Proceedings of the 2nd International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/132/2024.20529

Optimization of Distributed UAV Swarm Placement for Target Localization Using Multiple Heuristic Algorithms Based on Compressive Sensing

Zeyu Du[1,6,†], **Luoyu Liu**[2,7,†], **Hangyu Zhou**[3,8,†], **Ziyi Chen**[4,9,†], **Yusen Yao**[5,10,*,†]

[1]School of Electronic and Information Engineering, Tongji University, Shanghai, China
[2]School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an, China
[3]College of Electrical and Electronic Information, XiHua University, Sichuan, China
[4]Department of Electronic Information Engineering, Shenzhen University, Shenzhen, China
[5]High School Affiliated to South China Normal University International Department, Guangzhou, China

[6]3022077861@qq.com
[7]certainyu@outllook.com
[8]daqingzpm668@qq.com
[9]1261012062@qq.com
[10]seeley0518@gmail.com
*corresponding author
†These authors contributed equally to this work and should be considered co-first authors.

Abstract. Collaborative UAV swarms are increasingly deployed as temporary base stations in emergency situations to relay communications. This paper designs intelligent optimization algorithms, which minimizing mutual coherence within a region of interest (ROI) aims to solving challenge in these scenarios is optimizing the UAVs' locations to avoid mutual signal interference and ensure high communication quality. Minimizing mutual coherence reduces the likelihood of signal interference between UAVs. We explore various optimize algorithum, including Heuristic Search (HS) and Ant Colony Optimization (ACO) and so on. The results provide insights into each algorithm's performance in dynamic environments, helping to identify the most suitable approaches for UAV deployment in emergency scenarios. This study contributes to the development of efficient UAV deployment strategies, enhancing the reliability of UAV-based communication systems during critical events.

Keywords: UAV swarms, Optimize algorithum, mutual coherence, MIMO radar system.

1. Introduction

UAVs are widely used in aerial photography, mapping, first aid and other scenarios. And UAV swarms are suitable for performing complex tasks, performing different functions separately and collaborating on operations. Relay is an important application scenario for UAV swarm collaboration. When the base station or other network communication equipment in a certain area is disconnected from the outside world due to a sudden disaster or other reasons, rescuers can send one or more groups of collaborative UAV swarms to travel to the area of communication disruption and establish a temporary base station in the target area.

To ensure compatibility among communication devices, all drones—both transmitting and receiving—must operate within the same frequency band. Consequently, mutual interference can occur when two UAVs are transmitting and receiving simultaneously. To mitigate this interference and maintain communication quality, UAVs must be rapidly deployed to optimal locations once the swarm reaches the target area. This deployment strategy not only minimizes inter-UAV interference but also ensures that every target point within the area can effectively receive and transmit radio signals via the temporary base stations established by the UAVs.

When the number of large UAVs is high, the UAV positions cannot be displayed. Based on this problem, a literature proposes to develop corresponding intelligent optimization algorithms and deploy them in UAV swarms to achieve automatic deployment and adjustment of UAV positioning, and tries to optimize the UAV positions using heuristic algorithms and gradient descent method respectively [1]. However, this paper argues that both the heuristic algorithm and the gradient descent method have their own drawbacks, and thus are not particularly suitable for relay communication applications in UAV swarms. In addition, this paper also argues that the deployment algorithm for UAV swarms must ensure that the endurance of the UAV swarms is as long as possible, since the UAV swarms cannot be continuously powered during operation and cannot receive solar energy by installing solar panels due to the limitation of the size of the UAVs. In this paper, we focus on the effect of the algorithm on the UAV range time by counting the number of operations of the algorithm. A smaller number of operations corresponds to a longer range time.

This research integrates two existing algorithms, the HS algorithm and the GD algorithm, to create a new hybrid algorithm that leverages the strengths of both. Additionally, the study explores and incorporates design principles from several other intelligent optimization algorithms to further enhance the relay communication capabilities and ultra-long endurance of collaborative drone swarms. The paper introduces a total of six algorithms, including the combined HS-GD algorithm, and evaluates their performance based on RSEM, the number of operations, and other key metrics. The algorithms examined are the HS-GD combination, ACO algorithm, DE algorithm, SA algorithm, GA algorithm, and PSO algorithm.

