



Obstacle Avoidance and Path Planning of a Wheeled Mobile Robot Using Hybrid Algorithm

Tahseen F. Abaas, Alaa H. Shabeeb*

Production & Metallurgy Engineering Dept., University of Technology-Iraq, Alsina'a Street, 10066 Baghdad, Iraq.

*Corresponding author Email: <mailto:alaa.h.shabeeb@uotechnology.edu.iq>

HIGHLIGHTS

- The proposed hybrid algorithm outperforms the PSO, IPSO, and GWO algorithms.
- The proposed method outperforms class PSO and GWO algorithms in determining the shortest and collision-free path for a mobile robot under the same environmental restrictions.
- The performance made the hybrid algorithm more effective in finding the best potential solution.

ARTICLE INFO

Handling editor: Omar Hassoon

Keywords:

Obstacle Avoidance; Path planning; Particle Swarm Optimization; Grey Wolf Optimization; Hybrid Algorithm.

ABSTRACT

In the mobile robot workplace, the path planning problem is crucial. Robotic systems employ intelligence algorithms to plan the robot's path from one point to another. This paper proposes the fastest and optimal path planning of the wheeled mobile robot with collision avoidance to find the optimal route during wheeled mobile robot navigation from the start point to the target point. It is done using a modern meta-heuristic hybrid algorithm called IPSOGWO by combining Improved Particle Swarm Optimization (IPSO) with Grey Wolf Optimizer (GWO). The principal idea is based on boosting the ability to exploit in PSO with the exploration ability in GWO to the better-automated alignment between local and global search capabilities towards a targeted, optimized solution. The proposed hybrid algorithm tackles two objectives: the protection of the path and the length of the path. During, Simulation tests of the route planning by the hybrid algorithm are compared with individual results PSO, IPSO, and GWO concepts about the minimum length of the path, execution time, and the minimum number of iterations required to achieve the best route. This work's effective proposed navigation algorithm was evaluated in a MATLAB environment. The simulation results indicated that the developed algorithm reduced the average path length and the average computation time, less than PSO by (1%, 1.7%), less than GWO by (1%, 1.9%), and less than IPSO by (0.05%, 0.4%), respectively. Furthermore, the superiority of the proposed algorithm was proved through comparisons with other famous path planning algorithms with different static environments.

1. Introduction

In robotics, two important issues are travel preparation (path planning) and collision avoidance, which have been studied and discussed by numerous researchers in the past three decades. The primary reason for motion planning is to discover an optimal or almost optimal track from the initial to final destination point with the potential for collision avoidance [1]. Path planning for autonomous mobile robots entails constructing a feasible path that allows all of the robots to reach their goal point without colliding. Therefore, route planning is an important aspect of designing a quick and efficient navigation procedure. Path planning algorithms are divided into two categories: One is referred to as centralized, while the other is referred to as decentralized. Centralized planning techniques of the mobile robot are used to identify the best path in the complicated environment [2]. The route planning algorithm is utilized to calculate the robot's restrictions and goal functions in this case. In the case of a decentralized method, each robot is given its configuration area for path planning. Mobile robots can deal with complex problems and harsh environments that humans find difficult to work in. It requires either a 2D or 3D workspace to move from the starting position to the target location [3].

A path is a collection of arranged points in a contiguous framework. The initial step of path planning is to seek optimal control points from the source to the destination in the workspace to ensure an efficient and feasible track suitable for real-time robot execution [4]. So, the algorithm's processing cost must be considered when considering obstacle collision and path length.

If an algorithm is computationally costly yet produces a path that is not considerably better than the competition, it loses its competitive edge. A credible algorithm should balance the amount of time it takes to run and the quality of the results it generates. Recently, researchers and scientists have developed a variety of nature-inspired algorithms [5]. The common objective of these algorithms is to improve the quality of the solutions, stability, and performance of convergence to solve and enhance the path planning of WMR. Path planning algorithms should be equipped with exploration and exploitation to accomplish this. The convergence capability of the algorithm near a good ideal solution (the most excellent solution) of the problem is referred to as exploitation, and an algorithm's capacity to identify entire areas of a problem search space is referred to as exploration. Finally, to find the best global optimum solution in the search space, all path planning algorithms strive to balance exploration and exploitation capabilities. This procedure continues for multiple generations (iterative process) until the best ecologically acceptable solutions are discovered. One way of achieving this balance is by using a hybrid algorithm that enhances performance by combining two techniques [6].

Recently, nature-inspired algorithms have been developed by researchers and scientists. These algorithms aim to increase the quality of the solutions, stability, and convergence performance to solve and improve WMR path planning. MBL Saraswathi et al. [7] developed a hybrid based on combining two cuckoo-search and bat algorithms for mobile robot path planning problems in an unknown environment. Compared to separate methods, the suggested technique takes less time to attain the aim. Muna Mohammed Jawad, and Esraa Adnan Hadi. [8] have developed the hybrid (FFCPSO) algorithm for the optimal path for mobile robots by combining the advantages of chaotic PSO (CPSO) and firefly (FF) to solve the global path-planning problem for the single and multi-robot environment. The simulation results are carried out in a MATLAB environment. The overall evaluated results indicated that the FFCPSO is a better choice than individuals' algorithms (PSO, FF). T-W Zhang et al. [9] illustrated the use of a new hybrid algorithm (GAFA) by combining both genetic (GA) and firefly (FA) algorithms of robot path planning. The best qualities of both the GA and the FF algorithm were merged to enhance finding the optimum path. In the present work, a modern hybrid meta-heuristic algorithm, IPSOGWO, is presented through improved Particle Swarm Optimization (IPSO) and Grey Wolf Optimization (GWO). It is implemented for WMR path planning and obstacle avoidance in stationary conditions. The goal of this research work is to develop a new algorithm to generate an optimal navigation path that would lead them to WMR in the safe and quickest time possible.

2. Environment Model and Obstacle Extended

At work, the environment contains many static obstacles. After detection and processing, all obstacles are surrounded and regulated by a circle shape to reduce the computation complexity and improve the system's accuracy. In this process, to ensure WMR safety when attempting mobility in the environment, the obstacle size will be increased by adding a safety distance value, as shown in Figure 1. After creating a 2D map with a start, destination point, and obstacles, it's used to construct WMR movement. WMR is represented in this map as a point by a collection of cartesian coordinate positions (x,y).

3. Fitness Function and Selection

The goal of the path planning issue is to discover the best route between a beginning point and a target point. The best road to take may be the shortest one that takes the least amount of time and uses the least amount of energy. In most path planning issues, the shortest path is considered the goal function. The x-y coordinates of the mobile robot change once it moves from one spot to the next. In this study, the objective function value for each particle/ agent used is given in Eq. (1) [10].

$$f(i) = \begin{cases} \sum \sqrt{(Rx_i - Rx_{i+1})^2 + (Ry_i - Ry_{i+1})^2} & \text{(for feasible path)} \\ \sum \sqrt{(Rx_i - Rx_{i+1})^2 + (Ry_i - Ry_{i+1})^2} + \text{Penalty} & \text{(for infeasible paths)} \end{cases} \quad (1)$$

Where; $f(i)$ is the fitness function of the summation distance between path points, Rx_i and Ry_i is the robot's current position, Rx_{i+1} and Ry_{i+1} is the robot's next position.

