Chapter 3
An Enhanced White Shark Optimization Algorithm for Unmanned Aerial Vehicles Placement

Amylia Ait Saadi, Assia Soukane, Yassine Meraihi,
Asma Benmessaoud Gabis, Amar Ramdane-Cherif, and Selma Yahia

3.1 Introduction

Over the past decade, the area of Unmanned Aerial Vehicles (UAVs) has experienced significant growth in the commercial, civilian, and military markets [1, 2]. This is primarily due to the tremendous mobility, autonomy, communication, and relatively low cost of UAVs. Therefore, manufacturers today work on fitting and embedding technologies to make UAVs more valuable and suitable for various missions. One of the most promising applications is the application of UAVs to offer various services to connected users in wireless networks. The main purpose of this application's challenging issue is to find the optimal position of UAVs that cover the maximum number of users while ensuring access to the network by connecting the maximum number of drones [3]. The problem of UAVs placement belongs to the group of NP-hard problems successfully solved and optimized by meta-heuristic

---

A. A. Saadi (✉)
LIST Laboratory, University of M'Hamed Bougara Boumerdes, Boumerdes, Algeria

LISV Laboratory, University of Paris-Saclay, Velizy, France
e-mail: a.aitsaadi@univ-boumerdes.dz

A. Soukane
ECE Paris School of Engineering, Paris, France

Y. Meraihi · S. Yahia
LIST Laboratory, University of M'Hamed Bougara Boumerdes, Boumerdes, Algeria

A. B. Gabis
Ecole nationale Supérieure d'Informatique, Laboratoire des Méthodes de Conception des Systèmes, Alger, Algeria

A. Ramdane-Cherif
LISV Laboratory, University of Paris-Saclay, Velizy, France

M. D. Hina et al. (eds.), *Future Research Directions in Computational Intelligence*,
EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-031-34459-6_3

algorithms. Therefore, in this context, various research based on meta-heuristics were conducted. Authors in [4] applied Elephant Herding Optimization (EHO) algorithm for maximum coverage under different numbers of drones. According to simulation results, EHO performs well in offering maximum coverage for users using a less number of drones. In [5], Ozdag et al. proposed four approaches (OFSAC-PSO, OFSAC-EML, OFSAD-PSO, and OFSAD-EML) based on Particle Swarm Optimization (PSO) and Electromagnetism-Like (EML) algorithms for improving the UAV placement. The performance of the proposed approaches was assessed under different distributions of users and evaluated based on different metrics such as fitness function values, coverage rates, drones altitudes, and 3D drones' locations. Results showed that the OFSAC-PSO algorithm outperforms other optimization methods. In the work of Chaalal et al. [6], Social Spider Optimization (SSO) Algorithm was applied to solve the UAV deployment problem. The effectiveness of SSO algorithm was assessed in three different areas serving different numbers of users and compared to random search (RS) method and uniform distribution application. Simulation results proved that the SSO algorithm outperforms other meta-heuristics in terms of fitness value, execution time, and covered users. Reina et al. [7] proposed a multi-layout multi-sub-population genetic algorithm (MLMPGA) to enhance the UAVs placement for maximum coverage and connectivity. The MLMPGA algorithm was evaluated in various scenarios with different numbers of drones and users and compared to Genetic Algorithm (GA), PSO, and Hill Climbing algorithm (HCA). Test results showed that the MLMPGA algorithm gives competitive results compared to state-of-the-art meta-heuristics regarding the fitness value, coverage, connectivity, and redundancy.

This chapter proposes an ameliorated version of WSO algorithm, called EWSO, based on the incorporation of Elite opposition-based scheme for solving the UAVs deployment problem. The proposed EWSO is tested using 23 cases with various numbers of UAVs and users in comparison with the classical WSO, GWO, and BA algorithms.

The remaining of this chapter is organized as follows. Section 3.2 gives the formulation of the UAVs deployment problem. Section 3.3 gives the description of the WSO algorithm and EOBL strategy. Section 3.4 explains the structure of the proposed EWSO algorithm for solving the UAVs placement issue. Section 3.5 discusses the simulation findings. Finally, Sect. 3.5 shows the concluding remarks and future works.

