	20.14	30.12	4.21	4.12	3.25
12	24.03	19.42	29.42	24.97	20.42	30.42	3.76	4.90	3.29
13	23.94	19.64	29.14	25.01	20.46	30.76	4.28	4.01	5.27
14	23.64	19.61	29.46	24.74	20.81	30.97	4.45	5.77	4.88
15	23.16	20.19	28.97	24.36	21.06	30.19	4.93	4.13	4.04
16	23.18	19.67	29.11	24.38	20.49	30.42	4.92	4.00	4.31
17	23.24	19.69	29.67	24.19	20.83	30.48	3.93	5.47	2.66
18	24.1	20.19	29.43	25.11	21.08	30.84	4.02	4.22	4.57
19	23.25	19.76	29.94	24.34	20.85	30.91	4.48	5.23	3.14
20	23.61	19.34	29.68	24.74	20.31	30.75	4.57	4.78	3.48
21	23.37	19.48	29.47	24.43	20.42	31.05	4.34	4.60	5.09
22	23.97	19.18	29.68	25.15	20.17	31.18	4.69	4.91	4.81
23	23.38	19.41	28.98	24.36	20.41	30.41	4.02	4.90	4.70
24	23.26	19.87	29.34	24.16	20.68	30.79	3.73	3.92	4.71
25	23.51	19.64	29.18	24.67	20.49	30.49	4.70	4.15	4.30
26	23.63	19.87	29.37	24.78	20.61	31.27	4.64	3.59	6.08
27	23.13	19.67	29.42	24.28	20.37	30.98	4.74	3.44	5.04
28	23.14	19.97	29.47	24.12	21.1	30.62	4.06	5.36	3.76
29	23.84	19.67	28.83	25.16	20.67	30.51	5.25	4.84	5.51
30	23.95	19.36	29.31	25.34	20.34	31.11	5.49	4.82	5.79
Average=	23.46	19.59	29.23	24.52	20.47	30.46	4.32	4.29	4.04

Table IX. Path length and execution time in scene 5

Simulation path length (cm)	Simulation execution time (s)
243.15	23.32

Table X. Comparison of path lengths with Fuzzy-.PRM[45]

		Path length		
Sl No.	Environment	Fuzzy-PRM (pixels)	IWD-GA (pixels)	% Improvement
1	Comparison in environment 1 Fig. 20 (a) and (b)	883	787	**10.87**
2	Comparison in environment 2 Fig. 20 (c) and (d)	910	832	**8.57**
3	Comparison in environment 3 Fig. 20 (e) and (f)	921	835	**9.34**
Average improvement = 9.59%				

https://doi.org/10.1017/S0263574721000114 Published online by Cambridge University Press

Table XI. Comparison of path lengths with IQ-FPA[46]

Sl No.	Environment	Path length		% Improvement
		IQ-FPA (fig.(a)) (unit)	IWD-GA (fig.(b)) (unit)	
1	Comparison in environment 1 Fig. 21 (a) and (c)	29.33	24.41	16.77
2	Comparison in environment 2 Fig. 22 (a) and (c)	31.27	23.49	**24.88**
Average improvement = 20.83%				

Table XII. Comparison of path lengths with IQD[46]

Sl No.	Environment	Path length		% Improvement
		IQD (fig.(a)) (pixels)	IWD-GA (fig.(b)) (pixels)	
1	Comparison in environment 1 Fig. 21 (b) and (c)	26.27	24.41	**7.08**
2	Comparison in environment 2 Fig. 22 (b) and (c)	26.37	23.49	**10.92**
Average improvement = 9%				

