3D Path Planning Method for Multi-UAVs Inspired by Grey Wolf Algorithms

Farzad Kiani[1], Amir Seyyedabbasi[2], Royal Aliyev[3], Mohammed Ahmed Shah[4], Murat Ugur Gulle[3]

[1] Software Engineering Dept., Istinye University, Turkey
[2] Computer Engineering Dept., Beykent University, Turkey
[3] Computer Engineering Dept., Istanbul Aydin University, Turkey
[4] Computer Engineering Dept., Istanbul Ayvansaray University, Turkey
farzad.kiyani@gmail.com, amir.seyyedabbasi@gmail.com, raliyev@stu.aydin.edu.tr, ahmed.shah@gmail.com, muratgulle@stu.aydin.edu.tr

Abstract

Efficient and collision-free pathfinding, between source and destination locations for multi-Unmanned Aerial Vehicles (UAVs), in a predefined environment is an important topic in 3D Path planning methods. Since path planning is a Non-deterministic Polynomial-time (NP-hard) problem, metaheuristic approaches can be applied to find a suitable solution. In this study, two efficient 3D path planning methods, which are inspired by Incremental Grey Wolf Optimization (I-GWO) and Expanded Grey Wolf Optimization (Ex-GWO), are proposed to solve the problem of determining the optimal path for UAVs with minimum cost and low execution time. The proposed methods have been simulated using two different maps with three UAVs with diverse sets of starting and ending points. The proposed methods have been analyzed in three parameters (optimal path costs, time and complexity, and convergence curve) by varying population sizes as well as iteration numbers. They are compared with well-known different variations of grey wolf algorithms (GWO, mGWO, EGWO, and RW-GWO). According to path cost results of the defined case studies in this study, the I-GWO-based proposed path planning method ($PP_{I\text{-}GWO}$) outperformed the best with %36.11. In the other analysis parameters, this method also achieved the highest success compared to the other five methods.

Keywords: Path planning, Multiple UAV, Mobile robots, Metaheuristics

1 Introduction

Mobile robots are cutting-edge technologies that can be employed in numerous unprecedented research areas such as Internet of Things (IoT) [1-2], military [3], agriculture [4], and health [5]. These robots have also been used in Vehicle Ad-hoc Networks (VANETs) [6-7] and Flying Ad-hoc Networks (FANETs) [8], a subset of the Mobile Ad-hoc networks [9-10], Internet of Drones (IoD) [11] and Internet of Vehicles (IoV) [12] in the IoT category. In autonomous routing techniques of these systems, one of the aims is to find safe paths in shortest possible time, effectively using the resources. These robots consist of sensors and actuators. Furthermore, each robot (node) has a processor and a memory. As such it is adequate to state that these devices can function as an all-in smart agent. Therefore, artificial intelligence-based techniques can be easily implemented using these smart nodes. When appropriated in interconnected and successive systems, it is vital to plan a path for every single mobile autonomous device such as UAVs and drones. In this paper, the proposed path planning mechanisms, inspired by Incremental Grey Wolf Optimization (I-GWO) [13] and Expanded Grey Wolf Optimization (Ex-GWO) [13] are realized for each autonomous mobile robot (e.g. UAVs) in different environments, containing various obstacles without any collisions. Each proposed method attempts to find an almost optimal path by eliminating the process of creating complex environment models based on stochastic approaches. They can be faster and more successful in finding the most suitable solutions.

In literature, many studies have been conducted on path planning and mobile autonomous vehicles in recent years [14-19], especially on three-dimensional path planning [20-23]. The general taxonomy of 3D path planning methods is generally consisting of four basic areas; *sampling-based, node-based optimal algorithm, mathematical model-based and nature-inspired algorithms* [24]. The methods in the first three categories suffer from high time complexity and local minima capture [25], especially where mobile robots face multiple constraints when planning a path. Therefore, the nature-inspired algorithms, especially metaheuristics, can be the most appropriate methods in 3D path planning. So, in this section, mainly

*Corresponding Author: Farzad Kiani; E-mail: farzad.kiyani@gmail.com
DOI: 10.53106/160792642021072204003

metaheuristic-based studies are discussed.

In [14] has been presented a new method for a mobile robot in an uncertain environment based on Firefly Algorithm (FA). It solves the challenges of navigation by avoiding the random movement of fireflies and minimizing the computational cost. The authors defined an objective function, which is controlled by the trial and error method. With this function, the paths to be chosen are decided. In this study, which aim is crowded environments, the choice of the obstacle closest to each station is required for path planning and an equation has been defined for this purpose. However, this equation, which contains very few parameters, may not be successful in real and complex environmental conditions. Moreover, this study focused solely on 2D path planning. In [22] has been proposed ground robot based on a new version of Ant Colony Optimization (ACO) for 3D path planning. It defines a new phenomenon update mechanism and constructs various paths between the initial and target points of a robot. It avoids obstacles and is used for solving the easily falling into local optimum and long search times in 3D path planning problems. Performance analysis could not be comprehensive by comparing this proposed method with only the basic ACO method. In addition, generally, it has a higher time and space complexity than GWO-based methods, this is due to the nature of ACO.

The GWO algorithm may be more likely to be successful than other metaheuristic methods in this type of problem on various parameters due to its working mechanism. One of the most important and effective of the last studies is a GWO-based 3D path planning method. Some researchers [26] have proposed a new 3D path planning method for multi-UAVs. The main important issue is obstacle avoidance that the authors in this study, focused on obstacle avoidance and their main goal is to find the path with minimum cost. In their work, they used the GWO algorithm to find an optimal path with minimal cost. According to the results of the study, the better path cost with best time complexity in the GWO-based method in comparison to other methods such as Dijkstra, A*, D*, and a few other famous metaheuristic-based methods is obtained. In [26] there are three different maps with varying number of obstacles. Euclidean distance between stations, visited by UAVs, is used to calculate the cost of the path. Most of the obstacles in their study are located in the center of the maps so that the UAVs do not require much effort to reach the destination. According to this paper, GWO-based methods, with the specific features and advantage of the nature of its algorithm, perform a more balanced and better performance in similar problems. Therefore, GWO-based algorithms are sought after in many research and application areas due to their balanced behavior amongst various metaheuristic algorithms. For this reason, this paper uses two variants of GWO and propose two novel path planning methods for obstacle detection and avoidance, random movement avoidance, and optimal pathfinding. In addition, the used methods to compare with our proposed methods in this study in performance analysis will be GWO-based methods. Despite the advantages of the study, [26], two new methods that perform better in various sizes and conditions environments have been proposed in our paper. Ex-GWO based path planning method ($PP_{Ex\text{-}GWO}$) performs more successfully in larger and crowded environments, and I-GWO based path planning method ($PP_{I\text{-}GWO}$) method outperforms good results in smaller and less populated environments.

The rest of this paper is planned as follows: In Section 2, scenarios are defined including UAV positions, environment, and obstacle maps defined together with the necessary definitions regarding the problem. For this, the fitness function is defined where UAVs can find the appropriate path. The proposed methods are described together with the relevant problems in Section 3. In section 4, simulation results are analyzed and discussed. The final section contains the study's conclusions and future works.

2 Definitions

The main goal of a 3D path planning method is to find optimal paths in a predefined environment, containing various obstacles without any collisions. The difficulties of path planning in a 3D environment, unlike 2D path planning, increases exponentially due to the inherent kinematic nature of the environment. Furthermore, finding an optimal 3D path planning is a Non-Deterministic Polynomial-Time (NP-hard) problem. Because the suitable path planning mechanism arises from examining all possible paths. However, this can be a very costly process, so this study is inspired by metaheuristic algorithms.

