

# Autonomous navigation and obstacle avoidance of an omnidirectional mobile robot using swarm optimization and sensors deployment

Fatin Hassan Ajeil<sup>1</sup>, Ibraheem Kasim Ibraheem<sup>1</sup> ,  
Ahmad Taher Azar<sup>2,3</sup> and Amjad J Humaidi<sup>4</sup>

## Abstract

The present work deals with the design of intelligent path planning algorithms for a mobile robot in static and dynamic environments based on swarm intelligence optimization. Two modifications are suggested to improve the searching process of the standard bat algorithm with the result of two novel algorithms. The first algorithm is a Modified Frequency Bat algorithm, and the second is a hybridization between the Particle Swarm Optimization with the Modified Frequency Bat algorithm, namely, the Hybrid Particle Swarm Optimization-Modified Frequency Bat algorithm. Both Modified Frequency Bat and Hybrid Particle Swarm Optimization-Modified Frequency Bat algorithms have been integrated with a proposed technique for obstacle detection and avoidance and are applied to different static and dynamic environments using free-space modeling. Moreover, a new procedure is proposed to convert the infeasible solutions suggested via path the proposed swarm-inspired optimization-based path planning algorithm into feasible ones. The simulations are run in MATLAB environment to test the validation of the suggested algorithms. They have shown that the proposed path planning algorithms result in superior performance by finding the shortest and smoothest collision-free path under various static and dynamic scenarios.

## Keywords

Path planning, obstacle avoidance, bat swarm optimization, particle swarm optimization, sensors, mobile robot

Date received: 21 September 2019; accepted: 4 May 2020

Topic: Mobile Robots and Multi-Robot Systems

Topic Editor: Antonio Fernandez-Caballero

Associate Editor: Oleg Sergiyenko

## Introduction

Path planning is an essential topic in the robotics field due to the popularity of mobile robots in different applications such as military, industry, libraries, and security. It determines how safely the mobile robot reaches its goal position (GP) taken into account several criteria such as shortest distance and minimum energy. Several methodologies have been suggested to solve path planning problem, among them the classical approaches such as, mathematical programming,<sup>1</sup> cell decomposition,<sup>2</sup> roadmap approach,<sup>3</sup> and potential fields.<sup>4</sup> The most downsides of these techniques

<sup>1</sup> Department of Electrical Engineering, College of Engineering, University of Baghdad, Al-Jadriyah, Baghdad, Iraq

<sup>2</sup> Robotics and Internet-of-Things Lab (RIOTU), Prince Sultan University, Riyadh, Saudi Arabia

<sup>3</sup> Faculty of Computers and Artificial Intelligence, Benha University, Benha, Egypt

<sup>4</sup> Department of Control and Systems Engineering, University of Technology, Baghdad, Iraq

### Corresponding author:

Ibraheem Kasim Ibraheem, Department of Electrical Engineering, College of Engineering, University of Baghdad, Al-Jadriyah, Baghdad 10001, Iraq.

Email: [ibraheemki@coeng.uobaghdad.edu.iq](mailto:ibraheemki@coeng.uobaghdad.edu.iq)



Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (<https://creativecommons.org/licenses/by/4.0/>) which permits any use, reproduction and distribution of the work without

further permission provided the original work is attributed as specified on the SAGE and Open Access pages (<https://us.sagepub.com/en-us/nam/open-access-at-sage>).

are their inefficiency due to the high computational cost and inaccuracy due to the trapping in the local minimum. Using different heuristic techniques such as fuzzy logic system and swarm optimization algorithms, one can overcome the drawbacks of the above algorithms. A significant amount of research has been devoted to solving the path planning problem in recent years. Among them, the work of Han et al.<sup>5</sup> where the standard ant colony optimization (ACO) is combined with the influence value of the critical obstacles as the amount of initial pheromone on each arc to encourage ants to move in a path closest to the direct line that connects the start position (SP) and GP. Modified ACO considering the age of the ants has been exploited in the path planning of an omnidirectional mobile robot in static grid environments as presented in the work of Ibraheem and Ajeil.<sup>6</sup> The enhancement of the ACO calculations to accomplish effective search skills of navigation algorithms in complex environments for mobile robots has been introduced in the work of Dai et al.<sup>7</sup> The improved ACO makes use of the characteristics of the MAX-MIN ant system and the A\* technique. Mingle and Xiaoming<sup>8</sup> proposed an improved ACO-based path planning algorithm, namely, chaotic ACO, using the chaos which improves the global search capability. Moreover, particle swarm optimization (PSO)-based path planning algorithm has been demonstrated in dynamic environments as presented in the studies of Rath and Deepak,<sup>9</sup> Badmos et al.,<sup>10</sup> and Adamu et al.,<sup>11</sup> whereas multi-objective PSO-based different application are presented in the studies of Zhang et al. and Song et al.<sup>12–15</sup> Chołodowicz and Figurowski<sup>16</sup> demonstrated the optimization problem in dynamic and static environments, then a smooth path based on cubic splines is generated through interpolating the optimization solutions. To balance between global search and local search (LS), Tang et al.<sup>17</sup> hybridized nonlinear time-varying PSO with ranking-based self-adaptive differential evolution and applied the proposed method for mobile robot global path planning. The path planning offered in the study of Han and Seo<sup>18</sup> is achieved in two steps; in the initial step, an encompassing point set is produced to encircle the obstacle, at that point a preliminary possible path utilizing Dijkstra calculations is generated. In the subsequent step, a path improvement technique relying on the former and latter points has been carried out, in which each point in the path is relocated by two points on either side. Mac et al.<sup>19</sup> introduced a hierarchical approach to obtain the shortest and smoothest path based on the PSO algorithm in a cluttered environment. Genetic algorithm (GA) has been widely applied in the path planning problem<sup>20–23</sup> in different types of environments; wherein Liu et al.<sup>23</sup> proposed a tailored GA to plan an optimal path for the multi-goal visiting task. The comparison between two metaheuristic algorithms, that is, artificial immune system and ACO, is presented in the work of Rebeiro et al.<sup>24</sup> to tackle the problem of mobile robot path planning. The best path of the mobile robot studied in Contreras-Cruz et al.<sup>25</sup> is found in two sequential

stages; artificial bee colony is implemented in the first stage to find an initial feasible path, then a refinement of this path is accomplished by the evolutionary algorithm. Frog leaping behavior inspired algorithm is applied by Arshi et al.<sup>26</sup> to conduct multi-objective path planning. Modified simulated annealing is implemented as a path planner in small UAVs in the work of Behnck et al.<sup>27</sup> Path planning based on hybridization between different swarm optimization algorithms using multi-objective measures has been studied by Ajeil et al.<sup>28</sup> Bäsaca-Preciado et al.<sup>29</sup> proposed a high-accuracy localization based on dynamic triangulation, where a shorter and smoother trajectory for Pioneer 3-AT mobile robot is obtained. Sergiyenko et al.<sup>30–33</sup> proposed a method using fuzzy logic to share information among multi-robot group to increase the overall movement speed for all robots in an area with high density of obstacles. Finally, multi-objective evolutionary algorithm, memetic algorithms, and intelligent water drop have been introduced to find the best path for the mobile robot in the studies of Xue and Sun,<sup>34</sup> Zhu et al.,<sup>35</sup> and Salmanpour et al.,<sup>36</sup> respectively. It should be noted that mobile robot navigation including path planning can be regarded as a high-level motion planning task through which the mobile obtains data and reacts to its surroundings. This layer is built on the motion control lower-level layer, which controls the actuation of the mobile robot as commanded by the upper layer. Many linear and nonlinear control techniques are available for the design of the motion control layer.<sup>37–43</sup>

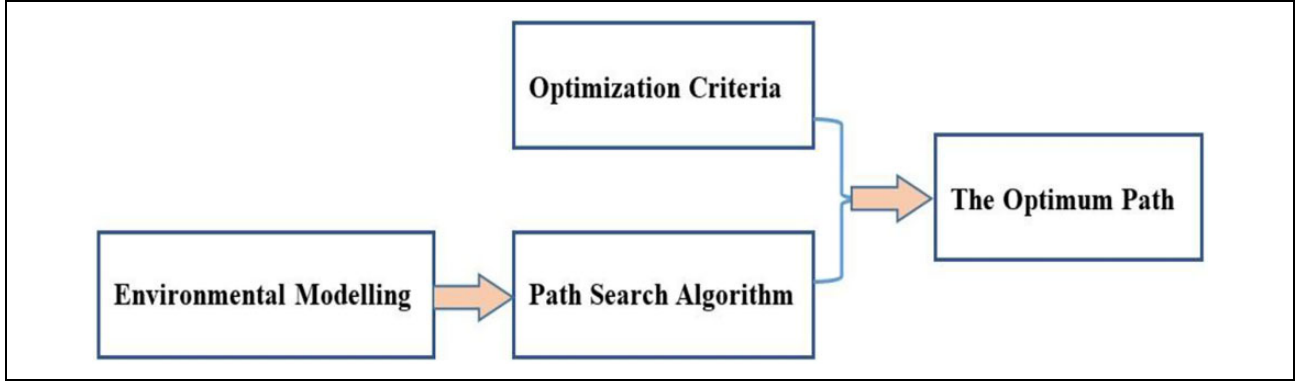
## Motivation

Some of the limitations of previous works are, the previous studies try to adopt an objective function like the shortest and/or smoothest path in a static or dynamic environment without considering robot size into account. Moreover, grid-based environment modeling suffers from inflexibility in dynamic environments and space wastage. Motivated by the pros and cons of the aforementioned studies, this article presents a collision-free shortest path planning algorithm considering the obstacle size in the path planning algorithm and fits in dynamic and static environments, where the considered environments are modeled in free-space modeling technique.