2. Literature Review

The introduction highlights the use of UAVs as airborne base stations for communication services and as data collection and dissemination nodes. One of the research priorities in the field of wireless communications is how to optimize the deployment of UAVs in three-dimensional space for various applications.

Early research focused on two-dimensional (2D) deployment of UAVs, with an emphasis on optimizing the horizontal position of the UAV at a fixed altitude to maximize coverage or minimize transmission power. However, this approach ignores the impact of altitude on communication performance.

As research progresses, 3D positioning becomes mainstream, combining altitude as a key optimization parameter with horizontal positioning. This allows a better balance between path loss and line-of-sight (LoS) connectivity. For example, literature investigates the trade-off between UAV flight altitude with path loss and LoS[2].

With the growing demand for networks, the deployment of collaborative UAV swarms has become a new research hotspot. Nowadays, researchers not only consider the optimal location of individual UAVs, but also the coordination among multiple UAVs to enhance the coverage and network capacity. Literature uses a greedy algorithm to determine the minimum number of UAVs and their optimal 3D locations [3], while literature introduces UAV clusters to optimize network performance through UAV cooperation [4].

Around 2017, the rise of artificial intelligence, especially reinforcement learning (RL), introduced new methods for optimizing UAV 3D positions. RL including Deep Reinforcement Learning (DRL) techniques such as Deep Q Networks (DQN) and Proximal Policy Optimization (PPO) have been applied to UAV positioning in NOMA networks [4]. In addition to RL, Evolutionary Algorithms (EAs) have been widely used for UAV localization using natural selection and genetic evolutionary processes. Algorithms such as Particle Swarm Optimization (PSO) and Genetic Algorithms (GA) have proven their effectiveness in 3D UAV localization, especially in terms of convergence speed and global search capability [5]. Despite significant progress, there are still some challenges and outstanding issues. Real-time UAV localization in complex, dynamic environments and the integration of multiple intelligent optimization algorithms, e.g., combining RL and EAs to leverage their respective strengths, are key areas for future research [5-7].

3. Signal and System Model

The signal model we use is linear frequency modulated continuous wave (LFMCW) chirp waveforms, because it is as one of the most frequently used modulation methods in UAVs, and it also has the ability to simplify the hardware and signal processing. In the LFMCW radar system, the frequency of the transmitted signal increases linearly with time, a property that allows the transmitted signal to be used in the form of a continuous wave for accurate range measurements to a target. At the same time, we place a point scatterer, which is placed at a distance d from the transmitter and the receiver. At the receiver, the received signal r(t) is mixed with the transmitted signal s(t). This results in an intermediate frequency (IF) signal, which is represented by the following equation.

$$IF(t) = \frac{\alpha}{d^2} e^{j2\pi\frac{(fc+st)}{c}d} \tag{1}$$

where α is the reflectance, the$\frac{d\phi(t)}{dt} = fc + st$is the emitted signal frequency, fc is the starting frequency and s is the rate of frequency modulation. This IF signal is subsequently used for target detection and distance estimation.

The system model for target localization in a distributed MIMO radar system uses piggyback LFMCW transmitters and receivers. Where N_t is the transmitters, the N_r One for the receiver, satisfying$N_t + N_r = U$. The positions of the transmitters and receivers are represented by $P_{Tx}^{(m)}$ and $P_{Rx}^{(m)}$, respectively. Assuming that there are K targets in a two-dimensional region of interest (ROI), and the position of each target is denoted byt_k.

To simplify receiver design, signals from different transmitters are made orthogonal, achievable in either the time or frequency domain. The transmitted signal is reflected by the target and captured by the receiver, where it is mixed with the transmitted signal to generate an intermediate frequency (IF) signal. The IF signal is sampled by an analog-to-digital converter (ADC) to produce discrete time samples. These samples are used to construct the measurement matrix Ψ, which, together with the reflection vector α and noise vector n, forms the observation vector y. The observation vector y is thus a combination of the reflection vector α and noise vector n.

$$y = \psi\alpha + n \tag{2}$$

4. UAV Placement Optimization

The main goal of this research is to minimize mutual coherence by strategically placing UAVs. Considering that UAV positions are restricted within a specific space (S_{UAV}), the optimization problem can be formulated as follows:

$$min \ _{p_{Tx1},\dots,p_{TxNT},p_{Rx1},\dots,p_{RxNR} \in S_{UAV}} \mu(\Psi_{grid}) \tag{3}$$

Given the complexity of solving this optimization problem directly, we propose six algorithms to approximate a solution.

