Journal of Computers Vol. 35 No. 4, August 2024, pp. 39-58
doi: 10.53106/199115992024083504004

Improved Gray Wolf Algorithm Based on African Vulture Multi-Strategy and Its Application in Channel Estimation

Ji-Hua Chai, Li-Yi Zhang, Ting Liu, Yun-Shan Sun, and Yong Zhang*

School of Information Engineering, Tianjin University of Commerce, Tianjin 300134, China
Cjh999L@outlook.com, {zhangliyi, liuting, sunyunshan, zhangyong}@tjcu.edu.cn

Received 3 May 2023; Revised 30 May 2023; Accepted 31 May 2023

Abstract. The Grey Wolf Optimizer (GWO) is known for its simple structure, easy implementation, and few control parameters, but it has problems such as slow convergence, local optimization of capture, and late reduction of population diversity. To address these shortcomings, this paper proposes fusing three different strategies. Firstly, the African vulture algorithm is combined with the GWO to exploit the fast convergence speed and high accuracy of the African vulture algorithm in finding the best solution. Secondly, the leading wolf guidance strategy is introduced to increase population diversity. Finally, adaptive weights are added to dynamically adjust the search step size and balance the global and local search ability of the GWO. The proposed algorithm is validated through testing on 23 benchmark test functions, demonstrating its superior performance. For millimeter-wave large-scale MIMO systems, a deep convolutional neural network is used for channel estimation, using the correlation between space and frequency to simultaneously input the corrupted channel matrix of adjacent subcarriers into the convolutional neural network. Furthermore, we address the drawbacks of manually setting the hyperparameters of convolutional neural networks, which may lead to overfitting and large errors. Therefore, we propose to use the improved grey Wolf algorithm (AVGWO) of African vulture to find the hyperparameters of the convolutional neural network model and improve the accuracy of the model prediction channel.

Keywords: grey wolf optimization algorithm, African vulture, bootstrap strategy, adaptive weighting, channel estimation

1 Introduction

Meta-heuristic optimization algorithm are optimization algorithms that achieve optimal solutions by simulating natural and human intelligence. Since their introduction in the 1960s, they have experienced rapid development and can be divided into four main categories.

The first category is nature-inspired evolutionary algorithms (NEA), which are optimization algorithms based on genetics and natural selection. In NEA, the population is composed of many individuals, each of which represents a solution. The algorithm chooses the best solution acting as a parent and generates new individuals through interchange and mutation. The algorithm then chooses the fittest individuals as part of the new generation population, and repeats this process until termination conditions, such as maximum iterations or finding the optimal solution. Examples of NEA include Genetic Algorithm (GA) [1] and Differential Evolution (DE) [2]. The social interactions and information sharing of biological swarms in nature serve as the basis for the second category of intelligence algorithms, known as swarm-based intelligence algorithms (AIs). They use swarm intelligence for collaborative search to find the optimal solution in the search space. Predation, reproduction, and hunting are three of the most typical and well-known social behaviors of animals. Numerous meta-heuristic optimization algorithm have been proposed based on these behaviors, including Particle Swarm Optimization (PSO) [3], Ant Colony Optimization [4], Artificial Bee Colony (ABC) [5], Artificial Fish Swarm Algorithm (AFSA) [6], Grey Wolf Optimizer (GWO) [7]. The third category pertains to meta-heuristic optimization algorithm enlightened by human behaviors, encompassing traits like teaching, socializing, learning, emotional intelligence, and managerial skills. Such as algorithms include Teaching-Learning-Based Optimization (TLBO) [8] and Political Optimizer (PO) [9]. Algorithms based on physics and chemistry comprise the fourth category, which majorly mimic the physical laws and chemical reactions of nature. In this category, algorithms emulate the laws of physics and chemistry to navigate the search space. They leverage concepts such as gravity, inertia, light refraction, and molecular dynamics to facilitate communication and collaboration among individuals. Noteworthy examples

* Corresponding Author

of these algorithms are Gravitational Search Algorithm (GSA) [10], Charged System Search (CSS) [11], Atom Search Optimization (ASO) [12].

The Grey Wolf optimizer was introduced by the distinguished scholar Mirjalili of Griffith University in Australia in 2014. It draws inspiration from the hunting behavior of wolves and is renowned for its simple structure, few control parameters, and ease of implementation, making it a popular subject of study among researchers. By emulating the adaptive and collaborative nature of wolves, GWO can efficiently explore complex search spaces and converge towards optimal solutions. The algorithm has already demonstrated significant potential in tackling diverse optimization challenges across a range of fields, and it continues to inspire new developments and improvements in the field of Meta-heuristic optimization.

In the domain of flow shop scheduling, Jeet [13] applied the GWO algorithm to the flow shop scheduling problem, the researchers compared the results obtained using GWO with those obtained using multi-objective genetic algorithms, particle swarm algorithms, and NSGA-II algorithms, and demonstrated the superior performance of GWO in resolving this scheduling issue. In another study, Lu et al. [14] introduced a boosted Grey Wolf algorithm, which amalgamates the diversity of cellular automat with the enhanced benefits of variable neighborhood search. This algorithm was applied to address the mixed workshop scheduling, achieving promising results. In the domain of path planning, the study by Luo et al. [15] presented a superior technique utilizing Grey Wolf Optimization to achieve automatically route planning for mobile automation in sophisticated circumstances. By leveraging the power of GWO, this algorithm showcased remarkable efficacy in navigating intricate environments and avoiding obstacles. These studies showcase the versatility and robustness of GWO in tackling complex optimization problems across diverse domains; Shi et al. [16] stated the AP-GWO approach to adaptive multi-drone path planning to boost the Grey Wolf algorithm for problems such as slow convergence of path planning and insufficient flight paths. In the domain of image classification, Convolutional Neural Networks (CNNs) were utilized by Ladi et al. [17] to categorize hyperspectral pictures by employing a novel method, where the GWO algorithm was utilized to boost six hyperparameters of the CNN. Compared with traditional models, the proposed Convolutional neural networks optimized by GWO had higher accuracy. In the domain of signal handling, Saxena et al. [18] advocated the Evolutionary Operators Equipped Grey Wolf Optimizer (E-GWO), which combines tournament selection operators, cross and alterations implemented during the position update step and applied it to harmonic estimation.

In this paper, we apply the GWO algorithm and the proposed AVGWO to channel estimation problems in the field of signal processing. As we all know, in communication, due to the non-ideal characteristics of the transmission channel and the influence of superimposed noise, it is difficult to guarantee the reception quality. Therefore, channel estimation is necessary to improve communication quality. With the increasing maturity of machine learning theory and technology, many scholars have applied machine learning to channel estimation. Orthogonal frequency-division multiplexing (OFDM) receivers are currently utilized in wireless communications. Gao et al. [19] proposed a new Deep Learning (DL) method that combines DL with expert knowledge to replace existing orthogonal frequency-division multiplexing receivers in wireless communications. The claimed DL receiver utilizes a block-by-block method of signal processing. Compared to the LMSE error algorithm, the claimed DL receiver provides more precise channel estimation. ChanEstNet, a cutting-edge Deep Learning approach, was constructed by Liao et al. [20] for channel estimation in wireless communication. Convolutional Neural Network (CNN) is utilized to derive feature vectors of the channel response, and a Recursive Neural Network (RNN) is utilized to carry out channel estimate. In comparison to conventional approaches, simulation observations demonstrate that the advocated channel estimation approach exhibits lower calculating sophistication and substantially boosted efficiency in rapid-fire mobile atmospheres. Dong et al. [21] proposed to forecast more genuine channels by utilizing a conditional Generative Adversarial Network through oppositional learning of two DL networks. Simulation Results demonstrate that the newly suggested cGAN-based method performs better than the current DL approaches and excellent reliability in systems utilizing massive MIMO.

This paper proposes the use of deep CNN for channel estimation in millimeter-wave massive MIMO-OFDM systems. The advocated method is based on a spatial-frequency CNN for channel estimation, taking advantage of the correlation between adjacent subcarriers in OFDM systems. Deep CNN have numerous parameters, such as learning rates, kernel sizes, and numbers, that directly impact their performance. Therefore, setting the optimal hyperparameters of deep CNN is critical. To generate the best network hyperparameter configurations, the paper introduces the GWO algorithm and a CNN network based on GWO is proposed to enhance the performance of deep CNN in channel estimation. As the GWO algorithm flaws comprise a poor rate of convergence and being susceptible to regional optima, the paper also employs the proposed AVGWO algorithm to optimize the hyperparameters of deep CNN and verify its advantages.

2 Basic Principles of Traditional Grey Wolf Optimizer

In 2014, Mirjalili et al [7] presented the GWO, a meta-heuristic optimization algorithm method. The α , β , and δ wolves as the optimal solutions, driving ω wolves to migrate and look for the overall optimal solution, which served as the inspiration for this algorithm. The three primary phases of wolf hunting in the wild are as follows.

2.1 Encircling Phase

During the encircling process, wolves use the following position update equation:

$$\mathbf{X}(t+1) = \mathbf{X}_p - \mathbf{A} \times \mathbf{D} \ . \tag{1}$$

Where t is the current iteration number, $\mathbf{X}_p$ is the position vector of the prey, $\mathbf{X}(t+1)$ is the position vector of the wolf in the next iteration, $\mathbf{D}$ is the distance between the wolf and the prey:

$$\mathbf{D} = |\mathbf{C} \cdot \mathbf{X}_{\mathrm{p}}(t) - \mathbf{X}(t)| \ . \tag{2}$$

Where $\mathbf{X}(t)$ is the position vector of the current wolf, **A** and **C** are coefficient vectors:

$$\mathbf{A} = 2 \cdot \mathbf{r}_1 \cdot \mathbf{a} - \mathbf{a} \ . \tag{3}$$

$$\mathbf{C} = 2 \cdot \mathbf{r}_2 \ . \tag{4}$$

Where $\mathbf{r}_1$ and $\mathbf{r}_2$ are random vectors in [0,1], the components of $\mathbf{a}$ decrease linearly from 2 to 0, as follows:

$$\mathbf{a} = 2 - \frac{2t}{\mathrm{t}_{\max}} \ . \tag{5}$$

Where $\mathrm{t}_{\max}$ is the maximum number of iterations.