3.2 UAV Placement Network Model and Problem Formulation

Consider a network system consisting of G users, $G = \{g_1, g_2, \ldots, g_m\}$. These users is served by a set of UAVs $U = \{u_1, u_2, \ldots, u_n\}$ for various applications. To ensure communication and reliability, each UAV is equipped with a radio interface with a maximum transmission range of R_{max} to communicate with ground users and other UAVs. UAVs can take any position defined as (x_j, y_j, h_j), $j \in \{1, 2, \ldots, n\}$ in $3D$ area of dimension $W \times L \times H$. The UAV height is limited by lower and upper bounds h_{min} and h_{max}, respectively. h_{min} is fixed by the user according to the application to protect UAVs from ground threats. h_{max} is related to coverage radius and visibility angle. Let us assume that the users are randomly located at a fixed position (x_i, y_i), $i \in \{1, 2, \ldots, m\}$, where $(x_i, y_i) \in W \times L$. Each user is equipped with a radio interface with a maximum transmission range of R_{max} to communicate with UAV.

The main objective is to find the best UAVs location for maximum user coverage and connectivity, which can be formulated mathematically by the following equation:

$$f(p_i) = \omega_1 . \frac{Cv}{m} + \omega_2 . \frac{Cn}{n}, \tag{3.1}$$

where Cv denotes the user coverage cost. Cn expresses the UAV connectivity cost. ω is the linear weight coefficient in the range [0, 1], so that $\sum_{i=1}^{2} w_i = 1$.

User Coverage Cost (*Cv*)

In this chapter, we consider that the users are static in the study area. Each UAV covers a number of users with cover radius defined in Eq. (3.2). We say that the user g_i is covered by the UAV u_j only if the distance between them $d(g_i, u_j)$, expressed in Eq. (3.4), is less than the coverage radius r_j. It is mathematically formulated in Eq. (3.3).

$$r_j = h_j . \tan\left(\frac{\theta}{2}\right), \tag{3.2}$$

where h_j is the UAV height. θ represents the visibility angle.

$$d(g_i, u_j) \leq r_j, \tag{3.3}$$

where $d(g_i, u_j)$ stands for the Euclidean distance between the UAV u_j and the user g_i. It is expressed in the equation below:

$$d(g_i, u_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \tag{3.4}$$

The total coverage cost by all UAV Cv is represented in the following equation:

$$Cv = \sum_{j=1}^{n} C_{g,u}, \tag{3.5}$$

where $c_{g,u}$ refers to the coverage cost of users by the UAV u_j. To make sure that one user is covered by exactly one UAV, the coverage cost is formulated as follows:

$$C_{g,u} = \begin{cases} 1, & if \min\{d(g,u)\}, \forall u \in U \\ 0 & \text{otherwise.} \end{cases} \tag{3.6}$$

UAV Connectivity Cost (*Cn*)

The main goal of UAVs wireless network is to provide access to different available services. The network is formed by connecting the maximum number of UAVs in a mesh topology to ensure redundancy and availability. We say that the UAV u_j is connected to the UAV u_j only if the distance between them $d(u_j, u_k)$ does not exceed twice the maximum transmission range R_{max}. The connectivity is mathematically represented in Eq. (3.7).

$$Cn = \sum_{j=1}^{n} N_{u_j}, \tag{3.7}$$

where N_{u_j} represents the number of connected UAVs with the UAV u_j in a single hop that is calculated as follows:

$$N_{u_j} = |u_k| d(u_j, u_k) < 2.R_{max}|, \tag{3.8}$$

where $d(u_j, u_k)$ represents the distance between UAVs u_j and u_k, which is expressed in Eq. (3.9).

$$d(u_j, u_k) = \sqrt{(x_j - x_k)^2 + (y_j - y_k)^2 + (h_j - h_k)^2}. \tag{3.9}$$

3.3 Preliminaries

This section describes the definition and concept of WSO algorithm and EOBL strategy.

3.3.1 White Shark Optimizer Algorithm

White Shark Optimizer (WSO) Algorithm is a newly meta-heuristic proposed by Braik et al. [8] in May 2022 for solving optimization problems. The WSO algorithm is a swarm intelligence meta-heuristic that mimics the behavior of the White Shark in hunting preys [8, 9] that can be summarized in three different actions as follows: (I) Movement toward the prey; (II) Random search for the prey; (III) Nearby location of the prey:

(I) Movement toward the prey:

In this behavior, the white shark tracks and locates the prey based on their senses. As a prey moves, a white shark hears wave hesitations that pinpoint the location of its prey and moves directly toward it. This behavior is represented mathematically in Eq. (3.10).