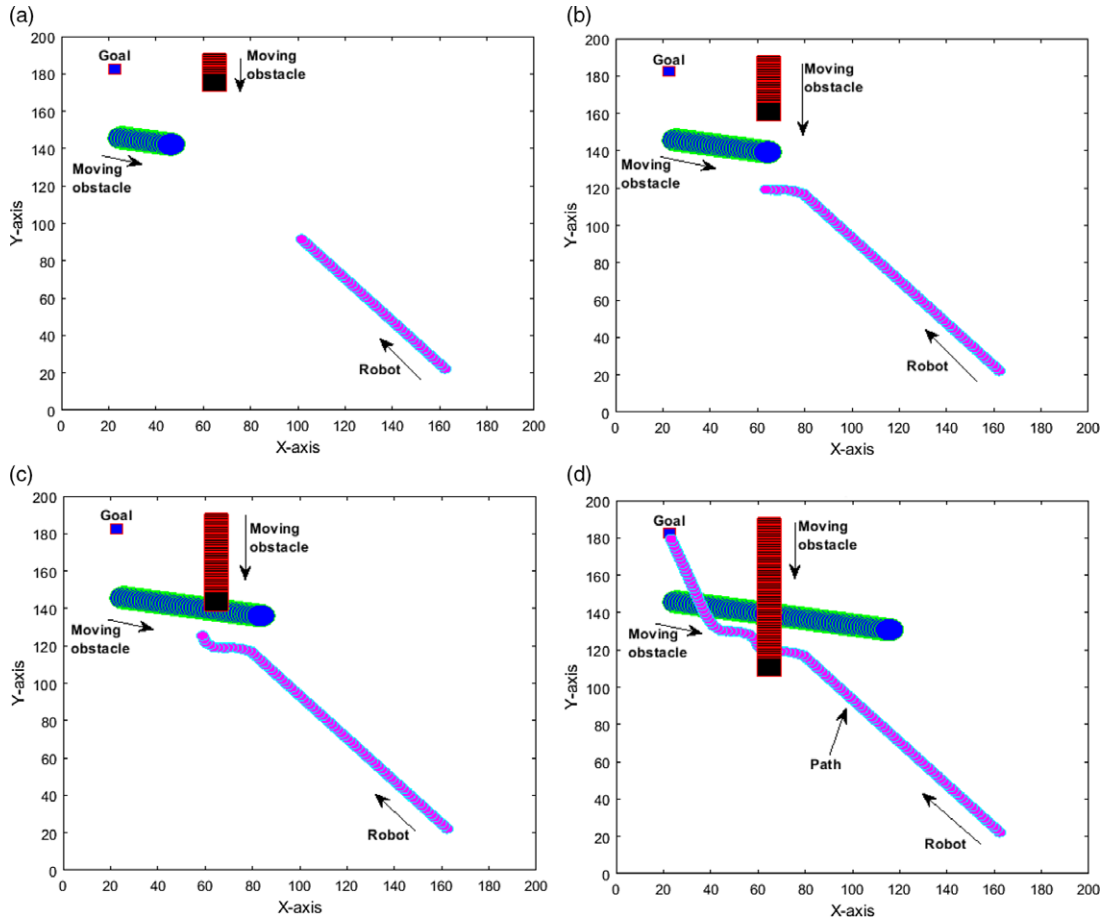

Fig. 19. IWD-GA in a dynamic environment (scene 5).

https://doi.org/10.1017/S0263574721000114 Published online by Cambridge University Press

Table XIII. Comparison of path lengths with ISLF algorithm[47]

Sl No.	Environment	Path length		% Improvement
		ISLFA (units)	IWD-GA (units)	
1	Comparison in environment 1 Fig. 23 (a) and (b)	23.75	19.11	**19.54**

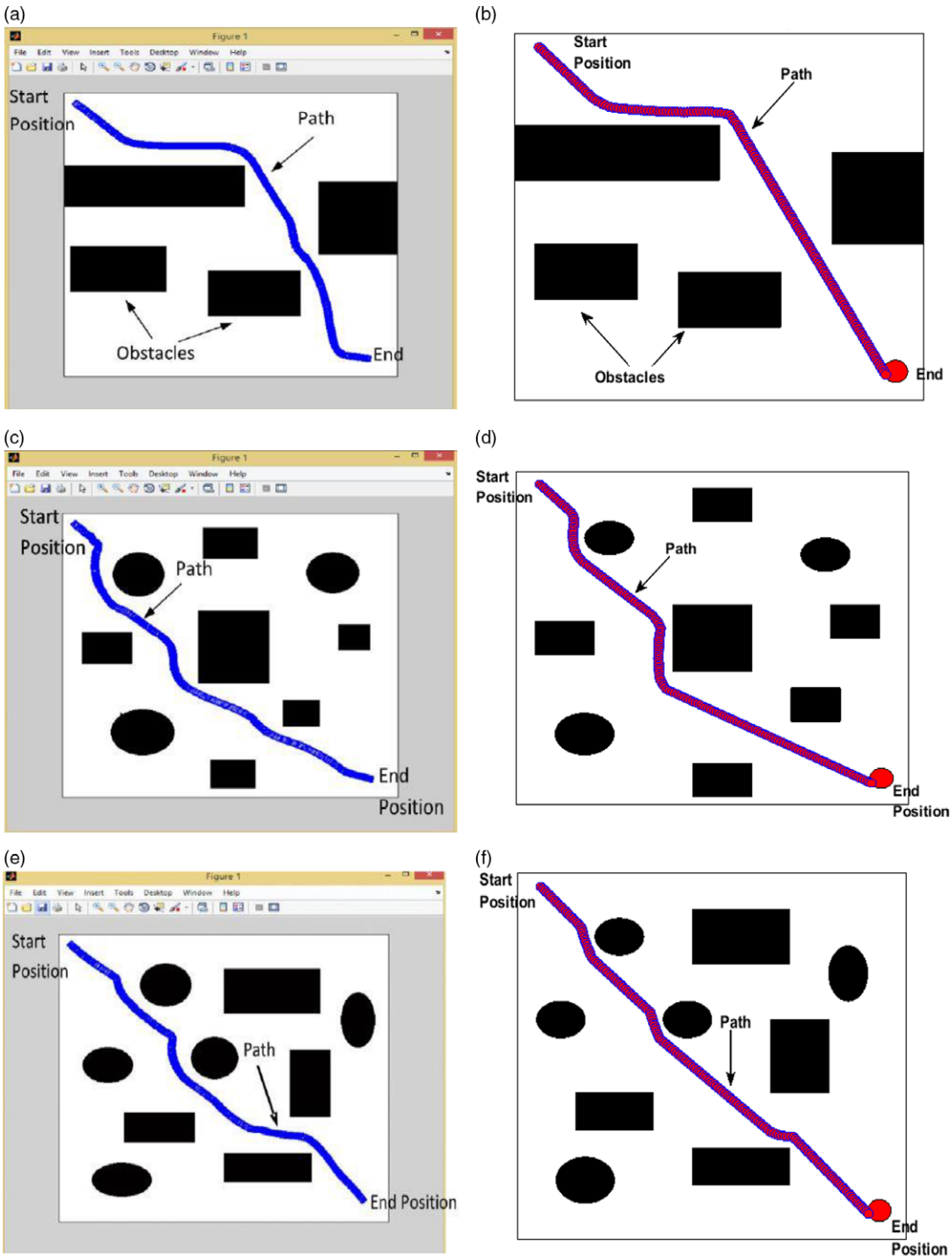

Fig. 20. (a), (c), (e) Path generated by Mohanta et al.[45] (b), (d), (f) Path generated by the proposed technique.

