Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.
Optimizing multiple ONUs placement in Fiber-Wireless (FiWi) access network using Grasshopper and Harris Hawks Optimization Algorithms

Puja Singh\(^1\), Shashi Prakash*\(^*\)

Photonics Laboratory, Department of Electronics & Instrumentation, Institute of Engineering & Technology, Devi Ahilya University, Khandwa Road, Indore 452001, India

**ABSTRACT**

FiWi network is a multi-domain network that integrates optical and wireless networks. Hybrid fiber-wireless (FiWi) access network endeavours at consolidating the huge amount of available bandwidth of optical networks and the ubiquity, mobility of wireless access networks with the motive of reducing cost and complexity. This manuscript entails investigations undertaken towards optimal placement of multiple Optical Network Units (ONUs) in FiWi network using two recent optimization algorithms named as Harris Hawks Optimization (HHO) and Grasshopper Optimization algorithms (GOA). The results of the investigations are then benchmarked with respect to the Whale Optimization algorithm (WOA). The outcomes demonstrate the superiority of the proposed HHO algorithm over GOA and WOA algorithms, and return the lowest value of cost function; WOA outperforms GOA in terms of improved convergence rate and time complexity. Additionally, the diversification and intensification features of HHO, GOA and WOA have been compared. In order to benchmark the performance of the HHO and GOA optimizers, a series of convergence curves corresponding to different values of controlling parameters are plotted, and their optimal values determined. The dependence of objective function value upon distribution of users and initial placement of ONUs is also studied; the random placement of users and deterministic placement of ONUs return the lowest value of objective function.

1. Introduction

In the current scenario, there is increasing demand for the latest technologies which can fulfil the need of huge bandwidth and the fast data rate for quad play applications. These include voice, video, Internet, wireless and premium rich-media applications (e.g., multimedia, interactive gaming, and metaverse) [1-4]. To serve the growing bandwidth demands, network technologies with higher capacity are embarked both in the wired and wireless network domains [5]. The extensive research and development activities have given birth to the two dominant broadband-access technologies [6]. These are optical and wireless access networks.

Wireless access network is one of the leading access technologies. It provides cost efficiency, robustness, flexibility, easy deployment and wide coverage [7,8]. However, this fails to maintain high quality of service while transmitting large amount of data. Wireless technology cannot support long haul communication as its data rate is bounded due to spectrum limitation. Wireless access network also suffers from impairments. On the other hand, optical fiber technology provides huge amount of bandwidth. It shows its superiority over its two precursors - Digital Subscriber Line (DSL) and Cable Modem (CM). The key features of DSL and CM are huge bandwidth capacity, reduced transmission loss and lesser susceptibility to interference. But the distance of users from the Central Office (CO) and service access time are the barriers in deciding their performance. The optical network improves the performance of DSL and CM by providing higher bandwidth and giving access to the subscribers situated more than 20 km from CO [9]. The data rates of optical network can be improved further by the latest technologies.

But optical networks suffer from cost inefficiency and do not provide mobility and flexibility.

The aforementioned two access technologies suffer from their pros and cons. To take the benefit of huge bandwidth capacity of optical access networks and ubiquity, mobility of wireless access network, hybrid FiWi (Fiber Wireless) access networks seem to be more attractive than relying on either standalone access solution. This hybrid network optimally serves end subscribers in a cost efficient way integrating optical backhaul and wireless fronthaul. Optical and Wireless Access Network (OWAN) [10] or Wireless and Optical Broadband Access Network (WOBAN) [6] are alternate terms for FiWi access network.

To date extensive research activities have been performed to...
integrate optical and wireless technologies in order to promote this combination ahead. There are various issues which are being researched. These are cloud-based radio over optical fiber networks (CRON) [11-14], gateways/ONU placement, survivability [15,16], flow and congestion control, routing, energy consumption [17], scheduling and allocation of bandwidth, etc. Optimal placement of ONUs is an important issue as it enhances the system throughput, makes system economic and allows optimal utilization of resources. To solve this optimization problem various strategies (random and deterministic) have been implemented. For optimizing the position of multiple ONUs, many deterministic (Greedy algorithm) and metaheuristic algorithms have been used. Analysis of these algorithms based on their outcomes was done in [7,18,19]. Simulated Annealing (SA), Moth-Flame Optimization (MFO) [20] and Whale Optimization algorithm (WOA) [21] are metaheuristic optimization algorithms. MFO and WOA are population based optimization algorithms, whereas SA algorithm is an individual based algorithm. These algorithms suffer from local optima stagnation. In SA algorithm, setting of parameters and handling of constraints is always a challenging task. There is trade off between computation time and quality of outcomes. To decide whether the obtained outcome is a global fitness value or local, is a difficult task. MFO and WOA are optimizers which may get stuck in local optima. The rate of convergence is also slow. Recently, an efficient metaheuristic optimizer, Harris Hawks Optimization algorithm (HHO) was proposed in [22]. Another algorithm named as Grasshopper Optimization algorithm (GOA) was proposed in [23]. In the present research, we used these two optimizers to optimally place multiple ONUs in FiWi access network. The results of HHO and GOA are compared with WOA. HHO algorithm shows excellent performance as far as exploitation, exploration, local optima avoidance and computational complexity are concerned.

The proposed work adds the following essential contributions:

1. The proposed research provides in depth performance analysis of different metaheuristic algorithms (WOA, GOA and HHO) for solving ONU placement problem in FiWi network. The comprehensive research has been performed to elaborate the characteristics of WOA, GOA and HHO optimizers based on exploration, exploitation, rate of convergence and computational complexity metrics. To the best of our knowledge, these algorithms (HHO and GOA) have not been reported for ONU placement in FiWi network, so far.

2. The outcomes of the algorithms are crosschecked by altering their controlling parameters. Different curves have been obtained showing the variation in fitness function with controlling parameters. For the present research, the optimum value of controlling parameters has been determined.

3. The computational complexity of all three algorithms has been calculated and on the basis of it, the algorithms have been compared.

4. On the basis of the nature of convergence curves, the performance of the optimizers has been analyzed.

The present research work has following outlines: Section 2 reviews the literature of ONU placement in FiWi network, WOA, GOA and HHO optimization algorithms. Section 3 covers system model, constraints, notations and objectives. Section 4 explains the fundamentals, and the implementation of GOA and HHO algorithms for the deployment of multiple ONUs in FiWi access network. Section 5 describes the computational complexity of WOA, GOA and HHO algorithms. Computational complexity of HHO is lesser as compared to GOA and WOA algorithms. GOA has higher value of computational complexity. Section 6 analyzes and evaluates the results of HHO, GOA and WOA algorithms. The results show the superiority of HHO algorithm over GOA and WOA algorithms. WOA performs better than GOA. The objective value obtained through HHO algorithm is minimum. It has been proved that HHO is potentially able to guide the initial random solutions, and let them converge to an optimal position in the search area. The performance evaluation of the HHO algorithm considering nature of the convergence curves and fitness value has been done by varying the random jump length in LF. For \( J = 2.5(1 - \epsilon) \), we get minimum value of the objective function value. In the present manuscript, the simulation results have been obtained by setting the aforementioned value of \( J \). In GOA, there are two parameters \( I \) and \( F \), which controls exploration and exploitation of GOA. These parameters are varied and their effect on the nature of the quality of the convergence curve has been checked. The results of the investigation justifies that we get minimum value of the fitness function in accordance with the predefined values set by the author in [23]. This section also discusses the weaknesses of GOA. Section 7 concludes the work and gives insight towards the future work.

