

Adaptive weight grey wolf algorithm application on path planning in unknown environments

Mustafa Salah Abed, Omar Farouq Lutfy, Qusay Fadhel Al-Doori
Department Control and Systems Engineering, University of Technology-Iraq, Baghdad, Iraq

Article Info

Article history:

Received Mar 13, 2022

Revised Jun 17, 2022

Accepted Jul 2, 2022

Keywords:

Adaptive variable weights

Grey wolf optimization

Mobile robot

Modification grey wolf

Obstacle avoidance

Path planning

ABSTRACT

Autonomous mobile robots developed using metaheuristic algorithms are increasingly becoming a hot topic in control and computer sciences. Specifically, finding the shortest route to the goal and avoiding hurdles are current subjects of autonomous mobile robots. The modified grey wolf optimization (MGWO) is demonstrated in this work using two approaches: first, the adaptive adjustment approach of the control parameters, and second, the adaptive variable weights method. Those two methods are utilized for updating the wolf position, accelerate convergence, and cut down on time. The proposed online optimization approach is used in three different environments including an environment with unknown static obstacles, dynamic obstacles, and an environment with a dynamic target. The online optimization method is performed using two phases which are the sensors reading phase and the path calculation phase. The proposed approach can solve a local minima problem in the static obstacles. A comparison study result between the proposed method and two other algorithms revealed that the proposed algorithm performed better in avoiding obstacles, which include the situation with the local minima. Finally, when put to comparison with hybrid fuzzy-wind driven optimization and adaptive particle swarm optimization the average improvement rates in route length are 2.86% and 4.70391%, respectively.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Mustafa Salah Abed

Department of Control and Systems Engineering, University of Technology-Iraq

Sinaa Street, Bagdad, Iraq

Email: cse.19.07@grad.uotechnology.edu.iq

1. INTRODUCTION

Route planning problem of the mobile robots has lately been a popular topic of research in mobile robots [1], [2]. The necessity to systematically design the route becomes a requirement for automated processes [3], [4]. Motion planning is a procedure for obtaining an intelligent and feasible path between the starting point and the destination. Planning the mobile robot's locomotion in an uncommon environment is typically divided into three categories. The first depends on information regarding potential obstacles, which may or may not be known [5]. The second is a reference to time [6], [7], which includes both offline and online planning. The third is determined by the sort of obstacles present in the environment, which could be moving or stationary [8], [9]. In navigating a differential wheel mobile robot (DWMR), various researchers have employed different methods. Supriya and Joshy [10] used the ant colony optimization (ACO) technique to determine a potential route in an unfamiliar environment with static obstacles. It can't be used on a continuous map or in complex settings. In an uncertain environment, Pandey and Parhi [11] proposed a type1-fuzzy control controller and evolutionary method (fuzzy-wind driven optimization algorithm). The flaw with this method is that it is incapable of resolving the local minima problem in a U or L style setting.

The whale optimization algorithm, which was utilized to discover the safest possible route, was modified by Chhillar and Choudhary [12]. The suggested approach was put to the test in a number of different search spaces. In serried situations, on the other hand, the algorithm does not guarantee a collision-free route. Hosseinnejad and Dadkhah [13] show how the adaptive particle swarm optimization (APSO) approach with the adaptive weight factor can improve the path planning of the mobile robots. However, this algorithm is prone to early convergence and cannot avoid local ex-tremity. Oleiwi *et al.* [14] utilized fuzzy logic control for avoiding collisions with dynamic obstacles in the cases of the partially unknown environments, and they utilized A-star technique in order to discover the route off-line. This method is incapable of dealing with completely unknown or maze-like settings. An intelligent APSO technique for robot path planning in unpredictable situations was suggested in another paper [15]. Yet, in order to work in dynamic or multi-robot systems and avoid early convergence in local minima, this approach must be adjusted.

From previous studies, it was note that various optimization approaches produce low-quality solutions. These trouble are observed in particle swarm optimization (PSO) [15], ant colony [8], and grey wolf optimization (GWO) [15], therefore there is a need to study and improve these methods to overcome their limitations. This article exhibits a development path planning approach that utilizes the modification of grey wolf optimization for navigations of the mobile robots in static and dynamic unknown environments with a dynamic target. The proposed online optimization approach is used in three different environments including an environment with unknown static obstacles, unknown dynamic obstacles, and a dynamic target. The proposed approach can solve a local minima problem in the environment with static obstacles and has fast convergence in finding the shortest path taking less time compared with other methods.

2. PROBLEM STATEMENT

Finding the best route between the robot's starting position and the goal is one of the most important topics in the field of search and rescue applications that use robots or quadcopters. In this work, develop the GWO algorithms to solve the drawbacks of the original GWO algorithm such as low solving accuracy, the disappointing capacity of local searching, and slow convergence rate, and employed in an online procedure to adaptively handle three different environments that include unknown static obstacles, unknown dynamic obstacles, and a dynamic target.

3. METHOD AND MODELLING

In this section, the methodology of the work and the modeling of environments are described considering a two-dimensional (X, Y) square map. Then molding the mobile robots and sensor. Finally described the modification of the algorithm. In particular, the modeling stage includes the following parts.

3.1. Obstacles

This section describes the architecture of the obstacles. There are N obstacles in the environment, O_1, O_2, \dots, O_N . The obstacles' coordinates are shown as $(X_{O1}, Y_{O1}), (X_{O2}, Y_{O2}), \dots, (X_{ein}, Y_{ein})$ and this presentation includes static and dynamic obstacles. In particular, in the case where an obstacle is stationary, hen its speed becomes zero, and when it moves, its velocity (v) is over X and Y axles. Each obstacle's speed is determined at random and can be large than, less than, or equal to the speed of robot. The robot is unaware of the obstacle vectors' speed and position (velocity and direction).

3.2. Robot and sensing

Generally, when faced with a local path planning problem, the robot tries to set the next safest position in the indefinite environment in steps from start to finish. As illustrated in Figure 1, the robot location utilized in the present paper has been based upon a relative location, in which the robot calculates its next location depending on its present corner, position, and speed over a period of time. In specifically, (1) and (2) formulate the robot's next place [16], [17].

$$X_{next} = X_{current} + Vln * Cos(\theta_i) \quad (1)$$

$$Y_{next} = Y_{current} + Vln * sin(\theta_i), \quad (2)$$

Where $X_{current}$ and $Y_{current}$ represent coordinates regarding existing, position and X_{next} and Y_{next} are the next position coordinates to n th robot in Cartesian coordinate system. In addition to that, the speed of a robot can be specified by Vln and θ_i .