2.2 Hunting Phase

In the GWO algorithm, the α , β , and δ wolves are considered as the optimal solutions, and the rest of the wolves are influenced by these leaders to move towards the global optimal solution. During the hunting process, the α , β , and δ wolves have a better understanding of the prey's position. Therefore, the positions of the α , β, and δ wolves are used as the prey's location, and each wolf changes its position using the following equation:

$$\mathbf{X}(t+1) = \frac{\mathbf{X}_1 + \mathbf{X}_2 + \mathbf{X}_3}{3} \ . \tag{6}$$

Where $\mathbf{X}_1$, $\mathbf{X}_2$ and $\mathbf{X}_3$ are calculated as follows:

$$\begin{cases} \mathbf{X}_1 = \mathbf{X}_\alpha - \mathbf{A}_1 \cdot \mathbf{D}_\alpha \\ \mathbf{X}_2 = \mathbf{X}_\beta - \mathbf{A}_2 \cdot \mathbf{D}_\beta \\ \mathbf{X}_3 = \mathbf{X}_\delta - \mathbf{A}_3 \cdot \mathbf{D}_\delta \end{cases} . \tag{7}$$

Where $\mathbf{X}_1$, $\mathbf{X}_2$ and $\mathbf{X}_3$ are the positions of the top three wolves in the population at time t , and $\mathbf{A}_1$, $\mathbf{A}_2$, and

$\mathbf{A}_3$ are coefficient vectors, which can be calculated using equation (3). $\mathbf{D}_\alpha$, $\mathbf{D}_\beta$, and $\mathbf{D}_\delta$ are the distances between grey wolf and α , β , and δ wolves, calculated as follows:

$$\begin{cases} \mathbf{D}_\alpha = \left|\mathbf{C}_1 \cdot \mathbf{X}_\alpha - \mathbf{X}\right| \\ \mathbf{D}_\beta = \left|\mathbf{C}_2 \cdot \mathbf{X}_\beta - \mathbf{X}\right| \\ \mathbf{D}_\delta = \left|\mathbf{C}_3 \cdot \mathbf{X}_\delta - \mathbf{X}\right| \end{cases} . \tag{8}$$

Where $\mathbf{C_1}$, $\mathbf{C_2}$, and $\mathbf{C_3}$ are coefficient vectors, calculated using equation (4).

2.3 Attacking Phase

When the prey has ceased running, the wolf will attack it, and this process is controlled by a. As equation (3) shows, the range of the vector a is [0, 2], and as a decreases, the components of the vector $\mathbf{A}$ also decrease. When |a|<1, the wolf approaches the prey and launches an attack; when |a|>1, the wolf moves away from the prey and seeks new prey.

3 Principle of African Vulture-based and Multi-strategy Fusion GWO Algorithm

The GWO algorithm boasts several advantages, including its simple principles, minimal parameter tuning requirements, ease of implementation, and straightforward operations. However, like many algorithms, the GWO is not without its shortcomings, including poor solution accuracy and sluggish convergence, insufficient population diversity, and a propensity for local optima. In response, this article presents a series of novel improvement strategies aimed at addressing these issues. By implementing these enhancements, we intend to boost the overall efficacy and capability of the GWO algorithm, enabling it to provide more accurate and efficient solutions across a broader range of applications.

3.1 Improvement of Bootstrap Location

Search Mechanism Based on the African Vulture Algorithm. One of the main drawbacks of the GWO is slow convergence, and one reason for this slow convergence is that the grey wolf optimizer only considers the distance between the current wolf and the target solution when updating the wolf position, without considering the distance between other wolves and the target solution and other factors. This local update strategy may cause the algorithm to get trapped in regional optimal solutions, which diminishes convergence. In order to accelerate the convergence speed, this paper combines the grey wolf optimizer with the African vulture algorithm and introduces a search mechanism based on the African vulture algorithm in the pursuit phase of the grey wolf optimizer. The African vulture algorithm [22] simulates the foraging behavior of African vultures for problem optimization. African vulture search for food, and if vultures are in a satiated state, they have high energy and can fly lengthy distances to find food. Conversely, if African vulture are starved, vulture will not have sufficient energy to flight lengthy distances. The fullness rate is expressed in equation (9).

$$\mathbf{F} = (2 \cdot r_3 + 1) \cdot \gamma \cdot \left(1 - \frac{t}{t_{\max}}\right) + \mathbf{H} . \tag{9}$$

Where γ is the random coefficient of [-1, 1] and the equation for $\mathbf{H}$ is the following description:

$$\mathbf{H} = \mathbf{h} \cdot \left(\sin^{\lambda}\left(\frac{\pi}{2} \cdot \frac{t}{\mathrm{t}_{\max}}\right) + \cos\left(\frac{\pi}{2} \cdot \frac{t}{\mathrm{t}_{\max}}\right) - 1 \right) . \tag{10}$$

Where $\mathbf{h}$ is the random vector of [-2, 2] and λ is a constant.

In the pursuit phase of the GWO, since wolfs will search for prey in different regions, these regions based on two diverse policies as follows and use the parameter to choose any strategy, $p1$ is a constant whose value is between 0 and 1, while $rand$ is a casual number in the range 0 to 1. When $p1 \geq rand$, the grey wolf optimizer selects to randomly search for prey in the area of the best wolves, and when $p1 \geq rand$, the grey wolf chooses to find its prey randomly in the area of (lb , up) , as in equation (11).

$$\mathbf{Y}(t+1)=\begin{cases}\mathbf{X}_\alpha-\mathbf{D}_\alpha\cdot\mathbf{F} & \text{if}\,(p1\geq rand)\\ \mathbf{X}_\alpha-\mathbf{F}+\mathbf{r}_4\cdot\left(\left(\mathbf{up}-\mathbf{lb}\right)\cdot\mathbf{r}_5+\mathbf{lb}\right) & \text{if}\,(p1< rand)\end{cases}. \tag{11}$$

Where $\mathbf{Y}(t + 1)$ is the next generation of the wolf's position vector, $\mathbf{r}_4$ is a random vector from 0 to 1, **up** and **lb** is the upper and lower bounds of the population, and $\mathbf{r}_5$ is a random vector from 0 to 1 in order to enhance randomness.

Search Mechanism Based on the Leading Wolf Guidance Strategy. During the pursuit phase of the GWO, updating positions are performed using only the places of the α , β , and δ wolves. However, when the α, β, and δ wolves are trapped in local optima, this can decrease in population diversity and an early convergence to a local optimum. Additionally, in the GWO, the α wolf plays a fundamental part as the target solution for the iteration to end. To address these issues, the Grey Wolf optimizer introduces a weight vector based on to give greater weight to the α wolf when changing the location of the wolves. Therefore, when changing the location of the wolves, a greater emphasis is placed on the **u** update base vector for the α wolf, assigning it more weight. Moreover, to increase population diversity, an update perturbation vector **Δ** and a random adjustment factor $\mathbf{r}_6$ are incorporated, as described in equations (13).

$$Z(t+1)=u+r_6\cdot\Delta\ , \tag{12}$$

$$\begin{cases}u=\frac{1}{2}\cdot X_1+\frac{1}{3}\cdot X_2+\frac{1}{6}\cdot X_3\\ \Delta=\frac{1}{2}\cdot\left(X_\alpha-X\right)+\frac{1}{4}\cdot\left[\left(X_\beta-X\right)+\left(X_\delta-X\right)\right]\end{cases}. \tag{13}$$

In the above improved strategy, the search approach based on the African vulture algorithm and the leading wolf guidance strategy, eventually adopting a greedy mindset to select one as the location for the next iteration of the current grey wolf.

$$\mathbf{X}(t+1)=\begin{cases}\mathbf{Y}(t{+}1) & if\,(\mathrm{F(Y)}\leq\mathrm{F(Z)})\\ \mathbf{Z}(t{+}1) & if\,(\mathrm{F(Y)}>\mathrm{F(Z)})\end{cases}. \tag{14}$$

Where F(Y) and F(Z) are the current adaptation values of the grey wolf, respectively.

3.2 Adaptive Adjustment of Weights

The role of inertia weight in the Grey Wolf Optimization algorithm cannot be overstated, as it plays a fundamental part in determining the algorithm's convergence velocity and optimization precision. If the weight of linear inertia is not appropriately selected, it can negatively impact the algorithm's convergence velocity. Therefore, this article proposes an adaptive approach to changing the weight based on the distribution of the GWO population. In Equation (16), the inertia weight $\mathbf{w}$ can significantly affect the algorithm's ability to conduct both global and regional searches, also accelerate convergence velocity and improve optimization precision. When the inertia weight is high, the algorithm tends to favor global explore, while a low inertia weight favors regional search. In the algorithm, the inertia weight $\mathbf{w}$ is used to influence the three alpha wolves' update positions with respect to other wolves.

$$\begin{cases} \mathbf{X}_1 = \mathbf{w}\cdot\mathbf{X}_\alpha - \mathbf{A}_1\cdot\mathbf{D}_\alpha \\ \mathbf{X}_2 = \mathbf{w}\cdot\mathbf{X}_\beta - \mathbf{A}_2\cdot\mathbf{D}_\beta \\ \mathbf{X}_3 = \mathbf{w}\cdot\mathbf{X}_\delta - \mathbf{A}_3\cdot\mathbf{D}_\delta \end{cases} . \tag{15}$$

$$\mathbf{w} = \mathrm{d}_1\cdot\left(\mathbf{X}_{\mathrm{worst}} - \mathbf{X}_{\mathrm{best}}\right) + \mathrm{d}_2\cdot\frac{\mathbf{up}-\mathbf{lb}}{t+1} . \tag{16}$$

In this algorithm, d_1 and d_2 are two constant parameters, $\mathbf{X}_{\mathrm{worst}}$, indicate the location vectors of the worst, $\mathbf{X}_{\mathrm{best}}$ indicate the location vectors of the finest wolves in the current population. When the GWO algorithm's early iterations, if the wolf pack becomes trapped in a regional optimum, and the disparities between the best and worst solutions is insignificant, increasing $\mathbf{w}$ can boost the algorithm's global search capability. Given that the latter portion of $\mathbf{w}$ is inconsequential to the search, more sizable strides can be taken to amplify the search scope, thereby obviating the risk of being ensnared in regional optimum. With the amplification of iterations, the impact of the latter half progressively abates. If the optimum solution has not yet been obtained at this point, the first half of $\mathbf{w}$ dominates, enabling larger steps to be taken to seek out the optimum solution.