## Article contribution

This research proposes a new path planner for an omnidirectional mobile robot where its structure consists of three modules to plan a safe, feasible, and optimal path for a mobile robot in dynamic and static environments. These modules are:

1. *First module.* A module which includes path generation algorithm using Modified Frequency Bat (MFB) and Hybrid PSO-MFB-based optimization techniques.
2. *Second module.* The second module accomplishes the conversion of the infeasible solutions that might



**Figure 1.** The principle of mobile robot path planning.<sup>44</sup>

be generated by the proposed swarm optimization-based path-planning algorithm in the first module into feasible ones.

3. Finally, the *third module* provides obstacle detection and avoidance (ODA) procedures.

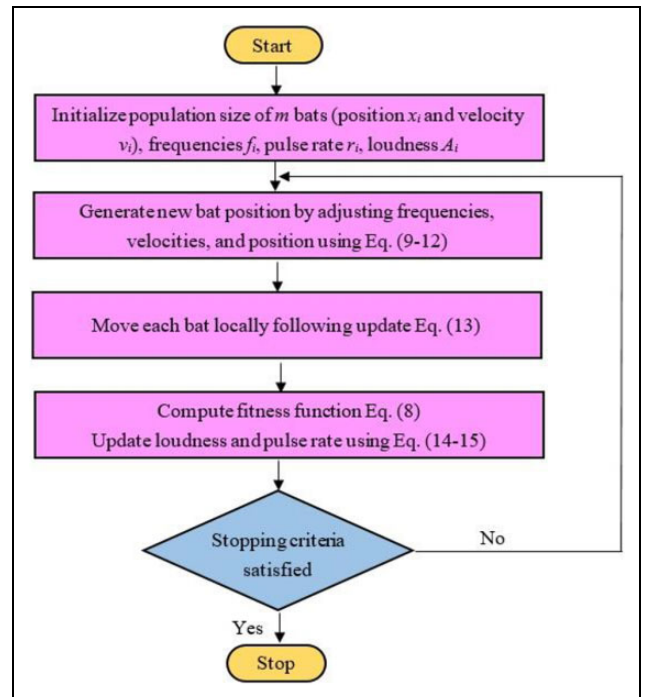
To the best of our knowledge, no work has been found in the literature that solves the problem of the mobile robot path planning using the claimed modified swarm optimization techniques with integrated ODA procedures.

The current research is organized as follows: The second section presents the environment modeling. The third section presents the objective measures considered in this article. The proposed modified swarm optimization techniques and swarm-based path planning algorithms are introduced in the fourth section. In the fifth section, the simulation results are introduced to check the validity of the path planning problem. Finally, the conclusions and recommendations for future works are presented in the sixth section.

## Environment modeling

The basic requirements to find the optimal path for a mobile robot are illustrated in Figure 1. It is apparent that to proceed in the optimum path planning, the first step is to establish the environment model. The mobile robot's workspace is represented as a 2D Cartesian coordinates  $(x, y)$ . The origin of the environment is in the lower left corner  $(0, 0)$ , Where the mobile robot is located at the SP and needs to arrive at its GP safely without colliding with static and/or dynamic circular obstacles (with radius  $r_{Obs}$ ) exist in the environment. The actual size of the mobile robot ( $r_{MR}$ ) is considered by adding it to the size of the obstacle to get an obstacle with new size ( $r_{obs}$ ). For a dynamic environment, suppose that the obstacles move linearly at constant speed as ( $v_{obs}$ ) and direction ( $\varphi_{obs}$ ) according to the following relationship

$$x_{obs\_New} = x_{obs} + v_{obs} \times \cos \varphi_{obs} \quad (1)$$



**Figure 2.** Flowchart of the proposed MFB swarm optimization algorithm. MFB: Modified Frequency Bat.

$$y_{obs\_New} = y_{obs} + v_{obs} \times \sin \varphi_{obs} \quad (2)$$

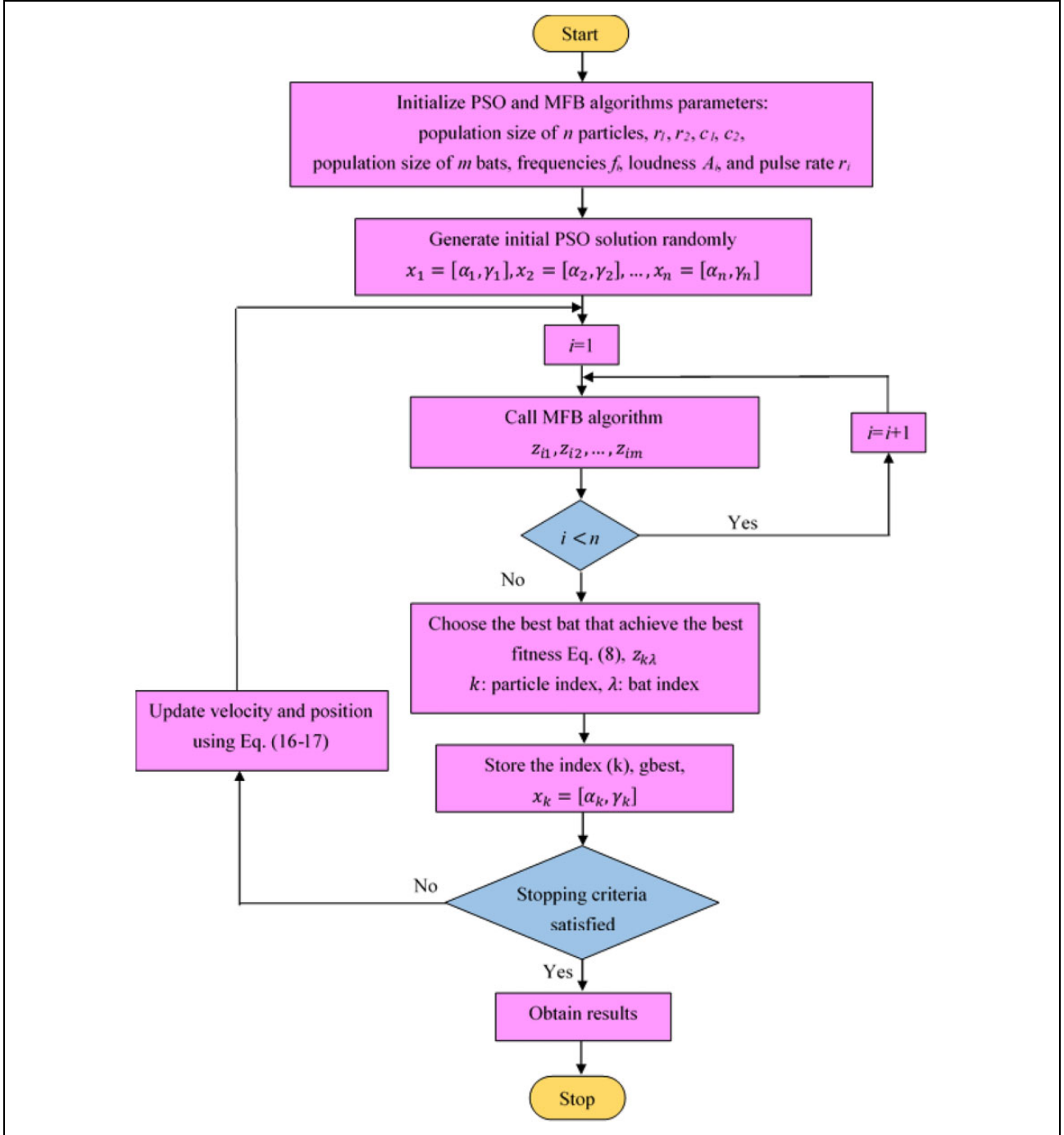
where  $\varphi_{obs}$  is the angle of the linear motion.

## Optimization criteria

The main purpose of the mobile robot path planning is searching for an optimal/near-optimal path via satisfying some criteria such as:

- (A) Shortest distance

In the field of mobile robot navigation, we always seek for the “Shortest Distance” which indicates



**Figure 3.** Flowchart of the suggested Hybrid PSO-MFB swarm optimization algorithm. MFB: Modified Frequency Bat; PSO: particle swarm optimization.