https://doi.org/10.1017/S0263574721000114 Published online by Cambridge University Press

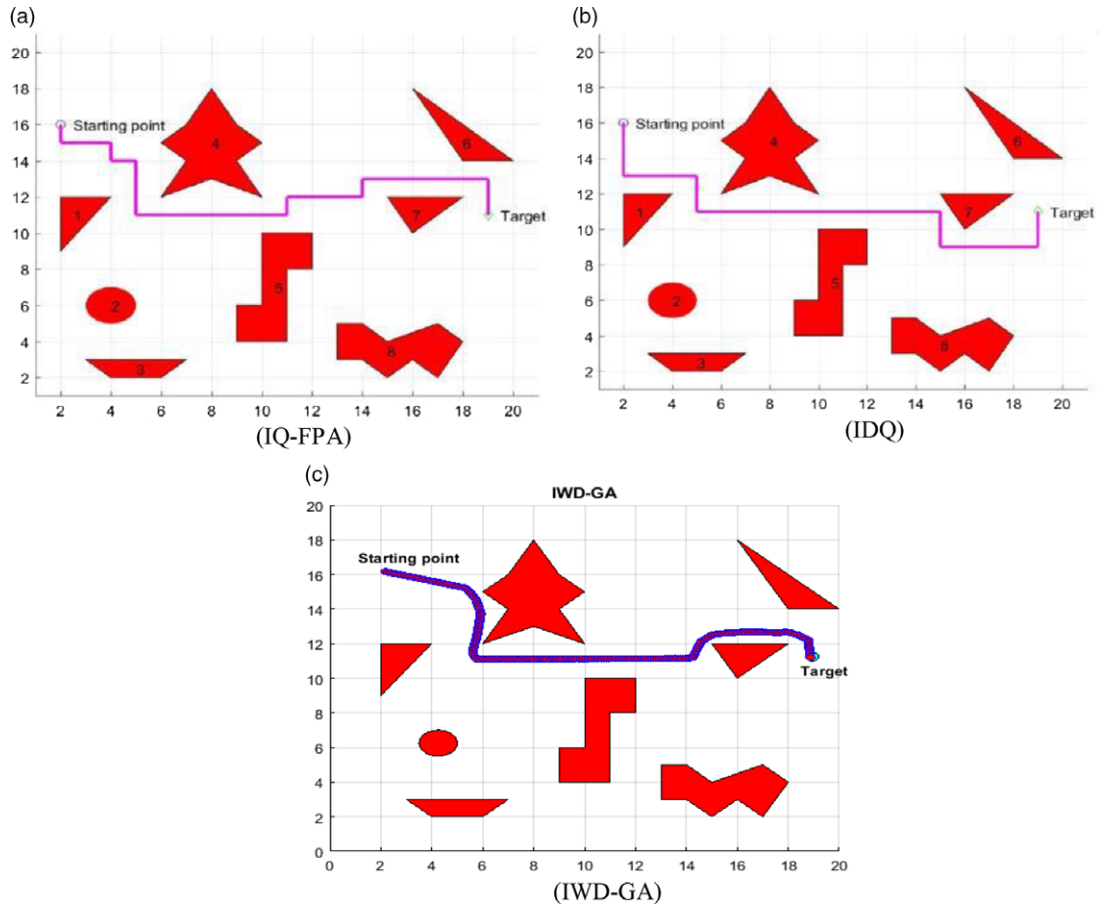

Fig. 21. (a), (b) Path generated by Low et al.[46] (c) Path genrated by the proposed technique.

4.3. Navigation in a dynamic environment

Apart from those investigations, it is also required to check the performance in a dynamic environment. A dynamic environment is created as shown in Fig. 19. During a navigation, no collision is observed, and the robot successfully reached the target. Ten runs have been applied to get an optimal result, and optimal results are inscribed in Table IX.

Deviations in the results of the platforms somehow differed. A leading cause of deviation is the condition taken during simulation and real-time experiments. In simulations, all conditions are considered ideal, but in real-time experiments, wheel slippage, friction on wheels, uneven path of robot during navigation are the main factors causing some deviation in results. The results show an average deviation of about 5%, which is acceptable.

5. Comparative Analysis between the Proposed IWD-GA and Other Existing AI Techniques

The effectiveness of the proposed technique is tested by simulation and experimental results. At the same time, it is also essential to test the effectiveness of the technique with other available techniques. For this, IWD-GA is compared with Fuzzy-PRM,[45] improved Q-learning algorithm,[46] IQ-FPA[46] and ISLF.[47] The results are depicted in the Tables X, XI, XII, and XIII.

From the comparisons, it is clear that the proposed technique (IWD-GA) gives better results than other AI techniques. Furthermore, the proposed technique is compared individually with IWD and GA techniques, and the results of comparison confirm the efficacy of the proposed technique. The comparative analysis is shown in Figs. 24 and 25. The result is recorded in Table XIV.