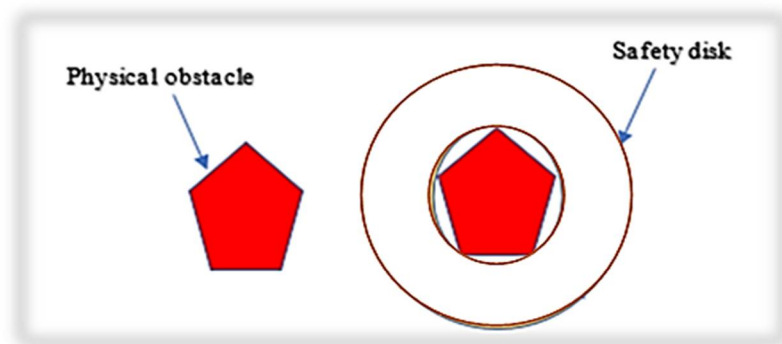


Figure 1: Increase the Size of The Obstacles

4. The Path Planning Algorithms

The presented algorithms' path planning is presented in this section and investigated. We present four swarm optimization algorithms; the first is the classical particle swarm optimization (PSO) algorithm, the second is modified of the classical PSO (IPSO) algorithm, and the third algorithm is the grey wolf optimization (GWO) algorithm. In contrast, the fourth algorithm is the proposed hybridized IPSOGWO algorithm.

4.1 Particle Swarm Optimization (PSO)

PSO is a research multipoint-based technique that mimics the social behavior of a flock of birds, a school of fish, and other animals. It was proposed by Kennedy and Eberhart in 1995 [11]. The first step is to generate the initial swarm or population, and each particle or individual (potential solution) has its velocity and location. The fitness function of each particle is then calculated. Then store each particle's best fitness value as P_{best} (best position in a particle), and store the particle's best fitness value between all particles is denoted as g_{best} (best position of all particles). Then, using Eq. (2) and Eq. (3), update the velocity and location of each particle. As a result, each particle's P_{best} and g_{best} are both updated. This procedure is repeated until the termination condition is met. Finally, g_{best} becomes the best answer for all particles [10].

$$V_i^{t+1} = w V_i^t + C_1 r_1 (X_{pbest,i} - x_i^t) + C_2 \times r_2 \times (X_{gbest} - x_i^t) \quad (2)$$

$$x_i^{t+1} = x_i^t + V_i^{t+1} \quad (3)$$

Where w is the inertia weight factor that specifies the rate at which a particle's prior velocity at the current iteration (t) contributes to its velocity at the current iteration $t+1$. V_i^t : Represent the speed of particle i at time step t . x_i^t : Represent position of particle i at time step t . C_1 is the individual knowledge and, C_2 is the collective knowledge. r_1 and r_2 are random numbers [0-1]. w value is a constant value in classical particle swarm optimization (PSO). In the current work. We proposed the linear decreasing approach of the inertia weight value as a mechanism to improve the performance of the classical PSO through control of the exploration and exploitation of the swarm, Eq. (4) shows how the w new value is updated based on linear decreasing inertia weight (LDIW).

$$w_{new} = \left(\frac{w_{max} - w_{min}}{iter_{max}} \right) iter + w_{max} \quad (4)$$

Where: w_{max} : the larger value of the inertia weight. w_{min} : the lower value of the inertia weight, $iter$: current iteration. For a balance between global and local search exploration, a small value of w aids the algorithm in local search. In contrast, a large value of w aids individual members of the swarm in exploring the issue space more readily and improving global search. Figure.2 shows a flowchart of the PSO algorithm used in this study.

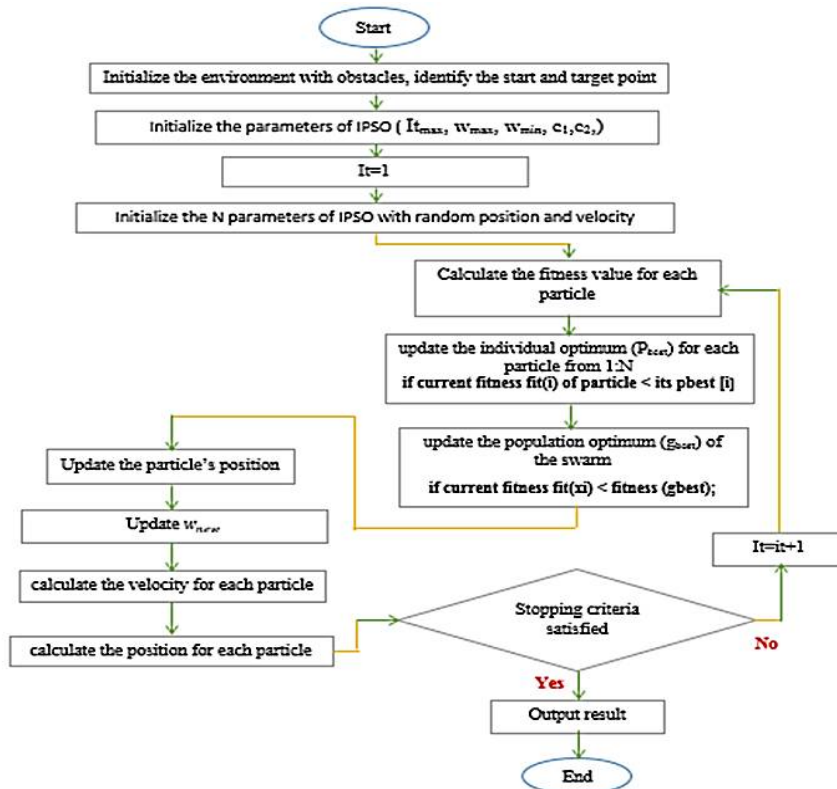


Figure 2: Flowchart of Path Planning based on IPSO Algorithm [10]

4.2 Grey Wolf Optimization (GWO)

The GWO method, proposed by Mirjalili et al., is a newcomer in the field of nature-inspired optimization algorithms. Grey wolves' hunting tactics and social order are used to create the Grey Wolf Optimizer (GWO). Grey wolves are divided into four groups: alpha, beta, delta, and omega wolves, according to the hierarchy. The alpha wolf is the group's leader or dominant wolf, and alpha wolves follow the other wolves in the pack. The alpha is the best wolf in terms of managing the group. Beta wolf is the second in the wolf group's social structure. Beta assists the alpha wolf in a variety of duties. The delta wolf must subordinate to the alpha and beta wolves, but the omega wolves are judged by the delta wolf [12].