2. Related literature

With the increasing demands for the huge bandwidth applications, new technologies are required to make the existing network flexible, cost efficient, fast and be able to utilize resources efficiently. To achieve the aforementioned objectives, cloud computing, Internet of things (IoT), fifth generation of mobile networks (5G) are appearing as prominent technologies. The benefits of FiWi network are combined with cloud networking and this has given origin to recent cloud radio access network (C-RAN) [11-14]. To enrich C-RAN, various research activities are in progress to enhance resource utilization and quality of services. The promising research attributes of C-RAN seem to be very effective to fulfill the future needs and have great influence in the areas of communication and networking.

There are various key issues of FiWi access network. These riddles are optimal ONU placement, routing, survivability, throughput, delay and QoS etc. These core elements of research in FiWi access network have been explored and analysed using wide range of algorithms. Optimal placement of multiple ONUs/gateways is an important research topic as this is an interface between optical and wireless domains. ONUs are not only the interfacing devices between optical and wireless domains, but optimal positioning of ONUs enhances the performance of the network also. The solution of this optimization problem leads to cost and resource optimization [7]. Many research outcomes have been obtained considering this problem. In [24], the authors explored two global optimization algorithms – SA and Hill Climbing (HC) for placement of multiple ONUs in FiWi access network. The results of SA and HC are compared with Greedy algorithm. The authors concluded that the results of Greedy algorithms are close to SA and HC optimizers, at the expance of much lower processing requirements. In [25], a mixed integer programming (MIP) based model for placing the Base Stations (BS) and ONUs, optimally, in FiWi network (primal problem) was proposed. The authors defined various constraints which were needed to be satisfied. For getting suitable solution of MIP, Lagrangean Relaxation method was used. The authors in [10], suggested the insights of FiWi network architecture. Performance evaluation of FiWi network incorporating design, connectivity and fault-tolerant behavior was accomplished. The authors solved the placement problem of multiple ONUs using Greedy and SA algorithms in [1]. The authors reported that FiWi network is a more cost efficient solution than Passive Optical Networks (PON) alone. In [26], a novel Load Balanced ONU Placement (LBOP) algorithm was proposed. LBOP algorithm minimized number of ONUs and satisfied the constraints of hop number and load balancing. In [19], the authors proposed the use of MFO algorithm, a population based paradigm, to optimally place multiple ONUs in FiWi access network and compared its results with Greedy and SA optimizers. The results elucidated that MFO algorithm outperforms Greedy and SA algorithms in terms of minimum fitness value (overall average distance). In [27,28], the authors explored MFO algorithm by incorporating variants of spiral paths for estimating the optimal positions of ONUs. In [7], a novel swarm based WOA was implemented for placing multiple ONUs optimally in FiWi access network. The performance
analysis of WOA was done by comparing its results with Greedy and MFO algorithms. The authors reported improved performance of WOA over Greedy and MFO algorithms in terms of improved cost function and faster convergence rate.

Optimization is on demand always. It is a mathematical technique for finding a maximum or minimum value of a variable or variables belonging to a given problem and subject to a set of constraints. There are two classes of algorithms: Single solution based (SA) and population based algorithms (MFO, WOA etc). In the former case, with the help of one processed solution, the optimization is performed. In the later case, a group of solutions are generated in each iteration and updated with some stochastic phenomenon. Population based algorithms are being used mostly because of their superiority in finding more optimal value. In this context over the last few decades, there is arrival of new optimizers. For example - Marine Predators algorithm (MPA) [29], Coronavirus optimization algorithm (CVOA) [30], Red Deer algorithm (RDA) [31], Slime Mould algorithm (SMA) [32], Group Teaching optimization algorithm (GTOA) [33], Equilibrium Optimizer (EO) [34], Black Widow Optimization Algorithm (BWO) [35], Harris Hawks optimization (HHO) [22], Salp Swarm Algorithm (SSA) [36], Grasshopper Optimization algorithm (GGA) [23], Sine Cosine algorithm (SCA) [37] Moth Flame Optimization (MFO) [20], Whale Optimization algorithm (WOA) [21], etc. The performance of these optimizers has been tested in different cases considering various metrics and their results demonstrated the outstanding performance on different search landscapes.

Among above metaheuristic algorithms, HHO is one of the efficient population based algorithms. The results of HHO algorithm report that it is very effective with meta-heuristic optimizers and show superiority over conventional techniques. Several authors used HHO algorithm for various applications in engineering [38–46]. The performance of HHO algorithm has been tested with other optimizers in [22]. It has been successfully used for the design of microchannel heat sinks [38], for solving the coordination problem in the directional over current relays (DOCRs) [39], for extracting the unknown parameters of the three-diode photovoltaic (TDPV) model [40], for manufacturing optimization problems [41], for accurate and reliable air pollutant forecasting model [42], for solving reference point multi-objective problems [43], for image processing [44], for satellite image segmentation [45] etc. In [46], the authors proposed hybrid models by combining artificial neural network (ANN) with GOA and HHO for predicting soil compression coefficient (SCC). These models are GOA-ANN and HHO-ANN. It has been reported that in some cases both GOA and HHO showed the best performance. On the other side the results obtained through HHO and GOA are dependent on some specified conditions.

In the present research, GOA has been applied to optimally place multiple ONUs in the FiWi access network. In [23,47–50], GOA has been used for solving several problems in engineering and application. Further in [23], the authors proposed GOA and accomplished the performance analysis of GOA. GOA has provided satisfactory solutions and has the potential to significantly outperform several reported algorithms. The authors implemented GOA in multi-objective problems [47], in unconstrained and constrained test functions [48], in Economic Load Dispatch (ELD) [49], in satellite image segmentation [50] etc. These works report the efficacy of GOA in comparison with the reported optimizers.

3. System model

This section deals with the system model and discusses the optimization problem of multiple ONUs positions causing minimal deployment cost for the FiWi network. To solve this problem we designed a system model which is based on the system model given by Suman Sarkar in reference [1]. Here the inputs are the area of the search space (area of FiWi network) in sq-meters (1000 × 1000), the number of ONUs − 3 and number of wireless users − 100. The optimal position of ONUs (X and Y coordinates) and value of the overall cost function (overall average distance) are the output. These users are considered as static (not moving). In this system, wireless routers and users mean the same entities, and are used interchangeably. They are defined by their x and y coordinates in FiWi access network. Further for estimating the cost function, first the distance from each of the ONUs to each of the users is calculated. The users which are closest to an ONU, are called its premium users and that ONU is called as the primary ONU for those users. We find the average of the x and y coordinates of the premium users, and the resultant corresponds to the optimum value of the X and Y locations of that primary ONU. We take the mean of the distance among primary ONUs and their premium users by diving it by total number of premium users. This is defined as individual cost or individual cost function value of an ONU. We follow this procedure for rest of the ONUs. We sum up the individual costs of all n ONUs and take the average of it (dividing it by total number of ONUs). The result is overall objective function value and is defined as the “distance” from ONUs to users.