5.1. Significant contributions

The approach is tested on simulation and real-time experimental platforms, and the observed results show a good agreement between both platforms. Further, this technique is compared with existing

https://doi.org/10.1017/S0263574721000114 Published online by Cambridge University Press

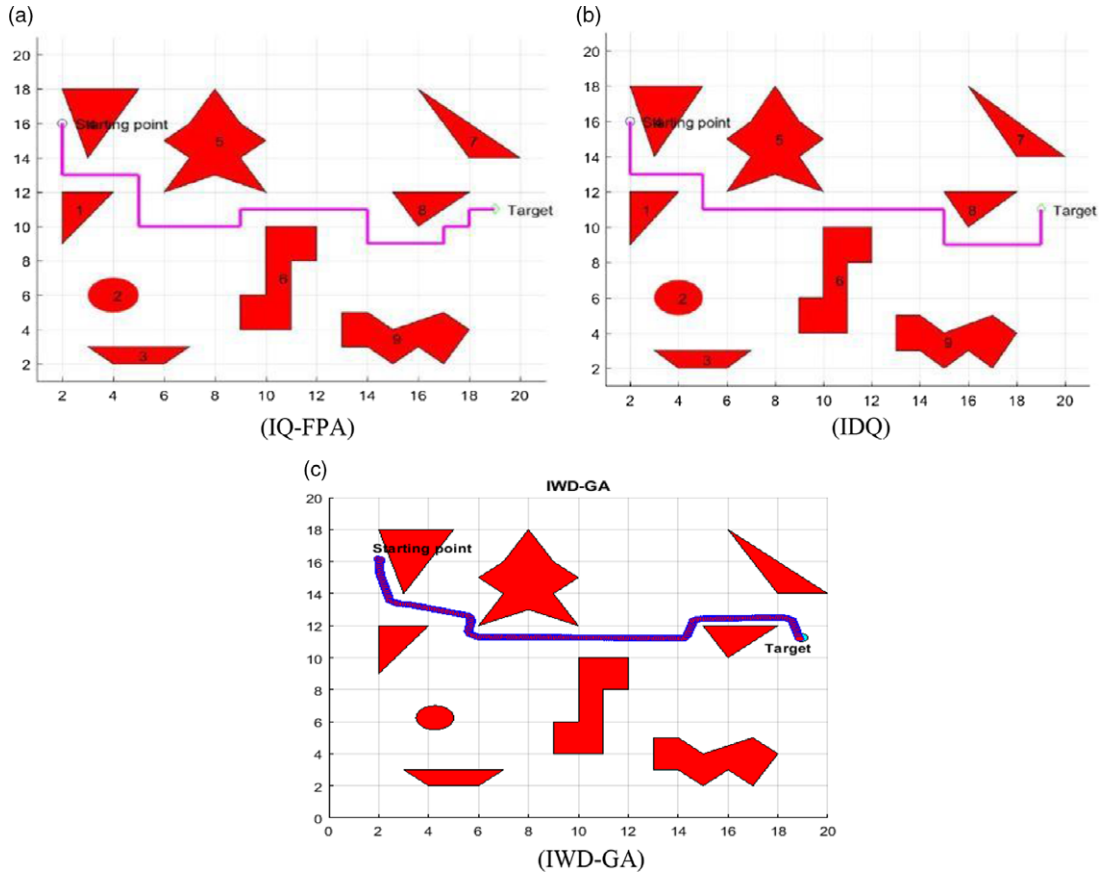

Fig. 22. (a), (b) Path generated by Low et al.[46] (c) Path genrated by the proposed technique.

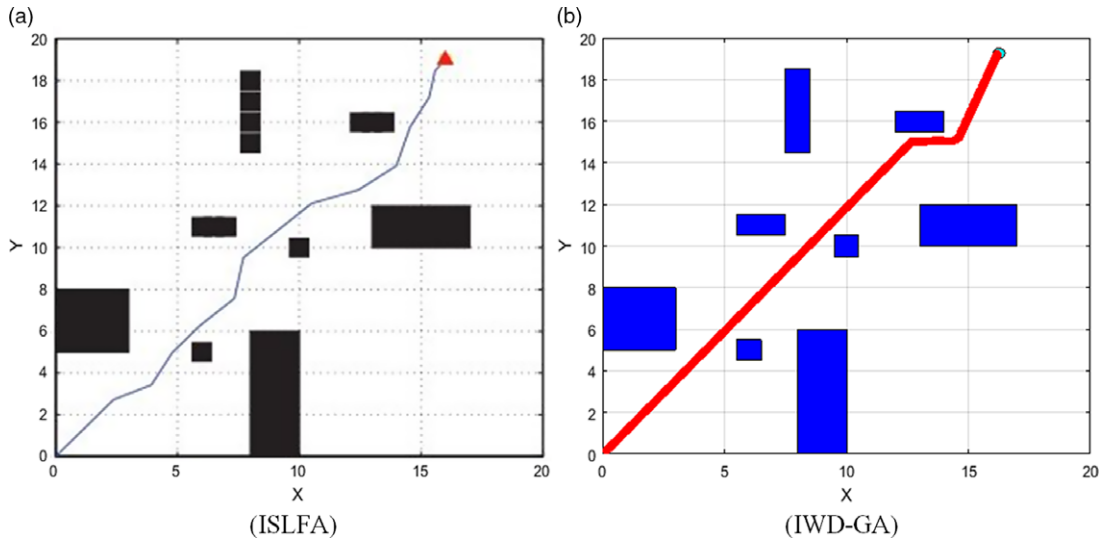

Fig. 23. (a) Path genrated by Ni et al.[47] (b) Path generated by the proposed technique.

techniques such as fuzzy-PRM,[45] improved Q-learning,[46] IQ-FPA[46] and ISLF[47] and showed a significant improvement of 9.59%, 20.83%, 9% and 19.54%, respectively, in terms of path length. Again, the approach is tested individually with the IWD and GA techniques, showing an average improvement of 10.67% in path length and 11.19% in execution time. From these comparisons, it can be concluded that the proposed approach performs well and optimizes path length during navigation and shows better control over robots.

https://doi.org/10.1017/S0263574721000114 Published online by Cambridge University Press