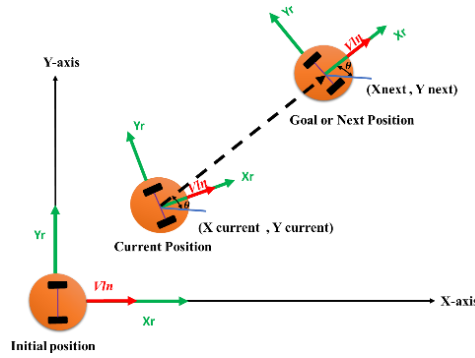


Figure 1. Positions of the mobile robot

The robot, which recognizes the obstacle within its range, determines the space between the obstacle and the robot and estimates the direction of the moving obstacle [18], [19]. If the position of the obstacle does not change, the obstacle is static. Otherwise, the obstacle is dynamic. Figure 2 shows the path-finding procedure of the mobile robot [20].

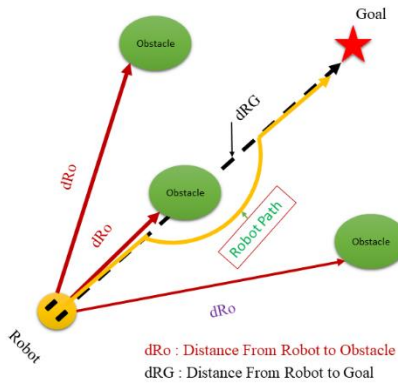


Figure 2. The path finding model of the mobile robot

3.3. Target-pursuing behavior

This section explains how the mobile robot detects the position of the goal, as illustrated in Figure 3. We can compute the minimum distance between the robot and its goal (dRG) which is defined by (3). Specifically, if the robot reaches the target, this means the distance (dRG) is equal to or near to zero, in which case the robot must be stopped [21].

$$dRG = \sqrt{(Xr - Xg)^2 + (Yr - Yg)^2} \tag{3}$$

Where Xg & Yg represents target coordinates in the environment, and Xr & Yr represent existing robot coordinates in the environment.

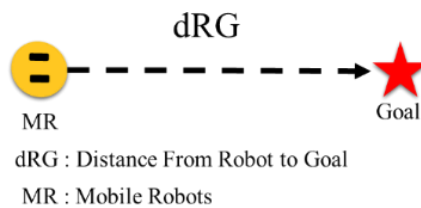


Figure 3. Distance between the robot and its goal

3.4. Obstacle-pursuing behavior

When an obstacle is found using the sensors, as depicted in Figure 4, the online optimization method can estimate the maximum distance between robot and the obstacle (dRo), which is defined by (4). Particularly, if the robot detects an obstacle, it tries to avoid it and then continues to reach the target:

$$dRo = \sqrt{(Xr - Xo)^2 + (Yr - Yo)^2} \quad (4)$$

where Xo and Yo represent obstacle's coordinates in environment, and Xr & Yr represent current robot's position coordinates in the environment.

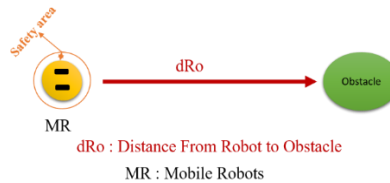


Figure 4. Distance between robot and obstacle

3.5. Objective function (cost function)

For the safety of the robot, the space between obstacle and the robot has to be as large as possible, while the space between the target and the robot must be as small as possible. Depending on these two perspectives, the cost function must ensure the best route. The cost function equation is used in particular in (5) [15]:

$$Cost\ Function = [\alpha_1 \times dRG + \alpha_2 \div dRO + \alpha_3 \times \theta] \quad (5)$$

in this equation, parameter α_1 has been set so that the robot covers minimal distance then reaches a point of the target [15]. Similarly, α_2 is responsible for controlling space between robot and obstacles [15]. α_3 represents parameter of path smoothing for avoiding the sharp turning [15]. dRO and dRG are utilized for the calculation of space between robot and obstacle, and space between robot and its point of destination, respectively with the use of distance formulas (4) and (5). Moreover, θ in the corner of the variation is wanted by robot for the detection of the next iteration location in an environment. In particular, θ has been represented by (6).

$$\theta = \tan^{-1} \frac{Yg - Yr}{Xg - Xr} \quad (6)$$

3.6. Path planning optimization algorithms

This section explained the detail the classical GWO work. Then explained modification grey wolf optimization one (MGWO). In particular, the algorithm stage includes the following parts:

3.6.1. Grey wolf optimization (gwo)

GWO has been inspired by the natural leadership structure of the grey wolves as well as their hunting method. Four sorts of wolves are employed to simulate the leadership hierarchy: beta (β), alpha (α), omega (ω), and delta (δ). In addition, the three basic processes of hunting are built: searching for prey, encircling it, and then attacking target. There are mathematical models for social hierarchy, also pursuit, encircling, and striking the target [22].

a. Mathematical model of prey encircling

In the case when hunting, wolves bound their victim. Equations (7) and (8) are utilized to mathematically model this situation. Thus, the hunt with new wolf position will be included:

$$\vec{D} = |\vec{C} \cdot \vec{Xp}(t) - \vec{X}(t)| \quad (7)$$

$$\vec{X}(t+1) = \vec{Xp}(t) - \vec{AX} \cdot \vec{DX}, \quad (8)$$

here, t symbolized the existing iteration \vec{A} and \vec{C} represent coefficient vectors, \vec{Xp} represents position vector of the prey, \vec{D} represents distance between the wolf and the victim, and \vec{X} symbolized the situation vector of the gray wolf. \vec{A} , \vec{C} and \vec{a} are determent as given in (9), (10), and (11):

$$\vec{A} = 2\vec{a} \cdot \vec{r1} - \vec{a} \tag{9}$$

$$\vec{C} = 2 \cdot \vec{r1} \tag{10}$$

where the pramer of \vec{a} are linearly decreased from 2 to 0 throughout iterations and $\vec{r1}$ & $\vec{r2}$ represent random vectors in [0,1].