The description of the constraints in the present research problem is in accordance with the following lines:

(i). The lower and upper bounds of the search area are the constraints of the problem and optimized locations of ONUs must not be beyond these limits.

(ii). The optimal positions of the ONUs should not be at the common points because these locations will be considered as invalid locations.

The structure of the system model is based on following considerations.

• Area of network is taken as a × a.
• Grid area is considered as gx' × gy'.
• The center position of ONUs in the grid defines deterministic placement.
• Any position other than centre region of the grid is random placement of the ONUs.

3.1. Notations

The model of the proposed FiWi network is defined as a directed graph G = (N, E). In this graph, N constitutes OLT, group of ONUs and wireless mesh routers. The abbreviation E is used for total number of links. Particularly, we denote

n : Sum of ONUs in the network.
K : Sum of users in the network.
NC : Number of centers in the network.
(Dx,yj) : User j’s locations along X/Y-axis,
(Dx,yi) : ONU i’s locations along X/Y-axis,
(Dx,yco) : ONU’s optimized X/Y-coordinates,
Dg : Distance between ONU i and user j,
SDj : Set of values of distances between user j and all initial ONU locations.

ONUy Primary : Primary ONU, (minimum-distance ONU) of user j,

kii : Premium users for ONUi (where\(\sum_{j} k_{ii} = K\))

Dyj ONU : Distance of ONU from the users.

DPrim ONUN : Distance between primary ONU and premium users.

\(\bar{D}_{ONU}\) : Average value of distance of ONUi, which is the distance of premium users from ONUs’ optimized locations.

\(\bar{D}_{ONU}\) : Overall average distance of ONUs.

3.2. Objectives

The present research work applies HHO and GOA algorithms to minimize the value of fitness function. Implementing these algorithms, the following objectives should be achieved using aforementioned
optimizers:

• Find the “Euclidean” distance (cost) to position the i-th ONU with respect to its users:

\[
D_{\text{ONU}_i}^{\text{init}} = \sum_{j=1}^{n_{\text{ONUs}}} \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}
\]

• Decide premium users and the mean of the X/Y-coordinates of these users. This defines the optimal positions of ONUs.

\[
\text{Optimize } (X_{\text{ini}}, Y_{\text{ini}}) = \sum_{j=1}^{n_{\text{ONUs}}} x_j / k_j \sum_{i=1}^{n_{\text{users}}} y_j / k_j
\]

• Estimate the overall cost of placing the ONUs by summing the individual costs of all n ONUs and taking the average of it (dividing it by total number of ONUs).

\[
D_{\text{Overall}}^{\text{init}} = \frac{1}{n} \sum_{i=1}^{n} D_{\text{Primary ONUs}_i}
\]

• Analyze the performance of these two algorithms considering basic methodology, characteristics, principles, and compare the outcomes of HHO, GOA and WOA algorithms based on some predefined metrics considering different initial distribution of users and ONUs.

4. Algorithms

This portion covers the details of the optimizers which are explored to optimally place multiple ONUs in FiWi access network.

4.1. WOA used for optimal placement of multiple ONUs in FiWi access network:

WOA has been used for the deployment of multiple ONUs in FiWi network. WOA is a combinatorial optimizer and simulates bubble-net attacking behavior of humpback whales when they are hunting their preys. The detailed algorithm of WOA has been discussed in [21].

In the process of optimizing multiple ONUs positions, the search area has been arranged in various non overlapping sections named as grids. The initial placement of users and ONUs based on the positions in grid, will define deterministic or random positions. The number of users, ONUs and their initial placements are the input to the system. The optimized locations of ONUs are the output of the system. The value of overall average distance of the ONUs concerning their users (premium users) defines the value of the cost function [7].

4.2. HHO algorithm

In the present research work Harris Hawks Optimization algorithm (HHO) [22] has been used to find the optimal positions of multiple ONUs. HHO is a population based gradient free optimization technique; hence it can be applied to any optimization problem subject to a proper formulation [22]. This optimization strategy is inspired by the cooperative behaviour and chasing style of Harris hawks in nature (desert site) called surprise pounce. Fig 1 shows the Harris’ hawk environment. The hawks follow a prudent technique to swoop a prey (rabbit) from different directions so that it may be astonished. These hawks can demonstrate different pursuing models in accordance with the dynamically changing scenarios and the ways the prey escape. The present optimization algorithm mathematically imitates these dynamic patterns and behaviour of Harris hawks to frame and develop an optimization algorithm. There are three main phases in HHO: exploring a prey, surprise pounce, and different kinds of attacking strategies of Harris hawks. These attacking techniques are defined for the exploitation process. The concept of the levy flight (LF) is utilized in exploitation phase. Further the struggle of the prey depends on its energy. In the exploration phase, the energy is sufficient to face the attacking situation but as time progresses the content of the energy decreases (in exploitation phase), and at last the prey will be caught and consumed. At that instant the energy of the prey is the lowest. The final position of the rabbit (prey) where its energy is the minimum signifies the optimum position.

The proposed work first applies HHO algorithm to find the potential sites of ONUs in the network. Here we start by initializing N, the population of hawks which represent initial locations of ONUs and are generated randomly. The desert site represents the search area in FiWi network. The upper and lower bounds of the problem are defined by the extreme points of the desert area. The fitness value or the objective function value of the problem is calculated by finding the energy of every hawk. Here hawk represents random solution and its fitness value represents the objective function value considering every location of ONUs. Among these values, we find the best location which returns minimum value of objective function (for each hawk). A few equations have been designed for exploration, exploitation and transition between these two phases. These modelling equations are based on the energy of the rabbits and chance of the rabbits to escape. The optimal value of fitness is achieved when the value of the objective function is the least. There are three main phases in HHO: exploring a prey, surprise pounce, and many kinds of attacking strategies of Harris hawks. The details of each phase are described as follows.