The obstacles in the environment are in a different position. In the movement space, UAVs must consider the Z dimension alongside the X and the Y dimensions. Each UAV in the environment find a trajectory between the initial (source) and final (destination) stations. source and destination denote relative coordinates of the source (X_{source}, Y_{source}, Z_{source}) and the destination ($X_{destination}$, $Y_{destination}$, $Z_{destination}$) positions on the map. Each path has a cost during motion from a source to a destination. There are different parameters that determine this cost between the two points. In most studies, cost is calculated using consumption of energy, altitude, air pressure, Euclidean distance, and velocity. In this paper, the cost between initial and final stations (states) is calculated based on the sum of the possible tuples Euclidean distance. In each map, the positions of the mobile robot marking its trajectory can be defined using positions [p_s, p_1, p_2, …, p_D]. The cost of the optimal path, where each optimal path is the sum of distances between tuples from source to the destination is obtained based on Eq.1. Where $distance_{i,j}$ demonstrates the distance between two stations [27].

$$Cost_{(i,j)} = \sum_{i=s}^{j=D} distance_{i,j} \quad \textbf{(1)}$$

Figure 1 shows the trajectory between the initial and final stations. In the trajectory, there are some stations that the UAVs movement over stations. The *S* indicates the initial station, and the *D* is the final station. The p presents the possible stations that UAVs can move in the environment. The optimized path is generated without collision with the obstacles. In each proposed method in this study, the cost of each tuple on the path is calculated by a fitness function. Therefore, it is possible to find an optimum or close to an optimal path with a minimum cost between two points. The possible optimal path with the proposed methods is represented as a sample in Figure 1. The structure of the proposed methods is described in detail in section 3.

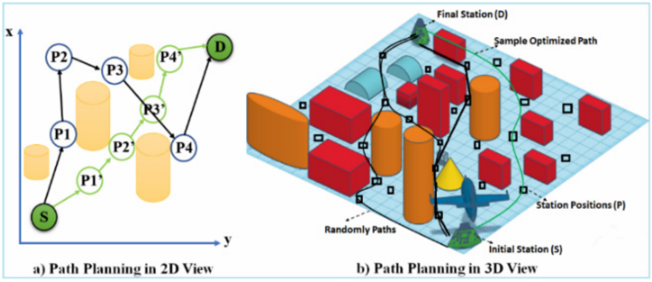


Figure 1. The generated sample optimal path between initial and final stations

2.1 Maps

Typically, the first step in path planning is to represent the workspace as a map. The presence of obstacles in the maps makes the task of finding an optimal path a bit complex for the UAVs, but with this definition, the scenario becomes more realistic. Where the challenge is to avoid the obstacles and to reach the final destination giving minimum costs. In this paper, two maps have been considered to evaluate proposed methods; a medium map and a large map. In Table 1, map boundary for both maps is presented. Furthermore, there are three UAVs with distinct initial and final positions. These three-dimensional points of UAVs are given in Table 2. Furthermore, the number of obstacles for each map is different, positions of the obstacle are listed in Table 3.

Table 1. 3D map boundary

Map	Start boundary	End Boundary
Medium map	(0,0,0)	(100,100,100)
Large map	(0,0,0)	(150,150,150)

Table 2. 3D map UAV initial and final positions

	Small map			Large map		
	UAV 1	UAV 2	UAV 3	UAV 1	UAV 2	UAV 3
Initial	(0, 0, 0)	(0, 0, 0)	(0, 0, 0)	(0, 0, 0)	(0, 50, 0)	(0, 100, 0)
Final	(50, 50, 50)	(150, 150, 150)	(150, 150, 150)	(150, 150, 150)	(100, 50 100)	(100, 0, 100)

Table 3. 3D obstacles coordinate for each map

Obstacle numbers	Medium map	Large map
1	(5, 7.5,4) - (10,20,15)	(5, 7.5,4) - (10,20,15)
2	(20, 5, 10) - (44,44,36)	(20, 5, 10) - (44,44,36)
3	(8, 30, 4) -(10,34,28)	(1,5,14) – (3,6,16)
4	(17, 14, 17) - (19,16,21)	(0,15,1) − (7,16,3)
5	(22, 5, 0) - (23, 5, 100)	(0,18,4) − (10,13,6)
6	(15, 5, 0) - (16, 0, 100)	(0,7,2) − (4,5,5)
7	(19, 1, 0) - (19, 6, 100)	(4,0,2) − (7,5,6)
8	(25, 6, 0) - (25, 8, 100)	(0,15,0) − (10,20,1)
9		(7,8,9) – (10,11,12)
10		(5,89,140) - (5,90,144)
11		(9,15,2) – (12,19,6)
12		(85, 15, 23) – (86, 16, 26)

3 Proposed Methods

As mentioned before, one of the most significant and contemporary problems in robotics is 3D path planning for mobile robots (e.g. UAVs). It is necessary to find an optimal (or close to optimal) path between initial and final stations for robots to move without any intervention. The obtained path contains the tuple of solutions. The sum of each possible tuple in the obtained path gives the path cost and the least costly path is used as the optimal solution for each method. As such, in this paper new 3D path planning methods are presented for autonomous UAVs. This study proposes two novel path planning methods for obstacle detection and avoidance, random movement avoidance, and optimal pathfinding. These methods are also intended to be useful for various purposes in different environments. The name of these methods, which have been inspired by the Incremental Grey Wolf Optimization (I-GWO), and Extended Grey Wolf Optimization (Ex-GWO) metaheuristic algorithms [13], are Path Planning based on I-GWO ($PP_{I\text{-}GWO}$) and Path Planning based on Ex-GWO ($PP_{Ex\text{-}GWO}$). In these proposed methods, each UAV will use the equations of the relevant metaheuristic algorithm, which are explained in the below subsection, to decide the most appropriate selection at the next station transition from the current to each next station. In each step, the cost of each possible path between tuple stations is calculated. This process will continue till reaches to a destination station. In the end, the obtained path cost of selected all stations will be also calculated by the defined fitness function.

3.1 I-GWO and Ex-GWO: Grey Wolf Algorithms

These metaheuristic algorithms are inspired by grey wolves in their natural habitat, where their natural behavior is mathematically modeled. The main behavioral traits of the wolves are encircling, hunting, and attacking the prey. There are four types of wolves in each pack; alpha (α), beta (β), delta (δ), and omega (ω). Each wolf has different responsibilities in the pack. The wolf's responsibilities are different in the group. Grey wolves encircle the prey during the hunt. In both algorithms, the omega wolves, which are a set in the pack, update their positions according to the defined equations. To model this behavior mathematically in I-GWO and Ex-GWO, Eq. 2 and 3 are proposed. Both equations are applied similar to these algorithms. Where t indicates the current iteration, T demonstrates the total iterations, $\vec{X}$ indicates the position vector of a wolf. Also, D is a vector that depends on the location of the target. The coefficient vectors $\vec{A}$, and $\vec{C}$ are considered to lead in encircling their prey (Eq. 4 and 5). These parameters control the tradeoff between exploration and exploitation phase in both I-GWO and Ex-GWO. Also, $\vec{a}$ is linearly decreased from 2 to 0 over the courses of iteration. It is used to get closer to the solution range and r_1 and r_2 are the random vectors in range of [0, 1]. The $\vec{a}$ is defined by Eq. 6 for the I-GWO and Eq. 7 for the Ex-GWO algorithm. Besides, there are two positions for each leader (alpha) in the pack; attack, or search. When $|\vec{A}|$ is less than 1, the wolves in the pack attack to hunt, otherwise they try to find prey to be hunted. In this way, these algorithms try to find possible solutions in the whole area.