Table XIV. Comparison of path lengths with IWD and GA

Sl No.	Scene	AI technique	Path length (cm)		% Improvement		Execution time (s)		% Improvement	
1	1	IWD	307.42		9.49		21.08		8.35	
2		GA	316.81		12.16		22.43		13.86	
3		IWD-GA	278.26				19.32			
			R-1	R-2	R-1	R-2	R-1	R-2	R-1	R-2
4	3	IWD	393.42	369.92	11.73	7.98	27.12	25.23	11.39	8.36
5		GA	402.63	378.61	12.55	10.09	27.87	26.11	13.77	11.45
6		IWD-GA	352.12	340.41			24.03	23.12		

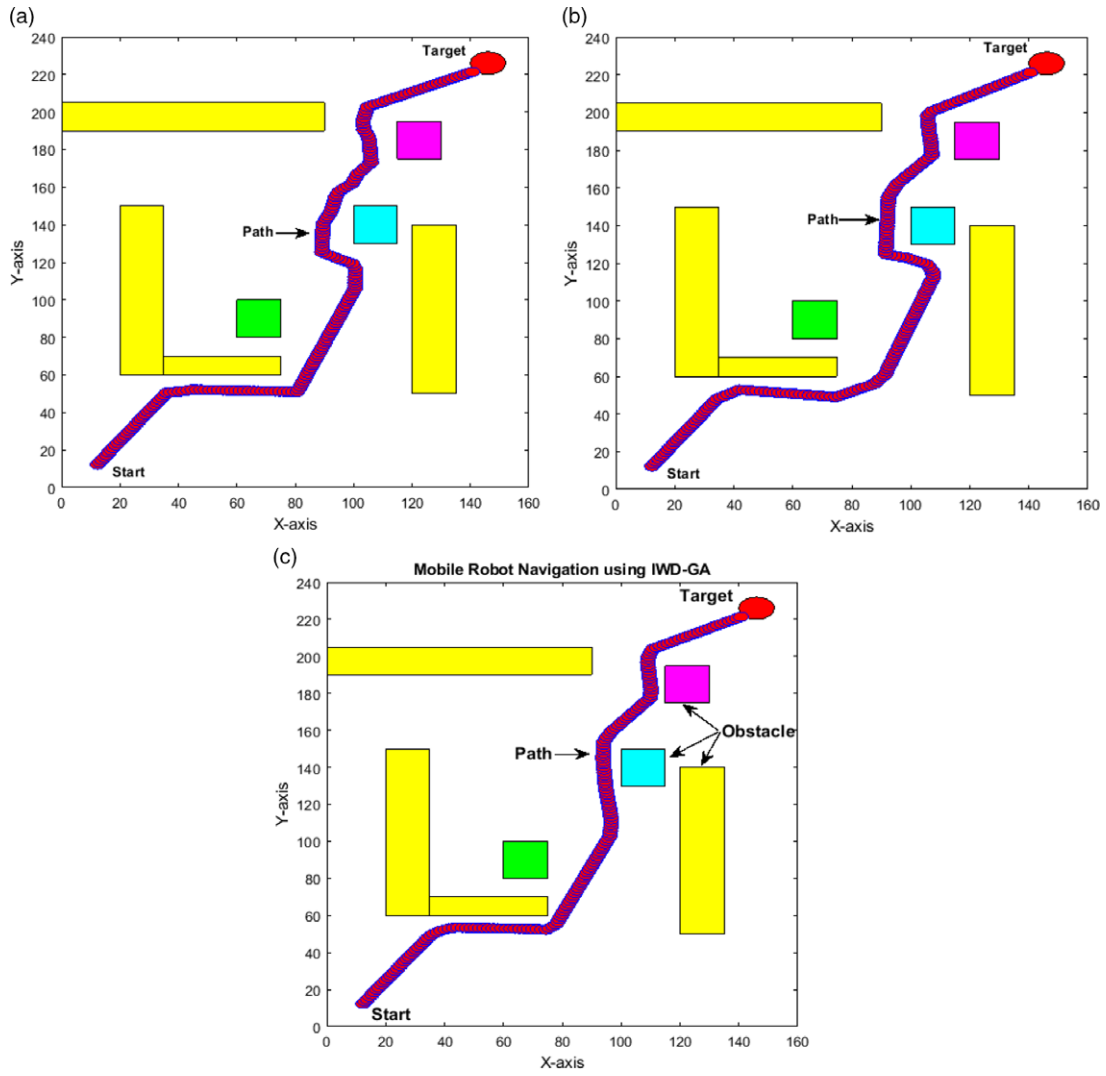

Fig. 24. Path genrated by (a) IWD, (b) GA, (c) proposed technique in scene 1.

6. Conclusions

Given the popularity of mobile robots in various places, it is required to generate the best path and control over navigation. Here, a hybrid IWD-GA is proposed for navigational control and path planning of mobile robots. Both simulation and real-time environments are considered to investigate the effectiveness and efficiency of the proposed technique. The navigation of single and multiple robots is tested in static and dynamic environments with the developed technique and the results are compared. The average deviation in both platforms is within the acceptable limit of 6%. With the

https://doi.org/10.1017/S0263574721000114 Published online by Cambridge University Press