3.3 Flowchart of AVGWO Algorithm

Fig. 1. Illustrates the flow diagram of the AVGWO algorithm

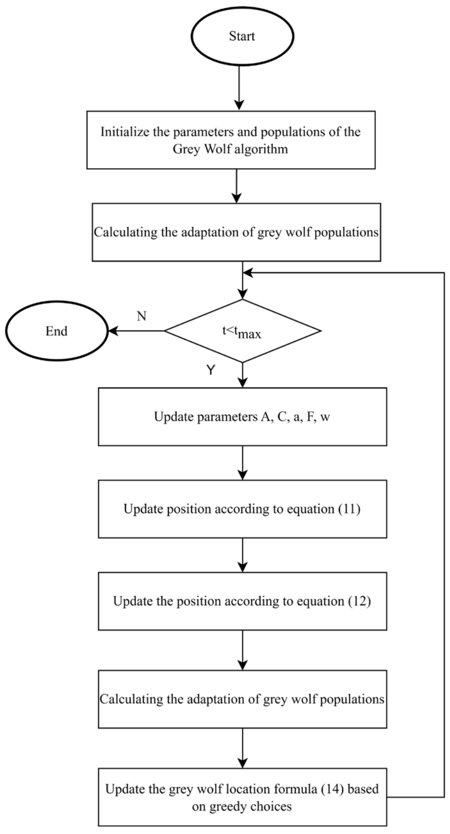

Start
Initialize the parameters and populations of the Grey Wolf algorithm
Calculating the adaptation of grey wolf populations
End
N
t<tmax
Y
Update parameters A, C, a, F, w
Update position according to equation (11)
Update the position according to equation (12)
Calculating the adaptation of grey wolf populations
Update the grey wolf location formula (14) based on greedy choices

Fig. 1. Flow graph of AVGWO algorithm

4 Performance Testing Based on African Vulture and Multi-Strategy Fusion Grey Wolf Optimization Algorithms

This study evaluated the manifestation of the AVGWO algorithm by conducting experiments on 23 benchmark test functions, In the Table 1, such as single-peak, multi-peak, fixed-dimension multi-peak test functions. AVGWO was compared with traditional GWO and other enhanced GWO, such as Modified Grey Wolf Optimizer (MGWO) that Padhy et al. [23] suggested, Improved Grey Wolf Optimizer (I-GWO) that Nadimi-Shahraki et al. [24] suggested, Soloklo et al. [25] suggested Fast-Dynamic GWO (FDGWO), and Aquila Grey Wolf Optimizer (AGWO) that Ma et al. [26] suggested. The number of populations was set to 50 and the number of iterations was set to 1000, which were all carried out under identical circumstances. To minimize the impact of randomness, each method was independently evaluated 30 times toward each test function.

Table 1. Benchmark functions

Function	Dim	Range	F_{min}
$F_1(x)=\sum_{i=1}^{Dim} x_i^2$	30	[-100,100]	0
$F_2(x)=\sum_{i=1}^{Dim}\lvert x_i\rvert+\prod_{i=1}^{Dim}\lvert x_i\rvert$	30	[-10,10]	0
$F_3(x)=\sum_{i=1}^{Dim}\left(\sum_{j-1}^{i} x_j\right)^2$	30	[-100,100]	0
$F_4(x)=\max x_i\left\{\lvert x_i\rvert,1\le i\le Dim\right\}$	30	[-100,100]	0
$F_5(x)=\sum_{i=i}^{Dim-1}\left[100\left(x_{i+1}-x_i^2\right)^2+\left(x_i-1\right)^2\right]$	30	[-30,30]	0
$F_6(x)=\sum_{i=i}^{Dim}\left(\left[x_i+0.5\right]\right)^2$	30	[-100,100]	0
$F_7(x)=\sum_{i=i}^{Dim} ix_i^4+rand\left[0,1\right)$	30	[-1.28,1.28]	0
$F_8(x)=\sum_{i=1}^{Dim}-x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)$	30	[-500,500]	−418.9829×Dim
$F_9(x)=\sum_{i=1}^{Dim}\left[x_i^2-10\cos\left(2\pi x_i\right)+10\right]$	30	[-5.12,5.12]	0
$F_{10}(x)=-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos\left(2\pi x_i\right)\right)+20+e$	30	[-32,32]	0
$F_{11}(x)=\frac{1}{4000}\sum_{i=1}^{Dim}x_i^2-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	30	[-600,600]	0
$F_{12}(x)=\frac{\pi}{n}\left\{10\sin\left(\pi y_1\right)+\sum_{i=1}^{Dim-1}\left(y_i-1\right)^2\left[1+10\sin^2\left(\pi y_{i+1}\right)\right]^2\right\}+\sum_{i=1}^{Dim}u\left(x_i,10,100,4\right)$	30	[-50,50]	0
$y_i=1+\frac{x_i+1}{4}u\left(x_i,a,k,m\right)=\begin{cases}k\left(x_i-a\right)^m & x_i>a\\ 0 & -a<x_i<a\\ k\left(-x_i-a\right)^m & x_i<-a\end{cases}$			

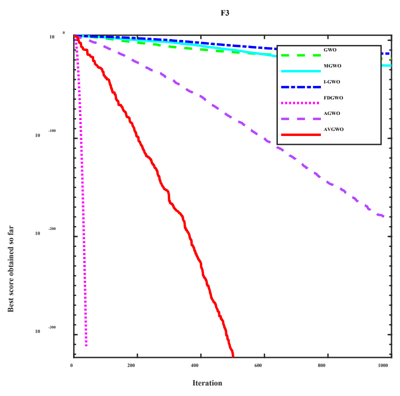
$F_{13}(x)=0.1\left\{\begin{matrix}\sin^2(3\pi x_1)+\sum_{i=1}^{Dim}(x_i-1)^2\left[1+\sin^2(3\pi x_i+1)\right]+\\(x_n-1)^2\left[1+\sin^2(2\pi x_n)\right]\end{matrix}\right\}$ + $\sum_{i=1}^{Dim}u(x_i,5,100,4)$	30	[-50,50]	0
$F_{14}(x)=\left(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}\left(x_i-a_{ij}\right)^6}\right)^{-1}$	2	[-65,65]	0.998
$F_{15}(x)=\sum_{i=1}^{11}\left[a_i-\frac{x_1\left(b_i^2+b_ix_2\right)}{b_i^2+b_ix_3+x_4}\right]^2$	4	[-5,5]	0.00030
$F_{16}(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	2	[-5,5]	-1.0316
$F_{17}(x)=\left(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6\right)^2+10\left(1-\frac{1}{8\pi}\right)\cos x_1+10$	2	[-5,5]	0.398
$F_{18}(x)=\left[1+\left(x_1+x_2+1\right)^2\left(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2\right)\right]$ $\times\left[30+\left(2x_1-3x_2\right)^2\times\left(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2\right)\right]$	2	[-2,2]	3
$F_{19}(x)=-\sum_{i=1}^{4}c_i\exp\left(-\sum_{j=1}^{3}a_{ij}\left(x_j-p_{ij}\right)^2\right)$	3	[1,3]	-3.86
$F_{20}(x)=-\sum_{i=1}^{i=4}c_i\exp\left(-\sum_{i=1}^{0}a_{ij}\left(x_j-p_{ij}\right)^2\right)$	6	[0,1]	-3.32
$F_{21}(x)=-\sum_{i=1}^{5}\left[\left(X-a_i\right)\left(X-a_i\right)^T+c_i\right]^{-1}$	4	[0,10]	-10.1532
$F_{22}(x)=-\sum_{i=1}^{7}\left[\left(X-a_i\right)\left(X-a_i\right)^T+c_i\right]^{-1}$	4	[0,10]	-10.4028
$F_{23}(x)=-\sum_{i=1}^{10}\left[\left(X-a_i\right)\left(X-a_i\right)^T+c_i\right]^{-1}$	4	[0,10]	-10.5363

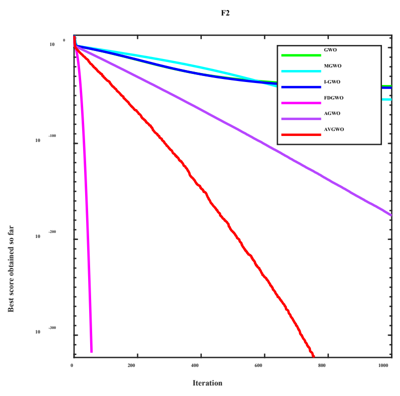
4.1 Comparison of AVGWO Algorithm with Other Improved Grey Wolf Optimizers

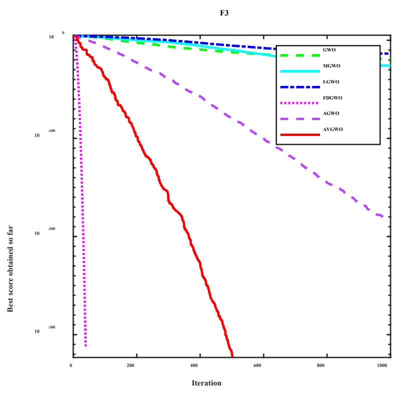
Table 2 displays the average and standard deviation of AVGWO algorithm and other improved Grey Wolf optimizers. It is capable of being illustrated that AVGWO performs better than GWO, MGWO, I-GWO, and AGWO in F1–F13. In single-peak functions F1-F4 and multi-peak functions F9 and F11, both AVGWO and FDGWO obtain the finest solution, but FDGWO is faster to converge than AVGWO. However, in F5, F6, F8, F12, and F13, FDGWO showed poorer performance. In F17, F18, and F20, AVGWO was less effective than the others but better than FDGWO, and in F21, F22, and F23, AVGWO was more effective. Fig. 2 displays the convergence curves of AVGWO, GWO, MGWO, I-GWO, FDGWO, AGWO in the benchmark functions F1-F15, F17-F23.