$$\vec{D} = |\vec{C} \cdot \vec{X_p}(t) - |\vec{X}(t)|, \tag{2}$$

$$\vec{X}(t+1) = \vec{X_p}(t) - \vec{A} \cdot \vec{D}, \tag{3}$$

$$\vec{A} = 2\vec{a} \cdot \vec{r_1} - \vec{a}, \tag{4}$$

$$\vec{C} = 2 \cdot \vec{r_2}, \tag{5}$$

$$\vec{a} = 2(1 - \frac{t^2}{T^2}), \tag{6}$$

$$\vec{a} = 2(1 - \frac{t}{T}) \tag{7}$$

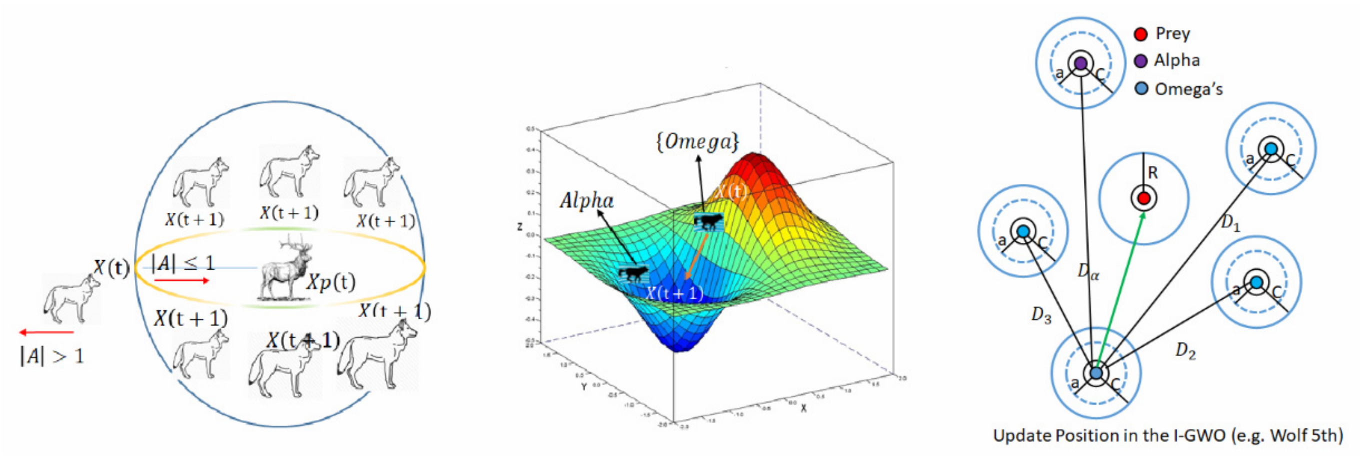
The hunting mechanism I-GWO algorithm for the wolves in the pack is based on the leader (alpha) wolf. In the I-GWO algorithm, the wolf at the top of the hierarchy is the leader wolf and the remaining wolves in the pack are others. In the prey hunting mechanism, the other wolves in the pack followed the leader wolf. It is assumed that the leader (alpha) has the best knowledge about the prey position. In this way, the remaining wolves in the pack update their own position based on this leader position. The other wolves in the pack that should update their own position to prey on the prey follow Eq. 8 to 10.

$$\vec{D_\alpha} = |\vec{C_\alpha} \cdot \vec{X_\alpha} - \vec{X}|, \tag{8}$$

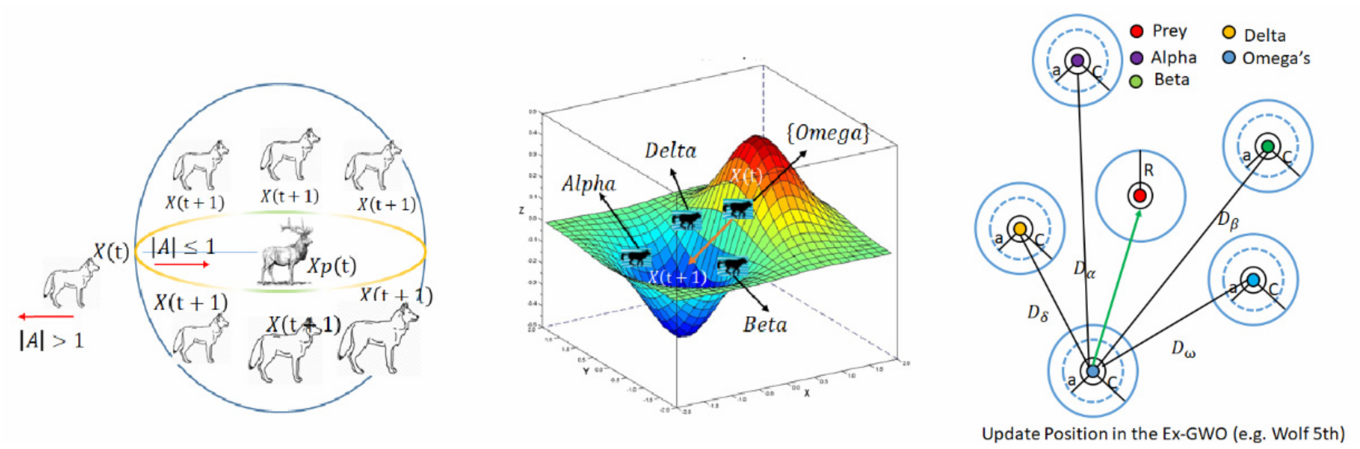
$$\vec{X_1} = \vec{X_\alpha} - \vec{A_1} - \vec{D_\alpha}, \tag{9}$$

$$\vec{X_n}(t+1) = \frac{1}{n-1} \sum_{i=1}^{n-1} X_i(t); n = 2, 3, \ldots, m \tag{10}$$

In Ex-GWO first three wolves; alpha, beta, and delta wolves, all have the best knowledge about the prey. The fourth wolf in the pack updates its own position based on alpha, bets, and delta wolves. The fifth wolf updates its own position using the positions of the first three wolves and the fourth wolf. As such, the nth wolf updates its own position based on the first three wolves in the pack and the n-3 wolves before it. So, the wolves benefit more from the pack's knowledge in order to hunt and attack. In Ex-GWO, the hunting mechanism follows Eq. 11 to 13. Figure 2 shows the working mechanisms of both algorithms considering exploration and exploitation phases.



(a) I-GWO



(b) Ex-GWO

Figure 2. Working mechanism by considering exploration and exploitation

$$\vec{D}_1 = |\vec{C}_1 \cdot \vec{X}_1 - \vec{X}|, \vec{D}_2 = |\vec{C}_2 \cdot \vec{X}_2 - \vec{X}|, \vec{D}_3 = |\vec{C}_3 \cdot \vec{X}_3 - \vec{X}|, \quad \textbf{(11)}$$

$$\vec{X}_1 = \vec{X}_1 - \vec{A}_1 \cdot \vec{D}_1, \vec{X}_2 = \vec{X}_2 - \vec{A}_2 \cdot \vec{D}_2, \vec{X}_3 = \vec{X}_3 - \vec{A}_3 \cdot \vec{D}_3, \quad \textbf{(12)}$$

$$\vec{X}_n(t+1) = \frac{1}{n-1}\sum_{i=1}^{n-1} X_i(t); n=4, 5, \ldots, m \quad \textbf{(13)}$$

3.2 $PP_{I\text{-}GWO}$ and $PP_{Ex\text{-}GWO}$: Path Planning Methods based on I-GWO and Ex-GWO

The structures and defined equations for the above metaheuristic algorithms are used to propose 3D path planning methods. In the proposed methods of study, each UAV starts from the initial stations based on the predefined configurations. Then, the next station for UAVs should be selected. In this step, the next station of each UAV is elected according to the defined suitable equations in $PP_{I\text{-}GWO}$ and $PP_{Ex\text{-}GWO}$. The selected stations are aimed to be the choices that will create the optimal path. With each proposed method, all stations are selected and their costs are calculated. According to the working mechanisms of the proposed methods, a path is determined for each wolf at the end of maximum defined iteration and the path of alpha wolf is accepted as the best solution to each UAV. The optimal selection of stations is obtained in $PP_{I\text{-}GWO}$ are based on Eq. 10 while the $PP_{Ex\text{-}GWO}$ are based on Eq. 13, as is presented in Algorithm 1. The path achieved will be a collision-free and optimal cost path. Some of the advantages of the proposed methods are simplicity, flexibility, derivation-free mechanism, and local optima avoidance. In addition, fewer parameters are needed to control them. Therefore, they may be effectively used for real problems with expensive or unknown derivative information. The mechanism of the iterations of both methods is presented in Figure 3 as an example of optimal path selection between each UAV between the *S* and *D* stations.