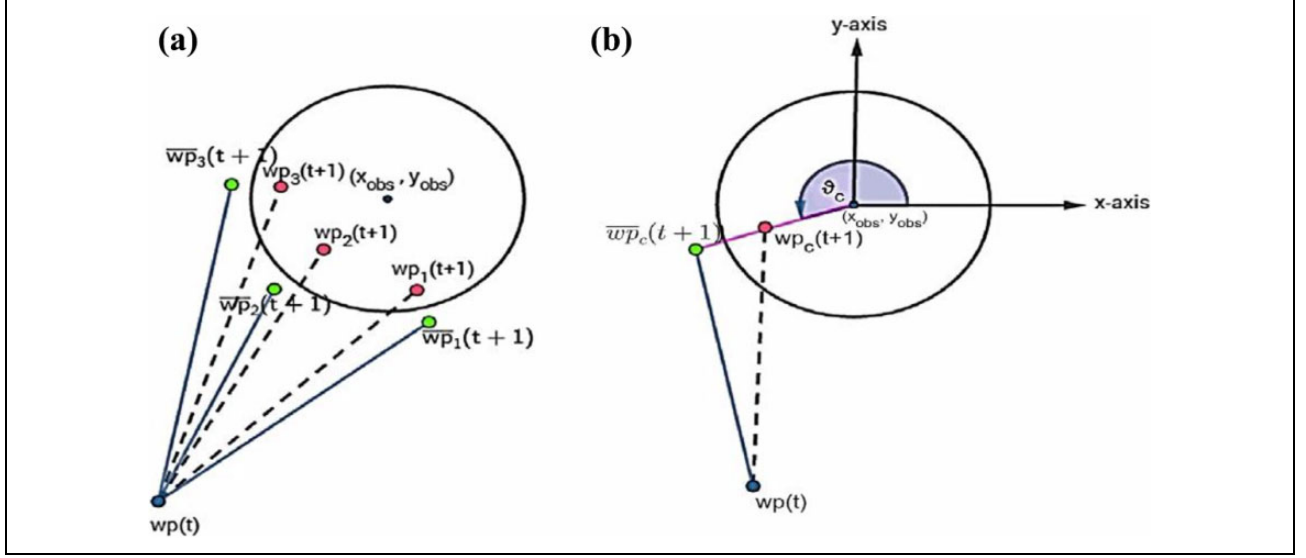
decreasing the length of the track from the SP to GP, the objective function used for this purpose is as follows

$$f_1(x, y) = d(wp_j(t), wp(N)) \quad (3)$$

The point  $wp_j(t)$  will be chosen as the candidate point provided that it has the shortest distance to the GP. Given a

set of mid-points  $(wp_j(2) \dots wp_j(N-1))$  produced by the proposed swarm-inspired optimization algorithm, the sum of the distances between these points is called the shortest path length (SPL) and given by

$$SPL = \sum_{t=1}^{N-1} d(wp_j(t), wp_j(t+1)) = \sum_{t=1}^{N-1} d_t \quad (4)$$



**Figure 4.** Infeasible path: (a) the point lies inside the obstacle and (b) the proposed solution.

where  $wp(1)$  is the SP while  $wp(N)$  is the GP,  $j$  is the best solution index, and  $d_i$  is given as

$$d_i = \sqrt{(x_{wp_j(t+1)} - x_{wp_j(t)})^2 + (y_{wp_j(t+1)} - y_{wp_j(t)})^2}$$

(B) Path smoothness

It implies reducing the angle difference in the directed segments (target-present point and candidate points-present point), and is given by

$$f_2(x, y) = \sum_{t=1}^{N-1} \left| \theta_{(wp(t), wp_j(t+1))} - \theta_{(wp(t), wp(N))} \right| \quad (5)$$

where  $\theta_{(wp(t), wp_j(t+1))} = \tan^{-1} \frac{y_{wp_j(t+1)} - y_{wp(t)}}{x_{wp_j(t+1)} - x_{wp(t)}}$ ,  $\theta_{(wp(t), wp(N))} = \tan^{-1} \frac{y_{wp(N)} - y_{wp(t)}}{x_{wp(N)} - x_{wp(t)}}$ .

The total weighted sum multi-objective optimization is given by

$$f(x, y) = w_1 f_1(x, y) + w_2 f_2(x, y) \quad (6)$$

where  $w_1$  and  $w_2$  are weighing coefficients for the aforementioned performance indices. They must ensure that

$$w_1 + w_2 = 1 \quad (7)$$

The total fitness function is given by

$$\text{fitness} = \frac{1}{f(x, y) + \epsilon} \quad (8)$$

where  $\epsilon$  is a small number (i.e.  $\epsilon = 0.0001$ ).

## Main results

The proposed swarm optimization and path planning algorithms are presented and investigated in this section.

### Proposed swarm optimization algorithms

In this section, we present two proposed swarm optimization algorithms; the first is the MFB, which is a modification from the standard bat algorithm (BA), while the second one is the Hybridized PSO-MFB swarm optimization algorithm.

**MFB swarm optimization algorithm.** BA is a bio-motivated strategy authored in the work of Yang et al.<sup>45</sup> It depends on the echolocation or biosonar attributes of the microbat. Echolocation is a basic component of bat manner, that is, the bats produce sound pulse and hearken the echoes reflecting from the obstacle while flying. The following introduces the modified BA by controlling the frequency of the individual bats.

#### (1) Bats movement

The position update of bats has been modified as

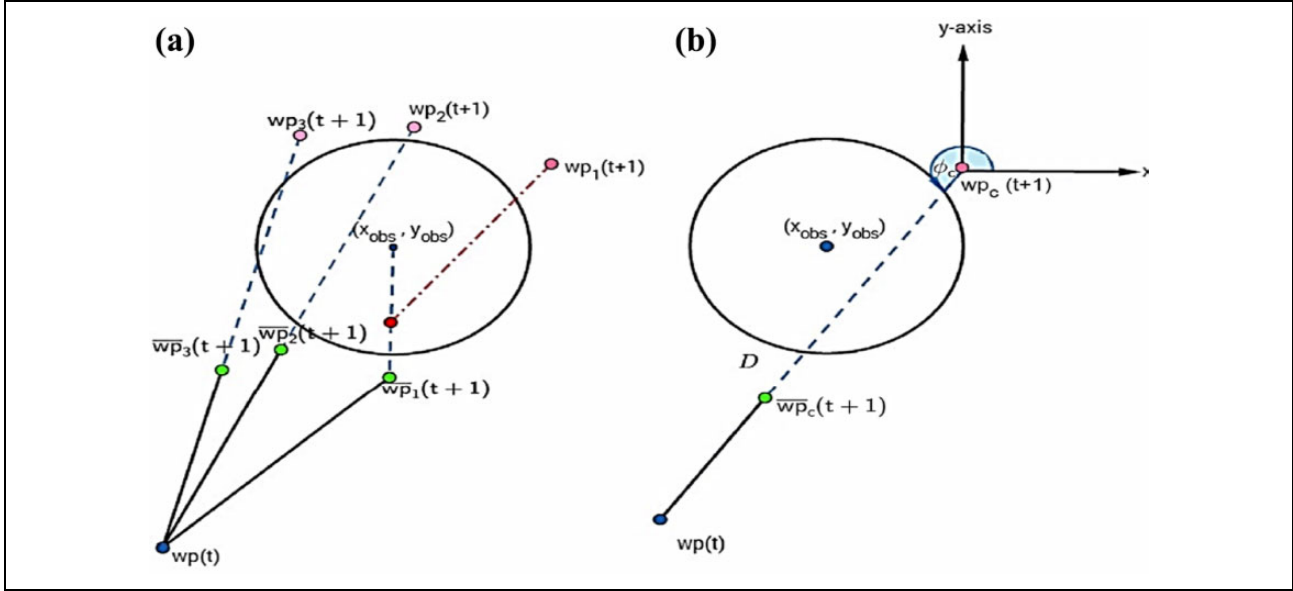
$$f_i = f_{\min} + (f_{\max} - f_{\min}) * \beta_i \quad (9)$$

$$\beta_i = te^{-\rho r} \quad (10)$$

$$v_i(t+1) = v_i(t) + (z_i(t+1) - z^*)f_i \quad (11)$$

$$z_i(t+1) = z_i(t) + v_i(t+1) \quad (12)$$

where the parameter  $\beta_i$  is a random number that increases through time for every individual bat and producing low-frequency pulses at the beginning periods of the searching task. These frequency pulses increase through time to enhance the global hunting convergence,<sup>46</sup>  $r$  is a random number (0,1),  $t$  is iteration index,  $\rho = 0.01$  (in our work),  $z^*$



**Figure 5.** Infeasible path: (a) the line segment passes through the obstacle and (b) the proposed solution.

is the current best position solution, obtained by comparing all  $m$  competing solutions. For exploitation phase, when a point is nominated among the present best points, a new point is produced for every individual bat locally using a random walk

$$z_{new} = z_{old} + \sigma \epsilon A(t) \quad (13)$$

where  $\sigma$  is a weight factor used to adjust the step size,  $A(t)$  is the average loudness of all the bats at time step  $t$ , and  $\epsilon$  is a random number within  $[-1, 1]$ .