4.2.1. Exploration phase

This subpart mainly covers the mathematical modelling of the Harris hawks - how they wait, search and detect the prey. In this optimization process, Harris hawks are considered as the candidate solutions and the position of prey is defined as the optimal or near optimal solution. This phase is mathematically defines as follows:

\[
X(T + 1) = \begin{cases} 
X_{\text{rand}}(T) - r_2 |X_{\text{rand}}(T) - 2r_3 X(T)| & p \geq 0.5 \\
(X_{\text{rabbit}}(T) - X_{\text{ini}}(T)) - r_3 (UBO - LBO) & p < 0.5 
\end{cases}
\]

where X(T + 1), is the position vector of hawks in the T + 1 iteration, X_{\text{rabbit}}(T) is the position of rabbit, X(T) is the current position of hawks, r_2, r_3, r_4, r_5 and p are the random numbers vary from 0 to 1. The values of these random numbers are updated in each iteration. UBO and LBO are the upper and lower bounds of the variables. X_{\text{rand}}(T) is random selection of the hawk from the current population and X_{\text{ini}}(T) is the average position of the hawks and can be calculated using following expression:

\[
X_{\text{ini}}(T) = \frac{1}{n} \sum_{i=1}^{n} X(T)
\]

where the location of each hawk is represented by X_{\text{ini}}(T) and n denotes the count of hawks.

4.2.2. Transition from exploration to exploitation

The escaping energy of the rabbit (prey) defines the transition from exploration to exploitation and different patterns of exploitation. There is considerable decrement in the escaping energy of the rabbit. The following equation defines the energy:

\[
E_{\text{rabbit}} = 2E_{\text{rabbit}} \left( \frac{T}{T_{\text{max}}} \right)
\]

where E_{\text{rabbit}} denotes the escaping energy of the prey, T_{\text{max}} is the maximum size of the iterations and E_{\text{rabbit}} is the initial state of energy. E_{\text{rabbit}} ranges from −1 to 1. HHO performs exploration and exploitation when |E| ≥ 1 and |E| < 1 respectively.

4.2.3. Exploitation

The state of exploration, exploitation and different intermediate stages during exploitation, largely depends on the escaping energy of
the rabbit $E_{rabbit}$ and the chance of the target's escaping probability. Based on the escaping strategies, the preys and pursuing styles of hawks, four different strategies are proposed to frame the assauling behaviour in HHO algorithm.

- **Soft besiege:** When $r > 0.5$ and $|E_{rabbit}| \geq 0.5$, this technique can be framed by eq. (4) and eq. (5).

\[
X(T + 1) = \Delta X(T) - E_{rabbit}|\nabla X_{rabbit}(T) - X(T)| \tag{4}
\]

\[
\Delta X(T) = X_{rabbit}(T) - X(T) \tag{5}
\]

where $\Delta X(T)$ is the difference between the position vector of the rabbit and the present location in iteration $T$, $J = 2(1 - r)$ is the value of the random jump of the rabbit all across the escaping action, $r$ is a random number changes from 0 to 1. $J$ takes random value in every iteration in accordance with rabbit movements.

- **Hard besiege:** When $r \geq 0.5$ and $|E_{rabbit}| < 0.5$, the rabbit gets very tired during self protection and its escaping energy becomes low. The Harris hawks targeted the prey to implement surprise pounce. This phase can be modeled by eq. (6).

\[
X(T + 1) = X_{rabbit}(T) - E|\Delta X(T)| \tag{6}
\]

- **Soft besiege with progressive rapid dives:** When $r < 0.5$ and $|E_{rabbit}| \geq 0.5$, the target can protect itself as it has still enough energy to escape. As a result the hawk can perform the next move which is given by the following eq. (7).

\[
Y = X_{rabbit}(T) - E|\nabla X_{rabbit}(T) - X(T)| \tag{7}
\]

In HHO algorithm, the levy flight (LF) is utilized in exploitation phase. It has been proved that LF-based activities are the optimal searching strategies of scavengers for food in non-destructive conditions \cite{51,52}. The following equation will define the hawks’ dive based on LF:

\[
\dot{X}(t) \sim U(0, \sigma) \quad \text{where} \quad \sigma = \frac{C_1 \cdot T_0}{\sqrt{2 \pi}} \tag{8}
\]

where, $D$ represents the dimension of the problem and $V$ is a random vector of 1 by D size and LF is the levy flight function, which is calculated by following relation:

\[
LF(x') = 0.01 \times \frac{u \times \theta}{|v|^2}, \quad \theta = \left(\frac{(1 + 2 \beta) \times \sin \left(\frac{\pi}{2} \frac{|v|^2}{\sigma^2}\right)}{((1 + 2 \beta) \times \beta \times \sqrt{\pi} \sigma^2)}\right)^{\frac{1}{2}} \tag{9}
\]

where $u$, $v$ and $\theta$ are random values range from 0 to 1, $\beta$ is a default constant set to 1.5.

The last mechanism to change the position of hawks can be given by the following equation:

\[
X(T + 1) = \begin{cases} 
Y'fF(Y) < F(X(T)) \\
L'fF(L') < F(X(T)) 
\end{cases} \tag{10}
\]

In eq. (10), $Y$ and $L$ can be defined by eqs. (7) and (8).

- **Hard besiege with progressive rapid dives:** When $r < 0.5$ and $|E_{rabbit}| < 0.5$, in this stage there is considerable decrement in the energy of the rabbit and a hard besiege is conducted before surprise pounce. The modelling of this behaviour is as follows:

\[
X(T + 1) = \begin{cases} 
Y'fF(Y) < F(X(T)) \\
L'fF(L') < F(X(T)) 
\end{cases} \tag{11}
\]

where $Y'$ and $L'$ are defined by the following eqs. (12) and (13).

\[
Y' = X_{rabbit}(T) - E|\nabla X_{rabbit}(T) - X_{m}(T)| \tag{12}
\]

\[
L' = Y' + V \times LF(D) \tag{13}
\]

where $X_{m}(T)$ is obtained using eq. (2).

The pseudo-code of the HHO algorithm for solving the optimal placement problem of multiple ONUs is as follows:

*Start by initializing random population for searching: N number of hawks (initial random search locations of ONUs);

While $T < T_{max}$

Calculate the objective function values considering every location of ONUs (fitness of every hawk);

Find the best location which returns minimum value of objective function

for each location of ONU (for each hawk)

Update the initial energy $E_{rabbit}$ and $J$

Update $E_{rabbit}$ using eq. (3)

if $|E_{rabbit}| \geq 1$

Update the location of ONUs using eq. (1)

else if $r \geq 0.5$ and $|E_{rabbit}| \geq 0.5$

Update the location of ONUs using eq. (4)

else if $r < 0.5$ and $|E_{rabbit}| < 0.5$

Update the location of ONUs using eq. (6)

else if $r < 0.5$ and $|E_{rabbit}| < 0.5$

Update the location of ONUs using eq. (10)

else if $r < 0.5$ and $|E_{rabbit}| < 0.5$

Update the location of ONUs using eq. (11)

Display the most optimum location of ONUs obtained till yet.