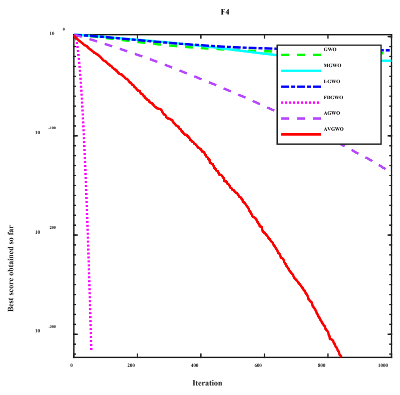
Table 2. The results of GWO, MGWO, I-GWO, FDGWO, AGWO and AVGWO on 23 benchmark problems

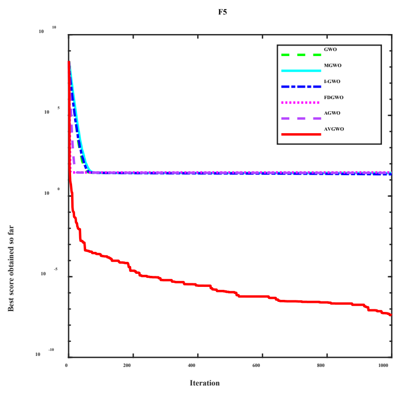
Function		GWO	MGWO	I-GWO	FDGWO	AGWO	AVGWO
F1	Average	2.55E-70	1.10E-93	5.63E-71	0	4.18e-321	0
	Std	3.72E-70	5.15E-93	2.11E-70	**0**	**0**	**0**
F2	Average	5.01E-41	6.62E-55	1.19E-42	0	4.76E-179	0
	Std	5.02E-41	7.78E-55	1.73E-42	**0**	0.00E+00	**0**
F3	Average	3.68E-20	2.16E-26	5.61E-14	0	2.74E-182	0
	Std	8.26E-20	4.58E-26	2.22E-13	**0**	**0**	**0**
F4	Average	1.95E-17	7.15E-25	1.14E-14	**0**	4.12E-140	0
	Std	3.40E-17	1.17E-24	2.41E-14	**0**	1.44E-139	**0**
F5	Average	2.64E+01	2.62E+01	2.24E+01	2.90E+01	2.72E+01	4.06E-08
	Std	5.31E-01	6.60E-01	3.40E-01	4.61E-07	4.46E-01	2.71E-08
F6	Average	3.43E-01	3.84E-01	1.04E-05	3.54E+00	2.56E+00	1.15E-10
	Std	3.39E-01	3.01E-01	2.42E-06	2.48E-01	6.19E-01	2.94E-10
F7	Average	4.18E-04	3.22E-04	7.63E-04	2.52E-05	6.78E-05	4.43E-05
	Std	1.60E-04	1.82E-04	3.40E-04	1.77E-05	4.70E-05	2.55E-05
F8	Average	-6.35E+03	-5.63E+03	-9.56E+03	-2.36E+03	-3.53E+03	**-1.26E+04**
	Std	7.13E+02	1.07E+03	1.36E+03	3.83E+02	3.88E+02	5.83E-06
F9	Average	1.90E-01	0	1.49E+01	**0**	**0**	**0**
	Std	1.02E+00	**0**	7.35E+00	**0**	**0**	**0**
F10	Average	1.23E-14	7.99E-15	9.77E-15	8.88E-16	4.44E-15	8.88E-16
	Std	3.13E-15	1.32E-15	2.99E-15	**0**	**6.38E-16**	**0**
F11	Average	1.35E-03	**0**	3.04E-03	**0**	**0**	**0**
	Std	3.51E-03	0	6.83E-03	0	0	0
F12	Average	2.25E-02	2.87E-02	7.44E-07	3.71E-01	1.86E-01	**1.62E-11**
	Std	1.12E-02	1.08E-02	2.25E-07	1.08E-01	9.58E-02	**2.87E-11**
F13	Average	3.23E-01	3.71E-01	1.61E-05	3.00E+00	1.92E+00	**7.35E-11**
	Std	1.82E-01	1.49E-01	5.02E-06	0	2.35E-01	1.26E-10
F14	Average	2.57E+00	4.65E+00	9.98E-01	7.05E+00	3.98E+00	9.98E-01
	Std	2.64E+00	4.38E+00	**0**	3.30E+00	3.29E+00	1.01E-13
F15	Average	4.32E-03	1.04E-03	**3.07E-04**	9.07E-02	2.05E-03	4.23E-04
	Std	8.02E-03	3.60E-03	**5.36E-10**	4.48E-02	5.90E-03	6.17E-05
F16	Average	-1.03E+00	-1.03E+00	-1.03E+00	-9.25E+00	-1.03E+00	-1.03E+00
	Std	1.11E-09	1.64E-08	6.66E-16	6.57E-02	4.80E-08	5.20E-05
F17	Average	3.98E-01	3.98E-01	3.98E-01	2.00E+00	3.98E-01	4.14E-01
	Std	7.12E-08	3.94E-07	**0**	1.32E+00	2.97E-06	2.00E-02
F18	Average	3.00E+00	3.00E+00	3.00E+00	6.92E+00	3.00E+00	3.01E+00
	Std	3.37E-06	4.68E-06	**9.49E-16**	3.92E+00	7.40E-08	1.55E-02
F19	Average	-3.00E-01	-3.00E-01	-3.00E-01	-3.00E-01	-3.00E+00	-3.00E-01
	Std	2.22E-16	2.22E-16	2.22E-16	2.56E-16	2.22E-16	0
F20	Average	-3.23E+00	-3.25E+00	-3.30E+00	-1.78E+00	-3.24E+00	-2.87E+00
	Std	7.51E-02	7.62E-02	**4.13E-02**	5.51E-01	5.80E-02	2.23E-01
F21	Average	-9.23+00	-9.47E+00	-1.01E+01	-5.34E-01	-8.89E+00	**-1.02E+01**
	Std	1.49E+00	1.73E+00	1.67E-01	1.83E-01	2.65E+00	1.93E-07
F22	Average	-1.02E+01	-1.02E+01	-1.04E+01	-8.17E-01	-1.04E+00	**-1.04E+01**
	Std	1.67E-04	9.54E-01	**4.23E-09**	4.10E-01	2.02E-03	2.30E-07
F23	Average	-1.05E+01	-1.03E+01	-1.05E+01	-8.25E-01	-9.45E+00	**-1.05E+01**
	Std	1.89E-04	9.62E-01	**2.84E-12**	2.52E-01	2.94E+00	5.84E-08