The gray wolves attack the target and try to prevent the movement of the target throughout a hunting gear. Such mechanism has been modeled with the use of (9). \vec{A} represents a random vector, and its amount ranges in $[-a, a]$, with value of \vec{a} detracton linearly pending iterations and existence in the $[2, 0]$ range, as in (11), [22], [23]:

$$\vec{a} = \left(2 - 2 * \frac{t}{t_{max}} \right) \tag{11}$$

where t_{max} symbolized the max number of the iterations and t represents the existing iteration.

b. Mathematical model of the hunting

Gray wolves are capable of encircling their prey. The numerical model implies the prey has no idea where it is. Thus, beta, alpha, and delta have a better idea of where the victims are located. Alpha (1st best solution), delta, and beta are the 3 best candidate answers. Omega wolves follow the upper layer wolves and reposition themselves. The next (12) and (13) are suggested in this approach [24]:

$$\vec{D}\vec{\alpha} = |\vec{C1}\vec{X} \cdot \vec{X}\vec{\alpha1} - \vec{X1}|, \vec{D}\vec{\beta} = |\vec{C2} \cdot \vec{X}\vec{\beta2} - \vec{X2}|, \vec{D}\vec{\delta} = |\vec{C3} \cdot \vec{X}\vec{\delta3} - \vec{X3}| \tag{12}$$

$$\vec{X1} = \vec{X}\vec{\alpha1} - \vec{A1}, \vec{D}\vec{\alpha}, \vec{X2} = \vec{X}\vec{\beta2} - \vec{A2}, \vec{D}\vec{\beta}, \vec{X3} = \vec{X}\vec{\delta3} - \vec{A3}, \vec{D}\vec{\delta} \tag{13}$$

the wolf update place can be determined using (14).

$$\vec{X}(t + 1) = \frac{\vec{X1} + \vec{X2} + \vec{X3}}{3} \tag{14}$$

where $\vec{X}(t + 1)$ the updated the wolf location.

3.6.2. Modification grey wolf optimization (MGWO)

The path can be produced using rout planning algorithms and methods. In prewise section 2.6.1, the basic theory of the classical GWO has been explained. While in this section, the proposed modification on the classical GWO is presented. In particular, the MGWO algorithm stage includes the following parts:

a. First modification: adaptive adjustment approach of the control parameters

The first modification was proportion to [25], where non-linear parameter has been suggestion deepened upon cosine function and adaptive parameter \vec{a} has been organized, as given [13]:

$$\vec{a} = 1 - \cos \left(\left(1 - \frac{t}{t_{max}} \right)^k * \pi \right) \tag{15}$$

where t_{max} represents the max number of the iterations and t represent existing iteration. $k=2$ represents parameter of non-linear adjustment. The algorithms with 1st and 2nd adjustments have been dubbed as MGWO-1 and MGWO-2, respectively, for comparison purposes.

b. Second modification: adaptive variable weights (AVW) approach

Despite the modifications in the first modification, The averaging weight in (14) the effect is not sufficient in our implementation, thus suggesting another modification by (16) was needed. In classic GWO, the location vector regarding a grey wolf is guided equally through positions of β , α , and δ wolves as it has been presented by (14). In the presented study, a higher value of the weight is given to α wolf, which is

succeeded by β and δ wolves in suggested adjustment for calculating location vector regarding the gray wolf as:

$$\vec{X}(t+1) = \frac{w\alpha * X1 + w\beta * X2 + w\delta * X3}{6} \quad (16)$$

in which $w\beta$, $w\alpha$, and $w\delta$ represent weight values for β , α , and δ wolves, respectively. The adaptive variable weight (AVW) is utilized for reducing solution effort and time, and the technique for taking into consideration the weights, as well as the number of search iterations, is described in the next step [26]:

c. Step 1: selection weights' condition

The value regarding all of the wights is limited in (1) and summed of wight calculate through (17).

$$w\alpha + w\beta + w\delta = 1 \quad (17)$$

The weights of beta wolf $w\beta$, alpha wolf $w\alpha$, and the delta wolf $w\delta$ must always satisfy $w\alpha \geq w\beta \geq w\delta$. As the search progresses, the alpha weight will decrease from 1 to 0.33. From 0 to 0.33, beta and delta weights rise. The cos function may be entered in order to depict $w\alpha$ when we limit the angle θ to vary in range $[0, \cos^{-1}(0.33)]$.

d. Step 2: calculating $w\alpha, w\beta$ and $w\delta$

The following is a proposed new update positions technique with varying weights [26]:

$$w\alpha = \cos(\theta) \quad (18)$$

$$w\beta = 0.5 * \sin(\theta) * \cos(\varphi) \quad (19)$$

$$w\delta = 1 - w\alpha - w\beta \quad (20)$$

where the ϑ and Θ are angular and theta angle, respectively estimated by [26]:

$$\vartheta = 0.5 * \tan^{-1}(t) \quad (21)$$

$$\theta = \frac{2}{\pi} * \cos^{-1} 0.33 * \tan^{-1}(t) \quad (22)$$

where t is the existing iteration, and the pseudocode of MGWO is described in Algorithm 1. The algorithmic flowchart regarding the suggested algorithm has been depicted in Figure 5.

```

Algorithm 1. MGOW
Initialize the grey wolf population Xi (i=1,2,...,n)
Initialize  $\vec{a}$ , r1, r2
Calculate A and C
Calculate the fitness value of each search agent
 $\vec{X}\alpha$  = the best search agent
 $\vec{X}\beta$  = the second search agent
 $\vec{X}\delta$  = the third search agent
while t < Maxiteration do
  for each agent
    Update current search position according to (16)
  Endfor
  MGWO : Update  $\vec{a}$  by (15)
  Update A, C by (9), (10)
  Evaluate the fitness value of each search agents
  Update postion of  $\vec{X}\alpha, \vec{X}\beta,$  and  $\vec{X}\delta$  (16)
t=t+1
End while
Return  $\vec{X}\alpha$ 