## (2) Loudness and pulse emission

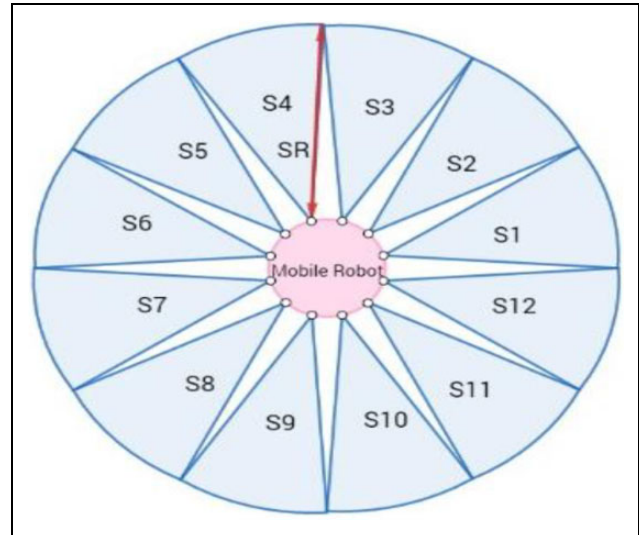
The rate of pulse emission  $r_i$  and loudness  $A_i$  are changed through iterations. Typically, the loudness reduced when a bat finds its prey, while the pulse emission rate rises according to

$$A_i(t+1) = \alpha A_i(t) \quad (14)$$

$$r_i(t+1) = r_i(0)[1 - \exp(-\gamma t)] \quad (15)$$

where  $0 < \alpha < 1$  and  $\gamma > 0$  are design parameters. Figure 2 shows the flowchart of the MFB algorithm.

**Hybrid PSO-MFB swarm optimization algorithm.** Hybrid algorithms have been improved by merging two or more algorithms to develop or enhance total search performance by using the benefits of single algorithms for the common good.<sup>47</sup> A hybridization between PSO and MFB algorithms has been developed to produce a new algorithm called Hybrid PSO-MFB algorithm. In the suggested Hybrid PSO-MFB technique, the auto-zooming capability include variations of loudness,  $A_i$ , and pulse emission rates,  $r_i$ , for the MFB optimization algorithm were adapted using PSO algorithm to balance between the exploration and



**Figure 6.** Sensors deployment of the mobile robot.

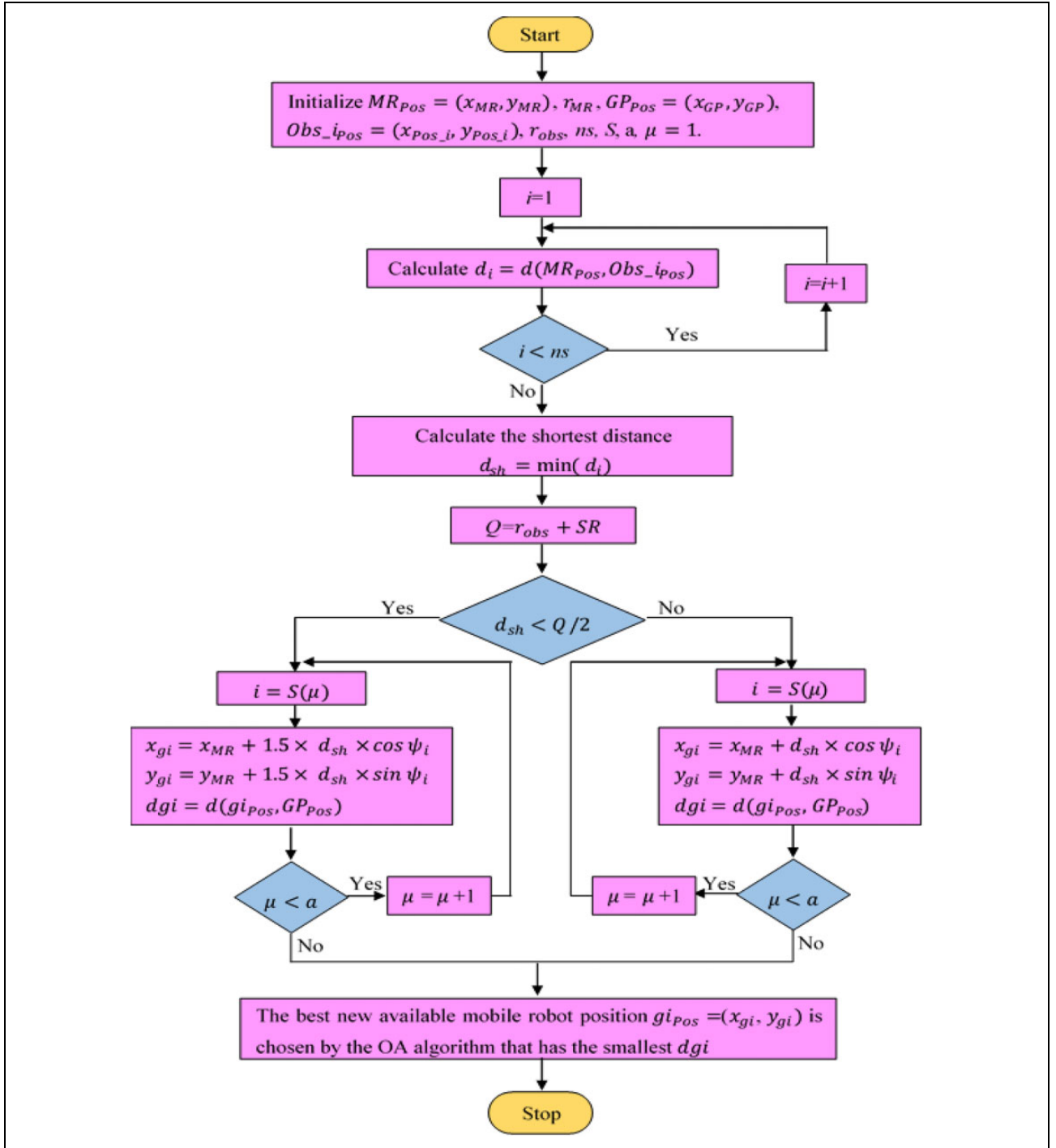
exploitation processes of the swarm optimization. Moreover, their values are application-specific and thus it is hard to define their nominal values.

PSO is a population-based stochastic optimization algorithm inspired by intelligent aggregate conduct of some animals such as flocks of birds or schools of fish, presented by Kennedy and Eberhart in 1995.<sup>48</sup> PSO accomplishes seeking through a herd of individuals that updates from iteration  $t$  to iteration  $t+1$ .

To find the optimal solution, each particle  $i$  moves in the direction to its previously best  $pbest_i$  position and the global best  $gbest$  position in the herd. The velocity  $v_i$  and position  $x_i$  of particle  $i$  are updated as follows

$$v_i(t+1) = v_i(t) + c_1 r_1 (pbest_i - x_i) + c_2 r_2 (gbest - x_i) \quad (16)$$





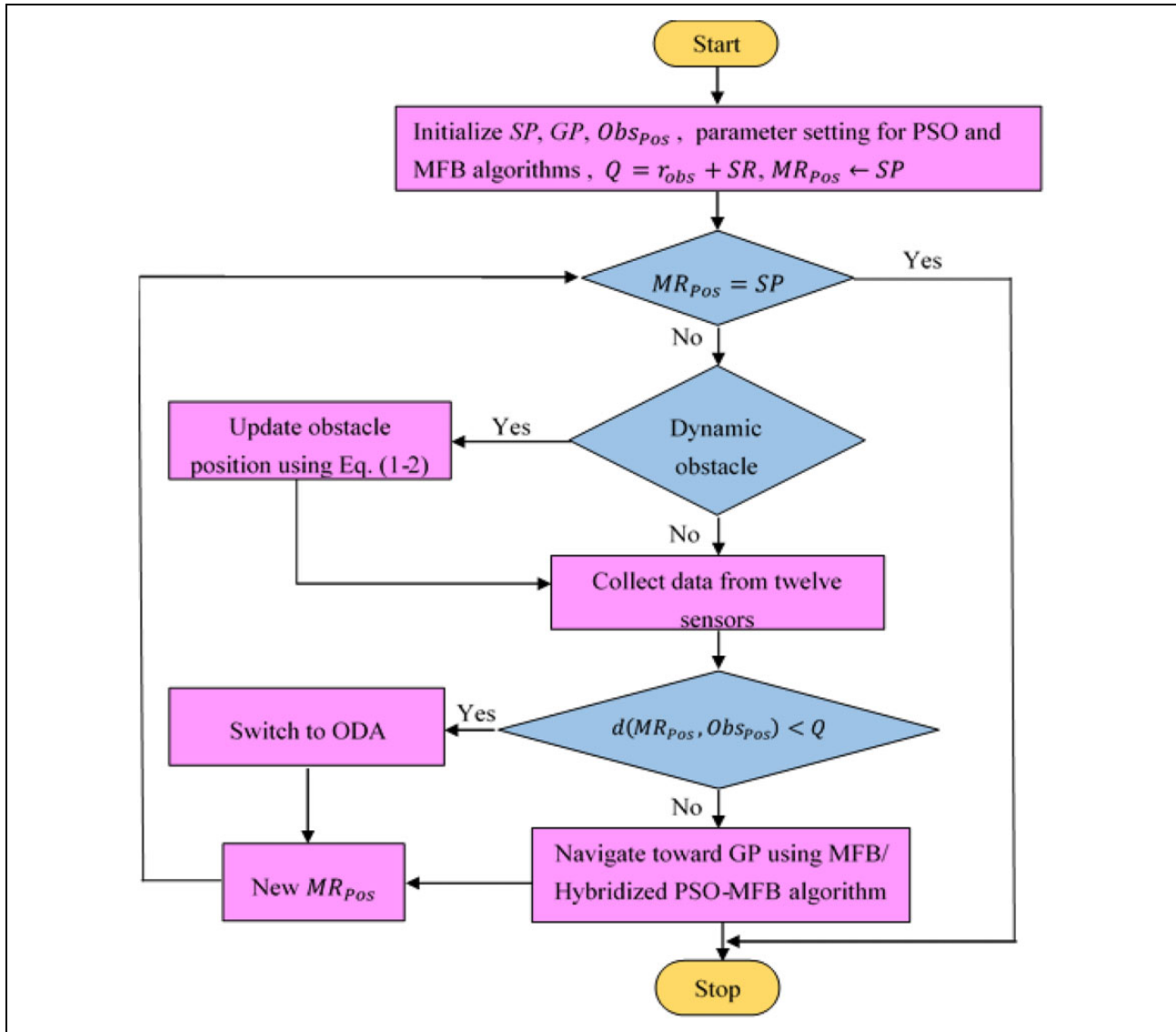
**Figure 7.** Flowchart of the proposed OA procedure. OA: obstacle avoidance.