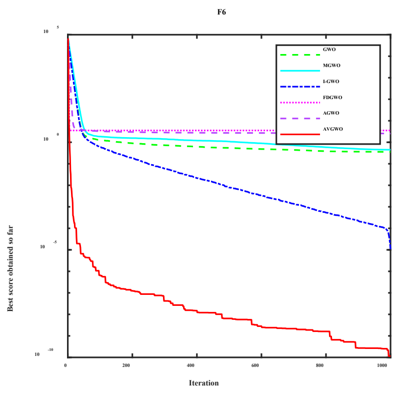

F3
GWO
MGWO
I-GWO
FDGWO
AGWO
AVGWO
Best score obtained so far
Iteration

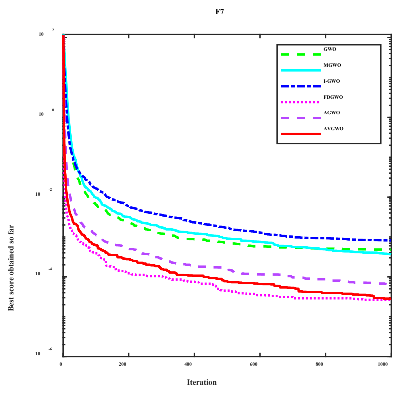

F2
GWO
MGWO
I-GWO
FDGWO
AGWO
AVGWO
Best score obtained so far
Iteration

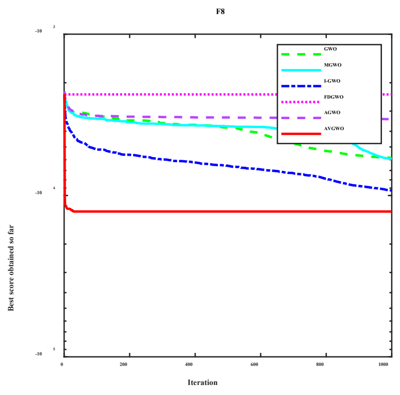

F3
GWO
MGWO
I-GWO
FDGWO
AGWO
AVGWO
Best score obtained so far
Iteration

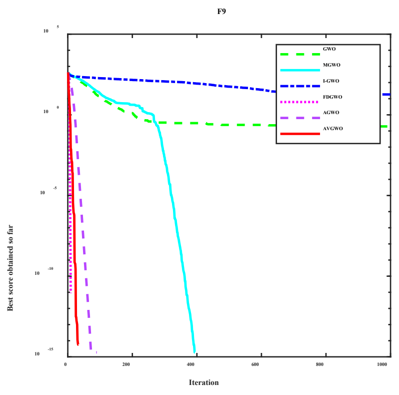

F4
GWO
MGWO
I-GWO
FDGWO
AGWO
AVGWO
Best score obtained so far
Iteration

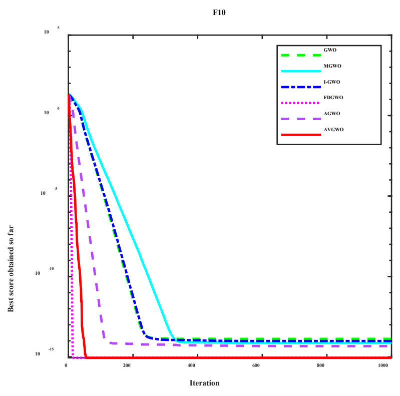

F5
GWO
MGWO
I-GWO
FDGWO
AGWO
AVGWO
Best score obtained so far
Iteration

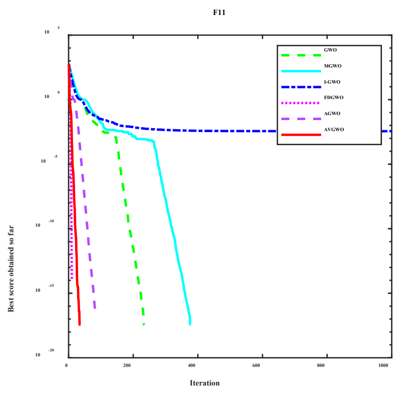

F6
GWO
MGWO
I-GWO
FDGWO
AGWO
AVGWO
Best score obtained so far
Iteration

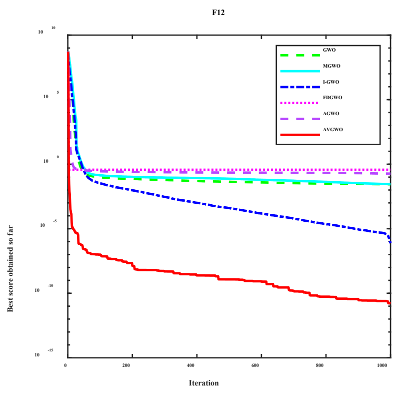

F7
GWO
MGWO
I-GWO
FDGWO
AGWO
AVGWO
Best score obtained so far
Iteration


F8
GWO
MGWO
I-GWO
FDGWO
AGWO
AVGWO
Best score obtained so far
Iteration


F9
GWO
MGWO
I-GWO
FDGWO
AGWO
AVGWO
Best score obtained so far
Iteration


F10
GWO
MGWO
I-GWO
FDGWO
AGWO
AVGWO
Best score obtained so far
Iteration


F11
GWO
MGWO
I-GWO
FDGWO
AGWO
AVGWO
Best score obtained so far
Iteration


F12
GWO
MGWO
I-GWO
FDGWO
AGWO
AVGWO
Best score obtained so far
Iteration

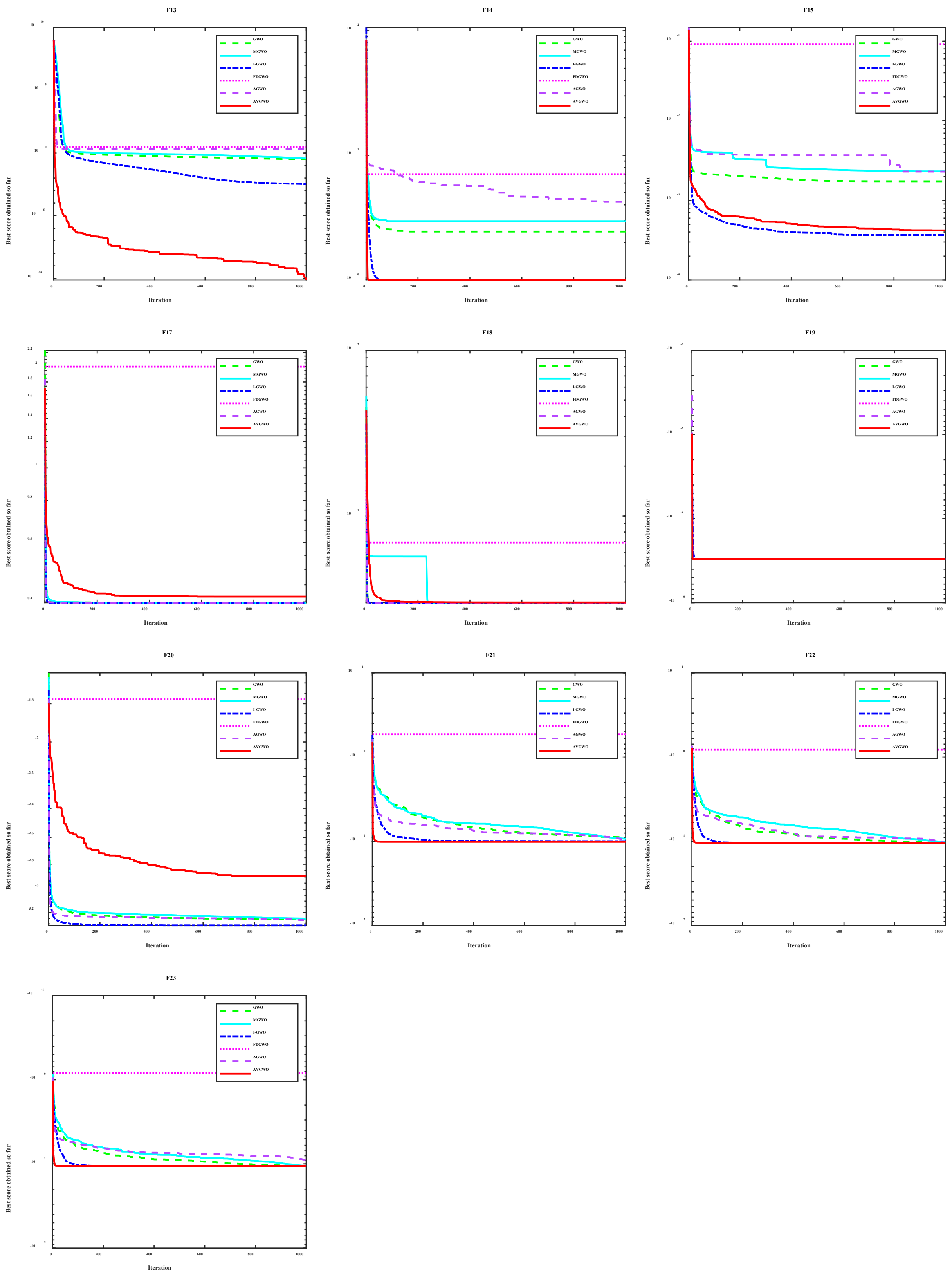


Fig. 2. Convergence curve

4.2 Comparison of AVGWO with Other Meta-heuristic Algorithms

To evaluate the suggested AVGWO algorithm against alternative metaheuristic algorithms, such as sine cosine algorithm [27], salp swarm algorithm [28], simulated annealing algorithm [29], whale algorithm [30], and Harris Hawk algorithm [31]. The experiments were conducted independently for 30 times. The outcomes of the experiment are displayed in Table 3. It can be noticed that AVGWO found the optimal solutions for F1-F4, F9, and F11 functions, indicated in bold font, which represents excellent performance among all testing algorithms.

Table 3. The results of AVGWO with metaheuristic algorithms on 23 benchmark problems

Function		SCA	SSA	SA	WOA	HHO	AVGWO
F1	Average	3.38E-03	9.08E-09	3.27E-12	1.01E-164	1.83E-193	0
	Std	1.49E-02	1.64E-09	9.87E-12	**0**	**0**	**0**
F2	Average	7.09E-06	3.96E-01	1.48E-07	2.48E-110	1.38E-99	0
	Std	1.68E-05	7.77E-01	1.77E-07	1.14E-109	7.39E-99	**0**
F3	Average	2.57E+03	5.16E+01	2.50E+03	9.88E+03	2.26E-165	0
	Std	4.13E+03	4.18E+01	7.87E+02	5.41E+03	**0**	**0**
F4	Average	1.50E+01	3.92E+00	4.44E-01	2.71E+01	7.28E-95	0
	Std	1.10E+01	2.28E+00	1.11E-01	2.75E+01	3.88E-94	**0**
F5	Average	2.60E+02	7.99E+01	7.65E+01	2.65E+01	8.01E-04	**4.06E-08**
	Std	8.75E+02	1.07E+02	7.72E+01	2.85E-01	1.08E-03	**2.71E-08**
F6	Average	4.39E+00	9.05E-09	**4.63E-12**	4.10E-03	1.53E-05	1.15E-10
	Std	6.33E-01	1.90E-09	**1.49E-11**	1.87E-03	1.88E-05	2.94E-10
F7	Average	2.96E-02	5.39E-02	7.19E-02	9.90E-04	4.63E-05	**4.43E-05**
	Std	3.03E-02	1.70E-02	2.31E-02	9.95E-04	3.90E-05	**2.55E-05**
F8	Average	-4.02E+03	-7.65E+03	-1.25E+04	-1.20E+04	-1.26E+04	**-1.26E+04**
	Std	2.85E+02	7.87E+02	1.03E+02	7.66E+02	2.12E-01	**5.83E-06**
F9	Average	1.97E+01	5.05E+01	5.01E-01	0	0	0
	Std	3.12E+01	1.81E+01	7.15E-01	**0**	**0**	**0**
F10	Average	1.12E+01	1.54E+00	9.41E-07	3.26E-15	**8.88E-16**	**8.88E-16**
	Std	9.51E+00	8.53E-01	1.48E-06	2.12E-15	**0**	**0**
F11	Average	1.51E-01	1.23E-02	2.95E-02	7.61E-04	**0**	**0**
	Std	2.07E-01	1.17E-02	2.43E-02	4.10E-03	**0**	**0**
F12	Average	1.88E+00	3.61E+00	**1.02E-14**	2.89E-03	5.93E-07	1.62E-11
	Std	4.00E+00	2.29E+00	**2.26E-14**	9.04E-03	6.47E-07	2.87E-11
F13	Average	4.08E+00	6.93E-03	**1.43E-12**	4.28E-02	1.13E-05	7.35E-11
	Std	6.15E+00	9.13E-03	**7.32E-12**	4.81E-02	1.78E-05	1.26E-10
F14	Average	1.26E+00	9.98E-01	9.98E-01	1.46E+00	9.98E-01	**9.98E-01**
	Std	6.74E-01	**2.55E-16**	8.39E-12	1.80E+00	7.78E-11	1.01E-13
F15	Average	7.69E-04	1.47E-03	5.46E-03	5.97E-04	4.53E-04	**4.23E-04**
	Std	3.33E-04	3.52E-03	8.15E-03	3.36E-04	2.71E-04	**6.17E-05**
F16	Average	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00
	Std	1.59E-05	**1.04E-14**	1.43E-11	1.20E-11	1.11E-12	5.20E-05
F17	Average	3.98E-01	3.98E-01	3.98E-01	3.98E-01	3.98E-01	4.14E-01
	Std	5.30E-04	**2.00E-15**	8.07E-12	1.53E-07	3.85E-09	2.00E-02
F18	Average	3.00E+00	3.00E+00	9.30E+00	3.00E+00	3.00E+00	3.01E+00
	Std	1.02E-05	**7.12E-14**	2.05E+01	2.38E-06	7.91E-10	1.55E-02
F19	Average	-3.00E-01	-3.00E-01	-3.00E-01	-3.00E-01	-3.00E-01	-3.00E-01
	Std	2.22E-16	2.22E-16	2.22E-16	2.22E-16	2.22E-16	**0**
F20	Average	-2.95E+00	-3.23E+00	-3.27E+00	-3.26E+00	-3.17E+00	-2.87E+00
	Std	3.57E-01	**5.29E-02**	5.93E-02	7.58E-02	9.17E-02	2.23E-01
F21	Average	-3.18E+00	-9.06E+00	-4.88E+00	-9.81E+00	-5.56E+00	**-1.02E+01**
	Std	1.77E+00	2.23E+00	2.60E+00	1.27E+00	1.53E+00	**1.93E-07**
F22	Average	-4.76E+00	-8.86E+00	-4.74E+00	-9.57E+00	-5.61E+00	**-1.04E+01**
	Std	1.89E+00	2.86E+00	2.77E+00	2.15E+00	1.57E+00	**2.30E-07**
F23	Average	-4.64E+00	-8.70E+00	-4.11E+00	-9.00E+00	-5.49E+00	**-1.05E+01**
	Std	1.80E+00	3.10E+00	3.02E+00	2.85E+00	1.35E+00	5.84E-08

5 Channel Estimation Model

Millimeter-wave (mmWave) communication is a promising technology for achieving high data rates due to its wide bandwidth. However, the high propagation loss of mmWave signals presents a significant challenge, which can be overcome using MIMO techniques. In mmWave mass MIMO systems, providing each antenna with a specialized chain of radio frequency (RF) is difficult due to limited physical space and high power consumption. To address this issue, two-stage structures based on phase shifters, referred to as hybrid structures, are widely employed in transmitters and receivers to link a lot of antennas with less RF chains. This paper proposes using deep convolutional neural networks (CNNs) for channel estimation in mmWave large-scale MIMO-OFDM systems. To make use of relevance between channel of subcarrier adjacent to each other in OFDM, this article describes a method for channel estimation based on spatial frequency CNN (SF-CNN). Input coarse estimated channel matrix at neighboring subcarriers to CNN simultaneously to improve better accuracy and efficiency [32].