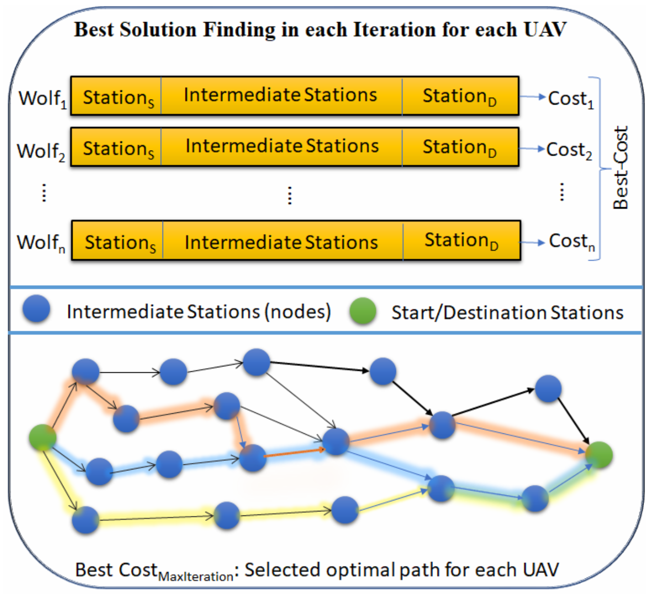


Figure 3. At the end of iterations, the best solution is chosen in proposed methods

For the realization of the proposed methods, we will use the map information and fitness function defined in section 2. The map, obstacles, and initial and final stations of each UAV are described in the previous section. In this study, the number and positions of stations (UAV's stopovers) and obstacles are predefined similar to other studies in literature [13, 23-26]. In proposed methods a station pool is employed, which includes defined stations. In the first step, these stations are randomly created, furthermore, they can be predefined by the user. Figure 4 shows a 3×n matrix depicting a station pool as a sample. Each station in the pool is a possible position for a UAV that can choose as the next station. This pool is used to control the UAV movement in the area. Besides, by using the information of this pool, it may be possible to avoid obstacles.

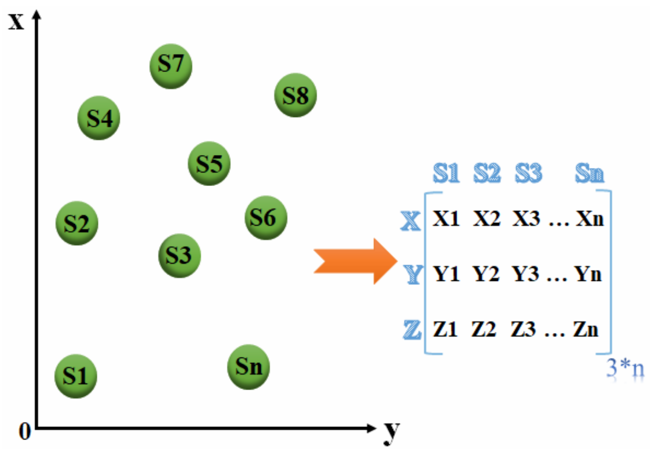


Figure 4. Stations pool that each state has ordinates

Primarily, the proposed methods initialize the random position matrix. Each row of position matrix defines the path, and the columns represent the number of steps, in the path, to the destination. These number of steps are denoted as p. The (x_n^m, y_n^m, z_n^m) presents coordinates of each station where m is the aforementioned index of stations and n is the number of search agents in each method (Table 4). The search agents are the configuration parameter of the metaheuristic algorithms. Then, for each metaheuristic algorithm, a search space, based on the position matrix, is initialized. The search space is shown in Table 5, which represents the distance between tuples. In this table, each row represents a path length. Each element of the row shows the distance between two points as $d_{(i,j)}^n$, where i is the current station and j is the previous station. Furthermore, n is in the number of search agents. Besides, in the proposed methods is calculated the path cost based on a fitness function that was presented in the Eq.1.

Table 4. The position matrix of each path

path	1	2	⋯	p
1	(x_1^1, y_1^1, z_1^1)	(x_1^2, y_1^2, z_1^2)	⋯	(x_1^p, y_1^p, z_1^p)
2	(x_2^1, y_2^1, z_2^1)	(x_2^2, y_2^2, z_2^2)	⋯	(x_2^2, y_2^2, z_2^2)
⋮	⋮	⋮	⋮	⋮
n	(x_n^1, y_n^1, z_n^1)	(x_n^2, y_n^2, z_n^2)	⋯	(x_n^p, y_n^p, z_n^p)

Table 5. The search space that represents distance between tuples

path	1	2	⋯	p
1	$d_{(1,s)}^1$	$d_{(2,1)}^1$	⋯	$d_{(p,p-1)}^1$
2	$d_{(1,s)}^2$	$d_{(2,1)}^2$	⋯	$d_{(p,p-1)}^2$
⋮	⋮	⋮	⋮	⋮
n	$d_{(1,s)}^n$	$d_{(2,1)}^n$	⋯	$d_{(p,p-1)}^n$

In the next step, the proposed methods calculate the distance between tuple for each station in the pool. In this case, we have a distance cost (d) between current station and next candidate stations. The d includes two values, first is the distance between current and next station, and second is the distance between next and destination station. However, the metaheuristic algorithms find one best solution for next station of each current station. If the distance of possible next stations is smaller than the obtained value from metaheuristic algorithms (w), the relevant station with minimum value is selected as elected next station. Otherwise, the UAV chooses the achieved solution of the metaheuristic algorithms as next station (Algorithm 1). The methods aim to reduce the cost of each path. The proposed methods try to find the optimal path with minimum cost for multi-UAVs. In this study, there are three UAVs that have a dissimilar start (initial) and final (destination) stations. The results obtained from this method are explained in the analysis and results section. Pseudocodes of proposed path planning methods ($PP_{I\text{-}GWO}$ and $PP_{Ex\text{-}GWO}$) can be found in

Algorithm 2 and Algorithm 3, respectively.

Algorithm 1. Pseudocode of node (station) selection

1. Station is array of candidate stations
2. w= distance obtained from metaheuristics *//Eq. 10 and 13*
3. d=The list of distances
4. **For** each station (*i*) in pool
5. d_i = distance between current and next stations + distance between next and destination stations
6. **End For**
7. MinDist=Min(d) *// Min function indicates minimum distance in the list*
8. **if** (MinDist < w)
9. Select station with minimum distance as next station
10. **Else**
11. Select station by metaheuristics as next station
12. **End if**

Algorithm 2. Pseudocode of path planning using I-GWO ($PP_{I\text{-}GWO}$)

1. Initialize grey wolf population Xi (i=1, 2, …, n)
2. Initialize A, C and a *//Eq. 4, 5, and 6*
3. Initialize Positions matrix and search space
4. Calculate fitness of each agent *//Eq. 1*
5. X_α = best search agent
6. **While** (t< Max number of iterations)
7. **For** each search agent
8. Update position of current search agent *//Alg. 1*
9. **End For**
10. Update a, A and C
11. Calculate fitness of all search agents
12. Update X_α
13. Insert X_α to best Positions matrix
14. Update search space matrix
15. t = t +1
16. **End While**
17. Return X_α