4.1. Ant Colony Optimization (ACO) Placement Optimization

The operational space of UAVs (represented by S_{UAV}) will be discretized into a set S_Q consisting of Q grid points, each representing a potential position of the UAV. Through this discretization, ACO iteratively searches for a near-optimal solution, emulating the foraging behavior of ants. Pheromone levels for transmitters and receivers are set to encourage exploration at first. The initial solution is assigned a high value to ensure that any feasible solution represents an improvement. Specifically, the initial pheromone level (τ_0) can be set as:

$$\tau_0 = \frac{1}{n} \tag{4}$$

where (n) is the number of potential UAV positions. This setting ensures that the pheromone levels are not too high initially, promoting broad exploration. Heuristic information, which guides the ants in constructing solutions, is calculated based on the distance between positions. The heuristic information (η_{ij}) for positions (i) and (j) is defined as:

$$\eta_{ij} = \frac{1}{d_{ij}} \tag{5}$$

where (d_{ij}) is the distance between positions (i) and (j). This ensures that closer positions are more attractive to the ants. In each iteration, transmitters and receivers are randomly placed on the grid based on pheromone levels and heuristic information, resulting in effective coverage. The probability (P_{ij}) of an ant moving from position (i) to position (j) is given by:

$$P_{ij} = \frac{(\tau_{ij})^{\alpha}(\eta_{ij})^{\beta}}{\sum_{k\in\text{allowed}}(\tau_{ik})^{\alpha}(\eta_{ik})^{\beta}} \tag{6}$$

where (τ_{ij}) is the pheromone level, (η_{ij}) is the heuristic information, (α) and (β) are parameters controlling their relative importance, and allowed is the set of feasible next positions.

The quality of each constructed solution is assessed by calculating its mutual coherence, which involves deriving the dictionary matrix (Ψ) based on UAV positions and the coherence matrix ($\Psi'\Psi$). The objective is to minimize the maximum off-diagonal element of this matrix. Pheromone levels are adjusted according to the solution's quality, reinforcing optimal solutions and guiding the algorithm towards better placements. If the current solution achieves lower mutual coherence than the best-known solution, the best solution is updated, ensuring the algorithm retains the optimal solution identified so far. Additionally, pheromone levels are gradually decreased to prevent premature convergence, maintaining a balance between exploration and exploitation by avoiding excessive attraction to current optimal solutions.

The pseudo-code for the ACO algorithm is given below:

```
Initialize pheromone levels
Set best_mu to a very high value
For iter = 1 to max_iter do
    For each ant do
        Generate solution p_tx, p_rx
        Calculate mutual coherence mu_temp
        If mu_temp < best_mu then
update best_mu, best_p_tx, best_p_rx
        End If
        Update pheromone levels for p_tx, p_rx
    End For
    Evaporate pheromone levels
End For
Return best_mu, best_p_tx, best_p_rx
```

It exploits the ability of the ACO algorithm to find high-quality solutions within complex search spaces, making the approach very appropriate for UAV placement optimization in the case of distributed sensing.

4.2. Differential Evolution (DE) Placement Optimization

The iterative structure of the DE-based placement algorithm closely resembles that of other evolutionary algorithms. Unlike heuristic searches, DE employs a population-based strategy to explore the search space, which helps maintain solution diversity and avoids convergence to local minima—particularly advantageous in high-dimensional, complex optimization problems like UAV placement.