Grey wolves' collective hunting method is another fascinating social characteristic. The grey wolves use the approach first to detect the location of prey and then encircle it under the guidance of the alpha wolf. In a mathematical grey wolf hunting strategy model, the alpha, beta, and delta wolves are assumed to have a greater understanding of probable prey locations. Consequently, the GWO algorithm updates the locations of wolves using the first three best solutions (alpha, beta, and delta). Figure. 3 shows a flowchart of the GWO algorithm. In the GWO code, there are no omega wolves. The following is a mathematical model of the grey wolf hunting mechanism [13]:

$$\vec{U}_\alpha = |\vec{C}_\alpha \cdot \vec{X}_\alpha - \vec{X}_t| ; \vec{U}_\beta = |\vec{C}_\beta \cdot \vec{X}_\beta - \vec{X}_t| ; \vec{U}_\delta = |\vec{C}_\delta \cdot \vec{X}_\delta - \vec{X}_t| \quad (5)$$

$$\vec{P}_\alpha = |\vec{X}_\alpha - \vec{A}_\alpha \cdot \vec{U}_\alpha| ; \vec{P}_\beta = |\vec{X}_\beta - \vec{A}_\beta \cdot \vec{U}_\beta| ; \vec{P}_\delta = |\vec{X}_\delta - \vec{A}_\delta \cdot \vec{U}_\delta| \quad (6)$$

$$\vec{X}_{(t+1)} = \frac{\vec{P}_\alpha + \vec{P}_\beta + \vec{P}_\delta}{3} \quad (7)$$

\vec{U}_α , \vec{U}_β , and \vec{U}_δ are the distance vectors between the prey and the wolf (alpha, beta, delta), \vec{X}_α , \vec{X}_β , and \vec{X}_δ represent the location vector of the prey For alpha, beta, and delta wolves. \vec{X} represents the grey wolf's location vector at t+1 iteration, the coefficient vectors of alpha, beta, and delta wolves are denoted by \vec{C}_α , \vec{C}_β , \vec{C}_δ , \vec{A}_α , \vec{A}_β and \vec{A}_δ respectively. The trial vector for an alpha, beta, and delta wolves is denoted by the letters \vec{P}_α , \vec{P}_β , and \vec{P}_δ . For alpha, beta, and delta wolves, the coefficient vectors are determined as follows:

$$\vec{A}_i = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \quad (8)$$

$$\vec{C}_i = 2 \cdot \vec{r}_2 \quad (9)$$

Where i is α , β , and δ . a denotes a linear reduction in the vector from 2 to 0 during optimization, the initial random vector in [0,1] is denoted as r. Members of the grey wolf pack change their locations based on alpha, beta, delta, and delta wolves, as well as prey. The grey wolves catch their victim and then attack it to conclude the hunt. This situation is defined as a decreasing vector in the mathematical model given below:

$$\vec{a} = 2 - \left(\frac{2 \cdot iter}{iter_{max}} \right) \quad (10)$$

4.3 Proposed Hybrid IPSOGWO Algorithm

Several techniques are established to improve the performance of the algorithms for enhanced exploration and exploitation behavior during iterations. The hybrid IPSOGWO is suggested in this paper, which combines Particle Swarm Optimization with Grey Wolf Optimization. The fundamental concept is to combine IPSO's exploitation capacity with GWO's exploration ability to strike a balance between exploration (global search) and exploitation (local search) to prevent large local optima. [14]. In our Hybrid IPSOGWO algorithm, the capability of exploitation is improved in IPSO using the capability of exploration in GWO. Additionally, the capability of exploration is improved in GWO using the capability of exploitation in IPSO. Integrating the performance results made the hybrid algorithm more effective in finding the best potential solution. The velocity and updated equation are produced as follows in the suggested approach to hybridize PSO and GWO variants:

$$V_i^{t+1} = w_{new} V_i^t + C_1 r_1 (X_1 - x_i^t) + C_2 \times r_2 \times (X_2 - x_i^t) + C_3 \times r_3 \times (X_3 - x_i^t) \quad (11)$$

The basic procedure of the hybrid algorithm used in this study is shown in Table 1.

5 Results and Simulation Analysis

5.1 Setting and Environment

Two environments are selected to validate the proposed approach. In all cases, the map dimensions are (2400 X 2400) mm, number of handle points is equal to 3 (via points). The mapping layout contained different objects, regular and irregular. These environments have static obstacles located at random positions and are unknown in the workspace environment. There is no prior information on the initial WMR location, obstacles, and target location. WMR environment information is defined based

on assuming using an external sensor (overhead Camera) to build the environment simulation as input to simulated, and path find depending on intelligent algorithms (PSO, IPSO, GWO, and hybrid PSOGWO). In the mass point simulation, WMR is considered a dimensionless point. Table 2 shows the parameter settings of algorithms parameters used in the simulation.

Table 1: Basic Procedure for Hybrid IPSOGWO algorithm.

1-Identify environment, Initial positions, and Goal position. 2- Initialize IPSO and GWO: set parameters to initialize the IPSO (population size of particles, Itmax ,r1,r2,r3, c1, c2, c3, Wmin, Wmax), set parameters to initialize GWO (population size of grey wolves, C, a, A). 3- Output: Way-points.
<pre> For each particle i (i=1 to N) do Position and velocity initialized by random End for iter ← 1 while (iter < iter_{max}) do for each particle i (i=1 to N) Fitness Function Assessment $f_{(i)}$ for each particle i. If current fitness $fit(i) < its\ pbest[i]$; Replace $pbest[i]$ with $fit[i]$ If current fitness $fit(xi) < fitness\ (gbest)$; Replace $gbest[i]$ with $fit[i]$ Update w_{new}, velocity V_i^{t+1} and position X_i^{t+1} based on Eq. (4), Eq. (11) & Eq. (3). Comparison of fitness value of each particle and choose the best three particles solution End for Given the updated positions to GWO. for each agent i (i=1 to N) Evaluation of the fitness function $f_{(final)}$ for each agent. If $wolf < \alpha$ wolf; Update α wolf. If $wolf < \beta$ wolf; Update β wolf. If $wolf < \delta$ wolf; Update δ wolf. Update location of the present search agent using Eq. (7). End for Update parameters A, a, and C Calculate the fitness of all search agents. Update $\vec{X}_\alpha = \vec{X}_1, \vec{X}_\beta = \vec{X}_2$, and $\vec{X}_\delta = \vec{X}_3$. Iter ← iter+1 End while </pre>

5.2 Simulation Results

The comparison is made with the proposed hybrid IPSO-GWO and individual algorithms in terms of best path length with an interval of 100 iterations. The best path length refers to the optimized path length. Therefor, each environment has four paths for safe paths as the first step. The distances of these four paths between the start point to the target point are determined using Eq. 1. A suitable algorithm can be chosen depending on concerning the priority of criteria to be applied to the path planning algorithm system. So, another comparison is made between the results for the proposed hybrid IPSO-GWO with individual algorithms (PSO, IPSO, and GWO) after executing the program 10 times (1000 iterations) to analyze the performance results of the proposed hybrid IPSOGWO with individual algorithms (PSO, IPSO, and GWO). The performance comparison of the presented methods was measured based on several iterations, average path length, standard division, and improved ratio (reduce in length path).