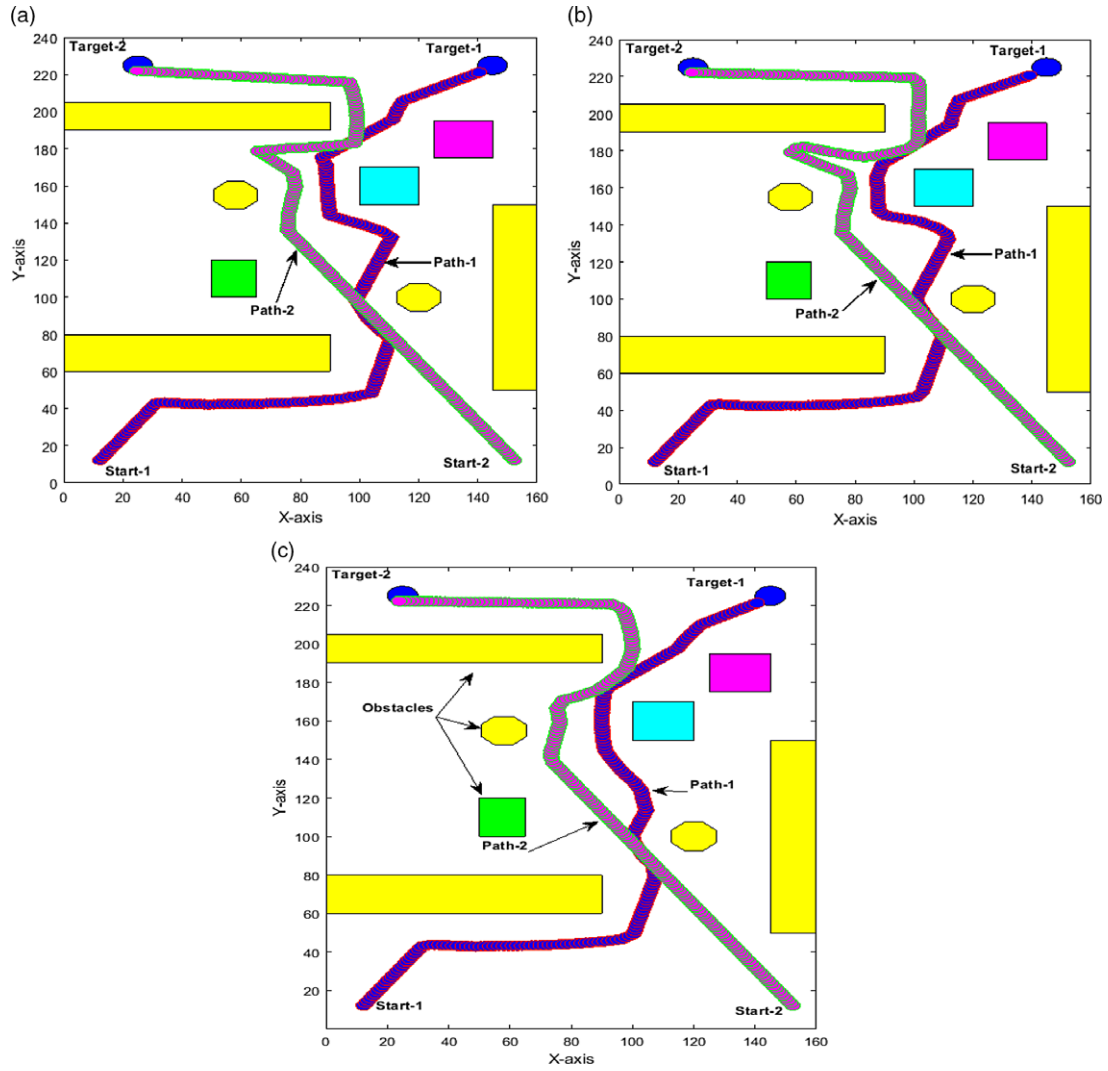

Fig. 25. Path genrated by (a) IWD, (b) GA, (c) proposed technique in scene 3.

proposed technique, the robots successfully achieved the target without any collision in both cases (single and multiple robots). In addition, the proposed method and other existing methods are compared with each other, and an average improvement of 13.14% is found. In future, this algorithm may be used in all real-time problems where autonomous movements of robots will be required, such as in industrial inspection, COVID-19 treatment at hospitals, military exercise, etc. Furthermore, this proposed technique can be used for any type of robot where path optimization is required. This work may be extended with mobile robot analysis in real dynamic environments.

References

1. S. Salmanpour, H. Omranpour and H. Motameni, "An intelligent water drops algorithm for solving robot path planning problem," *IEEE 14th International Symposium on Computational Intelligence and Informatics (CINTI)*, (2013) pp. 333–338.
2. H. Duan, S. Liu andJ. Wu, "Novel intelligent water drops optimization approach to single UCAV smooth trajectory planning," *Aerosp. Sci. Tech.* **13**(8), 442–449 (2009).
3. H. Shah-Hosseini, "An approach to continuous optimization by the intelligent water drops algorithm," *Procedia Soc. Behav. Sci.*, **32**, 224–229 (2012).
4. B. O. Alijla, L. P. Wong, C. P. Lim, A. T. Khader and M. A. Al-Betar, "An ensemble of intelligent water drop algorithms and its application to optimization problems," *Inf. Scie.* **325**, 175–189 (2015).
5. B. O. Alijla, L. P. Wong, C. P. Lim, A. T. Khader and M. A. Al-Betar, "A modified intelligent water drops algorithm and its application to optimization problems," *Expert Syst. Appl*. **41**(15), 6555–6569 (2014).

https://doi.org/10.1017/S0263574721000114 Published online by Cambridge University Press