In a millimeter-wave large-scale MIMO-OFDM system, as in Fig. 3, there are N_{T} antennas and $N_{\mathrm{T}}^{\mathrm{RF}}$ RF chains at the transmitter side and N_{R} antennas and $N_{\mathrm{R}}^{\mathrm{RF}}$ RF chains at the receiver side. A phase shifter is used to connect the antennas, in this paper $N_{\mathrm{T}}= 32$, $N_{\mathrm{R}} =16$, $N_{\mathrm{T}}^{\mathrm{RF}} = N_{\mathrm{R}}^{\mathrm{RF}} = 2$.

$$\mathbf{H}(\tau) = \sqrt{\frac{N_T N_R}{L}} \sum_{l=1}^{L} \alpha_l \delta\left(\tau - \tau_l\right) a_R\left(\varphi_l\right) a_T^H\left(\phi_l\right) . \tag{17}$$

Where $\alpha_l \sim \mathcal{CN}\left(0,\ \sigma_\alpha^2\right)$ is the spread gain of the first path, the mean power gain is σ_α^2, L is the count of paths, the latency of the l th path is indicated by τ_l, while φ_l and $\phi_l \in [0, 2]$ signifies the azimuths of arrival and departure (AoA/AoD) of the receiver and transmitter, respectively. In the case of a uniform linear array (ULA), the corresponding response vector can be denoted as:

$$\begin{cases} \mathbf{a}_{\mathrm{R}}\left(\varphi_l\right) = \dfrac{1}{\sqrt{N_{\mathrm{R}}}}\left[1, e^{-j2\pi\frac{d}{\lambda}\sin\left(\varphi_l\right)}, \ldots,\ e^{-j2\pi\frac{d}{\lambda}\left(N_{\mathrm{R}}-1\right)\sin\left(\varphi_l\right)}\right]^T \\ \mathbf{a}_{\mathrm{T}}\left(\phi_l\right) = \dfrac{1}{\sqrt{N_{\mathrm{T}}}}\left[1, e^{-j2\pi\frac{d}{\lambda}\sin\left(\phi_l\right)}, \ldots,\ e^{-j2\pi\frac{d}{\lambda}\left(N_{\mathrm{T}}-1\right)\sin\left(\phi_l\right)}\right]^T \end{cases} . \tag{18}$$

Where d and λ denote the separation between neighboring antennas and the wavelength of the carrier signal, respectively. Based on its channel model [33], the channel in the k th subcarrier's frequency region in OFDM is given by.

$$H_k = \sqrt{\frac{N_T N_R}{L}} \sum_{l=1}^{L} \alpha_1 e^{-j2\pi\tau_1 f_s \frac{k}{K}} a_R\left(\varphi_l\right) a_T^H\left(\phi_l\right) . \tag{19}$$

Where f_s denotes the sampling rate and K is the number of OFDM subcarriers.

To obtain $\mathbf{H}_k$ the pilot signal $x_{k,\ u}$, is delivered by way of beamforming vectors $\mathbf{f}_{k,\ u} \in \mathbb{C}^{N_{\mathrm{T}}\times 1}, u = 1,\ldots\ M_{\mathrm{T}}$ period M_{T} continuous time slots. Receiver adopts M_{R} combination vector, $\mathbf{w}_{k,\ v} \in \mathbb{C}^{N_{\mathrm{R}}\times 1},\ v = 1,\ldots\ M_{\mathrm{R}}$ to deal with each of the beamforming vectors. Then, the frequency guide signal matrix related to the k th subcarrier on the receiver's baseband can be obtained.

$$\mathbf{Y}_k = \mathbf{W}_k^H \mathbf{H}_k \mathbf{F}_k \mathbf{X}_k + \tilde{\mathbf{N}}_k . \tag{20}$$

Where $\mathbf{W}_k = \left[\mathbf{w}_{k,1}, \ldots, \mathbf{w}_k, M_{\mathrm{R}}\right]$ are the combination matrix, $\mathbf{F}_k = \left[\mathbf{f}_{k,1}, \ldots, \mathbf{f}_k, M_{\mathrm{T}}\right]$ are the beamforming matrix. $\mathbf{X}_k$ is an $M_{\mathrm{T}} \times M_{\mathrm{T}}$ diagonal matrix with its μ th elements of diagonal being $\mathrm{x}_{k,\mu}$ and $\tilde{\mathbf{N}}_k = \mathbf{W}_k^H \mathbf{N}_k$ represents user-combined noise, while $\mathbf{N}_k$ is additive white Gaussian noise with $\mathcal{CN}$ (0, 1) elements before combining.

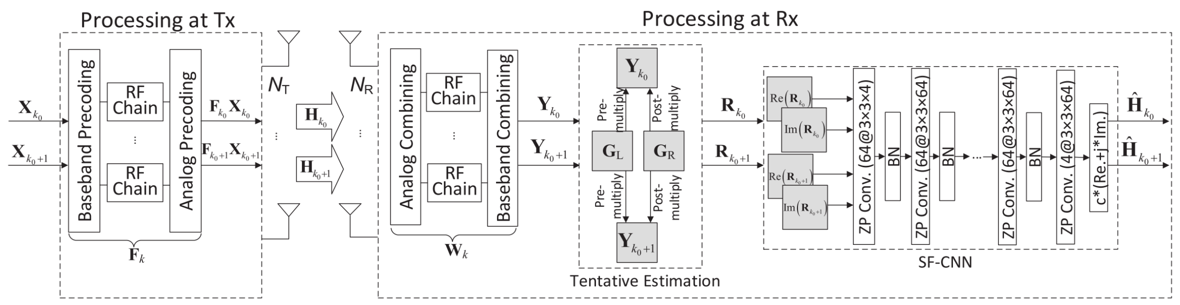


Fig. 3. SF-CNN channel estimation

6 Convolutional Neural Network-based Channel Estimation Method

6.1 Algorithm Description

Channel Pre-processing. To continue with generality while assuming the worst-case scenario that $\mathbf{F}_k = \mathbf{F}$, $\mathbf{W}_k = \mathbf{W}$, and $\mathbf{X}_k = \sqrt{P}\mathbf{I}$, where P denotes transmitted power. The conduction frequency signal's matrix, $\mathbf{Y}_k$, becomes:

$$\mathbf{Y}_k = \sqrt{P}\mathbf{W}^H\mathbf{H}_k\mathbf{F} + \tilde{\mathbf{N}}_k \ . \tag{21}$$

Fig. 3 illustrates the $\mathbf{Y}_k$ through estimation (TE) module, which processes $\mathbf{Y}_k$ and outputs a preliminary estimate $\mathbf{H}_k$ using two matrices $\mathbf{G}_\mathrm{L}$, $\mathbf{G}_\mathrm{R}$.

$$\mathbf{R}_k = \mathbf{G}_\mathrm{L}\mathbf{Y}_k\mathbf{G}_\mathrm{R} \ . \tag{22}$$

$$\begin{cases} \mathbf{G}_\mathrm{L} = \left(\mathbf{W}\mathbf{W}^H\right)^{-1}\mathbf{W} \\ \mathbf{G}_\mathrm{R} = \mathbf{F}^H\left(\mathbf{F}\mathbf{F}^H\right)^{-1} \end{cases} . \tag{23}$$

Then, the initially estimated channel matrices $\mathbf{R}_{\mathrm{k}0}$ and $\mathbf{R}_{\mathrm{k}0+1}$ are concurrently fed into the SF-CNN.

SF-CNN Offline Training. To the suggested SF-CNN, a training database made up of N_{tr} samples is created in the simulation environment based on a particular channel model, where ($\mathbf{R}_i$, $\mathbf{H}_i$) stands for the *i*th sample, $\mathbf{R}_i$ stands for input data, $\mathbf{H}_i$ stands for target data. $\mathbf{R}^i_{k_0'}$, $\mathbf{R}^i_{k_0'+1} \in \mathbb{C}^{N_\mathrm{R}\times N_\mathrm{T}}$ are the tentatively estimated channel matrices at subcarriers k_0' and $k_0'+1$, corresponds to the three-dimensional matrix which makes up $\mathbf{R}_i \in \mathbb{C}^{N_\mathrm{R}\times N_\mathrm{T}\times 2}$.

$\underline{\mathbf{H}}_i \in \mathbb{C}^{N_R \times N_T \times 2}$ is a 3D matrix consisting of $\frac{\mathbf{H}^i_{k_0'}}{c}$, $\frac{\mathbf{H}^i_{k_0'+1}}{c} \in \mathbb{C}^{N_R \times N_T}$, $\mathbf{H}^i_{k_0'}$, $\mathbf{H}^i_{k_0'+1}$ are the value of the associated real channel estimate, and $c > 0$ is the resizing constant. Then $\underline{\mathbf{R}}_i$ is delivered the SF-CNN to obtain the channel that approaches $\underline{\mathbf{H}}_i$.