Case 1:

Case study one contains four obstacles, as shown in Figure 4. First, the start point of WMR (red circle) and the target point (blue circle) are placed at (95.43, 1578.65) mm, and (2065.43, 391.25) mm, respectively, as shown in Figure 2.

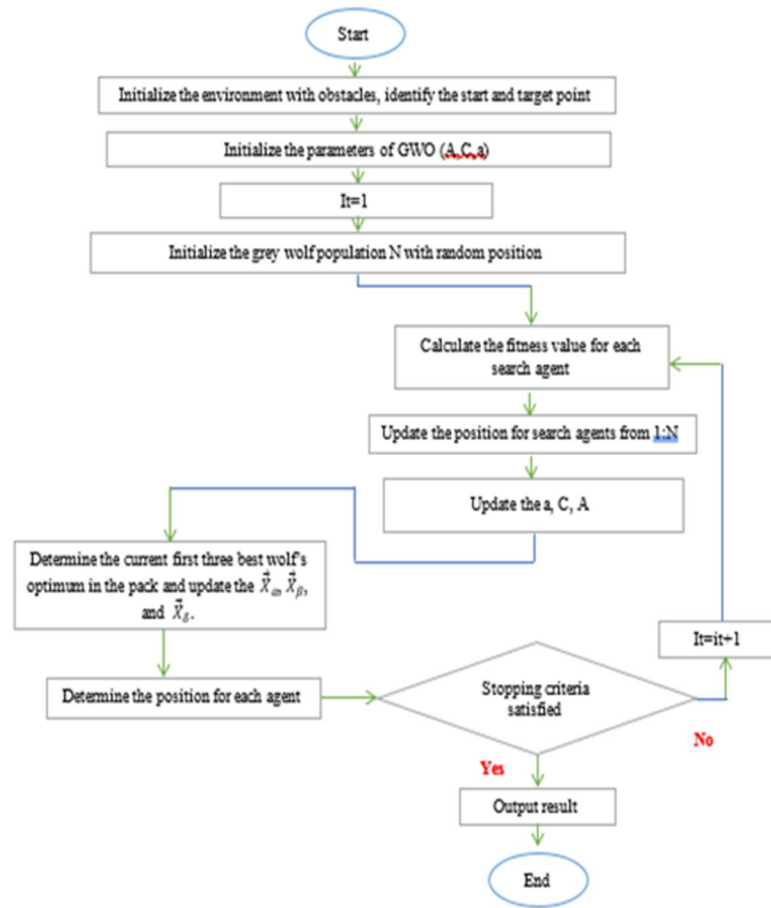


Figure 3: Flowchart of Path Planning based on GWO Algorithm [13].

Table 2: Parameters Setting of Algorithms.

Method	Parameters	Value
PSO and IPSO	Maximum iteration (it_{max})	100
	Number of particles (N)	40
	Acceleration Constant (c_1, c_2)	1.5, 2
	Constant Inertia Weight Factor (w)	0.9
	Min. Inertia Weight Factor (w_{min})	0.4
	Max. Inertia Weight Factor (w_{max})	0.9
	Random Values (r_1 & r_2)	0-1
GWO	Maximum iteration	100
	Number of agents (N)	40
Hybrid	Maximum iteration ($_{max}$)	100
	Number of particles (N)	40
PSO-GWO	Acceleration Constant (c_1, c_2, c_3)	1.5, 2, 1.5
	Min. Inertia Weight Factor (w_{min})	0.4
	Max. Inertia Weight Factor (w_{max})	0.9
	Random Values (r_1 & r_2 & r_3)	0-1

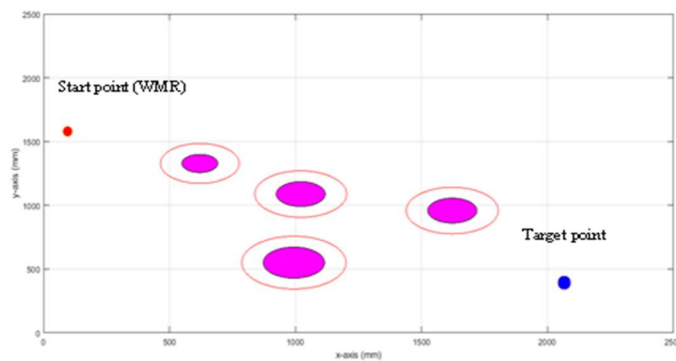


Figure 4: The Environment Case 1

The simulation results of the optimum route for the first case based on PSO, IPSO, GWO, and hybrid IPSOGWO algorithms are shown in Figures 5a – d, respectively. The black line path is the best path between the four optimal paths achieved by each algorithm in this work. The other colors lines paths are the second, third and fourth best optimal paths from 10 optimal paths (1000 iterations) achieved by each algorithm. In contrast, Figures 6a-d plotted the relation of the path length and the number of iterations to estimate the optimal value of the objective function for the presented algorithms. The output control point presented algorithms for this environment are tabulated in Table 3.