```

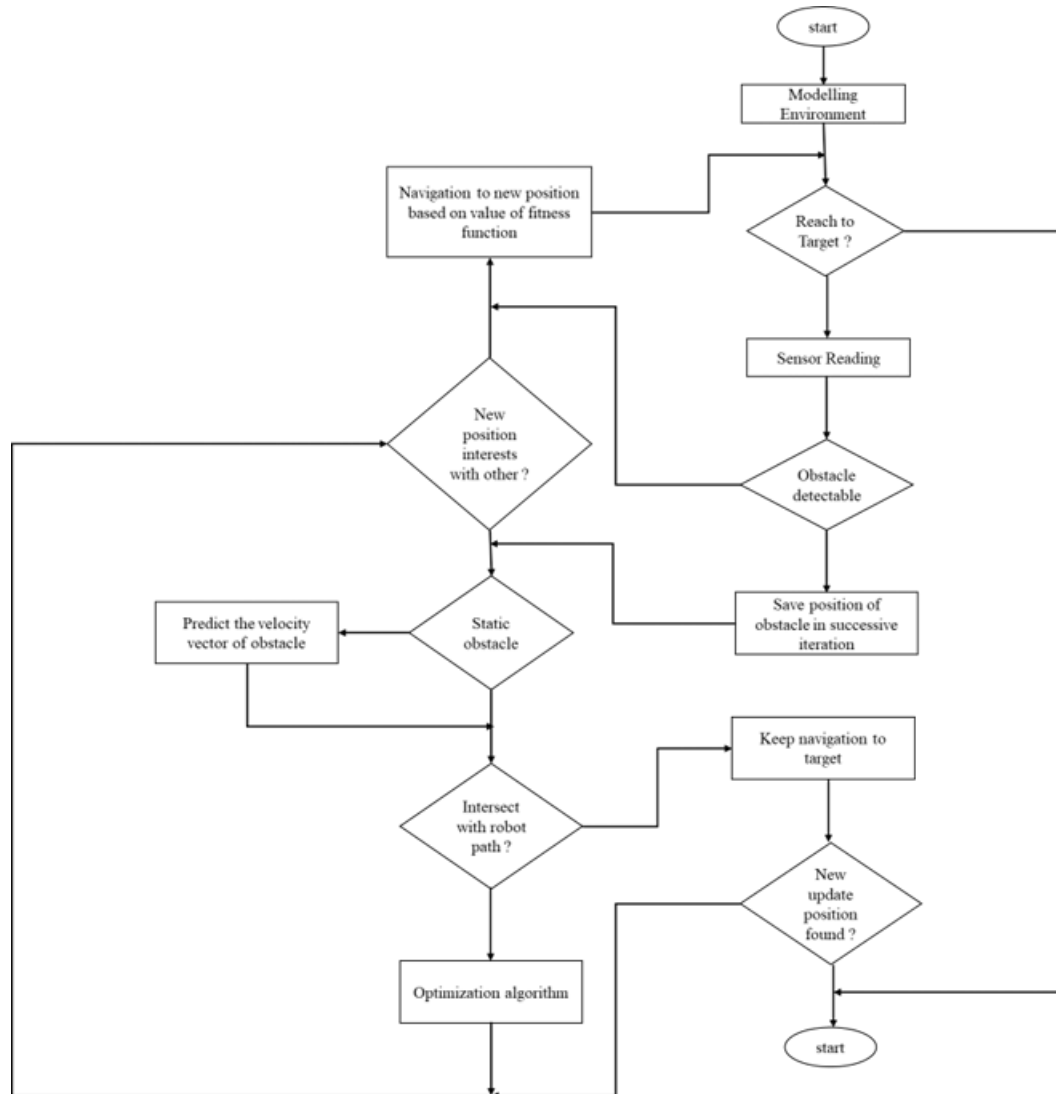


Figure 5. Flowchart of the proposed navigation algorithm in an unknown environment

4. RESULTS AND DISCUSSION

Feature of computer that using for test have, Win. 10 OS, Intel(R)Core (TM) i7-8550 U processor, 20 GB RAM and 1.80 GHz using for tests. where the simulation result test used of MATLAB package R2016 b. Table 1 lists the emulation parameters that were rated for the suggested approach.

Table 1. Factors tuning for simulations

Optimization parameters	
Number of the search agents	10
l_{max}	10
Dimension	1
Upper bound	5
Lower bound	-5

Use three cases for testing without obstacle, and (obstacle static, dynamic) unknown environment, the last example for the dynamic target, to further test performance of the MGWO algorithm upon specific issues. Previous research is compared to the suggested approach as seen in Figures 6-10 was comparison between fuzzy-wind driven optimization (FWDO) and APSO in subfigure (a) and MGWO result in subfigure (b). Table 1 summarizes the simulation findings for the static unknown map, where as Table 2 explains the reinforcement rates in the fitness function path lengths for the various approaches.

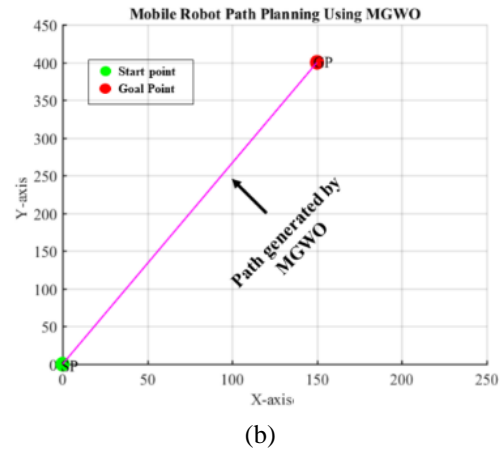
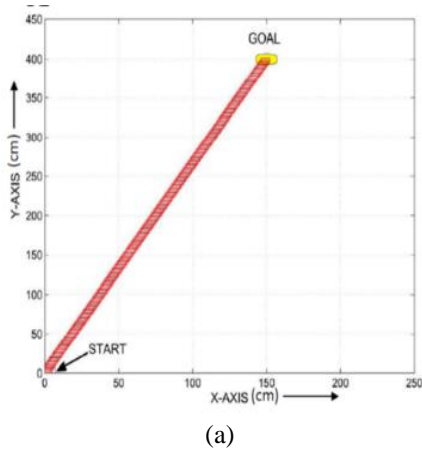


Figure 6. Map (1) simulation graph robot path planning without obstacle the starting point of (0,0) and an end-point at (400,150) in (a) FWDO [11] and (b) MGWO