4.3. GOA

GOA is a recent swarm intelligence algorithm, developed by Mirjalili at. el \cite{23}. It is a nature inspired paradigm, which has been utilized for optimal placement problem of ONUs in FiWi network. GOA considers the position of grasshoppers and implements the interaction of grasshoppers in nature \cite{23}. It is shown in Fig. 2. In the algorithm exploration occurs through random search in all possible directions with long step size. This is analogous to the long range, abrupt movement of grasshoppers in their adulthood phase. For exploitation, directional search is required with slow movement and small steps to perform local search. This phase of algorithm is accomplished by the slow motion and tiny steps of grasshoppers in larval phase. Food source seeking which is another important characteristic of the swarming of grasshoppers indicates the target achieving in the algorithm. In GOA, each grasshopper represents a solution in the population. The position of each grasshopper in the swarm is based on the three forces. The
social interaction between it and other grasshoppers, the gravity force on it, and the wind advection. Here only social interaction (attraction and repulsion) is taken into consideration in GOA. Initially there is repulsive force between the grasshoppers. The range of the distance where this force is active is from 0 to 2.079 units [23]. When a grasshopper is 2.079 units away from another grasshopper, there is neither attraction nor repulsion. This is called the comfort zone or comfortable distance. From 2.079 units to nearly 4 units of distance, attraction force increases. It is assumed that the fittest grasshopper (the one with the best objective value) during optimisation is the target. The next position of a grasshopper is determined based on its current position, the position of the target, and the position of all other grasshoppers. This algorithm uses an adaptive parameter. This parameter balances exploration and exploitation of the whole swarm around the target. It contributes to the reduction of repulsion/attraction forces between grasshoppers proportional to the number of iterations. It reduces the search coverage around the target as the iteration count increases. During the interaction, the location of the grasshopper which has minimum value of cost function is the optimal solution of the problem.

In GOA, we set the value of the parameters and maximum number of iterations as the stopping criteria. The initial population of swarms (initial random search locations of ONUs) is generated randomly and each solution in the population is evaluated by calculating its objective function. Initially there is repulsion force between the grasshoppers. This force enables exploration of the search space. The force of attraction increases for a particular unit of distance and then gradually decreases. This phase enables exploitation and promotes local search towards the target to get global optima. After evaluating all the solutions in the initial population or search agents (position of ONUs), the overall best solution is assigned according to its value. At each iteration the parameter value is updated. Each solution in the population is improved in accordance with a predefined coefficient and target value (global best solution). The above mentioned steps are repeated for all the solutions in the population. These are updated and evaluated and the best solution is assigned. The overall operations are repeated until the maximum number of iterations is reached, which is the termination criterion. The final position of the grasshopper which corresponds to the minimum value of the cost function denotes the optimum position. The details and relevant equations are described as follows.

The position of each grasshopper in the swarm is based on the three forces. The social interaction between it and other grasshoppers, the gravity force on it, and the wind advection. The final form of the three affected forces on each grasshopper can be defined as follows:

$$P_i = S_i + G_i + A_i$$  \hspace{1cm} (14)

where $S_i$, $G_i$ and $A_i$ are the social interaction, gravity force and air advection between two grasshoppers.

To build the randomness in the above equation, the eq. (14) can be rewritten as below:

$$P_i = r_S S_i + r_G G_i + r_A A_i$$  \hspace{1cm} (15)

Where, $r_S$, $r_G$ and $r_A$ are the random numbers in the range 0 and 1.

The social interaction force between grasshoppers can be defined as:

$$S_i = \sum_{j \neq i} S(d_{ij}) D_j$$  \hspace{1cm} (16)

where $d_{ij}$ is the distance between grasshopper i and grasshopper j and given as $|P_j - P_i|$. $S$ is a function defined as the strength of social forces and $D_j = \frac{P_j - P_i}{d_{ij}}$ is a unit vector from the $i^{th}$ grasshopper to the $j^{th}$ grasshopper.

$S$ is a function, and represents the strength of two social forces, attraction and repulsion between grasshoppers, it can be defined as follows:

$$S(r) = Fe^c - e^{\beta}$$  \hspace{1cm} (17)

where $F$ and $l$ are the intensity of the attraction and the attractive length scale. The social interaction changes with variation in $F$ and $l$ parameters.

The second factor of force that affects the position of the grasshopper is the gravity force which has been calculated as follows:

$$G_i = Ge_{G_i}$$  \hspace{1cm} (18)

Where G is gravitational constant and $e_{G_i}$ is a unit vector towards the centre of earth.

The $A_i$ component in eq. (14) is calculated as follows:

$$A_i = Ue_{W_i}$$  \hspace{1cm} (19)

Where $U$ is constant drift and $e_{W_i}$ is a unit vector towards wind. The position of grasshoppers can be calculated as follows:

$$P_i = \sum_{j=1, j \neq i}^{N} S(|P_j - P_i|) \frac{P_j - P_i}{d_{ij}} - Ge_{G_i} - Ue_{W_i}$$  \hspace{1cm} (20)

In order to solve optimization problem, and prevent the grasshoppers to reach the comfort zone quickly, the swarm does not converge to the target (global optimum), the previous eq. (20) can be proposed as follows:

$$P_i^d = c' \left( \sum_{j=1, j \neq i}^{N} c^' \frac{u_b - l_b}{2} S\left(\frac{|P_j^d - P_i^d|}{d_{ij}}\right) \frac{P_j - P_i}{d_{ij}} \right) + t_i$$  \hspace{1cm} (21)

where $u_b$ and $l_b$ are the upper and lower bound and $t_i$ is the target value in the $d^{th}$ direction. The parameter $c'$ is called the decreasing coefficient and it is responsible to decrease the comfort zone, repulsion zone and attraction zone. The value of $c'$ decreases with the number of iterations to balance the exploration and exploitation in the GOA and it can be calculated as below:

$$c' = c_{max} - \frac{t}{L} c_{min}$$  \hspace{1cm} (22)

Where $c_{max}$ and $c_{min}$ values, $t$ is the current iteration and $L$ is the maximum number of iterations.

The pseudo-code of GOA for ONU placement is as follows:

If number of iterations = < maximum limit

Initialize the swarms (a randomly distributed population or candidate solutions in the search space: Number of initial random locations of multiple
ONUs);
Initialize $c_{max}$, $c_{min}$, maximum number of iterations;
else
Select n number of the best solutions obtained so far in the initial iterations and formulate a new population.
end if
Calculate the fitness of each search agent (with every position of ONUs, find the value of the objective function).
$T^* =$ best search agent (the location of ONU for which we get minimum cost function value)
while (L' $<$ maximum number of iterations)
Update c using eq. (22)
for each search agent (position of ONU)
Normalize the distance between grasshoppers (distance obtained among ONUs);
Update the position of the current search agent by the eq. (21);
Correct the position of the current grasshopper if it is beyond the search (if the obtained position is not in accordance with the constraint) space;
end for
Update $T'$ in case of better solution;
Increase the number of iteration;
end while
Display $T'$ (most optimal location of ONUs which returned minimum value of overall cost function).

5. Computational complexity of algorithms

Computation complexity of an algorithm has a major role for determining its run time, which can be defined on the basis of the structure and implementation of the algorithm. Hence, to compute the efficiency of the algorithm for reaching to a solution, we calculate the computational complexity of each of the algorithms.