Initially, a population of potential solutions is generated, with each solution representing positions for transmitters and receivers. This population is then randomly dispersed across the UAV space to ensure broad coverage. To evaluate the quality of each solution, mutual coherence must be calculated, which involves two key tasks: computing the dictionary matrix (Ψ) based on UAV positions and computing the coherence matrix ($\Psi'\Psi$). The goal is to minimize the maximum off-diagonal element in the coherence matrix, a critical factor in UAV placement problems.

For each individual in the population, a mutant vector is created by combining the positions of three randomly selected solutions, scaled by a mutation factor (F). The mutant vector ($\boldsymbol{v}_i$) is generated as follows:

$$\boldsymbol{v}_i = \boldsymbol{x}_{r1} + F \cdot (\boldsymbol{x}_{r2} - \boldsymbol{x}_{r3}) \tag{7}$$

where ($\boldsymbol{x}_{r1}, \boldsymbol{x}_{r2}, \boldsymbol{x}_{r3}$) are randomly selected individuals from the population. A trial vector is then generated by combining the mutant vector with the current solution, guided by a crossover probability (CR). The trial vector ($\boldsymbol{u}_i$) is formed as:

$$u_{ij} = \begin{cases} v_{ij} & \text{if } rand_j \leq CR \\ x_{ij} & \text{otherwise} \end{cases} \tag{8}$$

where ($rand_j$) is a uniformly distributed random number. The trial vector is checked to ensure it is within the UAV space boundaries.

It compares the fitness of the trial vector with the current solution; in case of a smaller mutual coherence, the trial vector replaces the current solution in the population. In this manner, the population will be driven toward optimal placements with only very good solutions retained. The best solution found across generations is tracked to guarantee that the algorithm converges toward optimal placements.

The two important issues in the performance of a DE algorithm are appropriate choices of mutation factor (F) and crossover probability (CR). The amplitude of the mutation vector is usually controlled by a mutation factor, F, which falls between 0 and 1. Small values of F facilitate fine-tuning in the later stages of the algorithm; large values of F provide broader exploration in the early stages. A typical strategy is that F has been taken between 0.5 and 0.9; this value may be chosen by experimental tuning.

The crossover probability, usually denoted by CR, is used to determine how much of the trial vector should come from the mutant vector. This recommended range is usually between 0 and 1. A higher value of (CR) would give more weight to the mutant vector, exploring more new solutions, while a lower value will maintain more of the characteristics of the current solution. Common ranges for this parameter are between 0.1 and 0.9; the optimal value has to be empirically determined.

Higher values of F and CR can increase the convergence speed but raise the possibility of premature convergence. Lower values can enhance solution precision but may lower the speed of convergence. With higher values of F and CR, solution diversity increases, which enables the process to perform a global search. With lower values, solution diversity goes down, so it favors local search. With improper choice of parameters, instability results, and large fluctuations in the quality of the solution are noted. Experimental tuning for F and CR can improve stability.

Experiments on the parameter tuning either by grid search or random search can be done to find the optimal values of. Then, one can compare the different performances of the algorithms under these various sets of different parameters to find out which configuration is the best for the problem at hand.

The pseudo-code for the DE algorithm is as follows:

```
Initialize population
Set initial fitness values to a very high value
**For** gen = 1 to max_generations do
    **For** each individual in the population do
        Evaluate the fitness of current solution
```

```
            Create mutant vector
            Generate trial vector through crossover
            Ensure the trial vector is within boundaries
            Evaluate the fitness of trial vector
            If the trial vector is better than the current solution then
                Replace the current solution with the trial vector
            **End If**
        **End For**
        Update best solution found so far
    **End For**
    **Return** best_mu, best_p_tx, best_p_rx
```

4.3. Simulated Annealing (SA) Placement Optimization

In the SA-based placement algorithm, the iterative structure is designed to mimic the annealing process in metallurgy. This method effectively explores the search space and avoids local minima by allowing occasional uphill moves.

Initially, positions for transmitters and receivers are generated randomly within the defined UAV space, ensuring a broad initial search area. The initial temperature (T) and cooling rate (α) are set. The initial mutual coherence is then calculated by computing the dictionary matrix (Ψ) based on the initial UAV positions and evaluating the coherence matrix ($\Psi'\Psi$). The goal is to minimize the maximum off-diagonal element of this matrix, representing the mutual coherence.