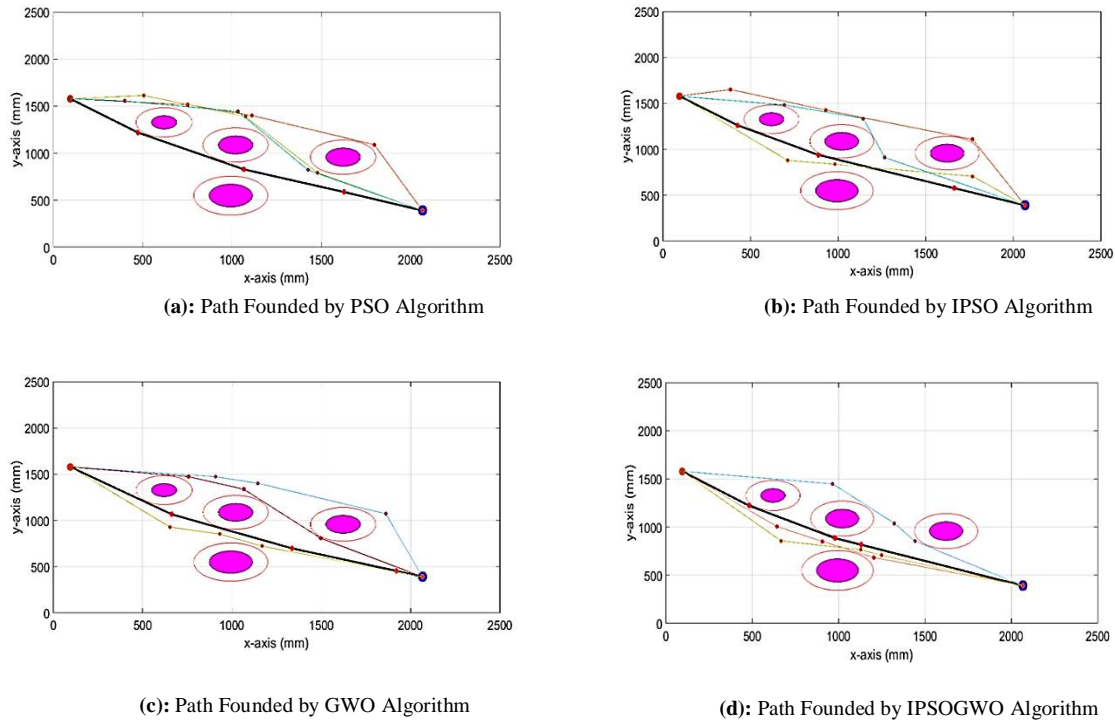


Figure 5: Path Found by Presented Algorithms for Case 1

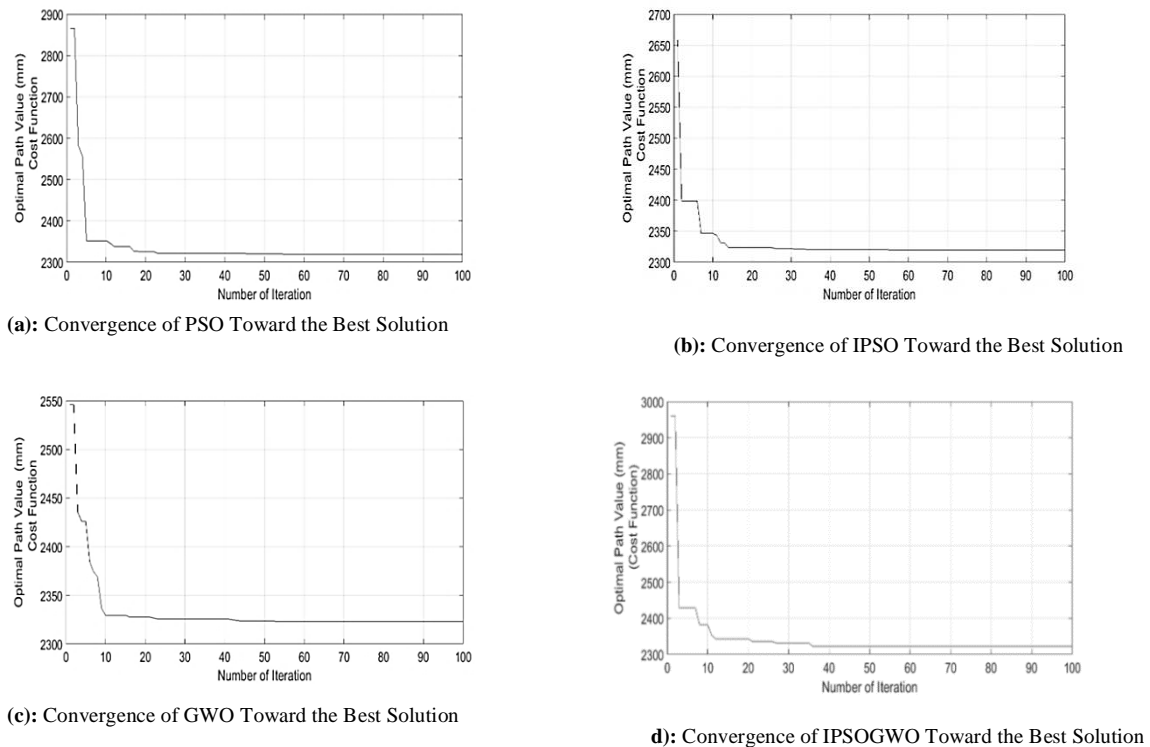


Figure 6: Best Distance function Founded by Presented Algorithms for Case 1

Table 3: Via Points Coordination for Case 1

Type of Intelligent Algorithm	P1		P2		P3	
	X (mm)	Y (mm)	X (mm)	Y (mm)	X (mm)	Y (mm)
PSO	474.73	1218.97	1067.81	827.81	1626.40	588.25
IPSO	4262.19	1262.46	886.87	939.79	1662.82	577.61
GWO	663.13	1066.29	1337.05	698.00	1919.47	455.27
IPSOGWO	483.00	1221.64	980.34	884.34	1129.54	817.37

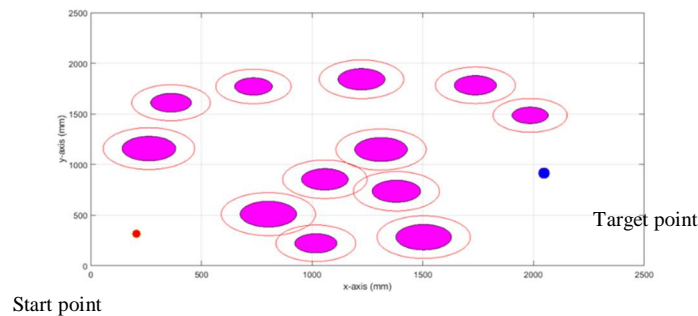
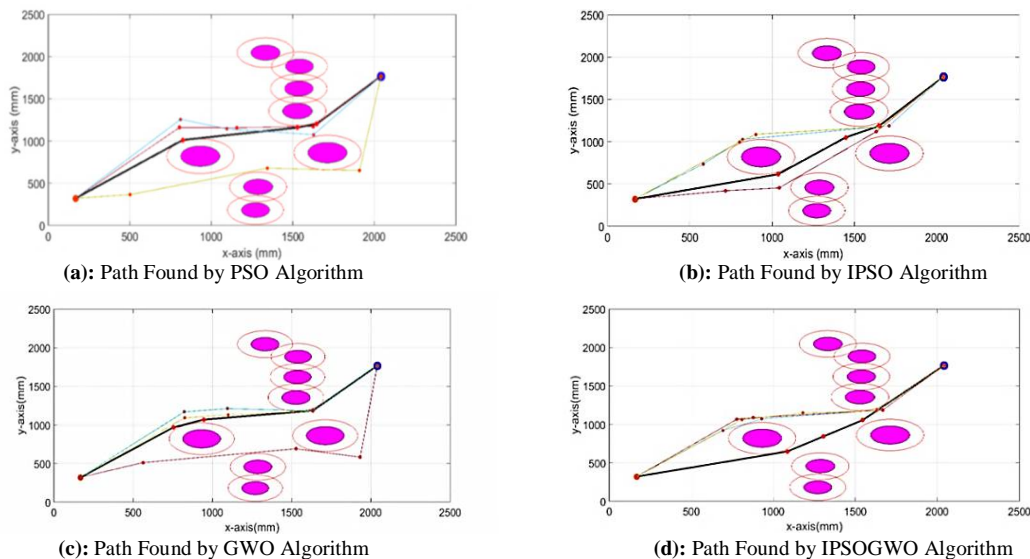
In the simulation results of case one, for the PSO algorithm, the shortest path is equal to 2321.99 mm with an average computation time of 8.38 sec, obtained in iteration no. 55, which has higher fitness. For IPSO, iteration no. 34 has higher fitness, with the shortest path equal to 2319.83 mm with an average computation time of 8.25 sec. The best path of the GWO algorithm is equal to 2322.87 mm with an average computation time of 8.41 sec. It is obtained in iteration no. 52, which has higher fitness. For Hybrid IPSOGWO, the best path is obtained in iteration no. 36, and equal to 2319.66 mm with an average computation time of 8.13 sec., which achieved the optimal solution (better fitness).