In order to create four 16×32 real-valued matrices for the millimeter-wave large-scale MIMO system, the SF-CNN first estimates the complex channel matrices $\mathbf{R}^i_{k_0'} \in \mathbb{C}^{16\times 32}$ and $\mathbf{R}^i_{k_0'+1} \in \mathbb{C}^{16\times 32}$ as the importation and splitting their real and virtual components. These four matrices are processed in the convolutional layer using 64 3×3×4 convolutional filters and the activation function RELU to create 64 16×32 real-valued matrices. Each feature matrix is processed with zero-padding (ZP) to maintain its dimensionality after convolution. The subsequent addition of a batch normalized (BN) layer prevents gradient spread and overfitting. The feature matrix transmitted from the preceding layer is used in each of the next eight convolution layers, which each employ 64 3×3×64 convolutional filters to produce 64 16×32 real-valued feature matrices. These eight layers receive the activation function ReLU, and a BN layer comes after each layer. The 64 16×32 real-valued feature matrices are processed by four 3×3 ×64 convolutional filters on the output layer to produce the real and imaginary components of the k_0', $k_0'+1$ scaled channel matrices. The output layer maps the output to the range [-1, 1] using hyperbolic tangent activation functions. The appropriate real and imaginary sections are scaled, and then the resulting 16×32 complex estimated channel matrices are created.

The goal of SF-CNN off-line training is used to reduce the MSE loss function.

$$\mathrm{MSE}_{\mathrm{Loss}} = \frac{1}{N_{\mathrm{tr}} c^2} \sum_{i=1}^{N_{\mathrm{tr}}} \sum_{q=1}^{2} \| \mathbf{H}^i_{k_0'+q-1} - \hat{\mathbf{H}}^i_{k_0'+q-1} \|_F^2 . \quad \textbf{(24)}$$

Online Deployment. After off-line training, the SF-CNN and the TE module will be positioned at the receiver to output the estimated channel matrix $\mathbf{Y}_{k_0}, \mathbf{Y}_{k_0+1}, \ldots, \mathbf{Y}_{k_0+Q-1}$. By processing the guide frequency matrix, in order to output the estimated channel matrix $\hat{\mathbf{H}}_{k_0}, \mathbf{H}_{k_0+1}, \ldots, \mathbf{H}_{k_0+Q-1}$.

6.2 Optimized Convolutional Neural Network Structure Based on AVGWO.

Structure of Convolutional Neural Networks. The typical framework of a CNN composed of convolutional layers, pooling layers, and fully connected layers. The convolutional layer is employed to extract local features, the pooling layer is utilized to minimize the dimensionality of the feature maps, and the fully connected layers is employed for tasks such as classification or regression. In this paper, we take CNN: convolutional layer (C1) - batch normalization layer-convolutional layers (C2) …convolutional layers (C10) as an example. The hyperparameters of the CNN, as presented in Table 4, have a straightforward effect on the network's performance. In this paper, the activation function type for layer C1, the activation function type and number of convolution filters for C10 have been determined. The selection of these hyperparameters determines the structure of the CNN, which ultimately affects its ability to perform. Therefore, the choices made regarding these hyperparameters play a fundamental role in shaping the CNN's architecture and performance.

Table 4. Hyperparameters in CNN

Symbols	Hyperparameters
$x_1 - x_9$	Number of convolution filters in C1-C9
$x_{10} - x_{19}$	Convolutional kernel size in C1-C10
$x_{20} - x_{27}$	Types of activation functions in C2-C9
x_{28}	Batch size for training
x_{29}	Learning rate of CNN

Algorithmic Ideas. The structure of CNN is very complex and its performance depends heavily on many hyperparameters. the hyperparameters of CNN determine its structure, and the optimization of CNN structure is essentially the optimization of hyperparameters in CNN, and in turn, the optimization of hyperparameters can achieve the purpose of optimizing CNN. In this study, the primary aim is to reduce the minimum mean square error MSE of CNN, which can be expressed as a function of the variation of CNN hyperparameters.

Algorithm Steps and Flow Chart. AVGWOCNN algorithm specific steps

Step1: Basic parameters setting, including population scale of AVGWO, maximum number of iterations and other parameters of AVGWO;

Step2: The hyperparameters that CNN needs to optimize are the grey wolf individuals of the AVGWO algorithm, initialize the grey wolf individual;

Step3: Use each grey wolf as a CNN structure, determine the CNN through training, use the test error of the CNN as the adaptation values for grey wolves, and determine the á wolves.

Step4: Upon attaining the maximum number of iterations, stop iteration and the position of á wolf is used as the optimal hyperparameter of the CNN, otherwise, continue to update the grey wolf position and return to Step3;

The flow graph of the AVGWOSF-CNN algorithm is shown in Fig. 4.

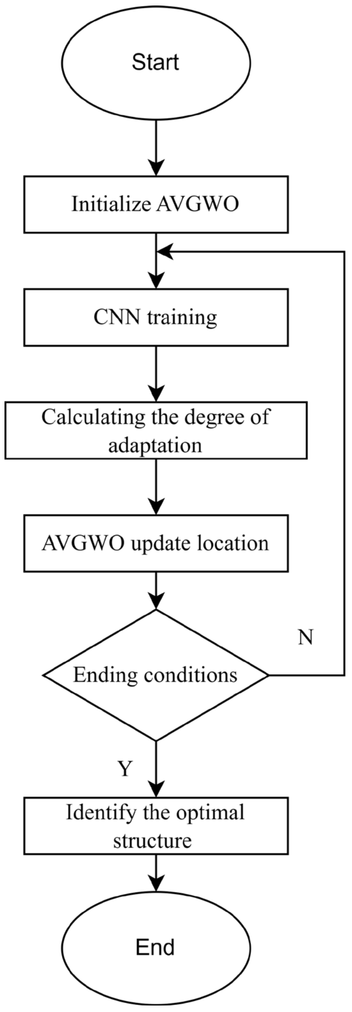


Fig. 4. Flow graph of AVGWOSF-CNN

7 Simulation Results

The channel data is generated based on the Clustered Latency Line model with frequency of carrier wave f_c =28 GHz, percentage of sampling f_s =100 MHz, main path number $L = 3$, and subcarrier number $K = 64$, according to the Third Generation Partnership Project (3GPP) TR 38.901 version 15 channel model [34].

For SF-CNN, the training set, validation set and test set incorporates 50,000, 5,000 and 8,000 samples, Adam is utilized as the optimizer, the number of training times is defined as 300, the learning rate is defined as 0.01 and the scaling constant c is defined as 2. In this paper, suggest a technique that combines modified grey wolf optimizer with CNN to optimize the architecture of CNN without the need for manual parameter tuning. By automating the procedure for searching optimal parameters, aim to reduce the amount of human trial and error involved in CNN optimization, and improve the performance of CNN. Specifically, Use the modified Grey Wolf optimizer to determine which is finest values of the hyperparameters in CNN, such as the number of filters, filter size.

To test the channel estimation capabilities, the normalized MSE (NMSE) is used, which is stated as:

$$\text{NMSE} = \mathbb{E}_{\mathbf{H}}\left\{ \| \mathbf{H} - \hat{\mathbf{H}} \|_F^2 / \| \mathbf{H} \|_F^2 \right\} . \tag{25}$$

$\mathbf{H}$ is the genuine channel, $\hat{\mathbf{H}}$ is the predicted channel.

In Fig. 5, a comparison is made between single-carrier and multi-carrier channel estimation methods based on SF-CNN, which verifies that frequency correlation can improve channel accuracy.

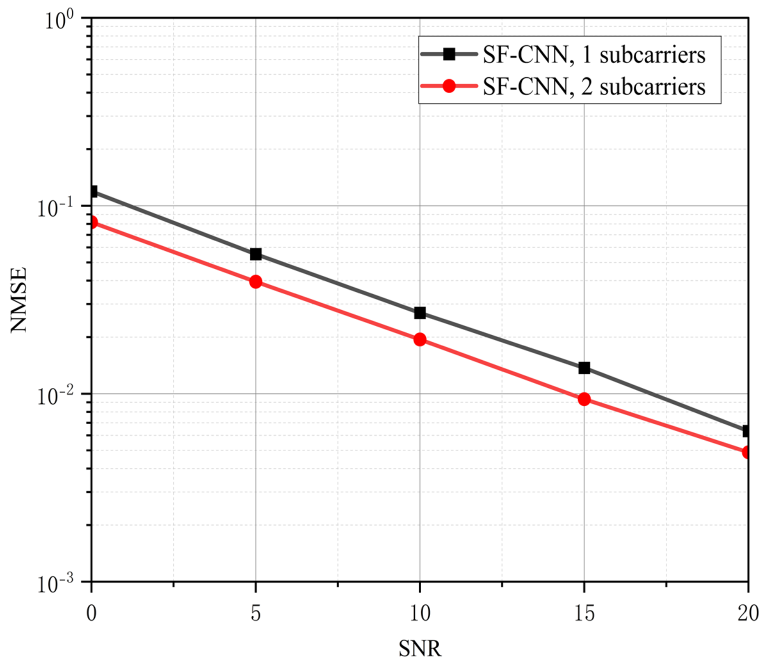


Fig. 5. Comparison of single-carrier and two-carrier channel estimation methods based on SF-CNN

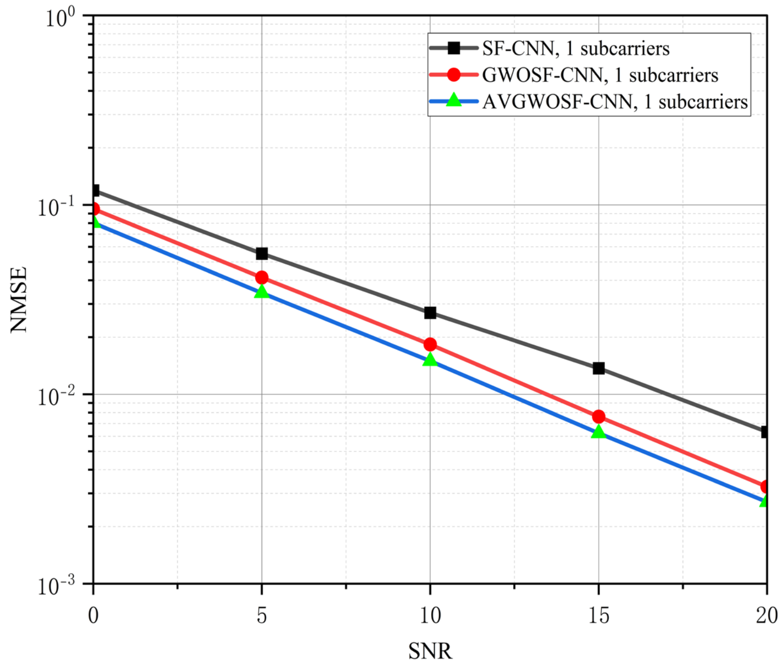


Fig. 6. NMSE and SNR for channel estimation based on SF-CNN, GWOSF-CNN and AVGWOSF-CNN