5.1. WOA

The computational complexity of the WOA is a function of the number of whales, variables, maximum number of iterations, and sorting mechanism of preys in each iteration. Due to the implementation of the Mergesort algorithm, the sort’s computational complexity has come out to be O(NlogN). Considering these, the overall computational complexity is explicated as below [7]:

$$O(\text{WOA}) = O((O(\text{Mergesort}) + O(\text{positionupdate})))$$

$$O(\text{WOA}) = O((t N \log N + d t))$$

$$O(\text{WOA}) = O(t N \log N + d t)$$

$$O(\text{WOA}) = O(N \log N)$$

where N, t and d stand for the number of whales, maximum number of iterations, and number of variables respectively.

5.2. GOA

In case of GOA, computation complexity is defined in terms of search agents n, maximum number of iterations k and dimension of the problem d. Computation complexity of GOA can be calculated as follows [23,50]:

For one iteration and single search agent, complexity of the algorithm is

$$O(\text{GOA}) = O(\text{fitnessevaluation})$$

$$O(\text{GOA}) = O(n^2 \ast d) + O(n \ast Cof)$$

For k iterations, complexity of the algorithm is

$$O(\text{GOA}) = O(k \ast (n^2 \ast d) + O(n \ast k \ast Cof))$$

If supposek = n, then complexity is

$$O(\text{GOA}) = O((n^3 \ast d) + O(n^2 \ast Cof))$$

$$O(\text{GOA}) = O(n^3)$$

Hence, computational complexity of GOA has the highest value among WOA and HHO and its value increases with n.

5.3. HHO algorithm

The computational complexity of HHO algorithm is based on initialization, fitness evaluation and updating of hawks. For initialization phase with N hawks computation complexity is O(N). For updating mechanism it is given by O(T*N) + O(T*N*D) which is composed of searching for the best location and updating the location vector of all hawks, where, T is the maximum number of iterations and D is the dimension of specific problems. Therefore, computational complexity of HHO is [22]:

$$O(\text{HHO}) = O(N \ast (T + T \ast D + 1))$$

In accordance with the computational complexity of WOA, GOA and HHO algorithms, it does not depend on the system configuration. The value of it depends on the step count only. This indicates the number of times each statement in algorithm is executed. Time complexity is a function of input size and we used input size for complexity calculation. As size of the input increases, the complexity of the algorithm also increases and it results in a cost inefficient algorithm.

The time of execution of an algorithm defines its run or execution duration of a program. For executing the program in this manuscript, the simulations have been performed on a Laptop with the following configurations: CORE i3, CPU 1.7 GHZ, 4.0 GB RAM, Windows 7. The simulation experiments have been conducted based on HHO, GOA and WOA algorithms. The analysis of the algorithms is accomplished based on the value of the objective function and run time for multiple ONUs deployment problem. Table 5 depicts the execution time for the aforementioned algorithms considering the similar parameters of simulations.

6. Results & discussion

In this research work, MATLAB software has been used to carry out simulation. The outcomes of the proposed strategies for multiple ONUs placement have been compared. For simulation, we have used the same parameters for all the three algorithms WOA, GOA and HHO as shown in Table 1 that is why in all upcoming figures, the initial points are identical. This means that each algorithm starts from the same random population/point. This shows that comparison is fair. The objective function values for the individual ONU1, ONU2 and ONU3 are represented by $M_1$, $M_2$ and $M_3$ respectively. Table 2 contains $M_1$, $M_2$ and $M_3$ values implementing HHO, GOA and WOA algorithms for different placement schemes of wireless subscribers and ONUs.

In this manuscript, the simulation experiments have been conducted using following input conditions: the network search area was assumed in sq-meters (1000 $\times$ 1000), the number of ONUs $- 3$ and number of

| Table 1 | Simulation parameters for WOA, GOA and HHO algorithms. |
|---------|--------------------------------------------------------|
| Number of candidate solutions | 100 |
| Lower Bound | 0,0,0,0,0,0 |
| Upper Lower | 1000,1000,1000,1000,1000 |
| Dimension | 6 |
| Number of iterations | 150 |
wireless users — 100. The whole area of the network was divided into non overlapping grids. The grid size was set as 10 by 10. The results were undertaken by taking the average of identical 150 combinations of inputs. The optimal positions of multiple ONUs considering different placement schemes were obtained. The values of the cost function of various placement strategies are compared. These strategies are: initial random placement and deterministic placement. In initial random placement, the starting position of ONU's and users are anywhere in the grid. In initial deterministic placement – ONU's and users are placed in the “center” of each grid. Following these placement schemes, there are four combinations of placement of ONU's and users in the present research work. Table 1 displays the simulation parameters for all three proposed algorithms – HHO, GOA and WOA. Table 2, exhibits the outcomes of the results of HHO, GOA and WOA with initial deterministic and random placement of users and ONUs. The lowest value of cost function is obtained for the initial deterministic placement of ONUs and random placement of users. Simulation experimental evidences reveal that the random placement of the users often result in optimized value of cost function in comparison with uniform distribution of users. The reason for this may be accounted as the removal of bias due to randomization process adopted for locating the users [53]. Hence, this configuration is used as reference to estimate the individual cost improvement for different algorithms. From Table 2, it is elicited that HHO algorithm performs better than the other two algorithms. HHO algorithm exhibits the better performance because of its competencies in balancing exploration and exploitation. WOA amends the cost of individual ONU deployment also. According to Table 3, for this configuration, the values of cost saving of HHO over GOA are 26%, 61%, 51% for ONU # 1, ONU # 2 and ONU # 3, respectively. Similarly, the values of cost improvement of HHO over WOA algorithm are 3.7%, 98%, 38% for ONU # 1, ONU # 2 and ONU # 3, respectively. The HHO algorithm improves the overall objective function value by minimizing it in comparison with WOA and GOA algorithms. The percentage of overall cost improvement of HHO in comparison with WOA and GOA algorithms for various placement schemes of ONUs and users is given in Table 3.

HHO algorithm proves its superiority over GOA and WOA for every placement scheme of user and ONU. The most significant improvement of HHO over GOA and WOA algorithms has been that of 19.18% and 4.31%, respectively, in case of randomly placed users and uniformly placed ONUs. The percentage of cost improvement of fitness value of WOA over GOA has been shown in Table 4.

The execution times of HHO, GOA and WOA algorithms are presented in Table 5. It is clear from Table 5 that HHO algorithm is the fastest optimizer when compared with WOA and GOA.

The performance analysis of GOA is accomplished by varying its controlling parameters F and l. The fitness values of GOA are displayed in Table 6 for different values of F and l.