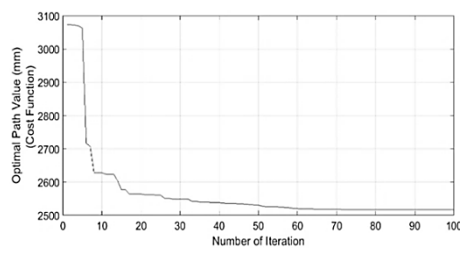
Case 2:

Case study two contains eight obstacles. The start point of WMR is the red circle, and the goal point (blue circle) is placed at (168.46, 322.19) mm, and (2039.59, 1766.56) mm, respectively, as shown in Figure 7. The simulation results of the optimum route for the first case study based on PSO, IPSO, GWO, and PSOGWO algorithms are shown in Figures 8a – d, respectively. In this case, the black line is the optimal path between the four best paths achieved by each algorithm.

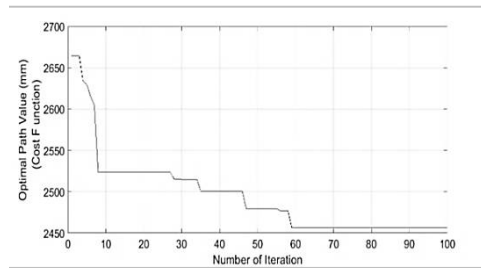
Table 4: Via Points Coordination for Case 2

Type of Intelligent Algorithm	P1		P2		P3	
	X (mm)	Y (mm)	X (mm)	Y (mm)	X (mm)	Y (mm)
PSO	825.62	1012.69	1527.59	1159.07	1646.08	1202.30
IPSO	804.03	1059.41	1157.46	1157.67	1625.67	1174.42
GWO	755.95	969.44	946.38	1069.52	1634.36	1186.10
IPSOGWO	1084.68	647.77	1304.80	843.03	1544.98	1056.31

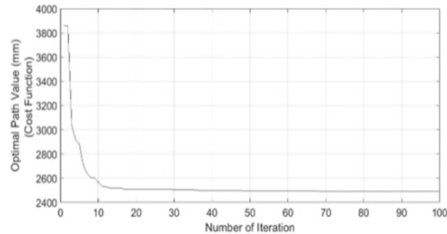

Figure 7: The Environment Case 2

Figure 8: Path Found by Presented Algorithms for Case 2



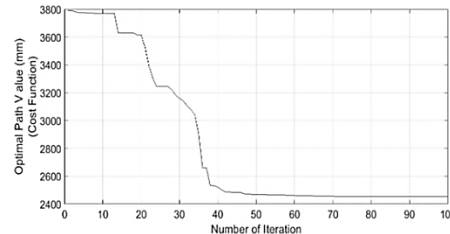
(a): Convergence of PSO Toward the Best Solution



(b): Convergence of IPSO Toward the Best Solution



(c): Convergence of GWO Toward the Best Solution



(d): Convergence of IPSOGWO Toward the Best Solution

Figure 9: Best Distance Function Found by Presented Algorithms for Case 2**Table 5:** Comparison of Algorithms performance for Case One

Type of intelligent Algorithm	Shortest Path in (mm)	Iteration of the best value	longest Path in (mm)	Average Computation Time in (sec)	Average Path in (mm)	Standard Deviation
PSO	2321.99	55	2451.77	8.38	2389.647	49.94448
IPSO	2319.83	34	2424.21	8.25	2349.179	46.24322
GWO	2322.87	52	2489.78	8.41	2398.388	57.16317
IPSOGWO	2319.66	36	2406.8	8.13	2338.446	26.54067

Figure 9a-d shows the relation between the path length and the number of iterations to estimate the optimal value of the objective function for the presented algorithm.

The output control point presented algorithms for this environment are tabulated in Table 4. In the simulation results of case two, for the PSO algorithm, the shortest path is equal to 2484.43 mm with an average computation time of 8.47sec, obtained in iteration no. 72, which has higher fitness. For IPSO, iteration no. 69 has higher fitness, with the shortest path equal to 2456.24 mm with an average computation time of 8.33 sec. The best path of the GWO algorithm is equal to 2493.03 mm with an average computation time of 8.48 sec., obtained in iteration no. 78, which has higher fitness. For Hybrid IPSOGWO, the best path is obtained in iteration no. 71, and equal to 2453.02 mm with an average computation time of 8.26 sec., which achieved the optimal solution (better fitness).

5.3 Performance Evaluation

The presented in this research work hybrid IPSOGWO algorithm generates a safe and short path for WMR in collide environments that contain multiple obstacles. Assessment is achieved in terms of the number of iterations, computation time, average path length, and standard division. Based on three criteria: (solution quality, stability, and convergence speed), These algorithms' performance was compared. The average fitness value is used to measure the solution quality. In work, it is defined as the average of ten optimal fitness values obtained from ten trials. The better the solution quality, the lower the average optimum fitness value. The standard deviation determines the algorithm's stability. The method is more stable if the standard deviation is smaller. Finally, the number of iterations necessary for the algorithm to converge to the optimal or suboptimal solution determines the convergence speed of the algorithm. The faster the method converges to the best solution with the least number of iterations given for a maximum number of iterations, the higher the convergence speed.

The results of the best performance of the presented algorithms for each case study are summarized in Tables 5 and 6 after executing the running of the proposed algorithm and individual algorithms 10 times (1000 iterations).