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (17)$$

In the suggested Hybrid PSO-MFB-based path planning algorithm, the results from PSO is a vector of two columns; the complete process of the suggested Hybrid PSO-MFB technique is presented in Figure 3, where  $x[1, 1]$  stands for  $\alpha$  and  $x[1, 2]$  stand for  $\gamma$ .

### The proposed swarm optimization-based path planning technique

This section proposed a three-level structure to construct an optimal/near-optimal path for a mobile robot in static and dynamic environments. These levels are path generation, local search (LS), and obstacle detection and avoidance (ODA) technique.



**Figure 8.** Suggested MFB/Hybrid PSO-MFB swarm optimization-based path planning algorithm for a mobile robot in static and dynamic environments. MFB: Modified Frequency Bat; PSO: particle swarm optimization.

**Table 1.** Parameters settings for MFB algorithm.

Factor	Value
Iteration number	8
Population size	8
$\alpha$	0.9
$\gamma$	0.9
$[f_{min}, f_{max}]$	[0, 10]
$\sigma$	0.3
SR	0.8 (m)

MFB: Modified Frequency Bat.

**Path generation.** Each time step  $t$ , the optimization algorithms (MFB, Hybrid PSO-MFB) suggest different solutions, the task of choosing the best point among the candidate feasible ones in every iteration is based on the balance between the two objective functions defined in

**Table 2.** Parameters settings for Hybrid PSO-MFB algorithm.

Factor	Value
Iteration number	5
Population size <sub>PSO</sub>	5
Population size <sub>MFB</sub>	5
$[c_1, c_2]$	[2, 2]
$[f_{min}, f_{max}]$	[0, 10]
$\sigma$	0.3
SR	0.8 (m)

MFB: Modified Frequency Bat; PSO: particle swarm optimization.

equations (3) and (5) for all suggested solutions. This task is ongoing until the mobile robot reaches its GP.

**Local search (LS).** The LS technique converts the invalid solutions generated by the swarm optimization algorithms



(MFB/Hybrid PSO-MFB) into valid ones. There are two cases of invalid solutions: the first case, *when the candidate solution lies inside the obstacle*, this case can be checked by calculating the Euclidean distance between the candidate point,  $wp_c(t+1)$ , and the center of the obstacle  $P_{obs} = (x_{obs}, y_{obs})$  as given in equation (3)

$$d(wp_c(t+1), P_{obs}) < r_{obs} \quad (18)$$

To avoid this situation, the solution is updated according to

$$\bar{x}_{wp_c(t+1)} = x_{wp_c(t+1)} + (r_{obs} + ds - d) \times \cos \vartheta_c \quad (19)$$

$$\bar{y}_{wp_c(t+1)} = y_{wp_c(t+1)} + (r_{obs} + ds - d) \times \sin \vartheta_c \quad (20)$$

where  $ds$  stand for the minimum safety distance,  $\vartheta_c$  is the angle between the obstacle center, and  $c$ th candidate point  $wp_c(t+1)$ .

Therefore, the red points in Figure 4 represent the candidate solutions generated by MFB and/or the Hybrid PSO-MFB algorithms, while the green ones are the updated ones according to equations (19) and (20), where  $\overline{wp}_c = (x_{\overline{wp}_c(t+1)}, y_{\overline{wp}_c(t+1)})$ .

The second one, *when the direct line between two consecutive points,  $wp(t)$  and  $wp_c(t+1)$ , passes through the obstacle region*. This situation is shown in Figure 5(a). To

manipulate this situation, the following equation is considered (green points in Figure 5(b))

$$\bar{x}_{wp_c(t+1)} = x_{wp_c(t+1)} + (\delta * D) \times \cos \vartheta_c \quad (21)$$

$$\bar{y}_{wp_c(t+1)} = y_{wp_c(t+1)} + (\delta * D) \times \sin \vartheta_c \quad (22)$$

$\delta$  is chosen to be 0.6,  $D$  is the distance between candidate waypoint ( $wp_c(t+1)$ ) and previous waypoint ( $wp(t)$ ).

### Obstacle detection and avoidance

The mobile robot traverses from its SP to GP using navigation algorithms. Sensors are an essential part of an autonomous mobile robot; without these devices, the mobile robot could not gather information about its environment; sensing method provides a higher level of intelligent capabilities for the mobile robot to execute its duties. The sensing method is achieved by equipping the mobile robot by virtual 12 sensors attached around the omnidirectional mobile robot, each sensor with  $30^\circ$  separated from the neighboring sensors and sensing range (SR) of 0.8 m as shown in Figure 6. Sensory vector  $Vs$  is a binary vector with length equal to the number of deployed sensors, i.e.,  $Vs = [a(1) \dots a(i) \dots a(12)]$ , where  $a(i)$ ,  $i \in 1, 2, \dots, 12$  are variables with binary values, and  $Vs$  reflects the status of an obstacle extant in an angle range  $Si$ ,  $i \in 1, 2, \dots, 12$ . For example, with  $a(1) = a(2) = a(7) = \text{logic "1"}$ , this indicates that obstacles are detected inside  $SR$  and in the angle range  $S1$ ,  $S2$ , and  $S7$  respectively, while a logic "0" in a certain  $a(i)s$  of  $Vs$  represents a free space in the corresponding angle ranges  $Sis$ . This can be done by computing the Euclidean distance between the mobile robot and the obstacle as shown in equation (23)

$$d(MR_{pos}, Obs_{pos}) < SR \quad (23)$$

Then, set the corresponding bit in  $Vs$  to logic 1, otherwise set to logic 0. After constructing sensory vector  $Vs$ , the obstacle avoidance phase is triggered and the avoidance is

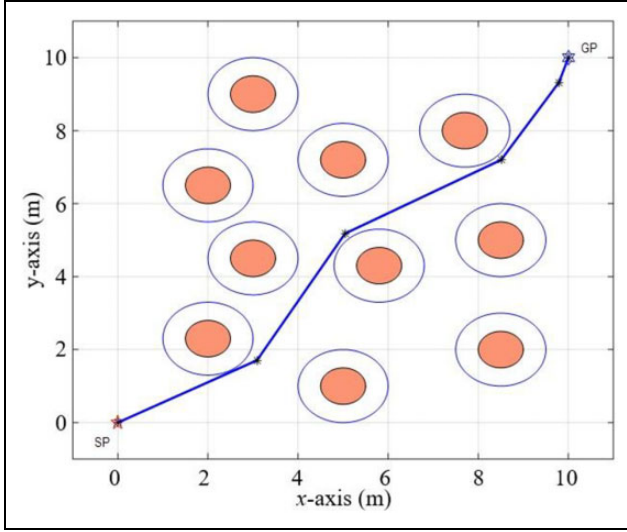
**Table 3.** Static environment settings.

Obstacle no.	Position (obs <sub>pos</sub> ) (m)
1	(2, 2.3)
2	(5.8, 4.3)
3	(8.5, 2)
4	(2, 6.5)
5	(8.5, 5)
6	(7.7, 8)
7	(3, 9)
8	(5, 1)
9	(3, 4.5)
10	(5, 7.2)

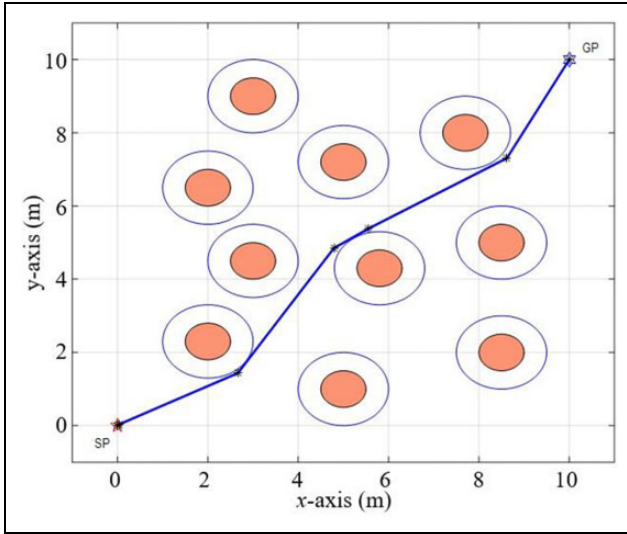
**Table 4.** The results of the proposed path planning algorithms.