It is obvious from equations (7), that F and l are controlling parameters and affect both exploitation and exploration of GOA. Different convergence curves have been plotted taking different values of F and l. Fig. 3 and Fig. 4 display the changes in convergence curve of GOA with respect to F and l parameters, against the average objective function value. According to Fig. 3, for F = 0.5, we get minimum value of cost function as the number of iterations increase; otherwise local optima stagnation occurs. The nature of the convergence curve does not show abruptness. For other values of F (0.2, 0.4, 0.6, 0.8 and 1.0), as shown in Fig. 3, the convergence curve exhibits poor exploration and exploitation, and returns local optima. In the similar manner, Fig. 4 depicts the variation in the cost function with changing values of l.

In GOA, the nature of the convergence curve changes with changing values of l. During exploration and exploitation phase of GOA, the fitness function value also varies with the change of l. As per Fig. 4, when l = 1.5, we get minimum value of the cost function. Other values of l, not only affect the rate of decay of convergence curve, but also alter the convergence value. In this way, dependency of exploitation and exploration of GOA on F and l parameters is also verified. In this manuscript, we accomplished the simulation of GOA by setting the values of F and l as 0.5 and 1.5, respectively. Fig. 5 shows the nature of the convergence curves of HHO, WOA and GOA algorithms. It is obvious from this figure that WOA and GOA get stuck in local optima with poor convergence, whereas HHO returns improved optimal solution with better convergence.

Further Fig. 6 demonstrates the change in fitness value of HHO algorithm by varying random jump length in LF. Table 7 shows the objective function value for different value of J in HHO algorithm.

The minimum value of objective function is achieved at J = 2.75(l — r2). Hence, in the present research work, simulation has been performed by setting the value of J as J = 2.75(l — r2)using HHO algorithm. The variation in the cost function value has been checked by exploring the modelling equation of the energy of the rabbit (eq. (3)). It is concluded that there is no considerable change in value of objective

Table 2
Cost improvement in FiWi for multiple deployments with WOA, GOA and HHO algorithms.

| Distribution of users | Placement of ONUs | Placement scheme | M   | M1  | M2  | M3  |
|-----------------------|-------------------|-----------------|-----|-----|-----|-----|
| Deterministic         | Deterministic     | WOA             | 214.8315 | 408.7299 | 191.1717 | 44.5929 |
|                       |                   | GOA             | 225.2700 | 196.0090 | 196.8843 | 196.9168 |
|                       |                   | HHO             | 197.3899 | 199.6692 | 196.7294 | 195.7442 |
| Random                | Deterministic     | WOA             | 197.4074 | 450.5129 | 118.8731 | 22.8361 |
|                       |                   | GOA             | 225.2700 | 196.0090 | 282.8843 | 196.9168 |
|                       |                   | HHO             | 190.0019 | 205.3757 | 208.6630 | 155.8052 |
| Random                | Deterministic     | WOA             | 179.5206 | 220.1031 | 203.3256 | 115.1332 |
|                       |                   | GOA             | 205.1152 | 170.1001 | 164.5350 | 280.7106 |
|                       |                   | HHO             | 172.1014 | 228.5645 | 202.5000 | 85.8776 |
|                       | Random            | WOA             | 209.9510 | 204.2620 | 202.6772 | 142.9136 |
|                       |                   | GOA             | 217.0819 | 199.9807 | 178.0709 | 274.1986 |
|                       |                   | HHO             | 191.2768 | 255.7750 | 165.5413 | 152.5143 |

Table 3
Overall Cost improvement in FiWi for multiple ONUs deployment with HHO, GOA and WOA algorithms.

| Placement schemes | Cost improvement (%) HHO over GOA | Cost improvement (%) HHO over WOA |
|-------------------|-----------------------------------|-----------------------------------|
| Deterministic placement of users and ONUs | 14.12 | 8.84 |
| Deterministic placement of users and random placement of ONUs | 18.56 | 3.91 |
| Random placement of users and deterministic placement of ONUs | 19.18 | 4.31 |
| Random placement of users and ONUs | 13.49 | 9.76 |
Based on No-Free-Lunch (NFL) theorem [54], the performance of all optimization algorithms changes from problem to problem. A single optimization algorithm cannot solve all types of problem. Therefore, with different problem and for their optimal performance we need various new optimizers [7,19].

ONU placement problem has been solved by many optimizers and those algorithms performed differently [1,7,19]. The salient features of this problem are: it is multimodal, real and structural optimization problem and comes under network planning issue. An optimal solution of the above problem defines the minimum value of the objective function. In this context HHO and GOA algorithms have been used to solve the afore stated problem, because in these domains these have been found to perform exceeding well based on various performance metrics. The authors proposed HHO algorithm in [22] and the findings of this research are as follows: HHO algorithm has excellent property to balance exploration and exploitation processes. It can avoid the problems of the search space like local optimal solutions, multimodality and deceptive optima. HHO optimizer searches for the optimal value in more promising regions in first half of the iterations and shows accelerated convergence rate in later half of the iterations. Hence, HHO algorithm balances exploitation and exploration very well. It uses the concept of Levy Flights (LFs) and this is the essence of this optimizer. LF is applied to imitate the actual random deceptive motions of rabbits (preys) during escaping phase and erratic, sudden and quick dives of hawks around the escaping rabbit (prey). It has been verified that LF-based movements are the optimal searching strategies for foragers/predators in non-destructive foraging situations. Furthermore, most of the animals like monkeys and sharks utilise LF-based patterns in their chasing activities. This feature of LF in HHO optimizer makes it different and more efficient than WOA and GOA algorithms. It is advantageous as it uses various length scales. The result of having small distances with a high probability, and long distances with a low probability is that each search will generally check a few solutions in an area before moving on to a far-off area to repeat the process. This explains that the HHO optimizer follows rigorous exploitation and does not allow it to get stuck in local optima. Moreover, HHO is much more efficient than constant-length random walk. On the contrary, LF makes algorithm more complex search algorithm in comparison with standard random walk. The results of computational complexity of HHO, WOA and GOA prove that HHO algorithm is reasonably fast, and yields competitive outcomes in comparison with WOA and GOA. HHO algorithm returns superior, competitive and high quality optimal solutions compared to GOA and WOA algorithms. The response of this optimizer is superb for multimodal functions also. HHO optimizer uses random, time varying dynamic values of escaping energy of rabbits which enable to explore and exploit the search area very effectively. HHO algorithm

![Convergence Curve for varying values of controlling parameter](image)