Table 6: Comparison of Algorithms performance for Case two

Type of Intelligent Algorithm	Shortest Path in (mm)	Iteration of the best value	longest Path in (mm)	Average Computation Time in (sec)	Average Path in (mm)	Standard Deviation
PSO	2484.43	72	2535.87	8.47	2504.406	33.45072
IPSO	2456.24	69	2527.55	8.33	2467.848	28.38103
GWO	2493.03	78	2591.31	8.48	2531.234	33.83359
IPSOGWO	2453.02	71	2496.6	8.26	2466.703	14.54526

Table 7: Results Comparison with Reference [8]

Figure no.	Method	Navigation path length (cm)
Fig (5-41, (a)).	PSO [8]	1148.8
Fig (5-41, (a)).	FF [8]	1158.4
Fig (5-41, (a)).	FF-CPSO [8]	1147.5
Fig (5-41, (b)).	IPSOGWO	1147.1

As may be seen, although the four algorithms succeed in generating a collision-avoidance path, hybrid IPSOGWO outperforms better on another presented algorithm. Where the convergence speeds of path planning for both PSO, IPSO, and GWO are slower compared with PSOGWO. The simulation results indicated that the developed algorithm reduced the average path length and the average computation time, less than PSO by (1%, 1.7%), less than GWO by (1%, 1.9%), and less than IPSO by (0.05%, 0.4%), respectively. Moreover, Path planners based on PSOGWO have higher solution quality than those based on PSO, GWO, and IPSO. IPSOGWO-based path planners have the least average fitness value, number of iterations, execution duration, and standard deviation, as shown in the statistical data tables to the comparison of algorithms performance, demonstrating great searchability, simplicity, rapid convergence, and robustness of IPSOGWO. Also, the IPSOGWO algorithm ensures getting on the shortest paths in minimum execution time. The IPSOGWO has great searchability, simplicity, and robustness based on the above. In other words, the performance of the hybrid algorithm is better than the other three tested algorithms.

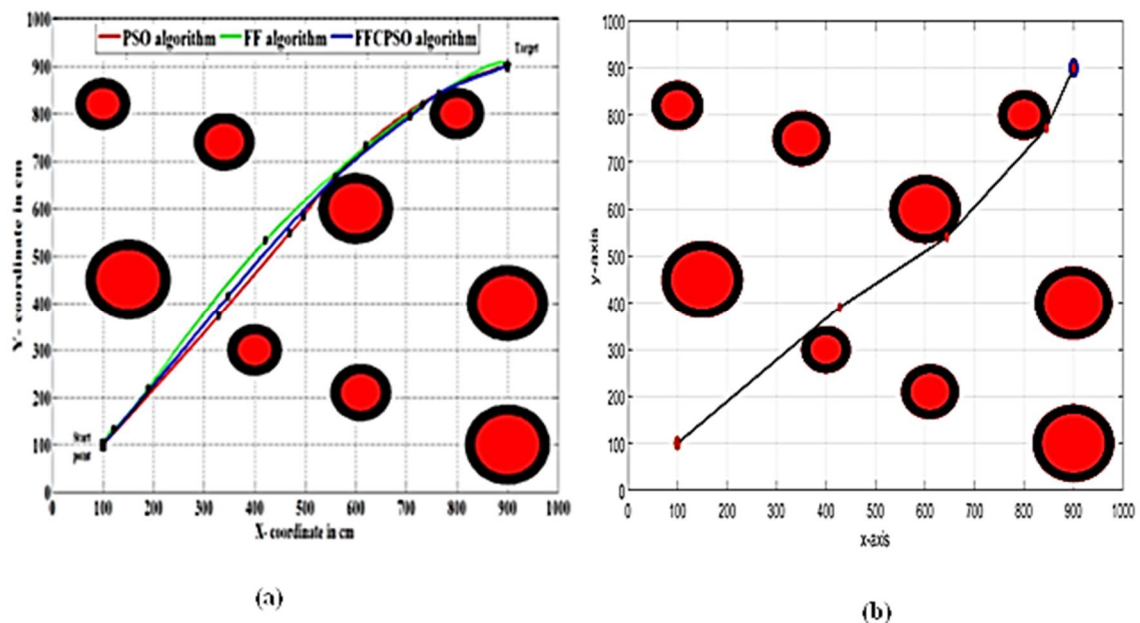
5.4 Evaluation of the Hybrid IPSOGWO

Executing the simulation process of any algorithm technique in different environments is not sufficient to assert that it is best. Instead, it should provide some proof compared with previously applied strategies to ensure that the proposed technique is better. From this point, the proposed algorithm was compared with other algorithms for the same tested cases to determine the response in the selected environment. The environment has been generated as stated by other authors, and the IPSOGWO algorithm has been applied to similar environments to demonstrate the efficiency of the proposed algorithm.

1. The first comparison investigation was conducted based on reference [8]. The navigation system was developed based on the three approaches, namely PSO, Firefly (FF), and developed hybrid FFCPSO. Figure 10a shows the simulation results obtained from the stated navigation methods. In contrast, the result obtained from using the developed algorithm is illustrated in Figure 10b. Table 7 summarizes the best path length, which can be achieved between the developed IPSOGWO algorithm and the reference algorithms [8].

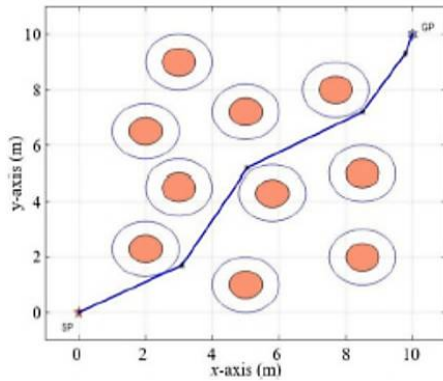
Table 8: Results Comparison with Reference [15]

Figure no.	Method	Navigation path length (cm)
Fig (5-43, (a)).	MFB [15]	15.073
Fig (5-43, (b)).	PSO-MFB [15]	14.798
Fig (5-43, (c)).	IPSOGWO	14.49

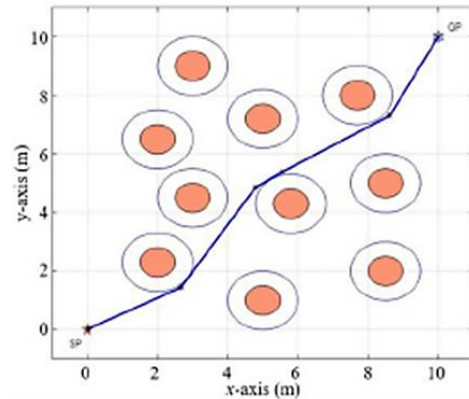
**Figure 10:** The Best Results Achieved by (a) Reference [8] Case A, and (b) IPSOGWO.

2. The fourth comparison study was achieved with reference [15]. The simulation results obtained using MFB and PSO-MFB algorithms are demonstrated in Figures 11a and 11b, respectively. In contrast, the result obtained from using the IPSOGWO is shown in Figure 11c. Finally, Table 8 summarizes the best path length, which can be achieved between the developed IPSOGWO algorithm and the reference algorithms [15].