Algorithm 3. Pseudocode for path planning using Ex-GWO ($PP_{Ex\text{-}GWO}$)

1. Initialize grey wolf population X_i (i=1, 2, …, n)
2. Initialize A, C and a *//Eq. 4, 5, and 7*
3. Initialize Positions matrix and search space
4. Calculate fitness of each agent *//Eq.1*
5. X_α = best search agent
6. X_β = second-best search agent
7. X_δ = third best search agent
8. **While** (t < Max iterations)
9. **For** each search agent
10. Update position of current search agent *//Alg. 1*
11. **End For**
12. Update a, A and C
13. Calculate fitness of all search agents
14. Update X_α, X_β and X_δ
15. Insert X_α to best Positions matrix
16. Update search space matrix
17. = t +1
18. **End While**
19. Return X_α

4 Simulation Results

This section presents the results of the proposed methods and evaluates their performance. The proposed methods are compared with GWO [26], mGWO [28], EGWO [29], and RW-GWO [30] in the same environmental conditions. The simulation and analysis presentation has been performed using MATLAB. The proposed methods and other used methods to compare are simulated on a Core i7-5500 U 2.4 processor with 8GB of RAM. In simulations, two maps (medium and large), presented in Table 1, with different starting and ending boundaries have been used. Furthermore, the initial and final stations of three UAVs were presented in Table 2, whereas the obstacles coordinates were explained in Table 3. All the coordinates are presented in three-dimensional space. The performance analysis parameters are cost analysis, execution time analysis, and convergence curve analysis. The population sizes and iterations numbers as given in couple-tuples form: (25, 40), (50, 100), (100, 100).

4.1 Analysis and Evaluation (Costs of Distance Traveled)

In this section, both of the proposed path planning methods are analyzed based on the cost function (Eq.1). All of the cost values obtained are in centimeters. As metaheuristic algorithms may obtain different as well as close to best solutions, we run each algorithm 10 times. The best, worst, and average cost values (distance traveled in cm) are presented with different population sizes and iteration numbers (see Table 6 and Table 7). Each proposed method has three UAVs with different starting and final positions. Briefly, each path planning method runs in the 3 different populations and iteration sizes on two maps.

Table 6. Simulation results of each path planning methods for medium map

Methods	pop	Iter.	UAV$_1$ COST (cm)			UAV$_2$ COST (cm)			UAV$_3$ COST (cm)			Overall time (sec)
			Best	Ave	Worst	Best	Ave	Worst	Best	Ave	Worst	
PP$_{GWO}$	25	40	261	324	377	204	217	233	241	277	307	**6.241**
PP$_{I\text{-}GWO}$	25	40	**255**	295	323	202	215	223	**224**	279	324	7.914
PP$_{EX\text{-}GWO}$	25	40	284	314	348	199	213	234	225	277	335	10.605
PP$_{mGWO}$	25	40	289	325	380	207	218	239	239	280	324	10.078
PP$_{EGWO}$	25	40	265	320	381	**198**	215	227	234	279	320	9.714
PP$_{RW\text{-}GWO}$	25	40	260	300	331	**198**	213	223	237	279	328	6.981
PP$_{GWO}$	50	100	239	297	341	**192**	213	225	236	241	246	49.601
PP$_{I\text{-}GWO}$	50	100	**236**	280	320	197	206	215	**222**	226	235	**43.491**
PP$_{EX\text{-}GWO}$	50	100	242	314	352	201	209	220	223	237	256	46.879
PP$_{mGWO}$	50	100	270	310	370	201	213	231	237	250	269	50.147
PP$_{EGWO}$	50	100	250	306	369	194	210	221	223	248	270	50.098
PP$_{RW\text{-}GWO}$	50	100	239	299	345	**192**	217	229	235	243	250	49.641
PP$_{GWO}$	100	100	**239**	293	346	191	208	220	**213**	224	234	128.26
PP$_{I\text{-}GWO}$	100	100	241	276	318	204	206	211	217	221	224	128.21
PP$_{EX\text{-}GWO}$	100	100	**239**	290	331	191	198	206	**213**	226	231	**118.05**
PP$_{mGWO}$	100	100	275	292	350	**190**	200	211	241	239	241	128.32
PP$_{EGWO}$	100	100	260	311	349	194	208	219	230	243	240	128.27
PP$_{RW\text{-}GWO}$	100	100	253	299	351	**190**	207	220	215	227	237	128.08

* The best values are bold.

Table 7. Simulation results of each path planning methods for large map

Algorithm	pop	Iter.	UAV$_1$ COST (cm)			UAV$_2$ COST (cm)			UAV$_3$ COST (cm)			Overall time (sec)
			Best	Ave	Worst	Best	Ave	Worst	Best	Ave	Worst	
PP$_{GWO}$	25	40	**363**	483	616	419	775	1204	397	432	455	7.933
PP$_{I\text{-}GWO}$	25	40	**363**	470	558	464	626	785	**388**	431	472	7.791
PP$_{EX\text{-}GWO}$	25	40	419	486	584	348	581	874	389	416	457	11.014
PP$_{mGWO}$	25	40	401	484	602	421	770	1219	400	435	458	9.147
PP$_{EGWO}$	25	40	420	489	581	360	619	903	397	427	472	8.172
PP$_{RW\text{-}GWO}$	25	40	370	488	601	410	690	1107	395	438	467	8.047
PP$_{GWO}$	50	100	374	444	528	**340**	489	674	362	381	399	84.569
PP$_{I\text{-}GWO}$	50	100	383	452	512	344	479	645	345	371	395	**72.626**
PP$_{EX\text{-}GWO}$	50	100	**358**	453	557	378	527	606	**343**	368	398	77.990
PP$_{mGWO}$	50	100	375	450	546	**340**	493	680	368	386	402	73.541
PP$_{EGWO}$	50	100	385	459	521	346	481	651	351	378	401	73.083
PP$_{RW\text{-}GWO}$	50	100	373	450	527	346	499	679	367	393	407	83.146
PP$_{GWO}$	100	100	365	433	468	328	385	429	348	356	360	153.763
PP$_{I\text{-}GWO}$	100	100	374	430	482	**322**	353	393	346	354	361	161.690
PP$_{EX\text{-}GWO}$	100	100	**352**	448	547	339	363	383	**338**	346	355	142.403
PP$_{mGWO}$	100	100	368	436	479	333	387	421	357	361	372	160.054
PP$_{EGWO}$	100	100	372	438	483	324	359	409	**338**	355	369	162.429
PP$_{RW\text{-}GWO}$	100	100	365	438	465	330	386	427	351	358	369	156.809

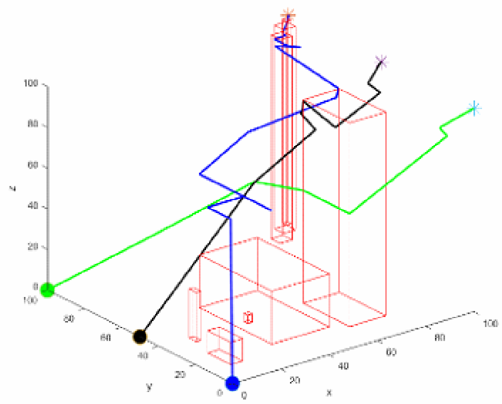
* The best values are bold.