$$v_{t+1}^{i} = \mu \left\{ v_t^i + w_1 \times c_1 \times \left(p_{gbest_t} - p_t^i \right) + w_2 \times c_2 \times \left(p_{best}^{v_t^i} - p_t^i \right) \right\}, \tag{3.10}$$

where v_{t+1}^i represents the i-th shark's velocity at $(t+1)$ iteration $t+1$. v_t^i is the current velocity of the i-th shark. p_{gbest_t} and p_t^i are the shark's best position and current i-th shark position at iteration t, respectively. $p_{best}^{v_t^i}$ stands for the best known position so far. v_t^i is the current i-th index vector of the white sharks reaching the best position, which is defined in Eq. (3.11). w_1 and w_1 are control parameters represented in Eqs. (3.12) and (3.13). c_1 and c_1 are two random variables. μ represents the constriction factor that is formulated in (3.14).

$$v = [n \times rand(1, n)] + 1, \tag{3.11}$$

where n donates the population size.

$$w_1 = w_{max} + (w_{max} - w_{min}) \times e^{-\left(\frac{4t}{T}\right)^2} \tag{3.12}$$

$$w_2 = w_{min} + (w_{max} - w_{min}) \times e^{-\left(\frac{4t}{T}\right)^2}, \tag{3.13}$$

where t and T are the current and the maximum number of iterations, respectively. w_{min} and w_{max} stand for the initial and subordinate velocities, respectively.

$$\mu = \frac{2}{|2 - \alpha - \sqrt{\alpha^2 - 4\alpha}|}, \tag{3.14}$$

where α is a fixed value at 4.125 that represents the acceleration coefficient.

(II) Random search for the prey:

In this case, the white shark follows the prey tracks based on the smelling and hearing in random positions. This movement is mathematically formulated by the following equation:

$$p_{t+1}^{i} = \begin{cases} p_t^i . \neg \oplus p_0 + ub.a + lb.b, & rand < ws \\ p_t^i + \frac{v_t^i}{f}, & rand \geq ws \end{cases}, \tag{3.15}$$

where p_{t+1}^{i} denotes the new position of the i-th white shark. $\neg$ is a negation operation. p_0 refers to a logical position vector defined in Eq. (3.18). ub and lb represent the upper and lower search space boundaries, respectively. C is a control parameter that balances the exploration and exploitation, which is expressed in Eq. (3.20). f stands for the frequency of the white shark's wavy motion that can be calculated as shown in Eq. (3.19).

$$a = sgn\left(p_t^i - ub\right) > 0 \tag{3.16}$$

$$b = sgn\left(p_t^i - lb\right) < 0 \tag{3.17}$$

$$p_0 = \oplus(a, b), \tag{3.18}$$

where $\oplus$ represents the bit-wise exclusive-or (XOR) operator.

$$f = f_{min} + \frac{f_{max} - f_{min}}{f_{max} + f_{min}}, \tag{3.19}$$

where f_{min} and f_{max} refer to the minimum and maximum frequencies of the white shark's wavy motion, respectively.

$$C = \frac{1}{\alpha_0 + e^{\left(\frac{T}{2} - t\right)\alpha_1}}, \tag{3.20}$$

where α_0 and α_1 are represented as two positive values to manage both exploration and exploitation behavior.

(III) Nearby location of the prey: In this method, the white shark uses a fish school technique and moves toward the shark that is closer to the prey. This movement is formulated by the following equation:

$$p_{t+1}^{\prime i} = p_{gbest_t} + r_1.\overrightarrow{D_p}.sgn(r_2 - 0.5), \quad r_3 < s, \tag{3.21}$$

where $p_{t+1}^{\prime i}$ is the updated i-th white shark's position. r_1, r_2, and r_3 are random variables in the range of [0, 1]. $\overrightarrow{D_p}$ represents the distance between the prey and the white shark. $sgn(r_2 - 0.5)$ is a parameter used to change the search direction. s expresses white shark's senses that are presented in Eq. (3.23).

$$\overrightarrow{D_p} = |rand.\left(p_{gbest_t} - p_t^i\right)|, \tag{3.22}$$

where $rand$ is a random variable within the range of [0, 1]. p_t^i stands for the current i-th white shark's position.

$$s = |1 - e^{\left(\frac{-\alpha_2 . t}{T}\right)}|, \tag{3.23}$$

where α_2 represents the control behavior parameter.
White shark's position update according to fish school behavior is given in Eq. (3.24).