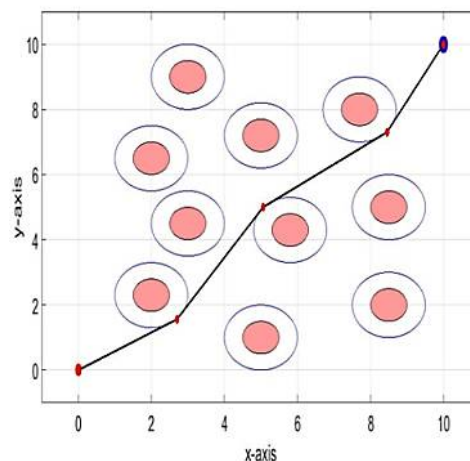
From the results obtained, it can be noticed that the developed algorithm (IPSOGWO) proved superior compared with algorithms applied by other researchers. Moreover, in simulation mode, the results proved that the proposed algorithm is effective in generating the optimal and shortest path, which is positively reflected in the promising applications of this proposed algorithm.



(a) The Best Path Achieved using the MFB Algorithm by [15] for Experiment 1



(b) The Best Path Achieved using the Hybridized PSO-MFB Algorithm by [15] for Experiment 1



(c) The Best Path Achieved using the IPSO-GWO Algorithm

Figure 11: The Best Results Achieved by (a-b) Reference [15] and (c) IPSOGWO

6 Conclusion

This work suggested path planning and obstacle avoidance based on the modern hybrid path planner algorithm. The IPSO and GWO algorithms are integrated to produce a plan optimal collision-free path. From the analysis of the results, we can conclude that the hybrid algorithm outperforms the PSO, IPSO, and GWO in:

- Generate the shortest path and fastest
- Reduce the number of iterations.
- Reduce time execution for generation safe and shortest path.

The capability of exploitation is improved in IPSO using the capability of exploration in GWO. In addition, the capability of exploration is improved in GWO using the capability of exploitation in IPSO. Integrating the performance results made the hybrid algorithm more effective in finding the best potential solution. In future works, we will apply our work through the experiment work of the proposed path planning algorithm in the physical world.

Author contribution

All authors contributed equally to this work.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data availability statement

The data that support the findings of this study are available on request from the corresponding author.

Conflicts of interest

The authors declare that there is no conflict of interest.

References

- [1] M. J. Reddy ,D. N. Kumar, Evolutionary algorithms, swarm intelligence methods, and their applications in water resources engineering: a state-of-the-art review, *H2Open J.*, 3 (2021) 135-188. <https://doi.org/10.2166/h2oj.2020.128>
- [2] A. S. Al-Araji, K. A. Attarid, M. K. Hamzah, Development of a Path Planning Algorithms and Controller Design for Mobile Robot, *Int. Conf. Electr. IEEE*, 2018, 72-77. <https://doi.org/10.1109/SCEE.2018.8684211>
- [3] B. B. Ayawli, R. Chellali, A. Y. Appiah, F. Kyeremeh. An overview of nature-inspired, conventional, and hybrid methods of autonomous vehicle path planning. *J. Adv. Transp.*, 2018. <https://doi.org/10.1155/2018/8269698>
- [4] S. Hosseinienejad , C. Dadkhah, Mobile robot path planning in dynamic environment based on cuckoo optimization algorithm, *Int. J. Adv. Rob. Syst.*, 16 (2019) 1–13. <https://doi.org/10.1177/1729881419839575>
- [5] H. Kong, J Sun, J Hu, Real-time Motion Planning Based on Layered Cost Map for AGV Navigation. In *Chinese Automation Congress (CAC)*, IEEE., 2020, 7624-7628. <https://doi.org/10.1109/CAC51589.2020.9327401>
- [6] S. Arora, H. Singh, M. Sharma, S. Sharma, P. Anand, A new hybrid algorithm based on grey wolf optimization and crow search algorithm for unconstrained function optimization and feature selection, *IEEE Access*, 7 (2019) 26343-26361. <https://doi.org/10.1109/ACCESS.2019.2897325>
- [7] M. Saraswathi, G. Bala Murali, B. B. V. L. Deepak, Optimal path planning of mobile robot using a hybrid cuckoo search-bat algorithm, *Procedia Comput. Sci.*, 133 (2018) 510-517. <https://doi.org/10.1016/j.procs.2018.07.064>
- [8] M. M. Jawad, E. A. Hadi, A Comparative study of various intelligent algorithms based path planning for Mobile Robots, *J. Eng.*, 25 (2019) 83-100. <https://doi.org/10.31026/j.eng.2019.06.07>
- [9] T-W. Zhang, G-H. Xu, X-S. Zhan, T. Han, A new hybrid algorithm for path planning of mobile robot, *J. Supercomput.*, 78 (2022) 4158–4181. <https://doi.org/10.1007/s11227-021-04031-9>
- [10] T. F. Abaas, A. H. Shabeeb, Path Planning Optimization of a Mobile Robot based on Intelligence Algorithm, *IOP Conf. Ser.: Mater. Sci. Eng.*, 881, 2020, 012047. <https://doi.org/10.1088/1757-899X/881/1/012047>
- [11] Blum, C., Groß, R. 2015. Swarm intelligence in optimization and robotics, In: Kacprzyk, J., Pedrycz, W. (eds) *Springer Handbook of Computational Intelligence*. Springer Handbooks, pp 1291–1309. Springer Handbooks. Springer, Berlin, Heidelberg.. https://doi.org/10.1007/978-3-662-43505-2_66
- [12] M. H. Nadimi-Shahraki, S Taghian, S. Mirjalili, An improved grey wolf optimizer for solving engineering problems, *Expert Syst. Appl.*, 166 (2021) 113917. <https://doi.org/10.1016/j.eswa.2020.113917>
- [13] T. F. Abaas , A. H. Shabeeb, Safe and Optimum Navigation of Wheeled Mobile Robot using Grey Wolf Optimization Algorithm, *IOP Conf. Ser. Mater. Sci. Eng.* 928, 2020, 022006. <https://doi.org/10.1088/1757-899X/928/2/022006>
- [14] F. Gul, W. Rahiman, S. S. Alhady, A. Ali, I. Mir, A. Jalil, Meta-heuristic approach for solving multi-objective path planning for autonomous guided robot using PSO–GWO optimization algorithm with evolutionary programming, *J. Ambient Intell. Hum. Comput.*, 12 (2021) 7873-7890. <https://doi.org/10.1007/s12652-020-02514-w>
- [15] F. H. Ajeil, I. K. Ibraheem, A. T. Azar, A. J. Humaidi, Autonomous navigation and obstacle avoidance of an omnidirectional mobile robot using swarm optimization and sensors deployment, *Int. J. Adv. Robot. Syst.*, 17 (2020) 1729881420929498. <https://doi.org/10.1177/1729881420929498>