**Fig. 3.** Convergence curve of GOA for different values of $F$ controlling parameter.

| Table 4 | Overall Cost improvement in FiWi for multiple ONUs deployment with WOA over HHO algorithms. |
|---------|----------------------------------------------------------------------------------------------------------------------------------|
| Placement schemes          | Cost improvement (%) WOA over GOA |
| Deterministic placement of users and ONUs     | 4.86 |
| Deterministic placement of users and random placement of ONUs | 14.11 |
| Random placement of users and deterministic placement of ONUs | 14.26 |
| Random placement of users and ONUs            | 3.40 |

| Table 5 | Execution time for WOA, GOA and HHO algorithms. |
|---------|---------------------------------------------------|
| Problem name | Name of the metaheuristic algorithm | Execution Time (sec) |
| ONU placement | WOA | 399.333 |
| ONU placement | GOA | 432.134 |
| ONU placement | HHO | 111.480 |

| Table 6 | Value of objective function with different values of $l$ and $F$ of GOA. |
|---------|-------------------------------------------------------------|
| $F$     | $l$     | Objective Function Value |
| 0.0     | 1.0     | 196.3920 |
| 0.2     | 1.2     | 205.1228 |
| 0.4     | 1.4     | 209.7219 |
| 0.5     | 1.5     | 179.4220 |
| 0.6     | 1.6     | 205.0808 |
| 0.8     | 1.8     | 180.6738 |
| 1.0     | 2.0     | 180.7352 |

| $F$     | $l$     | Objective Function Value |
|---------|---------|--------------------------|
| 0.0     | 1.0     | 211.0388 |
| 0.2     | 1.2     | 207.8197 |
| 0.4     | 1.4     | 204.8951 |
| 0.5     | 1.5     | 179.4220 |
| 0.6     | 1.6     | 211.2814 |
| 0.8     | 1.8     | 205.1170 |
| 1.0     | 2.0     | 179.4220 |
uses different serial searching phenomenon based on $r'$ and $|E_{habt}|$. The diversification power of HHO algorithm is very strong as it implements different exploration mechanisms in reference to the average location of hawks. This results in boosting the exploratory behaviour of HHO algorithm in initial iterations. The aforementioned features are missing in WOA and GOA algorithms. As HHO algorithm can efficiently search optimal solution locally and provides better value than WOA and GOA algorithms, the chances of getting stuck in local optima automatically get minimised. This feature is the main reason, why the HHO algorithm could return the optimal solution, even when WOA and GOA got stuck in local optima.

However, as far as the ONU placement problem is concerned, HHO outperformed the GOA. Since ONU placement problem is a multimodal problem, we need an optimizer which should have strong exploration and target seeking capability. HHO has the superb exploration and exploitation handling capability, and hence it returned more optimal solution.

Though HHO algorithm renders many advantages, it has limitations also. In this algorithm, it is noticeable that final value of solution depends strongly on the initially generated size of population (number of search agents). With too small population size, the algorithm gets stuck in premature convergence. For higher value of it, the time of calculation increases and results in degraded computational efficiency. This feature of HHO algorithm becomes a challenge to solve real-world problems and to determine the most perfect value of the scale [39]. Further in [23], the authors proposed GOA and tested its efficiency on various platforms. It has been reported that GOA is capable to solve almost all types of problems efficiently. In [47–50], the authors applied GOA to solve various problems. The results demonstrate the superiority of GOA in comparison with the tasted algorithms. In the light of the above facts, it has been confirmed and justified that these algorithms are suitable in the solving the ONU placement problem in FiWi network.

We analyzed the various features of WOA and GOA algorithms because of which they fail to yield good performance when compared with HHO algorithm.

The limitations of WOA are summarized as:

1. WOA exhibits poor convergence speed in both exploration and exploitation phases as these two metrics depend on a single parameter, which is $a'$ [7]. It is essential to improve the balancing between exploration and exploitation in WOA [55]. The failure in fine tuning of exploitation and exploration makes WOA inferior than HHO, as in...
this manuscript, and results in local optima entrapment. Although WOA shows faster convergence than GOA but exhibits higher value of computational complexity than HHO.

2. WOA may not be an efficient optimizer for various problems. These are highly complex problems [56], classification and dimensionality reduction problems [57], vehicle fuel consumption problem with complex environmental constraints [58] and single and multidimensional 0–1 knapsack problems [59].

3. WOA implements encircling mechanism in the search space. Due to this mechanism WOA gets trapped in local optima resulting in poor performance [60–62]. It also shows inferiority in improving the best solution after each iteration [63].

4. The value of the convergence of WOA largely depends on the optimal starting value of the controlling parameters \(a'\) and \(C'\). If these values are not set optimally, WOA may show premature convergence and fail to give optimum result. The nature of the convergence curve of WOA is good but still further improvements are possible, which may result in the sophisticated variants of the aforementioned algorithm [7,60].

Also, based on this research it has been concluded that, GOA could not return minimum value of cost function for multiple ONUs placement in comparison with WOA and HHO algorithms. The limitations of the GOA can be summarised on the basis of the following points:

1. GOA gets trapped in local optima and shows slow convergence speed.
2. The population diversity of GOA is poor, hence it suffers from weak local search ability. If an optimizer has poor population diversity, it may get stuck in local optima. To improve global exploration skill of GOA, there is need to enhance randomness of the search agents’ movement [64].
3. The value of the convergence of GOA largely depends on the optimal starting value of the controlling parameters \(l\) and \(F\). If these values are not set optimally, GOA may show premature convergence and fails to give optimum result. Figs. 1 and 2 show this behavior.
4. The exploration and exploitation of GOA extensively depends on \(F\) and \(l\). According to Figs. 1 and 2, for \(F = 0.5\) and \(l = 1.5\), we get the minimum value of cost function against total number of iterations respectively. The proper setting of \(F\) and \(l\) is required to avoid local optima.

7. Conclusion

In this research, multiple ONUs are optimally placed using HHO and GOA algorithms in FiWi access network. The results have been obtained by placing users, ONUs in deterministic and random manner in the network. The simulation outcomes have been compared with the WOA. Results exhibit that HHO algorithm performs better than WOA and GOA algorithms and provides the most optimal value of the objective function. The outcomes have been analyzed and evaluated considering nature of convergence curve, the percentage reduction in the cost function and computational complexity. For HHO and GOA algorithms, the optimal selection of the controlling parameters for the aforementioned problem is accomplished. Optimized locations of ONUs have several important effects on throughput, load balancing, cost efficiency and interference. Optimal placement of ONUs enhances throughput, cost efficiency, improves load balancing and reduces interference. In future, HHO and GOA shall be used for enhancing the system parameters using ONU placement problem in FiWi network.

CRediT authorship contribution statement

**Puja Singh:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Visualization, Writing - original draft, Writing - review & editing. **Shashi Prakash:** Supervision, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.
of Whale Optimization Algorithm, Hindawi Comput. Intelligence and Neurosci. 2019 (2019) 1–25.

[61] L.M. Pecora, T.L. Carroll, Synchronization of chaotic systems, Chaos: An Interdiscip. J. of Nonlinear Sci. 25 (2015) 1–12.

[62] S. Nagaraj, G. Raju, V. Srinadh, Data encryption and authentication using public key approach, Procedia Comput. Sci. 48 (2015) 126–132.

[63] Z. Xu, Y. Yu, H. Yachi, J. Ji, Y. Todo, S. Gao, A novel memetic whale optimization algorithm for optimization, Int. Conf. on Swarm Intelligence 10941 (2018) 384–396.

[64] S. Cheng, Y. Shi, Q. Qin3, Q. Zhang, R. Bai, Population Diversity Maintenance in Brain storm Optimization algorithm, JASCR 4 (2014) 83–97.