Run no.	Path length, MFB (m)	Fitness	Computation time (min)	Path length, Hybrid PSO-MFB (m)	Fitness	Computation time (min)
1	14.735	0.06786	2.2917381	14.6373	0.06831	4.4563969
2	14.7706	0.06770	2.4040951	14.6772	0.06813	4.4150342
3	14.7638	0.06773	2.4244711	14.649	0.06826	4.6669314
4	<b>14.7138</b>	<b>0.06796</b>	<b>1.375219</b>	14.6676	0.06817	5.5957896
5	14.755	0.06777	1.6405978	14.6596	0.06821	4.8929868
6	14.7416	0.06783	1.917876	14.6385	0.06831	4.2228934
7	14.7386	0.06784	1.868	14.6655	0.06818	4.136777
8	14.7355	0.06786	1.8630142	14.6614	0.06820	4.3344055
9	14.7512	0.06779	2.4853818	14.644	0.06828	3.9957471
10	14.7522	0.06778	2.1183458	<b>14.6268</b>	<b>0.06836</b>	<b>4.783996</b>

MFB: Modified Frequency Bat; PSO: particle swarm optimization.  
Bold figure represents best solution among the available results.



**Figure 9.** The best path achieved using the MFB algorithm static environment. The large circle represents the obstacle enlarged by the size of the mobile robot while the small one is the obstacle with its actual size. MFB: Modified Frequency Bat.



**Figure 10.** The best path achieved using Hybridized PSO-MFB algorithm static environment. The large circle represents the obstacle enlarged by the size of the mobile robot while the small one is the obstacle with its actual size. MFB: Modified Frequency Bat; PSO: particle swarm optimization.

accomplished by using gap vector  $Vg$ . It is similar to sensory vector  $Vs$ , where logic 1 indicates the occupancy gap (the mobile robot cannot pass through it) and logic 0 indicates free gap (the mobile robot may move through it, this depends on the nearest free gap to the GP). Each bit in gap vector  $Vg$  can be derived from the corresponding consecutive bits in the sensory vector  $Vs$  as follows

$$Vg_i = Vs_i + Vs_{i+1} \quad (24)$$

where  $i$  represents the index of the sensory and gap vectors, + represents OR gate. The procedure of ODA is illustrated

**Table 5.** Comparison results for MFB and Hybrid PSO-MFB algorithms.

Fitness (equation (8))	MFB	Hybrid PSO-MFB
Maximum	0.0679634	<b>0.0683676</b>
Minimum	0.067702	<b>0.0681328</b>
Mean	0.06781627	<b>0.06824682</b>
Standard deviation	0.05073727	<b>0.05023284</b>

MFB: Modified Frequency Bat; PSO: particle swarm optimization.

**Table 6.** Dynamic obstacles settings.

obs	Center (m)	Radius (m)	$V_{obs}$ (m/s)	$\theta^\circ$
1	(4, 2)	0.3	0.3	111.8°
2	(3, 7)	0.3	0.35	315°
3	(9, 4)	0.3	0.36	126.86°
4	(7, 9)	0.3	0.25	315°

in Figure 7. The path planner technique includes a three-level structure mentioned previously and is given in Figure 8.

## Simulation result

In the simulations, two experiments were performed; the first experiment included simulations of static-obstacle environment, the second experiment represented the simulation results in a dynamic-obstacle environment.

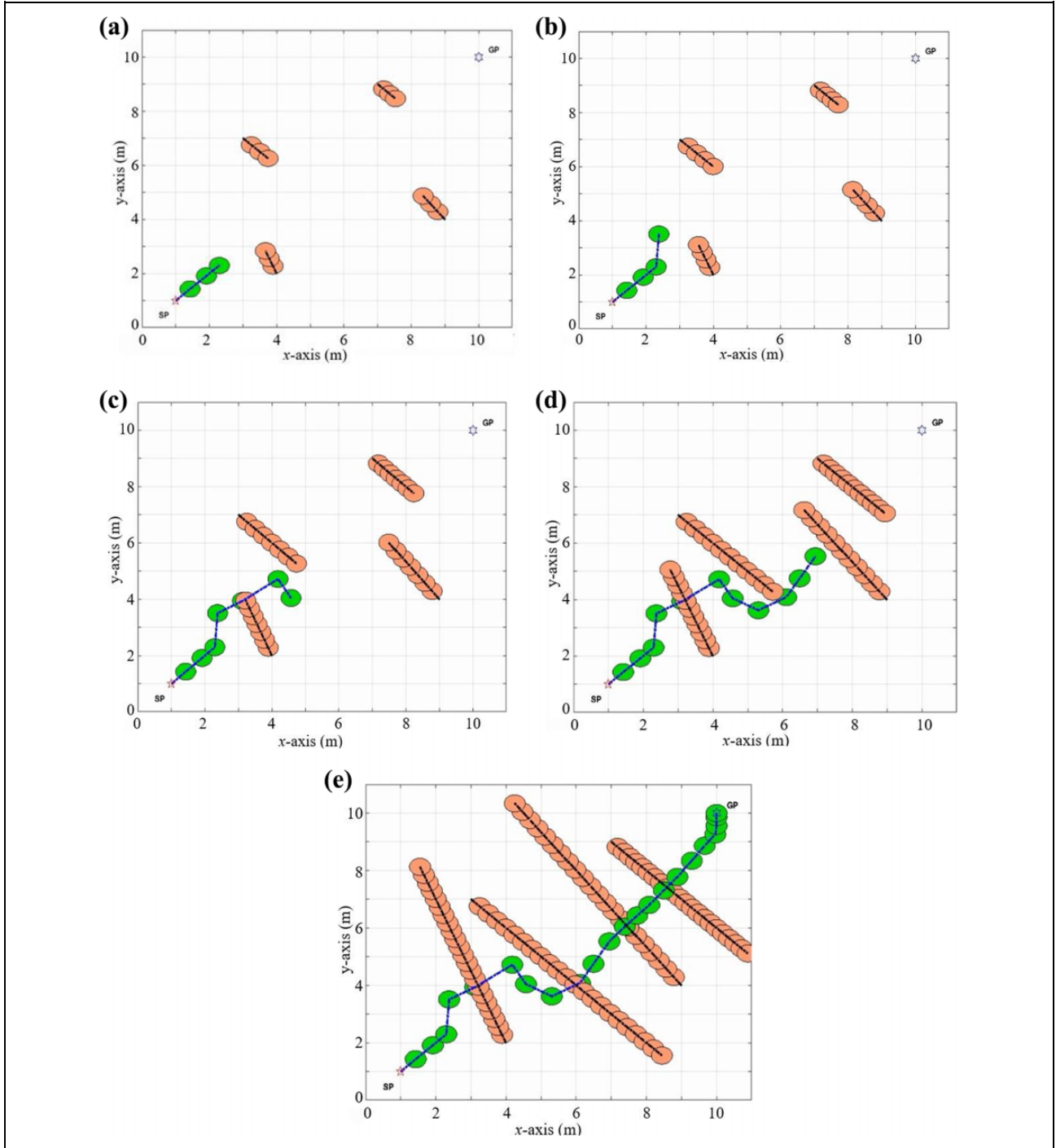
### Parameters settings of the optimization algorithms

The parameters settings for MFB and Hybrid PSO-MFB algorithms are tabulated in Tables 1 and 2, respectively.

### Experiment 1: static-obstacle environment

In this experiment, the static-obstacle environment is considered to illustrate the efficiency of the suggested technique for path planning problem. The static workspace of the mobile robot consists of 10 static obstacles with the same sizes (radius 0.5). The following are the parameters of the workspace:  $SP_{pos} = (0, 0)$ ,  $GP_{pos} = (10, 10)$ , and  $r_{MR} = 0.5m$ . The proposed MFB and Hybrid PSO-MFB algorithms are implemented in the same environment. The positions and sizes of the obstacles are tabulated in Table 3, and the performance index is declared in equation (8). The simulation results for this environment are tabulated in Table 4.