$$p_{t+1}^i = \frac{p_t^i - p_{t+1}^{'i}}{2.rand}. \tag{3.24}$$

3.3.2 *Elite Opposition-Based Learning*

Opposition-based learning strategy (OBL) proposed by Tizhoosh [10] is a well-regarded intelligent strategy that aims to enhance the chance of finding more effective solution by checking simultaneously the initial solution and its corresponding opposite solution. Let us consider a given candidate solution p in one dimension search space delimited by $[Lb, Ub]$. Then, the opposite solution $\overline{p}$ is defined as follows:

$$\overline{p} = Ub + Lb - p. \tag{3.25}$$

Elite opposition-based learning (EOBL) is an improved version of OBL widely combined with several meta-heuristics. The basic concept of EOBL is to employ first an elite solution that has expectantly more information than other individuals and then generate the opposite of the current solution in the search area. The elite individual leads the population toward the promising area where the global solution can be found. The elite opposite solution can be formulated by the following expression:

$$\overline{p_i} = r.(Du_j + Dl_j) - p_j, \quad j = 1, \ldots, Dim, \tag{3.26}$$

where r is a random number in the range $[0 - 1]$. Du_j and Dl_j are dynamic boundaries that can be presented as follows:

$$Du_j = max(p_j), \quad Dl_j = min(p_j). \tag{3.27}$$

However, the elite solution may jump out of the search space boundaries $[Lb, Ub]$. Consequently, EOBL will fail to consider a valid solution. To overcome this issue, we address a random value for this kind of solution as follows:

$$\overline{p_i} = rand(Lb_j, Ub_j) \quad if \quad \overline{p_j} < Lb_j \parallel \overline{p_j} > Ub_j. \tag{3.28}$$

3.4 Elite Opposition-Based White Shark Optimization Algorithm for UAVs Placement

This section describes the implementation steps of our proposed EWSO algorithm for solving the UAVs placement problem. In this sense, the EOBL strategy was incorporated into WSO to enhance its optimization performance. The proposed EWSO involves mainly four steps including initialization, evaluation, update, and finally termination that displays the best UAVs positions found.

3.4.1 Initialization

The first step of implementing the proposed EWSO algorithm consists of initializing white sharks positions randomly in the search area that is bounded by the size of the deployment area for UAVs. The initial position of White sharks is represented in a $N \times D$ matrix as shown in Eq. (3.29).

$$\text{Positions} = \begin{bmatrix} Pos_{1,1} & Pos_{1,2} & \dots & Pos_{1,D} \\ Pos_{2,1} & Pos_{2,2} & \dots & Pos_{2,D} \\ \dots & \dots & \dots & \dots \\ Pos_{N,1} & Pos_{N,2} & \dots & Pos_{N,D} \end{bmatrix}, \tag{3.29}$$

where N and D stand for the population size and the problem dimension, respectively. Pos represents the i-th shark position that expresses the UAVs positions. It is formulated in the following equation:

$$Pos_i = (\{x_{i,1}, x_{i,2}, \dots, x_{i,n}\}; \{y_{i,1}, y_{i,2}, \dots, y_{i,n}\}; \{z_{i,1}, z_{i,2}, \dots, z_{i,n}\}). \tag{3.30}$$

3.4.2 Evaluation

According to this population, the EWSO algorithm applies an evaluation by calculating the fitness value of the current population. Based on this evaluation, EWSO finds the initial best position that corresponds to the maximum fitness value as formulated in Eq. (3.31), considering the UAVs placement problem is a maximization problem.

$$Pos_{best} = \arg\max f(Pos). \tag{3.31}$$

After evaluation, the EWSO initializes the dynamic bounds $[Dl, Du]$ using Eq. (3.27) to process the next step.

Algorithm 1 Elite opposition-based white shark optimization algorithm for UAVs placement

```
Initialize EWSO parameters: Maximum number of iterations $T$, Population' size $N$, Dimension $Dim$, $\mu$, $v$, $f$, $etc.$
Initialize the population of EWSO: $Pos_i (i = 1, 2, ..., N)$
Calculate the fitness value $f(Pos_i)$
Determine the best position $Pos_{best}$
while ($t < T$) do
    for $i = 1, 2, \ldots, N$ do
        Find the $N$ opposite positions based on EOBL using Eqs. (3.27) and (3.26), and select the $N$ fittest positions using Eq. (3.32)
        $v_{t+1}^i = \mu\{v_t^i + p_1 \times c_1 \times (p_{gbest_t} - p_t^i) + p_2 \times c_2 \times (p_{best}^{v_t^i} - p_t^i)\}$
        Update the position using Eq. (3.15)
        if $rand < s$ then
            Update the position using Eq. (3.21)
            Update the final position using Eq. (3.24)
        end if
        Update the best position $Pos_{best}$
    end for
    $t = t + 1$
end while
return The best position $Pos_{best}$
```