Among the proposed methods in this study, PP$_{I\text{-}GWO}$ gives best results compared to the other methods. Table 8 presents the ranking summary of each method. This table shows the percentage of algorithms obtaining the minimum cost. According to the obtained simulation results, in general, PP$_{Ex\text{-}GWO}$ performed more successfully in larger and crowded environments, while PP$_{I\text{-}GWO}$ method gave good results in smaller and less populated environments. This is due to the fact that I-GWO only acts on the alpha wolf. However, Ex-GWO based methods involve all group members. Indeed, the PP$_{Ex\text{-}GWO}$ may be better in applications in an environment with a larger workspace and many obstacles. The main reason for this is that in the Ex-GWO method, almost all wolves in the pack have an important role in each other's position update. Therefore, the wolves in the pack minimize the escape paths of the hunt (prey), and hence, the hunts can be caught faster. The fact that this mechanism can be better than other methods can be seen more clearly in large and crowded environments. The I-GWO basic update process is very dependent on the alpha setup. Therefore, the speed of growth and the selection of the right places for the first wolf is of great importance. In this method, there is the possibility of finding problem solutions (preys) much faster in fewer iterations.

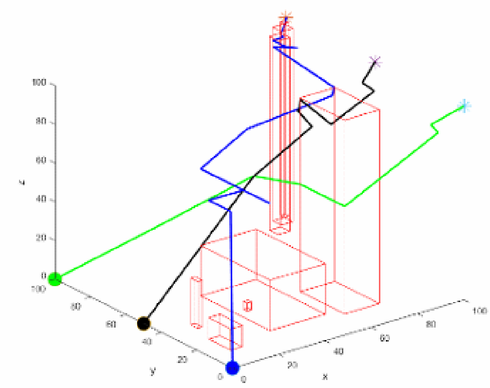
Table 8. Ranking summary of path planning methods in cost parameter

Algorithm	Success Rate (Percent)	Rank
PP_{GWO}	%13.89	3
$PP_{I\text{-}GWO}$	%36.11	1
$PP_{EX\text{-}GWO}$	%33.33	2
PP_{mGWO}	%5.55	5
PP_{EGWO}	%2.78	6
$PP_{RW\text{-}GWO}$	%8.34	4

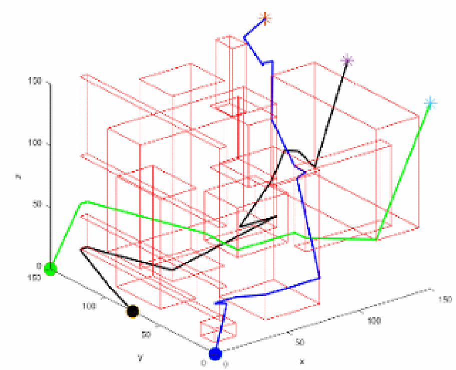
In Figure 5, all the paths generated for different maps have been shown in a perspective view. As mentioned before, there are three UAVs on each map. The balls show the initial state of each UAV and stars indicate the destination state of each UAV. The results show that both proposed methods generate optimum paths without any collision. This is a noteworthy issue to mention that is the experiments demonstrate that the metaheuristic algorithms need at least 40 iteration numbers and 25 population sizes for the optimal path. After that, the metaheuristic algorithms are less likely to decrease the cost of the path. Therefore, the movement model of the proposed methods on the maps is only be shown according to this number of populations and iterations.



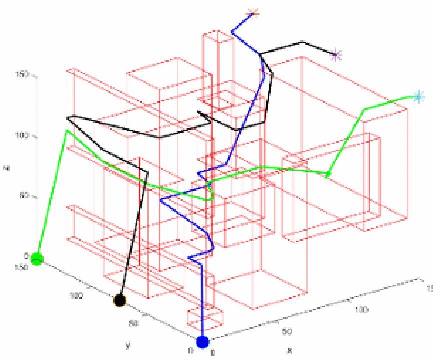
(a) 3D path planning by I-GWO in medium map



(b) 3D path planning by Ex-GWO in medium map



(a) 3D path planning by I-GWO in large map



(b) 3D path planning by Ev-GWO in large map

Figure 5. Generated optimal paths in $PP_{I\text{-}GWO}$ and $PP_{Ex\text{-}GWO}$ for medium and large map; population:25 and iteration: 40

4.2 Analysis and Evaluation (Execution Time and Complexity)

The second analysis parameter is execution times. The results of this parameter are presented in Figure 6 for the different populations and iterations of each of the five methods. The proposed algorithms should sort the entire possible set of stations for each element of the search space. The analysis of the time complexity for the proposed algorithms is $O(n^2)$. Also, the execution time of $PP_{I\text{-}GWO}$ is better than others. In the I-GWO, each wolf updates its own position based on all the wolves selected before. In the first step, there is one wolf. If there are n wolves in a pack, the nth wolf updates its own position based on n-1 wolves' position. Among the used algorithms, the I-GWO-based path planning algorithm, which employs three UAVs in parallel, takes the minimum time to reach the destination. In Ex-GWO algorithm, each pack member has more roles and contributions compared to other algorithms, which results in this algorithm consuming more execution time than $PP_{I\text{-}GWO}$. When all methods are evaluated in terms of execution time and complexity, it is shown that the I-GWO-based path planning method is a more suitable approach.