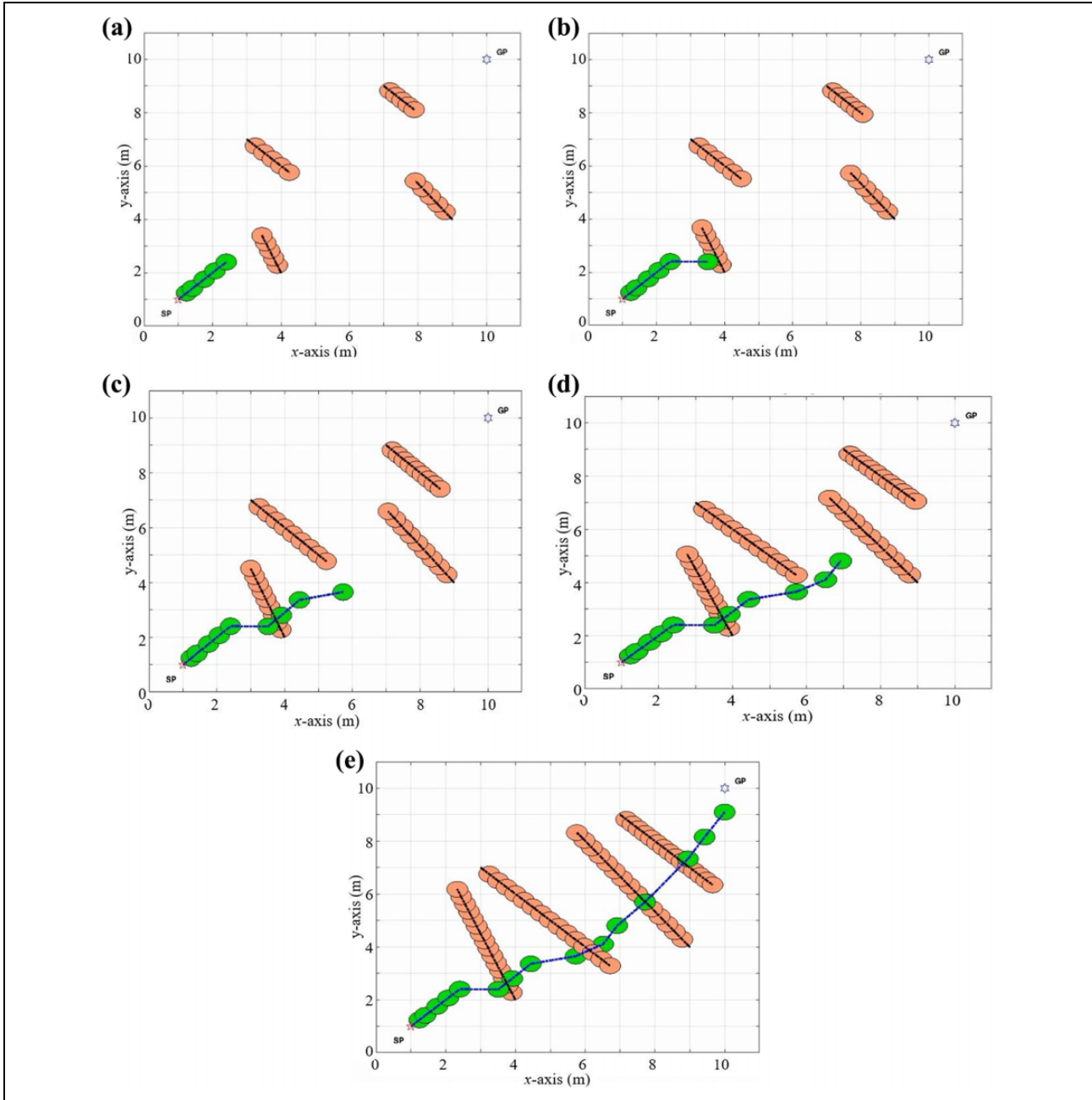
The best solution for the MFB algorithm is obtained in experiment no. 4 which has higher fitness. It is equal to 14.7138 m with computation time 1.375219 min as shown in Table 4, through points (3.0879, 1.7068), (5.0430, 5.1795), (8.5079, 7.1993), and (9.7918, 9.3244). For Hybrid PSO-MFB, the best solution (higher fitness) with



**Figure 11.** The best path achieved using the MFB algorithm in dynamic-obstacle environment, (a) Mobile robot navigates to its goal using MFB algorithm, (b) The Mobile robot detects an obstacle around it, switch to OA algorithm, (c) mobile robot avoids an obstacle using OA algorithm, (d) Mobile robot still in OA mode, (e) Mobile robot returns to MFB algorithm mode.

maximum smoothness and the shortest distance is conducted in experiment no. 10. It is equal to 14.6268 m with computation time 4.783996 min as shown in Table 4, through points (0.0305, 0.0305), (2.6628, 1.4365), (4.7907, 4.8512), (5.5495, 5.3801), and (8.6185, 7.3055). Figures 9 and 10 show the best path obtained for both MFB and Hybrid PSO-MFB, respectively.

Another comparison is made by summarizing the results for both algorithms after executing the program 10 times. The Hybrid PSO-MFB exceeds the pure MFB concerning maximum, minimum, and mean fitness as shown in Table 5. Overall, the shortest paths can be obtained using the Hybrid PSO-MFB methodology for the previous case study, while the execution time of the



**Figure 12.** The best path achieved using the Hybrid PSO-MFB algorithm in dynamic-obstacle environment, (a) Mobile robot navigates to its goal using Hybrid PSO-MFB algorithm, (b) Mobile robot detects an obstacle around it, switch to OA algorithm, (c) mobile robot avoids obstacle using OA algorithm, (d) mobile robot still in OA mode, (e) Mobile robot returns to Hybrid PSO-MFB algorithm mode.

MFB algorithm is less than that of the Hybrid PSO-MFB algorithm as seen in Table 4.

### Experiment 2: dynamic-obstacle environment

In this experiment, the suggested path planning has been checked under a dynamic workspace which consists of four dynamic obstacles with the following environment modeling parameters:  $SP_{Pos} = (1, 1)$ ,  $GP_{Pos} = (10, 10)$ , and  $r_{MR} = 0.3m$ . In this workspace, the obstacles move

linearly. The velocities, directions, and positions of the moving obstacles are given in Table 6. The mobile robot moves from (1, 1) as a SP using MFB and Hybrid PSO-MFB algorithms. When the obstacles show up in the sensing region, the mobile robot triggers the LS mode as shown in Figure 11. At that point, the mobile robot continues its searching process toward the GP. The best path obtained using MFB is shown in Figure 11 with path length equal to 14.9964 m and computation time 4.162019 s, while the best path using Hybrid PSO-MFB

is 13.6696 m with execution time 6.421085 s as shown in Figure 12.

## Discussion

Through simulations, it is apparent that the offered Hybrid PSO-MFB swarm optimization technique beats the others in terms of the shortest path. Especially, the Hybrid PSO-MFB algorithm beats the MFB one in terms of the shortest distance due to the dynamic interaction between the PSO and the MFB algorithms, where each individual particle in PSO algorithm requests for one complete MFB algorithm operation with the parameters ( $\alpha$  and  $\gamma$ ) being varying rather than constant. This results in a more sensitivity of the MFB algorithm to the loudness  $A_i(t)$  and the rate of the pulse emission  $r_i(t)$ , which raises the potential of the MFB algorithm to find better solutions. However, this achievement is on the account of the execution time of the algorithm, whereas with the Hybrid PSO-MFB algorithm it is higher than that of the MFB one. This shortcoming can be resolved via the adoption of an advanced high-speed H/W microprocessor boards for implementing the proposed algorithm.

## Conclusion

In this article, novel path planner algorithms for an omnidirectional mobile robot in environments with dynamic and static obstacles are suggested. It finds the shortest and smoothest collision-free path using swarm-based optimization techniques, namely, the MFB and Hybrid PSO-MFB algorithms. The suggested MFB and Hybrid PSO-MFB algorithms are integrated with the LS procedure to plan a path far from the obstacles and hence convert the invalid solutions into valid ones. With the sensors deployed around the mobile robot to sense the distance between it and the obstacles, the OD procedure can efficiently detect the obstacles and triggers the OA algorithm to find an alternative path to the GP. From the simulation results, we can conclude that the Hybrid PSO-MFB algorithm outperforms the MFB one concerning the length of the path because of the dynamic adjustment of the parameters  $\alpha$  and  $\gamma$  of the MFB algorithm using PSO steps. In terms of the computational complexity, the execution time of the Hybrid PSO-MFB technique is larger than that of the MFB one. Future works might include the application of the suggested path planning technique in moving target environments. Moreover, the H/W implementation of the suggested path planning technique is of great concern to validate the current work from the practical point of view.


## Declaration of conflicting interests


The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

## Funding

The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research is funded by Prince Sultan University, Riyadh, Kingdom of Saudi Arabia. Special acknowledgement to Robotics and Internet-of-Things Lab (RIOTU), Prince Sultan University, Riyadh, Saudi Arabia. We would like to show our gratitude to Prince Sultan University, Riyadh, Saudi Arabia.