3.4.3 *Update*

In this step, the EWSO algorithm searches for the elite opposite of the current population $\overline{Pos_i}$ using Eq. (3.26). Then, EWSO evaluates both Pos_i and $\overline{Pos_i}$ and selects N best candidates for updating according to the best fitness value as represented in Eq. (3.32).

$$Pos_i = \arg\max f(Pos_i \cup \overline{Pos_i}). \tag{3.32}$$

After positions' selection, the EWSO processes as original WSO for updating position by using Eqs. (3.10), (3.15), (3.21), and (3.24). The new positions are evaluated, and the dynamic bounds are updated according to the new positions. The EWSO algorithm repeats this step until the maximum number of iterations is reached.

3.4.4 *Termination*

This step represents the end of the process. The EWSO displays the best UAVs positions found represented in an array as follows:

$$Pos_{best} = (\{x_1, x_2, \ldots, x_n\}; \{y_1, y_2, \ldots, y_n\}; \{z_1, z_2, \ldots, z_n\}). \tag{3.33}$$

The pseudocode of our proposed EWSO algorithm for UAVs placement can be summarized in Algorithm 1.

3.5 Numerical Results

In this section, the evaluation of the proposed EWSO algorithm for solving the UAV placement is presented. The evaluation is done on several experiments and configurations as shown in Table 3.1 and compared to BA [11], GWO [12], and the original WSO algorithm. All simulations are running using MATLAB 2021B Software installed on Core i7 2.90 GHz, RAM 32 GB machine.

The effectiveness of the EWSO algorithm was evaluated by considering the fitness value, user coverage, and UAVs connectivity. The reported results represent the average of 50 runs for each metric found by each algorithm.

3.5.1 Impact of Varying the Number of UAVs

In the first scenario, the number of UAVs is varied from 4 to 24 with a step of 2, and the number of users is fixed at 200. The obtained results, in this case, are reported in Figs. 3.1, 3.2, and Table 3.2. We can clearly notice that the number of UAVs is proportional to the quality provided. Increasing the number of drones increases the fitness value. The EWSO algorithm reaches the highest fitness value of 0.9715 when the number of UAVs is fixed at $U = 20$. With 20 UAVs, EWSO covers more than 90% of users with 99.9% of connectivity, while BA, GWO, and WSO obtained the maximum fitness value at $U = 22$ and $U = 24$, respectively. Significantly, EWSO requires fewer UAVs than BA, GWO, and WSO to achieve the highest quality, positively impacting cost and energy.

Table 3.1 Description of experiences

Scenario parameters	Value
Population size	50
Maximum number of iterations	200
Weight coefficient w	[{0.5, 0.5}]
Number UAVs	[4−24]
Number of users	[50−300]
Maximum transmission range	100 m
Visibility angle	120
Area dimension	1000 m × 1000 m