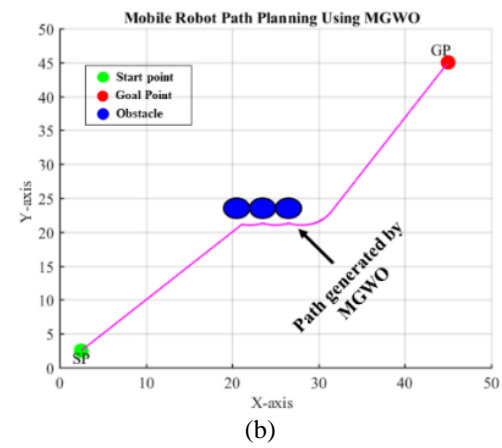
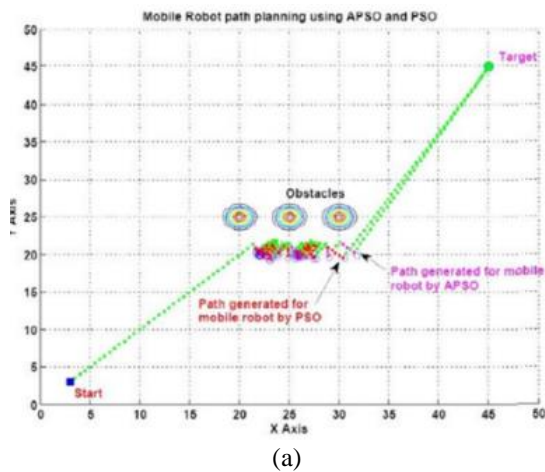


Figure 7. Map (2) simulation graph robot path planning in the narrow escaping environment with the starting point at (0,0) and end-point at (45,45) in (a) APSO [15] and (b) MGWO

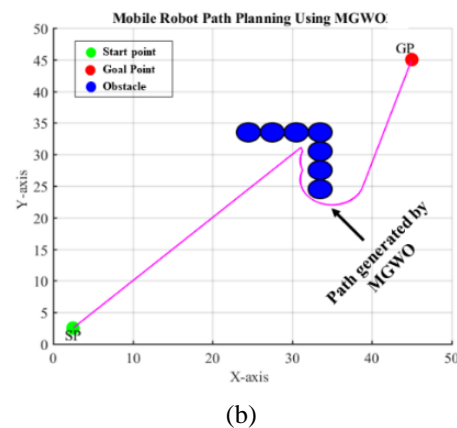
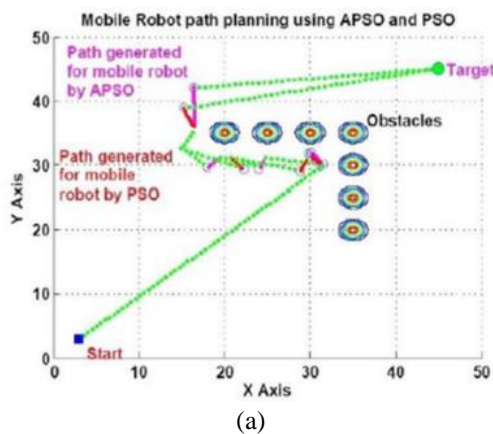


Figure 8. Map (3) simulation graph robot path planning in local minima environment with starting point at (0,0) and end-point at (45,45) in (a) APSO [15] and (b) MGWO

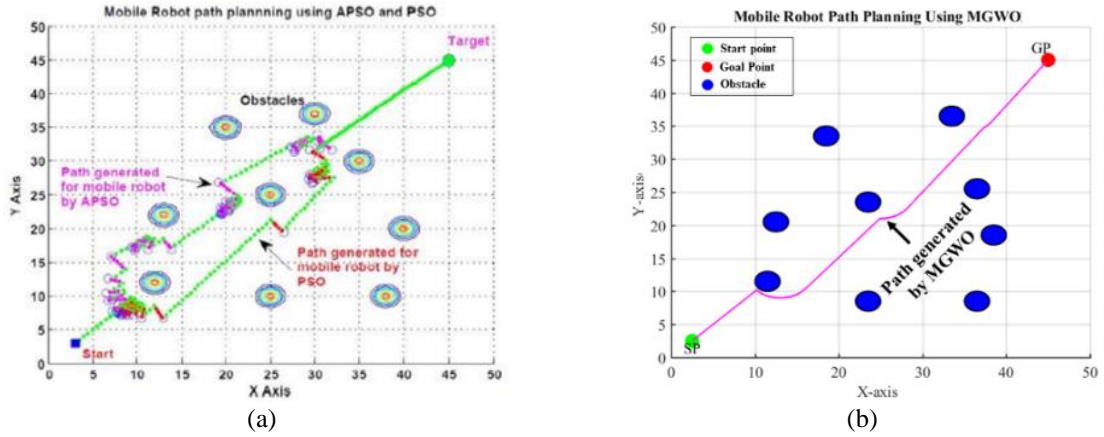


Figure 9. Map (4) simulation graph obstacle avoidance in trap condition with starting point at (0,0) and end-point at (45,45) in (a) APSO [15] and (b) MGWO

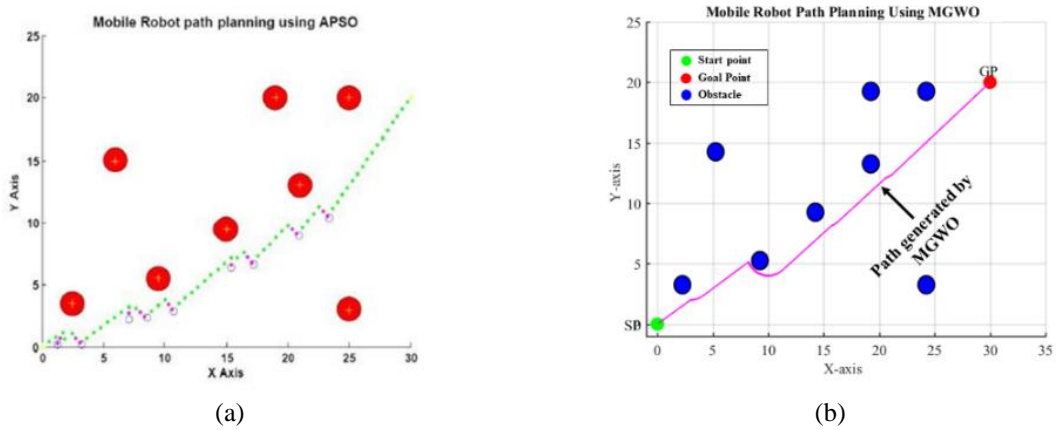


Figure 10. Map (5) simulation graph robot path planning in maze environment with starting point at (0,0) and end-point at (30,20) in (a) APSO [15] and (b) MGWO

As shown in Figure 6 the simulation results in without obstacle environment compared with FWDO, and in Figures 7-10 obstacle avoidance compared with APSO. Table 2 show the path length to reach the goal of the proposed algorithm in different maps with the latest literature APSO. From Table 2, it is obvious the enhancement rate in the path length. We can notice that the best result algorithm depends on the enhancement rate in path length. The best algorithm result was achieved by MGWO.