6. I. Kaur, P. Kaur and A. Verma, "Natural terrain feature identification using integrated approach of cuckoo search and intelligent water drops algorithm," *Int. J. Comput. Sci. Inf. Sec.* **15**(2), 199 (2017).
7. H. Shah-Hosseini, "The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm," *Int. J. Bio-Inspir. Comput.* **1**(1–2), 71–79 (2009).
8. H. Shah-Hosseini, "An approach to continuous optimization by the intelligent water drops algorithm," *Procedia Soc. Behav. Sci.* **32**, 224–229 (2012).
9. S. Salmanpour, H. Monfared and Omranpour H, "Solving robot path planning problem by using a new elitist multi-objective IWD algorithm based on coefficient of variation," *Soft Com.* **21**(11), 3063–3079 (2017).
10. S. Salmanpour and H. Motameni, "Optimal path planning for mobile robot using Intelligent Water Drops algorithm," *J. Intell. Fuzzy Syst.* **27**(3), 1519–1531 (2014).
11. R. Bansal, H. Agrawal, H. Afaq and S. Saini, "Implementation of Intelligent Water Drops Algorithm to Solve Graph-Based Travelling Salesman Problem," In *Proceedings of the Second International Conference on Soft Computing for Problem Solving (SocProS 2012)*, December 28–30, (2014) pp. 137–143.
12. D. C. Rao, M. R. Kabat, P. K. Das and P. K. Jena, "Hybrid IWD-DE: a novel approach to model cooperative navigation planning for multi-robot in unknown dynamic environment," *J. of Bionic Eng.* **16**(2), 235–252 (2019).
13. X. D. Xue, B. Xu, H. L. Wang and C. P. Jiang, "The basic principle and application of ant colony optimization algorithm," In *2010 International Conference on Artificial Intelligence and Education (ICAIE)*, (2010) pp. 358–360.
14. X. Chen, P. Zhang, G. Du and F. Li, "Ant colony optimization based memetic algorithm to solve bi-objective multiple traveling salesmen problem for multi-robot systems," *IEEE Access* **6**, 21745–21757 (2018).
15. H. Ismkhan, "Effective heuristics for ant colony optimization to handle large-scale problems," *Swarm Evol. Comput.* **32**, 140–149 (2017).
16. T. K. Priyambodo and T. Suhendra, "Moving robot path planning algorithm analysis on dynamic environment based difference method update on ant colony algorithm," In *2018 4th International Conference on Science and Technology (ICST)*, (2018) pp. 1–6.
17. H. Yang, J. Qi, Y. Miao, H. Sun and J. Li, "A new robot navigation algorithm based on a double-layer ant algorithm and trajectory optimization," *IEEE Trans. Ind. Electron.* **66**(11), 8557–8566 (2018).
18. D. R. Parhi and P. K. Mohanty, "IWO-based adaptive neuro-fuzzy controller for mobile robot navigation in cluttered environments," *Int. J. Adv. Manuf. Tech.* **83**(9–12), 1607–1625 (2016).
19. B. K. Patle, D. R. K. Parhi, A. Jagadeesh and S. K. Kashyap, "Application of probability to enhance the performance of fuzzy based mobile robot navigation," *Appl. Soft Comput*. **75**, 265–283 (2019).
20. A. Pandey, R. K. Sonkar, K. K. Pandey and D. R. Parhi, "Path planning navigation of mobile robot with obstacles avoidance using fuzzy logic controller," In *2014 IEEE 8th International Conference on Intelligent Systems and Control (ISCO)*, (2014) pp. 39–41.
21. D. R. Parhi and A. Chhotray, "Development and analysis of DAYANI arc contour intelligent technique for navigation of two-wheeled mobile robot," *Ind. Rob.: Int. J.* **45**(5), 688–702 (2018).
22. A. Pandey, S. Kumar, K. K. Pandey and D. R. Parhi, "Mobile robot navigation in unknown static environments using ANFIS controller," *Perspect. Sci.* **8**, 421–423 (2016).
23. P. Tan and Z. Cai, "Modelling and planning of mobile robot navigation control in unknown environment," In *2015 International Conference on Computational Intelligence and Communication Networks (CICN)*, (2015) pp. 1532–1536.
24. G. Li and W. Chou, "Path planning for mobile robot using self-adaptive learning particle swarm optimization," *Sci. China Inf. Sci.* **61**(5), 052204 (2018).
25. B. K. Patle, D. R. K. Parhi, A. Jagadeesh and S. K. Kashyap, "Matrix-Binary Codes based Genetic Algorithm for path planning of mobile robot," *Comput. Electr. Eng.* **67**, 708–728 (2018).
26. P. B. Kumar, C. Sahu and D. R. Parhi, "A hybridized regression-adaptive ant colony optimization approach for navigation of humanoids in a cluttered environment," *Appl. Soft Comput*. **68**, 565–585 (2018).
27. M. Nazarahari, E. Khanmirza and S. Doostie, "Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm," *Expert Syst. Appl*. **115**, 106–120 (2019).
28. A. Bakdi, A. Hentout, H. Boutami, A. Maoudj, O. Hachour and B. Bouzouia, "Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control," *Rob. Auton. Syst*. **89**, 95–109 (2017).
29. C. Lamini, S. Benhlima and A. Elbekri, "Genetic algorithm based approach for autonomous mobile robot path planning," *Procedia Comput. Sci.* **127**, 180–189 (2018).
30. J. H. Liang and C. H. Lee, "Efficient collision-free path-planning of multiple mobile robots system using efficient artificial bee colony algorithm," *Adv. Eng. Softw.* **79**, 47–56 (2015).
31. A. Pandey and D. R. Parhi, "Optimum path planning of mobile robot in unknown static and dynamic environments using Fuzzy-Wind Driven Optimization algorithm," *Def. Tech*. **13**(1), 47–58 (2017).
32. H. Qu, K. Xing and T. Alexander, "An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots," *Neurocomputing* **120**, 509–517 (2013).
33. A. Tuncer and M. Yildirim, "Dynamic path planning of mobile robots with improved genetic algorithm," *Comput. Electr. Eng.* **38**(6), 1564–1572 (2012).
34. R. K. Mandava, S. Bondada and P. R. Vundavilli, "An optimized path planning for the mobile robot using potential field method and PSO algorithm," In *Soft Computing for Problem Solving*, (2019) pp. 139–150.