## ORCID iDs

Ibraheem Kasim Ibraheem  <https://orcid.org/0000-0001-7009-3634>

Amjad J Humaidi  <https://orcid.org/0000-0002-9071-1329>

## References

1. Abichandani P, Ford G, Benson HY, et al. Mathematical programming for multi-vehicle motion planning problems. In: *2012 IEEE international conference on robotics and automation*, Saint Paul, MN, USA, 14–18 May 2012, pp. 3315–3322. IEEE.
2. Keil JM. Decomposing a polygon into simpler components. *SIAM J Comput* 2006; 14(4): 799–817.
3. Bhattacharya P and Gavrilova ML. Roadmap-based path planning—using the Voronoi diagram for a clearance-based shortest path. *IEEE Robot Autom Mag* 2008; 15(2): 58–66.
4. Kovács B, Szayer G, Tajti F, et al. A novel potential field method for path planning of mobile robots by adapting animal motion attributes. *Robot Auton Syst* 2016; 82: 24–34. DOI: 10.1016/j.robot.2016.04.007.
5. Han J, Park H, and Seo Y. Path planning for a mobile robot using ant colony optimization and the influence of critical obstacle. In: *Proceedings of the international conference on industrial engineering and operations management*, Detroit, MI, USA, 23–25 September 2016, pp. 178–186. IEOM Society.
6. Ibraheem IK and Ajeil FH. Path planning of an autonomous mobile robot using swarm based optimization techniques. *Al-Khwarizmi Eng J* 2016; 12(4): 12–25.
7. Dai X, Long S, Zhang Z, et al. Mobile robot path planning based on ant colony algorithm with A\* heuristic method. *Front Neurorobot* 2019; 13(15). DOI: 10.3389/fnbot.2019.00015.
8. Mingle X and Xiaoming Y. Chaotic ant system optimization for path planning of the mobile robots. *Comput Sci Eng Int J (CSEIJ)* 2016; 6(2/3). DOI: 10.5121/cseij.2016.6301.
9. Rath MK and Deepak BBVL. PSO based system architecture for path planning of mobile robot in dynamic environment. In: *2015 global conference on communication technologies (GCCT)*, Thuckalay, India, 23–24 April 2015, pp. 797–801. IEEE.
10. Badmos TA, Omolaye PO, Mebawondu J, et al. Robot path planning performance evaluation of a dynamic environment. *IOSR J Electron Commun Eng (IOSR-JECE)* 2018; 13(6): 19–26.
11. Adamu PI, Jegede JT, Okagbue HI, et al. Shortest path planning algorithm—a particle swarm optimization (PSO) approach. In: *Proceedings of the world congress on engineering*, WCE 2018, London, UK, 4–6 July 2018.

12. Zhang Y, Gong DW, and Ding Z. A bare-bones multi-objective particle swarm optimization algorithm for environmental/economic dispatch. *Inform Sci* 2012; 192: 213–227.
13. Song XF, Zhang Y, Guo YN, et al. Variable-size cooperative coevolutionary particle swarm optimization for feature selection on high-dimensional data. *IEEE Trans Evol Comput* 2020; 14(8): 1–14.
14. Zhang Y, Gong DW, and Cheng J. Multi-objective particle swarm optimization approach for cost-based feature selection in classification. *IEEE/ACM Trans Comput Biol Bioinform* 2015; 14(1): 64–75.
15. Zhang Y, Gong DW, and Zhang JH. Robot path planning in uncertain environment using multi-objective particle swarm optimization. *Neurocomputing* 2013; 103: 172–185.
16. Chołodowicz E and Figuirowski D. Mobile robot path planning with obstacle avoidance using particle swarm optimization. *Pomiary Autom Robot* 2017; 21: 59–68.
17. Tang B, Zhu Z, and Luo J. Hybridizing particle swarm optimization and differential evolution for the mobile robot global path planning. *Int J Adv Robot Syst* 2016; 13(3): 1–17. DOI: 10.5772/63812.
18. Han J and Seo Y. Mobile robot path planning with surrounding point set and path improvement. *Appl Soft Comput* 2017; 57: 35–47.
19. Mac TT, Copot C, Tran DT, et al. A hierarchical global path planning approach for mobile robots based on multi-objective particle swarm optimization. *Appl Soft Comput* 2017; 59: 68–76.
20. Lamini C, Benhlima S, and Elbekri A. Genetic algorithm based approach for autonomous mobile robot path planning. *Procedia Comput Sci* 2018; 127: 180–189.
21. Arora T, Yogita G, and Vijay A. Robotic path planning using genetic algorithm in dynamic environment. *Int J Comput Appl* 2014; 89(11): 8–12.
22. Alsouly H and Bennaceur H. Enhanced genetic algorithm for mobile robot path planning in static and dynamic environment. In: *Proceedings of the 8th international joint conference on computational intelligence (IJCCI 2016), ECTA*, Vol. 3, Porto, Portugal, 11 November 2016, pp. 121–131.
23. Liu F, Liang S, and Xian DX. Optimal path planning for mobile robot using tailored genetic algorithm. *TELKOMNIKA Indon J Electr Eng* 2014; 12(1): 1–9.
24. Ribeiro JMS, Silva MF, Santos MF, et al. Ant colony optimization algorithm and artificial immune system applied to a robot route. In: *20th international carpathian control conference (ICCC)*, Krakow-Wieliczka, Poland, 26–29 May 2019, pp. 1–6. IEEE.
25. Contreras-Cruz MA, Ayala-Ramirez V, and Hernandez-Belmonte UH. Mobile robot path planning using artificial bee colony and evolutionary programming. *Appl Soft Comput* 2015; 30: 319–328.
26. Arshi SS, Zolfaghari A, and Mirvakili SM. A multi-objective shuffled frog leaping algorithm for in-core fuel management optimization. *Comput Phys Commun* 2014; 185(10): 2622–2628.
27. Behnck LP, Doering D, Pereira CE, et al. A modified simulated annealing algorithm for UAVs path planning. *IFAC-PapersOnLine* 2015; 48(10): 63–68.
28. Ajeil FH, Ibraheem IK, Sahib MA, et al. Multi-objective path planning of an autonomous mobile robot using hybrid PSO-MFB optimization algorithm. *Appl Soft Comput* 2020; 89(106076): 1–13. DOI: 10.1016/j.asoc.2020.106076.
29. Básaca-Preciado LC, Sergiyenko OY, Rodríguez-Quinonez JC, et al. Optical 3D laser measurement system for navigation of autonomous mobile robot. *Optics Lasers Eng* 2014; 54: 159–169.
30. Sergiyenko OY, Ivanov MV, Tyrsva VV, et al. Data transferring model determination in robotic group. *Robot Auton Syst* 2016; 83: 251–260.
31. Sergiyenko O, Kartashov V, Ivanov M, et al. Transferring model in robotic group. In: *2016 IEEE 25th international symposium on industrial electronics (ISIE)*, Santa Clara, CA, USA, 8–10 June 2016, pp. 946–952. IEEE.
32. Ivanov M, Sergiyenko O, Tyrsva V, et al. Software advances using n-agents wireless communication integration for optimization of surrounding recognition and robotic group dead reckoning. *Program Comput Soft* 2016; 45(8): 557–569.
33. Ivanov M, Sergiyenko O, Tyrsva V, et al. Influence of data clouds fusion from 3D real-time vision system on robotic group dead reckoning in unknown terrain. *IEEE/CAA J Autom Sinica* 2020; 7(2): 368–385.
34. Xue Y and Sun JQ. Solving the path planning problem in mobile robotics with the multi-objective evolutionary algorithm. *Appl Sci* 2018; 8(9): 1425.
35. Zhu Z, Xiao J, Li JQ, et al. Global path planning of wheeled robots using multi-objective memetic algorithms. *Integr Comput Aided Eng* 2015; 22(4): 387–404.
36. Salmanpour S, Monfared H, and Omranpour H. Solving robot path planning problem by using a new elitist multi-objective IWD algorithm based on coefficient of variation. *Soft Comput* 2017; 21(11): 3063–3079.
37. Maher RA, Mohammed IA, and Ibraheem IK. Polynomial based  $H_\infty$  robust governor for load frequency control in steam turbine power systems. *Int J Electr Power Energy Syst* 2013; 57: 311–317.
38. Najm AA and Ibraheem IK. On the design of nonlinear PID controller for nonlinear quadrotor system. *Eng Sci Technol Int J* 2019; 22(4): 1087–1097.
39. Humaidi AJ and Ibraheem IK. Speed control of permanent magnet DC motor with friction and measurement noise using novel nonlinear extended state observer-based anti-disturbance control. *Energies* 2019; 12(9): 1651.
40. Maher RA, Mohammed IA, and Ibraheem IK. State-space based  $H_\infty$  robust controller design for boiler-turbine system. *Arabian J Sci Eng* 2012; 37(6): 1767–1776.
41. Ibraheem IK. A digital-based optimal AVR design of synchronous generator exciter using LQR technique. *Al-Khwarizmi Eng J* 2011; 7(1): 82–94.
42. Ibraheem GA and Ibraheem IK. Motion control of an autonomous mobile robot using modified particle swarm



- optimization based fractional order PID controller. *Eng Technol J* 2016; 34(13 Part (A)): 2406–2419.
43. Ibraheem IK. On the frequency domain solution of the speed governor design of non-minimum phase hydro power plant. *Mediterr J Meas Control* 2012; 8(3): 422–429.
44. Zhang HY, Lin WM, and Chen AX. Path planning for the mobile robot: a review. *Symmetry* 2018; 10(10): 450.
45. Yang XS, Cui Z, Xiao R, et al. *Swarm intelligence and bio-inspired computation theory and applications*. 1st ed. London: Elsevier, 2013.
46. Ajeil FH and Ibraheem IK. Path planning of an autonomous mobile robot in a dynamic environment using modified bat swarm optimization, <https://arxiv.org/abs/1807.05352> (2019, accessed September 2019).
47. Yang XS. *Nature-inspired optimization algorithms*. 1st ed. Amsterdam: Elsevier, 2014.
48. Kennedy J and Eberhart R. Particle swarm optimization. In: *Proceedings of ICNN'95—international conference on neural networks*, Perth, Australia, 27 November–1 December 1995, pp. 1942–1948. IEEE.