Table 2. The path length is covered by the latest literature and MGWO

No. Map	No. Figure	Type of algorithm	Average path length	
			Pixel	CM
MAP 1	Figure 6	Hybrid Fuzzy-WDO [11]	2237.31	59.19549375
		MGWO	2132.069	56.410992292
MAP 2	Figure 7	APSO [15]	1285.56	34.013775
		MGWO	1239.71275	32.800733177
MAP 3	Figure 8	APSO [15]	1784.71	59.19549375
		MGWO	1678.3087	44.405251021
MAP 4	Figure 9	APSO [15]	2237.31	59.19549375
		MGWO	2214.4432	58.590476333
MAP5	Figure 10	APSO [15]	470.45	12.447322917
		MGWO	466.1981	12.334824729

Table 3. Enhancement rate n the path lengths of our proposed algorithm

No. Map	Enhancement rates
MAP 1	4.70391%
MAP2	3.56633%
MAP3	5.96183 %
MAP4	1.02207%
MAP5	0.903794%

Figure 11 shows how mobile robots can readily avoid two moving obstacles in a dynamic environment utilizing MGWO, the beginning of the robot's mobility is seen in Figure 11(a), with dynamic obstacles moving in environment. The two obstacles in Figure 11(b) are extremely close to robot path. In order to prevent collisions, the robot detects an obstacle inside safety sensor range, as illustrated in Figure 11(c), and the algorithm begins working on avoidance by giving the robots a negative velocity, causing them to begin moving back to avoid obstacle 1. Figure 11(d) shows the robots steady progress toward the goal. Figure 12 displays the movement of mobile robots around a dynamic target. Figure 12 shows how unknown environment with a moving target.

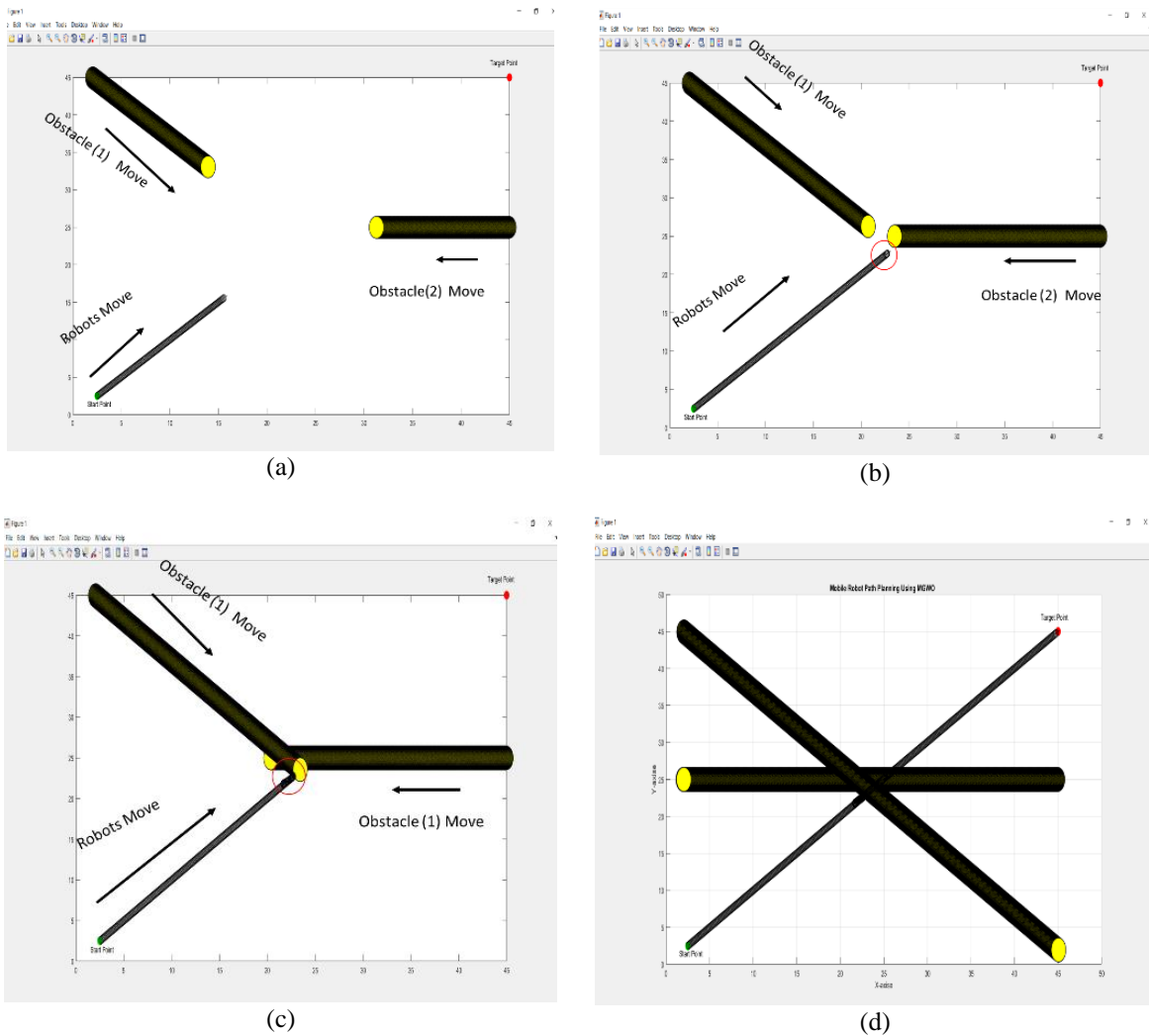


Figure 11. Simulation graph robot path planning with dynamic obstacle unknown environment test of proposed MGWO (a) start move, (b) obstacle near to robot, (c) obstacle far to robot, and (d) robots reach to the goal