35. J. C. Mohanta and A. Keshari, "A knowledge based fuzzy-probabilistic roadmap method for mobile robot navigation," *Appl. Soft Comput*. **79**, 391–409 (2019).
36. A. K. Rath, D. R. Parhi, H. C. Das, M. K. Muni and P. B. Kumar, "Analysis and use of fuzzy intelligent technique for navigation of humanoid robot in obstacle prone zone," *Def. Tech*. **14**(6), 677–682 (2018).
37. A. K. Rath, D. R. Parhi, H. C. Das, P. B. Kumar, M. K. Muni and K. Salony, "Path optimization for navigation of a humanoid robot using hybridized fuzzy-genetic algorithm," *Int. J. Intell. Unmanned Syst*. **7**(3), 112–119 (2019).
38. K. K. Pandey and D. R. Parhi, "Trajectory planning and the target search by the mobile robot in an environment using a behavior-based neural network approach," *Robotica*, (2019) pp. 1–15.
39. P. Panahandeh, K. Alipour, B. Tarvirdizadeh and A. Hadi, "A self-tuning trajectory tracking controller for wheeled mobile robots," *Ind. Rob.: Int. J. Rob. Res. Appl*. **46**(6), 828–838 (2019).
40. L. Li, J. Xiao, Y. Zou and T. Zhang, "Time-optimal path tracking for robots a numerical integration-like approach combined with an iterative learning algorithm," *Ind. Rob.: Int. J. Rob. Res. Appl*. **46**(6), 763–778 (2019).
41. G. Chen and J. Liu, "Mobile robot path planning using ant colony algorithm and improved potential field method," *Comput. Intell. Neurosci*. 10 (2019), Article ID 1932812. https://doi.org/10.1155/2019/1932812.
42. F. H. Ajeil, I. K. Ibraheem, M. A. Sahib and A. J. Humaidi, "Multi-objective path planning of an autonomous mobile robot using hybrid PSO-MFB optimization algorithm," *Appl. Soft Comput.* **89**, 106076 (2020).
43. J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. (MIT Press, Cambridge, MA, 1992).
44. D. E. Goldberg, "Genetic and evolutionary algorithms come of age," Commun. ACM **37**(3), 113–120 (1994).
45. J. C. Mohanta and A. Keshari, "A knowledge based fuzzy-probabilistic roadmap method for mobile robot navigation," *Appl. Soft Comput*. **79**, 391–409 (2019).
46. E. S. Low, P. Ong and K. C. Cheah, "Solving the optimal path planning of a mobile robot using improved Q-learning," *Rob. Auton. Syst*. **115**, 143–161 (2019).
47. J. Ni, X. Yin, J. Chen and X. Li, "An improved shuffled frog leaping algorithm for robot path planning," In *2014 10th International Conference on Natural Computation (ICNC)*, (2014) pp. 545–549.
48. S. Kumar, D. R. Parhi, M. K. Muni and K. K. Pandey, "Optimal path search and control of mobile robot using hybridized sine-cosine algorithm and ant colony optimization technique," *Ind. Rob*. **47**(4), 535–545 (2020).
49. U. Orozco-Rosas, O. Montiel and R. Sepúlveda, "Mobile robot path planning using membrane evolutionary artificial potential field," *Appl. Soft Comput*. **77**, 236–251 (2019).
50. O. Montiel, U. Orozco-Rosas and R. Sepúlveda, "Path planning for mobile robots using Bacterial Potential Field for avoiding static and dynamic obstacles," *Expert Syst. Appl*. **42**(12), 5177–5191 (2015).
51. O. Montiel-Ross, R. Sepúlveda, O. Castillo and P. Melin, "Ant colony test center for planning autonomous mobile robot navigation," *Comput. Appl. Eng. Educ*. **21**(2), 214–229 (2013).
52. M. P. Garcia, O. Montiel, O. Castillo, R. Sepúlveda and P. Melin, "Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation," *Appl. Soft Comput*. **9**(3), 1102–1110 (2009).
53. O. Castillo, L. Trujillo and P. Melin, "Multiple objective genetic algorithms for path-planning optimization in autonomous mobile robots," *Soft Comput*. **11**(3), 269–279 (2007).
54. A. Pandey, V. S. Panwar, M. E. Hasan and D. R. Parhi, "V-REP-based navigation of automated wheeled robot between obstacles using PSO-tuned feedforward neural network," *J. Comput. Des. Eng*. **7**(4), 427–434 (2020).
55. A. Pandey, A. K. Kashyap, D. R. Parhi and B. K. Patle,"Autonomous mobile robot navigation between static and dynamic obstacles using multiple ANFIS architecture," *World J. Eng. Emerald* **16**(2), 275–286, 2019.
56. S. Kumar, D. R. Parhi, A. K. Kashyap and M. K. Muni, "Static and dynamic path optimization of multiple mobile robot using hybridized fuzzy logic-whale optimization algorithm," *Proceedings of the Institution of Mechanical Engineers*, Part C: Journal of Mechanical Engineering Science, 0954406220982641. https://doi.org/10.1177/0954406220982641
57. M. K. Muni, D. R. Parhi, P. B. Kumar and S. Kumar, "Motion control of multiple humanoids using a hybridized prim's algorithm-fuzzy controller," *Soft Comput*. **25**, 1159–1180 (2021).