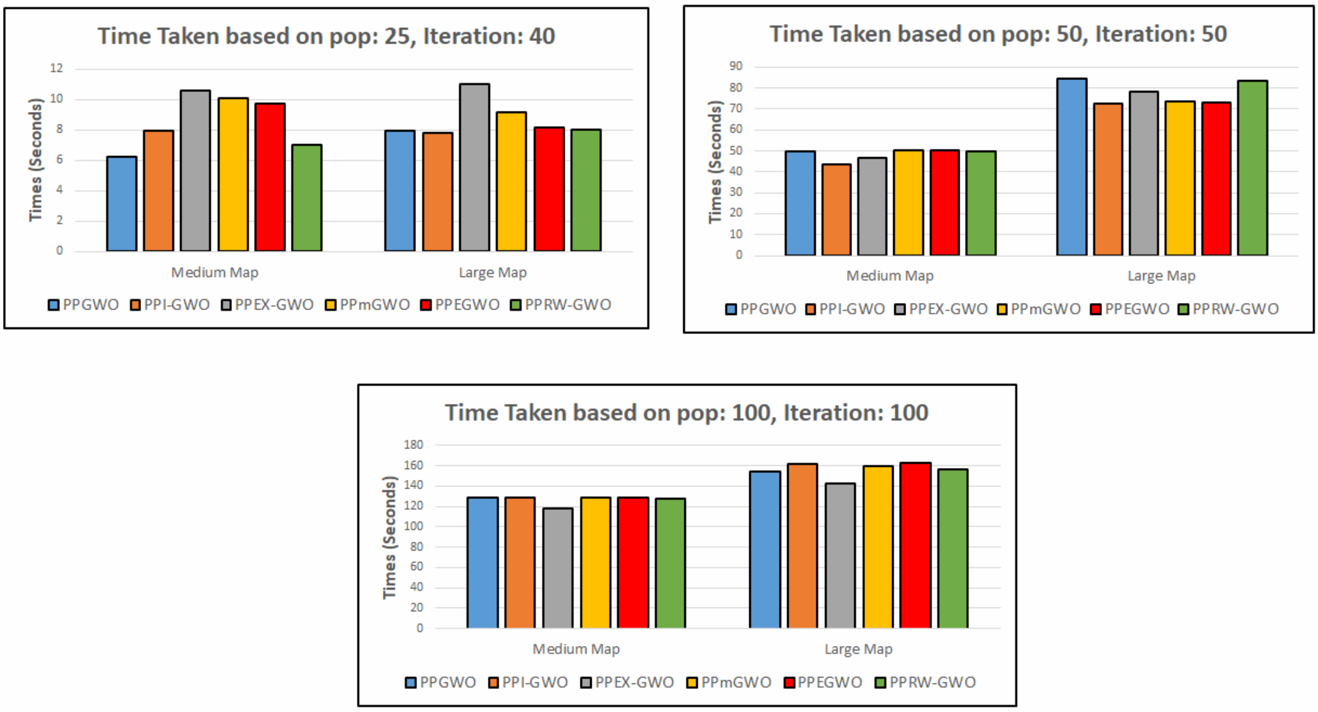


Figure 6. Execution time analysis for different maps in each method

4.3 Analysis and Evaluation (Convergence Curve)

Figure 7 present the convergence curve of each path planning method. As aforementioned, the obstacle numbers and the boundary sizes of the map are declared in the section 2. $PP_{I\text{-}GWO}$ and $PP_{Ex\text{-}GWO}$ methods have different structures with respect to exploration and exploitation. The presented figure illustrates the convergence curve of algorithms with 40 iterations and 25 population sizes. In $PP_{I\text{-}GWO}$, crossing from exploration to exploitation phase is faster compared to other methods. According to the results and the figure obtained, regarding $PP_{I\text{-}GWO}$ algorithm, UAVs reach near-optimal path earlier than when using other methods. In the analysis, we considered different iteration sizes to get the size of optimal iteration. As a result of observations, it was concluded that 40 iterations are enough to find the best path because achieved results have not a remarkable difference [26]. Continuing with further iterations, it was found that same results are being obtained. The acquired outcomes also indicate the execution time of algorithms is variable in the maps. Due to the number of obstacles and the map boundary, different results are collected. Also, while using three UAVs with different initial and final station behaviors in the convergence curve analysis, it is shown that the obstacle number has an effect on the path cost.

5 Conclusions and Future works

The novelty of this paper is to apply two new variants of the GWO algorithm to solve the 3D path planning problem for autonomous UAVs. They find collision-free paths with optimum cost. In this study, there are two different maps with various obstacles, furthermore, three UAVs with different start and final stations have been used. The proposed methods ($PP_{I\text{-}GWO}$ and $PP_{Ex\text{-}GWO}$) have been analyzed in terms of optimal path costs, time complexity, and convergence curve by varying population sizes as well as iteration numbers. The simulation results demonstrate that the proposed 3D path planning methods choose the optimal cost path across from the initial to final stations without collision. According to path cost results of the defined case studies in this study, $PP_{I\text{-}GWO}$ outperformed the best with %36.11. In the analysis of other parameters, this method also achieved the highest success compared to other methods. However, in general, $PP_{Ex\text{-}GWO}$ performs more successfully in larger and crowded environments, and $PP_{I\text{-}GWO}$ method outperforms good results in medium and smaller-sized environments. This is due to the nature of the working mechanisms of the proposed methods. These methods are convenient for the environment with distributed obstacles. In future work, the proposed 3D path planning methods can be employed for IoT, and connected vehicles with VANET and FANET structures.

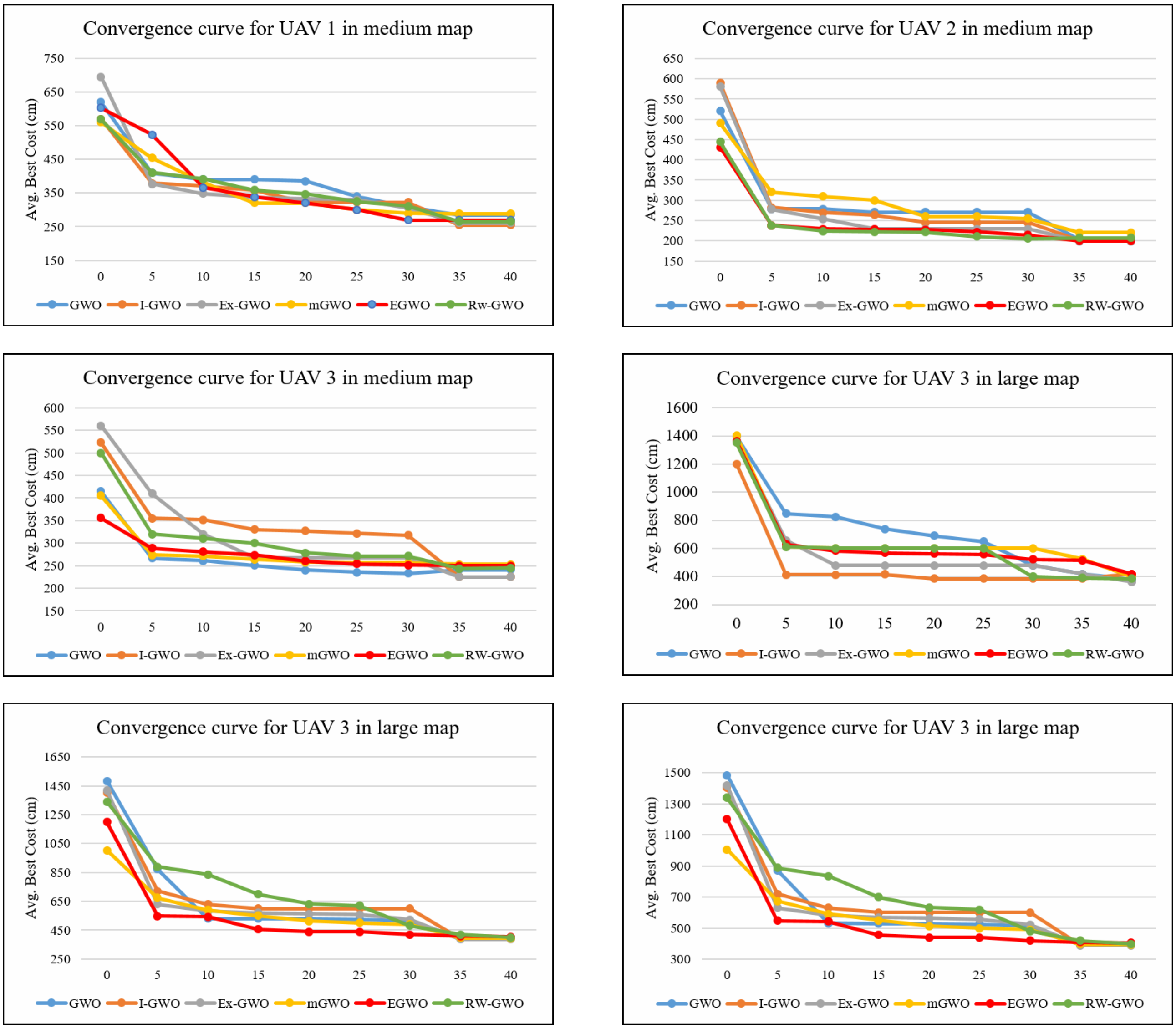


Figure 7. Convergence analysis for each UAV on the two different maps in 25 population and 40 iterations

References

[1] F. Kiani, S. Nematzadehmiandoab and A. Seyyedabbasi, Designing a dynamic protocol for real-time Industrial Internet of Things-based applications by efficient management of system resources, *Advances in Mechanical Engineering*, Vol. 11, No. 10, pp. 1-20, October, 2019.

[2] H. Cho, H. Wu, C. Lai, T. K. Shih and F. Tseng, Intelligent Charging Path Planning for IoT Network over Blockchain-based Edge Architecture, *IEEE Internet of Things Journal*, Vol. 8, No. 4, pp. 2379-2394, February, 2021.

[3] Q. P. Ha, L. Yen and C. Balaguer, Robotic autonomous systems for earthmoving in military applications, *Automation in Construction*, Vol. 107, Article No. 102934, November, 2019.

[4] F. Kiani and A. Seyyedabbasi, Wireless Sensor Network and Internet of Things in Precision Agriculture, *International Journal of Advanced Computer Science and Applications (IJACSA)*, Vol. 9, No. 6, pp. 99-103, June, 2018.

[5] F. Kiani, Reinforcement learning based routing protocol for wireless body sensor networks, *IEEE 7th International Symposium on Cloud and Service Computing (SC2)*, Kanazawa, Japan, 2017, pp. 71-78.