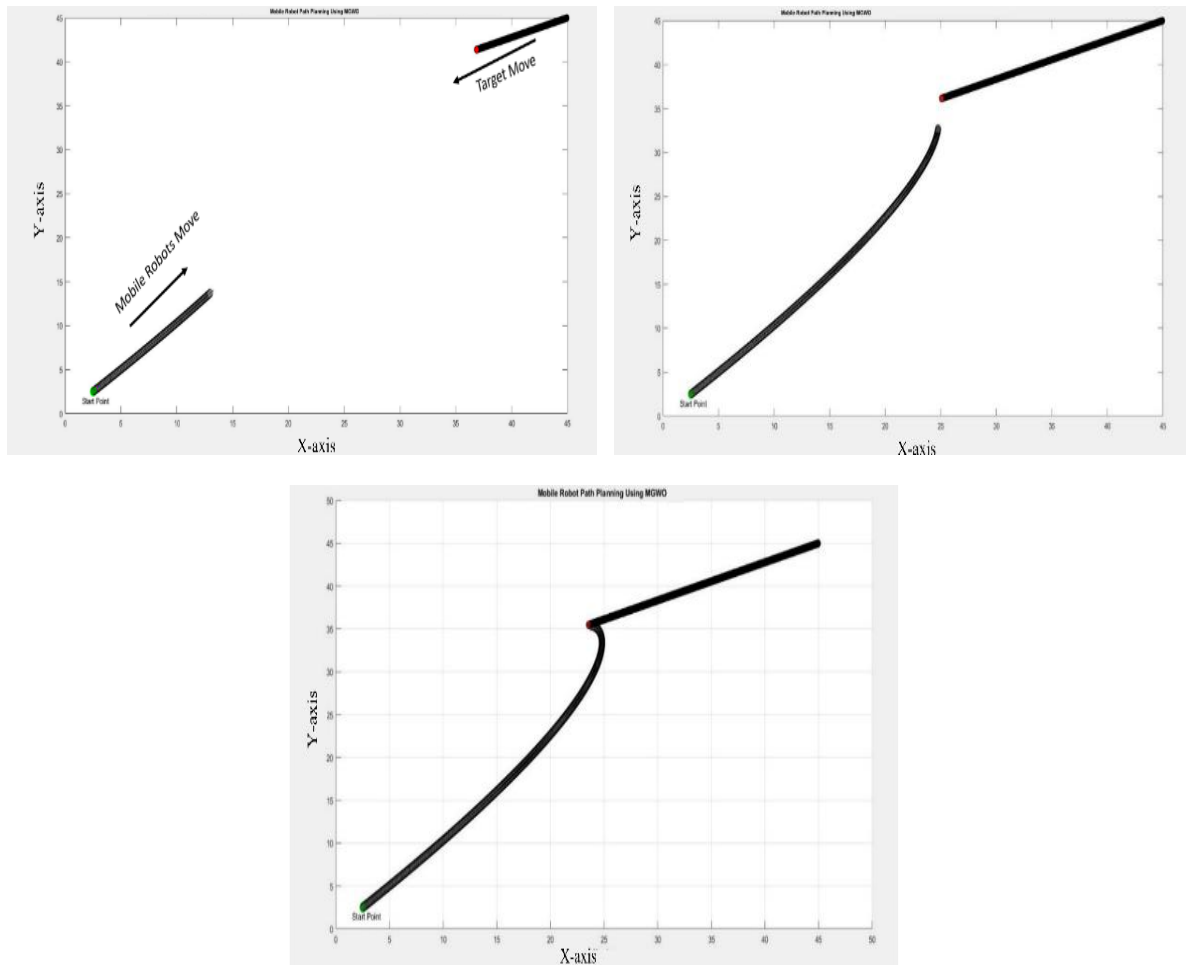


Figure 12. Simulation graph robot path planning with dynamic goal unknown environment test of proposed MGWO

The results that are shown in Figures 11 and 12 for the dynamic obstacle and the dynamic target the time took were 6.05453, 4.0969 with a path length of 30.459172 cm, 38.3 cm. The environment with a moving target has been successfully tested, in which the robot could successfully follow and reach the moving target. Moreover, the mobile robot has done well in avoiding moving obstacles in a dynamic environment




5. CONCLUSION

In an unknown environment with a range of dynamic and static obstacles, the path of effective navigation was created with the use of an online process and designed technique, which is MGWO. The adaptive adjusting technique regarding the MGWO control parameters achieves a balance between search as well as expansion capabilities related to GWO, also adaptive weighting is utilized for updating a wolf location, reducing time, and speeding up convergence. The suggested work is divided into two phases: the reading of sensors and the route account. Furthermore, this algorithm calculates the shortest route for a single mobile robot system with a successful obstacle avoidance mechanism in a short amount of time. In the case when there are moving obstacles and goals in the environment, the suggested method also performs best. The simulation results are compared with other intelligent approaches including the APSO with average enhancement path rates obtained by MGWO was 2.86%, and the enhancement path rate compared to the hybrid fuzzy-WDO was 4.70391%. Formation path planning is considered as one of the hottest topics nowadays which can be a common issue in UAV to be focus on in the future.