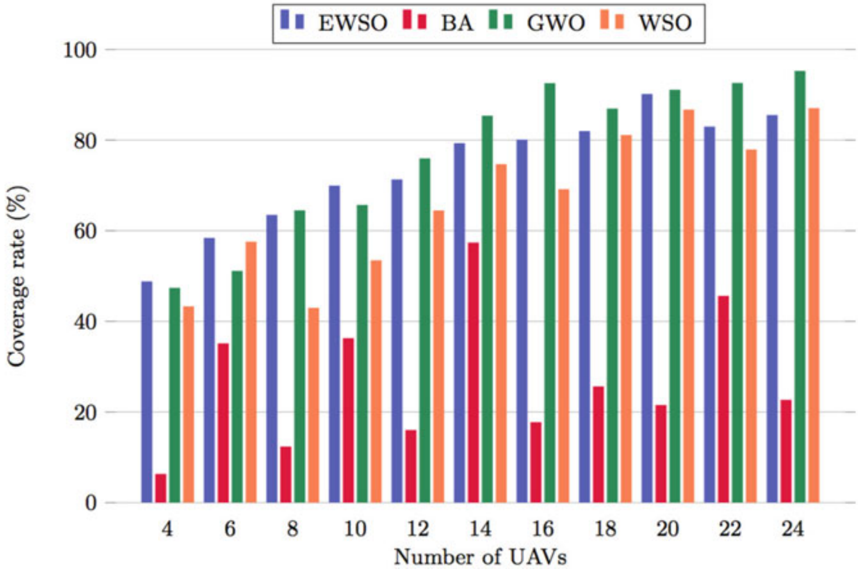

Fig. 3.1 The coverage rate using a different number of drones

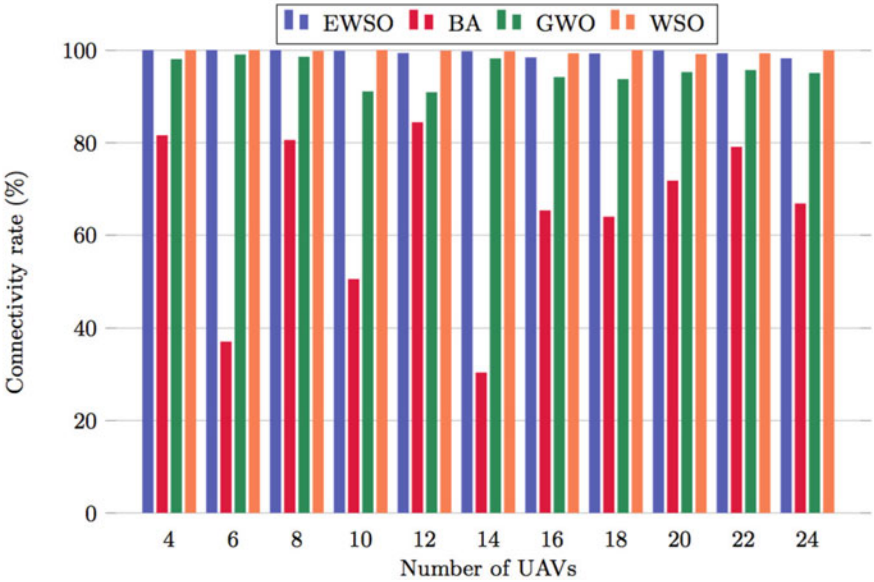

Fig. 3.2 The connectivity rate using a different number of drones

Table 3.2 Results obtained from different algorithms in the first case

Results	EWSO	BA	GWO	WSO
	$U = 4$			
Fitness	0.7617	0.4386	0.7264	0.7223
Coverage (%)	48.71	6.23	47.28	43.19
Connectivity (%)	100	81.5	98	100
	$U = 6$			
Fitness	0.8107	0.3602	0.7502	0.8024
Coverage (%)	58.31	35.04	51.03	57.47
Connectivity (%)	100	37	99	100
	$U = 8$			
Fitness	0.8336	0.4639	0.8172	0.7294
Coverage (%)	63.38	12.27	64.93	42.87
Connectivity (%)	100	80.5	98.5	99.75
	$U = 10$			
Fitness	0.8651	0.4327	0.7828	0.786
Coverage (%)	69.84	36.18	65.57	53.36
Connectivity (%)	99.8	50.4	91	100
	$U = 12$			
Fitness	0.8723	0.5013	0.8335	0.8347
Coverage	71.22	15.92	75.86	64.36
Connectivity	99.33	84.33	90.83	99.83
	$U = 14$			
Fitness	0.9173	0.4373	0.917	0.8904
Coverage (%)	79.21	57.2857	85.26	74.58
Connectivity (%)	99.71	30.29	98.14	99.71
	$U = 16$			
Fitness	0.9224	0.4145	0.9328	0.8643
Coverage (%)	79.98	17.65	92.44	69.06
Connectivity (%)	98.37	65.25	94.12	99.25
	$U = 18$			
Fitness	0.9316	0.4422	0.9025	0.922
Coverage (%)	81.87	24.55	86.84	81.01
Connectivity (%)	99.22	63.89	93.67	99.98
	$U = 20$			
Fitness	0.9715	0.4657	0.9306	0.9459
Coverage (%)	90.06	21.44	90.99	86.62
Connectivity (%)	99.9	71.7	95.2	99.1
	$U = 22$			
Fitness	0.9371	0.6225	0.9407	0.9106
Coverage (%)	82.36	45.5	92.5	77.8
Connectivity (%)	99.27	79	95.63	99.27
	$U = 24$			
Fitness	0.9512	0.4716	0.9508	0.9491
Coverage (%)	85.41	22.57	95.15	86.95
Connectivity (%)	98.17	66.75	95	99.91

3.5.2 Impact of Varying the Number of Users

In this scenario, the number of UAVs is fixed at 20, and the number of users varies from 50 to 300 with a step of 25. Figures 3.3, 3.4, and Table 3.3 describe the results