[6] L. Sumi and V. Ranga, An IoT-VANET-Based Traffic Management System for Emergency Vehicles in a Smart City, in: P. K. Sa, S. Bakshi, I. Hatzilygeroudis, M. N. Sahoo (Eds.), *Recent Findings in Intelligent Computing Techniques*, Vol. 708, Springer, Singapore, 2018, pp. 23-31.

[7] V. Sharma and K. Srinivasan, QoS-Aware Routing in Wireless Networks Using Aerial Vehicles, *Journal of Internet Technology*, Vol. 19, No. 1, pp. 73-89, January, 2018.

[8] M. Bacco, M. Colucci, A. Gotta, C. Kourogiorgas and A. D. Panagopoulos, Reliable M2M/IoT data delivery from FANETs via satellite, *International Journal of Satellite Communications and Networking*, Vol. 37, No. 4, pp. 331-342, July-August, 2019.

[9] B. Subbiah, M. S. Obaidat, S. Sriram, R. Manoharn and S. K. Chandrasekaran, Selection of intermediate routes for secure data communication systems using graph theory application and grey wolf optimization algorithm in MANETs, *IET Networks*, April, 2020. Doi:10.1049/iet-net.2020.0051.

[10] F. Kiani, A. Aghaeirad, M. K. Sis, A. Kut and A. Alpkocak,

EEAR: an energy effective-accuracy routing algorithm for wireless sensor networks, *Life Science Journal*, Vol. 10, No. 2, pp. 39-45, June, 2013.

[11] A. Nayyar, B. L. Nguyen and N. G. Nguyen, The Internet of Drone Things (IoDT): Future Envision of Smart Drones, in: A. Luhach, J. Kosa, R. Poonia, X. Z. Gao, D. Singh (Eds.), *First International Conference on Sustainable Technologies for Computational Intelligence*, *Advances in Intelligent Systems and Computing*, Vol. 1045, Springer, Singapore, 2020, pp. 563-580.

[12] Y. Chen, C. Lu and W. Chu, A Cooperative Driving Strategy Based on Velocity Prediction for Connected Vehicles with Robust Path-Following Control, *IEEE Internet of Things Journal*, Vol. 7, No. 5, pp. 3822-3832, May, 2020.

[13] A. Seyyedabbasi and F. Kiani, I-GWO and Ex-GWO: improved algorithms of the Grey Wolf Optimizer to solve global optimization problems, *Engineering with Computers*, Vol. 37, No. 1, pp. 509-532, January, 2021.

[14] B. K. Patle, A. Pandey, A. Jagadeesh and D. R. Parhi, Path planning in uncertain environment by using firefly algorithm, *Defence Technology*, Vol. 14, No. 6, pp. 691-701, December, 2018.

[15] B. K. Patle, L. G. Babu, A. Pandey, D. Parhi, and A. Jagadeesh, A review: On path planning strategies for navigation of mobile robot, *Defence Technology*, Vol. 15, No. 4, pp. 582-606, August, 2019.

[16] A. A. Maw, M. Tyan and J. W. Lee, iADA*: Improved Anytime Path Planning and Replanning Algorithm for Autonomous Vehicle, *Journal of Intelligent & Robotic Systems*, Vol. 100, No. 3-4, pp. 1005-1013, December, 2020.

[17] 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, February, 2017.

[18] A. V. Savkin and H. Huang, Bioinspired Bearing Only Motion Camouflage UAV Guidance for Covert Video Surveillance of a Moving Target, *IEEE Systems Journal*, pp. 1-4, October, 2020. DOI: 10.1109/JSYST.2020.3028577

[19] J. Cheng, B. Liu, K. Cai, X. Tang and B. Zhang, ETC Intelligent Navigation Path Planning Method, *Journal of Internet Technology*, Vol. 19, No. 2, pp. 619-631, March, 2018.

[20] J. Han, An efficient approach to 3D path planning, *Information Sciences*, Vol. 478, pp. 318-330, April, 2019.

[21] J. L. Sanchez-Lopez, M. Wang, M. A. Olivares-Mendez, M. Molina and H. Voos, A Real-Time 3D Path Planning Solution for Collision-Free Navigation of Multirotor Aerial Robots in Dynamic Environments, *Journal of Intelligent & Robotic Systems*, Vol. 93, No. 1-2, pp. 33-53, February, 2019.

[22] L. Wang, J. Kan, J. Guo and C. Wang, 3D Path Planning for the Ground Robot with Improved Ant Colony Optimization, *Sensors*, Vol. 19, No. 4, Article No. 815, February, 2019.

[23] L. Yang, J. Qi, J. Xiao and X. Yong, A literature review of UAV 3D path planning, *Proceeding of the 11th World Congress on Intelligent Control and Automation*, Shenyang, China, 2014, pp. 2376-2381.

[24] L. Yang, J. Qi, D. Song, J. Xiao, J. Han and Y. Xia, Survey of Robot 3D Path Planning Algorithms, *Journal of Control Science and Engineering*, Vol. 2016, pp. 1-22, July, 2016.

[25] A. Seyyedabbasi and F. Kiani, MAP-ACO: An efficient protocol for multi-agent pathfinding in real-time WSN and decentralized IoT systems, *Microprocessors and Microsystems*, Vol. 79, pp. 1-9, November, 2020.

[26] R. K. Dewangan, A. Shukla and W. W. Godfrey, Three dimensional path planning using Grey wolf optimizer for UAVs, *Applied Intelligence*, Vol. 49, No. 6, pp. 2201-2217, June, 2019.

[27] V. Sharma and R. Kumar, UAVs assisted queue scheduling in ground ad hoc networks, *International Journal of Ad Hoc and Ubiquitous Computing*, Vol. 30, No. 1, pp. 1-10, 2019.

[28] N. Mittal, U. Singh and B. S. Sohi, Modified grey wolf optimizer for global engineering optimization, *Applied Computational Intelligence and Soft Computing*, Vol. 2016, pp. 1-16, May, 2016.

[29] H. Joshi and S. Arora, Enhanced grey wolf optimization algorithm for global optimization, *Fundamenta Informaticae*, Vol. 153, No. 3, pp. 235-264, June, 2017.

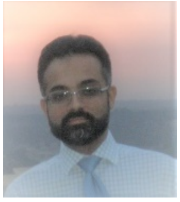
[30] S. Gupta and K. Deep, A novel random walk grey wolf optimizer, *Swarm and Evolutionary Computation*, Vol. 44, pp. 101-112, February, 2019.

Biographies

Farzad Kiani received bachelor, master, and Ph.D. degrees in computer engineering in 2006, 2009 and 2014 years, respectively. He is associate professor in computer engineering department. Dr. Farzad, who has many publications works on wireless communication, machine learning and trend technologies.

Amir Seyyedabbasi currently works as an assistant professor at Beykent University in computer engineering department. He received Ph.D. in Feb 2020 from IZU. He also received his B.Sc. and M.Sc. degrees in software engineering in 2012 and 2015. His research interests include wireless communications, optimization algorithms and IoT.

Royal Aliyev received bachelor degree at Azerbaijan State Oil and Industry University, and received master degree in robotics and automation engineering at University of Siena, Italy. He currently studies second master program at Istanbul Aydin University. His current research and master thesis includes the mobile robots and hybrid algorithms to control them.



Mohammed Ahmed Shah is an Assistant Professor in the department of computer engineering at Istanbul Ayvansaray University. He completed his undergraduate, graduate and doctorate studies at Eastern Mediterranean University. Dr. Shah’s Current research interests lie in the area of IoT, optimization algorithms, and reinforcement learning.

Murat Ugur Gulle has econometrics degree from Gazi University, Turkey. Currently, he is studying computer engineering master degree at Istanbul Aydin University. He has been conducting his research on reinforcement learning, optimization algorithms and game theory.