REFERENCES

- [1] O. A. R. A. Wahhab and A. S. Al-Araji, "An optimal path planning algorithms for a mobile robot," *IRAQI Journal of Computers, Communications, Control And Systems Engineering*, vol. 21, no. 2, pp. 44-58, 2021, doi: 10.33103/uot.ijccce.21.2.4.
- [2] A. M. Hasan and S. M. Rafaat, "Optimized formation control of the multi-agent system using PSO algorithm," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 20, no. 3, pp. 1591-1600, 2020, doi: 10.11591/ijeecs.v20.i3.pp1591-1600.
- [3] F. A. Raheem and U. I. Hameed, "Interactive heuristic D* path planning solution based on PSO for two-link robotic arm in a dynamic environment," *World Journal of Engineering and Technology*, vol. 7, no. 1, pp. 80-99, 2018, doi: 10.4236/wjet.2019.71005.
- [4] M. S. Abed, O. F. Lutfy, and Q. Al-Doori, "A review on path planning algorithms for mobile robots," *Engineering and Technology Journal*, vol. 39, no. 5A, pp. 804-820, 2021, doi: 10.30684/etj.v39i5A.1941.
- [5] M. I. Abdulkareem and F. A. Raheem, "Development of path planning algorithm using probabilistic roadmap based on ant colony optimization," *Engineering and Technology Journal*, vol. 38, no. 3A, pp. 343-351, Mar. 2020, doi: 10.30684/etj.v38i3A.389.
- [6] F. A. Raheem, S. M. Raafat, and S. M. Mahdi, "Robot path-planning research applications in static and dynamic environments," in *Earth Systems Protection and Sustainability*, Springer, 2022, pp. 291-325, doi: 10.1007/978-3-030-85829-2_12.
- [7] M. S. Abed, O. F. Lutfy, and Q. F. Al-Doori, "Online optimization application on path planning in unknown environments," *Journal Européen des Systèmes Automatisés*, vol. 55, no. 1, pp. 61-69, 2022, doi: 10.18280/jesa.550106.
- [8] R. T. Kamil, M. J. Mohamed, and B. K. Oleiwi, "Path planning of mobile robot using improved artificial bee colony algorithm," *Engineering and Technology Journal*, vol. 38, no. 9, pp. 1384-1395, Sep. 2020, doi: 10.30684/etj.v38i9A.1100.
- [9] N. A. Khaleq Zghair and A. S. Al-Araji, "A one decade survey of autonomous mobile robot systems," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 6, pp. 4891-4906, 2021, doi: 10.11591/ijece.v11i6.pp4891-4906.
- [10] P. Joshy and P. Supriya, "Implementation of robotic path planning using ant colony optimization algorithm," *2016 International Conference on Inventive Computation Technologies (ICICT)*, 2016, vol. 1, pp. 1-6, doi: 10.1109/INVENTIVE.2016.7823281.
- [11] A. Pandey and D. R. Parhi, "Optimum path planning of mobile robot in unknown static and dynamic environments using fuzzy-Wind Driven Optimization algorithm," *Defence Technology*, vol. 13, no. 1, pp. 47-58, 2017, doi: 10.1016/j.dt.2017.01.001.
- [12] A. Chhillar and A. Choudhary, "Mobile robot path planning based upon updated whale optimization algorithm," *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 2020, pp. 684-691, doi: 10.1109/Confluence47617.2020.9058323.
- [13] S. Hosseininejad and C. Dadkhah, "Mobile robot path planning in a dynamic environment based on a cuckoo optimization algorithm," *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, 2019, doi: 10.1177/1729881419839575.
- [14] B. K. Oleiwi, A. Mahfuz, and H. Roth, "Application of fuzzy logic for collision avoidance of mobile robots in dynamic-indoor environments," in *2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, 2021, pp. 131-136, doi: 10.1109/ICREST51555.2021.9331072.
- [15] P. K. Mohanty and H. S. Dewang, "A smart path planner for wheeled mobile robots using adaptive particle swarm optimization," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 43, no. 2, pp. 1-18, 2021, doi: 10.1007/s40430-021-02827-7.
- [16] S. K. Malu and J. Majumdar, "Kinematics, localization, and control of differential drive mobile robot," *Global Journal of Research In Engineering*, vol. 14, no. 1, 2014.
- [17] A. Madhloom, F. A. Raheem, and A. R. Kareem, "A modified kalman filter-based mobile robot position measurement using an accelerometer and wheels encoder," *Engineering and Technology Journal*, vol. 40, no. 1, pp. 267-274, Jan. 2022, doi: 10.30684/etj.v40i1.2082.
- [18] J. Liu, J. Yang, H. Liu, X. Tian, and M. Gao, "An improved ant colony algorithm for robot path planning," *Soft Computing*, vol. 21, no. 19, pp. 5829-5839, 2017, doi: 10.1007/s00500-016-2161-7.
- [19] F. A. Raheem and M. I. Abdulkareem, "Development of a* algorithm for robot path planning based on modified probabilistic roadmap and artificial potential field," *Journal of Engineering Science and Technology*, vol. 15, no. 5, pp. 3034-3054, 2020.
- [20] A. A. A. Rasheed, M. N. Abdullah, and A. S. Al-Araji, "A review of multi-agent mobile robot systems applications," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 4, 2022, doi: 10.11591/ijece.v12i4.pp3517-3529.
- [21] Z. D. Hussein, M. Z. Khalifa, and I. S. Kareem, "Optimize path planning for medical robot in iraqi hospitals," *Engineering and Technology Journal*, vol. 33, no. 5, pp. 1009-1022, Jun. 2015.
- [22] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46-61, 2014, doi: 10.1016/j.advengsoft.2013.12.007.
- [23] S. Saremi, S. Z. Mirjalili, and S. M. Mirjalili, "Evolutionary population dynamics and grey wolf optimizer," *Neural Computing and Applications*, vol. 26, no. 5, pp. 1257-1263, 2015, doi: 10.1007/s00521-014-1806-7.
- [24] F. J. Zwayyer, A. A. Abbood, and J. F. Hussein, "Optimization of automatic generation control and economic load dispatch for two area six unit interconnected power system," *Engineering and Technology Journal*, vol. 39, no. 10, pp. 1610-1624, Oct. 2021, doi: 10.30684/etj.v39i10.2158.
- [25] W. Gai, C. Qu, J. Liu, and J. Zhang, "An improved grey wolf algorithm for global optimization," in *2018 Chinese Control And Decision Conference (CCDC)*, 2018, pp. 2494-2498, doi: 10.1109/CCDC.2018.8407544.
- [26] G. Zheng-Ming and J. Zhao, "An improved grey wolf optimization algorithm with variable weights," *Computational Intelligence and Neuroscience*, vol. 2019, 2019, doi: 10.1155/2019/2981282.




BIOGRAPHIES OF AUTHORS

Mustafa Salah Abed    received a B.Sc. degree in mechatronics engineering at the Control and Systems Engineering Department, the University of Technology, Iraq in 2016 He is currently studying for an M.Sc. degree in mechatronics engineering at the Control and Systems Engineering Department, University of Technology. His research interests include intelligent control, artificial intelligence algorithms, and robotics. He can be contacted at email: cse.19.07@grad.uotechnology.edu.iq.



Prof. Dr. Omar Farouq Lutfy    has finished a B.Sc. degree in Control and Systems Eng. from Univ. of Technology in 2000 and a M.Sc. degree in Mechatronics Engineering for the Univ. of Technology in 2003. He has finished his Ph.D. in control and automation engineering at the Department of Electrical and Electronic Engineering, University Putra Malaysia (UPM), Malaysia in 2011. Currently, he has a Professor in the Control and Systems Engineering Department, Univ. of Technology, Iraq. His current research securities, artificial intelligence algorithms, robotics, and system identification. He can be contacted at email: 60157@uotechnology.edu.iq.



Lect. Dr. Qusay Fadhel Al-Doori    has finished B.Sc. and MSc degree in Electrical Engineering from the University of Baghdad, College of Engineering, Electrical Engineering Department in 1996 and 2000. He has finished his Ph.D. at the University of Salford, the United Kingdom in 2017 (UPM), Currently he has a Lecture in Control and Systems Engineering Department, University of Technology, Iraq. His current research securities, artificial intelligence algorithms, robotics Codedc Electronic circuits, paralyzation method, power dissipation in Cmos Circuit. He can be contacted at email: qusay.f.hasan@uotechnology.edu.iq.