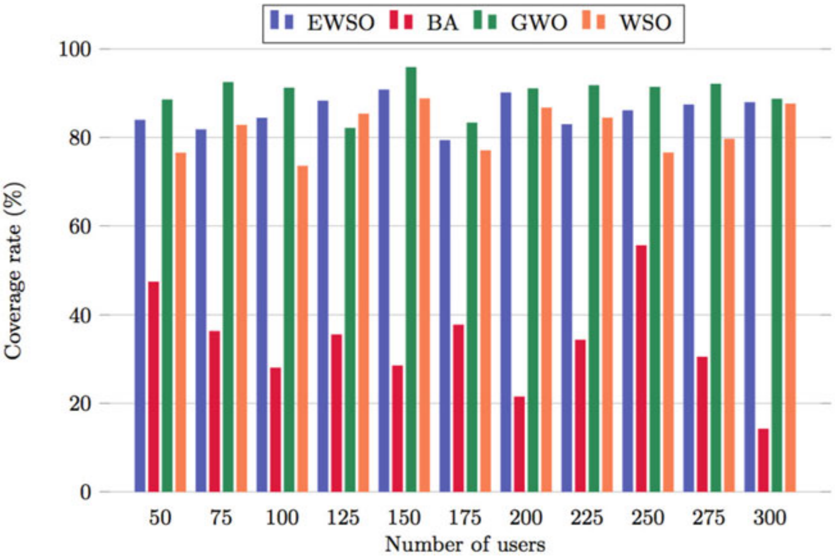

Fig. 3.3 The coverage rate using a different number of users

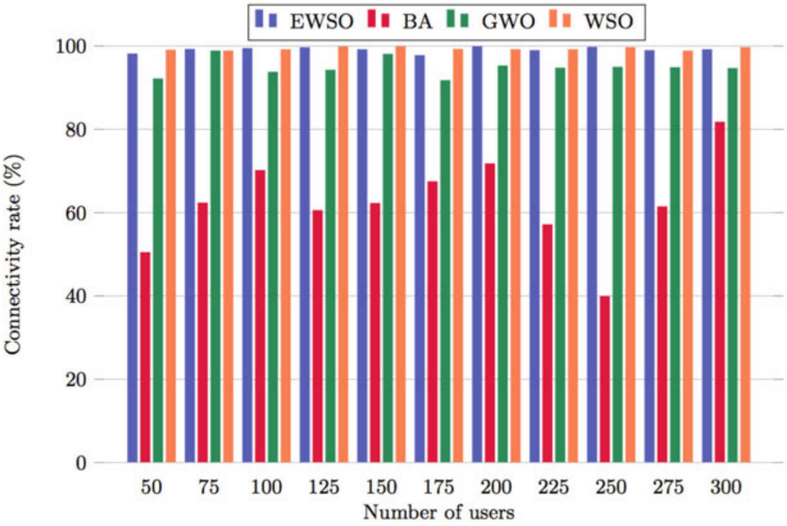

Fig. 3.4 The connectivity rate using a different number of drones

Table 3.3 Results obtained from different algorithms in the second test

Results	EWSO	BA	GWO	WSO
	$G = 50$			
Fitness	0.9476	0.4884	0.9029	0.9196
Coverage	83.88	47.28	88.48	76.44
Connectivity	98.1	50.4	92.1	99
	$G = 75$			
Fitness	0.9345	0.4928	0.9261	0.9423
Coverage	81.73	36.26	92.42	82.72
Connectivity	99.2	62.3	98.8	98.8
	$G = 100$			
Fitness	0.9435	0.4909	0.9242	0.8912
Coverage	84.34	27.98	91.14	73.48
Connectivity	99.4	70.1	93.7	99.1
	$G = 125$			
Fitness	0.9667	0.4804	0.8812	0.9476
Coverage	88.21	35.48	82.05	85.28
Connectivity	99.6	60.5	94.2	99.9
	$G = 150$			
Fitness	0.9767	0.4533	0.969	0.9602
Coverage	90.72	28.45	95.8	88.72
Connectivity	99.1	62.2	98	99.9
	$G = 175$			
Fitness	0.923	0.5255	0.8784	0.8983
Coverage	79.29	37.6914	83.24	76.97
Connectivity	97.7	67.4	91.7	99.2
	$G = 200$			
Fitness	0.9715	0.4657	0.9306	0.9459
Coverage (%)	90.06	21.44	90.99	86.62
Connectivity (%)	99.9	71.7	95.2	99.1
	$G = 225$			
Fitness	0.9343	0.4571	0.9321	0.9366
Coverage	82.88	34.3228	91.72	84.37
Connectivity	98.9	57.1	94.7	99.1
	$G = 250$			
Fitness	0.9549	0.4767	0.9312	0.896
Coverage	86.04	55.5	91.33	76.48
Connectivity	99.7	39.85	94.9	99.6
	$G = 275$			
Fitness	0.9571	0.4591	0.9341	0.9145
Coverage	87.33	30.4291	92.02	79.59
Connectivity	98.9	61.4	94.8	98.8
	$G = 300$			
Fitness	0.9605	0.4794	0.9162	0.9498
Coverage	87.86	14.1733	88.64	87.55
Connectivity	99.1	81.7	94.6	99.6