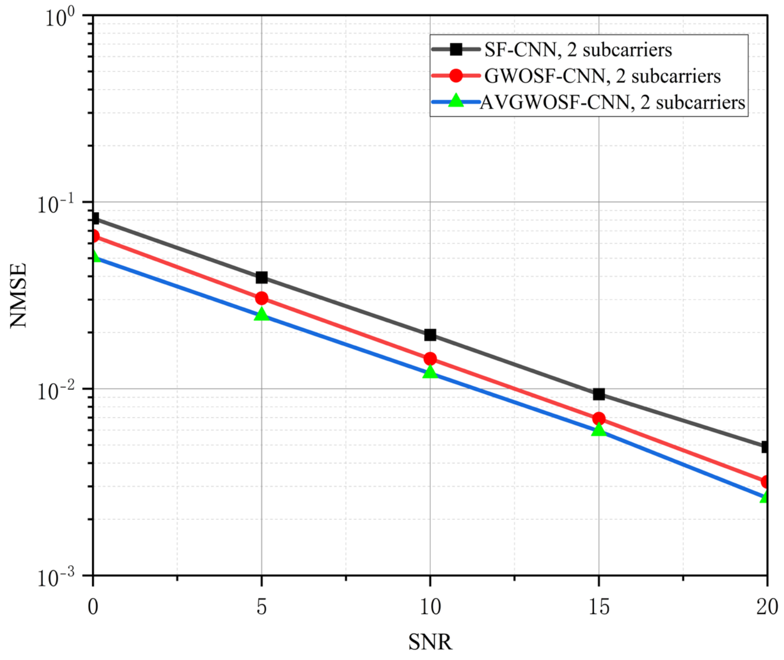


Fig. 7. NMSE and SNR for channel estimation based on SF-CNN, GWOSF-CNN and AVGWOSF-CNN

Fig. 6 and Fig. 7 shows the relationship between signal-to-noise ratio (SNR) and the channel estimation performance of the proposed SF-CNN-based channel estimation and the grey wolf algorithm optimized CNN-based channel estimation on a single subcarrier in the urban micro (UMi) non-line-of-sight (NLOS) scenario. Through offline training, the results show that under the same signal-to-noise ratio, the traditional grey wolf optimized CNN algorithm is superior to the SF-CNN algorithm in channel estimation, and using the AVGWO algorithm to optimize the CNN further improves the performance. Using the improved grey wolf algorithm to optimize the CNN not only eliminates the need for trial-and-error hyperparameter tuning, but also enhances the learning ability of the CNN, thus improving the accuracy of channel prediction.

8 Conclusion

In this paper, we propose a new improved gray wolf algorithm, AVGWO, which combines three different strategies, the search mechanism based on the African vulture algorithm, the search mechanism based on the bootstrap strategy, and the adaptive weights. The global search capability and local search capability of the algorithm are balanced while maintaining a high convergence speed. The performance of AVGWO is tested in 23 benchmark test functions and compared with the GWO algorithm and other improvements and other metaheuristics. Finally, in the massive MIMO-OFDM system, AVGWO and GWO are used to optimize SF-CNN for the proposed channel estimation based on convolutional neural network, the results show that both GWO and AVGWO improve the learning ability of CNN, AVGWO further improves the accuracy of channel estimation.

References

[1] S. Katoch, S.-S. Chauhan, V. Kumar, A review on genetic algorithm: past, present, and future, Multimedia Tools and Applications 80(5)(2021) 8091–8126.

[2] G.-H. Wu, X. Shen, H.-F. Li, H.-K. Chen, A.-P. Lin, P.-N. Sugathan, Ensemble of differential evolution variants, Information Sciences, 423(2018) 172–186.

[3] J. Kennedy, R. Eberhart, Particle swarm optimization, Proceedings of ICNN'95 - International Conference on Neural Networks, 1995.

[4] S.-C. Gao, Y.-R. Wang, J.-J. Cheng, Y. Inazumi, Z. Tang, Ant colony optimization with clustering for solving the dynamic location routing problem, Applied Mathematics and Computation 285(2016) 149–173.

[5] D. Karaboga, B. Gorkemli, C. Ozturk, N. Karaboga, A comprehensive survey: artificial bee colony (ABC) algorithm and applications, Artificial Intelligence Review 42(1)(2014) 21–57.

[6] M. Neshat, G. Sepidnam, M. Sargolzaei, A.-N. Toosi, Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinatorial and indicative applications, Artificial Intelligence Review 42(4)(2014) 965–997.

[7] S. Mirjalili, S.-M. Mirjalili, A. Lewis, Grey Wolf Optimizer, Advances in Engineering Software 69(2014) 46–61.

[8] R.-V. Rao, V.-J. Savsani, D.-P. Vakharia, Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems, Computer-Aided Design 43(3)(2011) 303–315.

[9] Q. Askari, I. Younas, M. Saeed, Political Optimizer: A novel socio-inspired meta-heuristic for global optimization, Knowledge-Based Systems 195(2020) 105709.

[10] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: A Gravitational Search Algorithm, Information Sciences 179(13)(2009) 2232–2248.

[11] A. Kaveh, S. Talatahari, A novel heuristic optimization method: charged system search, Acta Mechanica 213(3-4)(2010) 267–289.

[12] W.-G. Zhao, L.-Y. Wang, Z.-X. Zhang, Atom search optimization and its application to solve a hydrogeologic parameter estimation problem, Knowledge-Based Systems 163(2019) 283–304.

[13] J. Kawal, Fuzzy flow shop scheduling using grey wolf optimization algorithm, Indian Journal of Scientific Research, 7(2)(2017) 167-171.

[14] C. Lu, L. Gao, Q.-K. Pan, X.-Y. Lin, J. Zheng, A multi-objective cellular grey wolf optimizer for hybrid flowshop scheduling problem considering noise pollution, Applied Soft Computing 75(2019) 728-749.

[15] Y. Luo, Q. Qin, Z.-F. Hu, Y. Zhang, Path Planning for Unmanned Delivery Robots Based on EWB-GWO Algorithm, Sensors 23(4)(2023) 1867.

[16] J.-Q. Shi, L. Tan, H.-T. Zhang, X.-F. Lian, T.-Y. Xu, Adaptive multi-UAV path planning method based on improved grey wolf algorithm, Computers and Electrical Engineering 104(2022) 108377.

[17] S.-K. Ladi, G.-K. Panda, R. Dash, P.-K. Ladi, R. Dhupar, Correction to: A novel grey wolf optimization based CNN classifier for hyperspectral image classification, Multimedia Tools and Applications 82(18)(2023) 28669.

[18] A. Saxena, R. Kumar, S. Mirjalili, A harmonic estimator design with evolutionary operators equipped grey wolf opti-

mizer, Expert Systems with Applications 145(2020) 113125.
[19] X.-X. Gao, J. Shi, C.-K. Wen, G. Ye Li, ComNet: Combination of Deep Learning and Expert Knowledge in OFDM Receivers, IEEE Communications Letters 22(12)(2018) 2627–2630.
[20] Y. Liao, Y.-X. Hua, X.-W. Dai, H.-W. Yao, X.-Y. Yang, ChanEstNet: A Deep Learning Based Channel Estimation for High-Speed Scenarios, IEEE International Conference on Communications, 2019.
[21] Y.-D. Dong, H.-X. Wang, Y.-D. Yao, Channel Estimation for One-Bit Multiuser Massive MIMO Using Conditional GAN, IEEE Communications Letters 25(3)(2021) 854–858.
[22] B. Abdollahzadeh, F.-S. Gharehchopogh, S. Mirjalili, African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems, Computers & Industrial Engineering 158(2021) 107408.
[23] S. Padhy, S. Panda, S. Mahapatra, A modified GWO technique based cascade PI-PD controller for AGC of power systems in presence of Plug in Electric Vehicles, Engineering Science and Technology, an International Journal 20(2)(2017) 427–442.
[24] M.-H. Nadimi-Shahraki, S. Taghian, S. Mirjalili, An improved grey wolf optimizer for solving engineering problems, Expert Systems with Applications 166(2021) 113917.
[25] H.-N. Soloklo, N. Bigdeli, Fast-Dynamic Grey Wolf Optimizer for solving model order reduction of bilinear systems based on multi-moment matching technique, Applied Soft Computing 130(2022) 109730.
[26] C. Ma, H.-S. Huang, Q.-S. Fan, J.-A. Wei, Y.-M. Du, W.-S. Gao, Grey wolf optimizer based on Aquila exploration method, Expert Systems with Applications 205(2022) 117629.
[27] S. Mirjalili, SCA: A Sine Cosine Algorithm for solving optimization problems, Knowledge-Based Systems 96(2016) 120–133.
[28] S. Mirjalili, A.-H. Gandomi, S.-Z. Mirjalili, S. Saremi, H. Faris, S.-M. Mirjalili, Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems, Advances in Engineering Software 114(2017) 163–191.
[29] K.-M. El-Naggar, M.-R. AlRashidi, M.-F. AlHajri, A.-K. AI-Othman, Simulated Annealing algorithm for photovoltaic parameters identification, Solar Energy 86(1)(2012) 266–274.
[30] S. Mirjalili, A. Lewis, The Whale Optimization Algorithm, Advances in Engineering Software 95(2016) 51–67.
[31] A.-A. Heidari, S. Mirjalili, H. Faris, l. Aljarah, M. Mafarja, H.-L. Chen, Harris hawks optimization: Algorithm and applications, Future Generation Computer Systems 97(2019) 849–872.
[32] P.-H. Dong, H. Zhang, G.Y. Li, N. NaderiAlizadeh, I.-S. Gaspar, Deep CNN for Wideband Mmwave Massive Mimo Channel Estimation Using Frequency Correlation, ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing, 2019.
[33] Z. Gao, C. Hu, L.-L. Dai, Z.-C. Wang, Channel Estimation for Millimeter-Wave Massive MIMO With Hybrid Precoding Over Frequency-Selective Fading Channels, IEEE Communications Letters 20(6)(2016) 1259–1262.
[34] M. Rumney, P. Kyösti, L. Hentilä, 3GPP channel model developments for 5G NR requirements and testing, 12th European Conference on Antennas and Propagation, 2018.