obtained for different numbers of users covered by 20 UAVs. As the number of users increased, fitness value, coverage, and connectivity metrics increased until $G = 150$, where all algorithms except BA reached maximum performance. Therefore, the optimal number of UAVs to cover 150 users using this configuration is 20. For other user cases, solutions provided by EWSO, GWO, and WSO algorithms are good and acceptable.

In most cases, the proposed EWSO algorithm outperforms the others by giving the highest fitness results that reflect the best balance between the coverage and the connectivity objectives.

3.6 Conclusion

In this chapter, we have proposed an ameliorated version of WSO, named EWSO, for tackling the problem of UAVs placement in 5G networks. The Elite opposition-based learning strategy is incorporated into the original WSO to enhance its efficiency. The proposed EWSO was tested using 23 scenarios with 24 UAVs and 300 users compared to WSO, GWO, and BA algorithms. The simulation results proved the superiority and efficiency of EWSO considering fitness values, coverage, and connectivity parameters. For future work, several directions in which EWSO can be extended. Initially, the energy consumption problem for UAVs can be addressed in this context. Furthermore, the EWSO algorithm can be enhanced by combining other meta-heuristics for better quality.

References

1. M. Radmanesh, M. Kumar, P.H. Guentert, M. Sarim, Overview of path-planning and obstacle avoidance algorithms for UAVs: a comparative study. Unmanned Syst. **6**(02), 95–118 (2018)
2. S. Aggarwal, N. Kumar, Path planning techniques for unmanned aerial vehicles: a review, solutions, and challenges. Comput. Commun. **149**, 270–299 (2020)
3. I.A. Elnabty, Y. Fahmy, M. Kafafy, A survey on UAV placement optimization for UAV-assisted communication in 5G and beyond networks. Phys. Commun. **51**, 101564 (2022)
4. I. Strumberger, N. Bacanin, S. Tomic, M. Beko, M. Tuba, Static drone placement by elephant herding optimization algorithm, in *2017 25th Telecommunication Forum (TELFOR)* (IEEE, Piscataway, 2017), pp. 1–4
5. R. Ozdag, Multi-metric optimization with a new metaheuristic approach developed for 3D deployment of multiple drone-BSs. Peer-to-Peer Networking Appl. **15**(3), 1535–1561 (2022)
6. E. Chaalal, L. Reynaud, S.M. Senouci, A social spider optimisation algorithm for 3D unmanned aerial base stations placement, in *2020 IFIP Networking Conference (Networking)* (IEEE, Piscataway, 2020), pp. 544–548
7. D.G. Reina, H. Tawfik, S.L. Toral, Multi-subpopulation evolutionary algorithms for coverage deployment of UAV-networks. Ad Hoc Networks **68**, 16–32 (2018)
8. M. Braik, A. Hammouri, J. Atwan, M.A. Al-Betar, M.A. Awadallah, White shark optimizer: a novel bio-inspired meta-heuristic algorithm for global optimization problems. Knowledge-Based Syst. **243**, 108457 (2022)

9. M.A. Ali, S. Kamel, M.H. Hassan, E.M. Ahmed, M. Alanazi, Optimal power flow solution of power systems with renewable energy sources using white sharks algorithm. Sustainability **14**(10), 6049 (2022)
10. H.R. Tizhoosh, Opposition-based learning: a new scheme for machine intelligence, in *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, vol. 1 (IEEE, Piscataway, 2005), pp. 695–701
11. X.-S. Yang, A. Hossein Gandomi, Bat algorithm: a novel approach for global engineering optimization. Eng. Comput. **29**(5), 464–483 (2012)
12. S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer. Adv. Eng. Software **69**, 46–61 (2014)