During each iteration, new candidate solutions are generated by adding small random perturbations to the current positions of transmitters and receivers. This step explores the neighborhood of the current solution:

$$\boldsymbol{p}_{Tx}^{new} = \boldsymbol{p}_{Tx} + (\text{rand}(\text{size}(\mathbf{p}_{Tx})) - 0.5) \cdot \text{resolution}$$
$$\boldsymbol{p}_{Rx}^{new} = \boldsymbol{p}_{Rx} + (\text{rand}(\text{size}(\mathbf{p}_{Rx})) - 0.5) \cdot \text{resolution} \quad (9)$$

The new positions are checked to ensure they are within the defined boundaries of the UAV space, preventing the UAVs from being placed outside the operational area:

$$\boldsymbol{p}_{Tx}^{new} = max(min(\boldsymbol{p}_{Tx}^{new}, [xlength, ylength, zlength]), [0,0,0])$$
$$\boldsymbol{p}_{Rx}^{new} = max(min(\boldsymbol{p}_{Rx}^{new}, [xlength, ylength, zlength]), [0,0,0]) \quad (10)$$

The mutual coherence of the new solution is then calculated by computing the dictionary matrix (Ψ) and the coherence matrix ($\Psi'\Psi$). The acceptance probability is determined based on the mutual coherence and the current temperature, allowing occasional acceptance of worse solutions to escape local minima:

$$\text{acceptance\_probability}= exp\left(\frac{\mu-\mu_{\text{new}}}{T}\right) \quad (11)$$

If the new solution is accepted, the current positions and mutual coherence are updated. If the new solution is better than the best-known solution, the best solution is updated. The temperature is then reduced according to the cooling schedule:

$$T = T \cdot \alpha \quad (12)$$

Throughout the process, the best solution found across all iterations is tracked to ensure the algorithm converges towards the optimal placement. This involves continuously updating the best-known solution based on the fitness evaluations.

The pseudo-code for the SA algorithm is as follows:

```
Initialize positions and temperature
Evaluate initial solution
Set best solution to initial solution
**For** iter = 1 to max_iter do
    Generate new candidate solution
```

```
        Ensure new solution is within boundaries
        Evaluate new solution
        Calculate acceptance probability
        **If** new solution is accepted then
            Update current solution
        **End If**
        **If** current solution is better than best solution then
            Update best solution
        **End If**
        Cool down temperature
    **End For**
    **Return** best_mu, best_p_tx, best_p_rx
```

4.4. Genetic Algorithm (GA) Placement Optimization

In general, the basic structure of the GA-based iterative placement algorithm is very close to most of the evolutionary algorithms. This type of algorithm uses natural selection, crossover, and mutation to evolve generations of solution populations. It is very good at maintaining solution diversity and thus avoiding local minimums, which becomes very critical for high-dimensional complex optimization problems like UAV placement.

First, it creates a population of possible solutions. Each solution corresponds to a set of transmitter and receiver positions. This population will then be spread randomly to guarantee that the search covers a wide portion of the defined UAV space. Next, the mutual coherence for the quality evaluation of every solution in the population will be calculated. This step requires the computation of a dictionary matrix (Ψ)based on UAV positions and a coherence matrix ($\Psi'\Psi$). The objective should be such that the maximum off-diagonal element of this matrix, which is the mutual coherence, is minimized.

The solutions are then ranked based on their fitness values, and the top-performing solutions are selected to form a mating pool. This step ensures that the best solutions have a higher chance of passing their genes to the next generation. For each pair of solutions in the mating pool, crossover is performed to generate offspring. This involves exchanging segments of the parent solutions to create new solutions, with the crossover operation performed with a certain probability (CR), and the crossover points randomly selected:

$$\text{offspring}_1 = \text{parent}_1[1\colon crossover_{point}] + \text{parent}_2[crossover_{point} + 1\colon end]$$

$$\text{offspring}_2 = \text{parent}_2[1\colon crossover\_point] + \text{parent}_1[crossover\_point + 1\colon end] \tag{13}$$

Random mutations are introduced to the offspring with a certain probability (MR), involving randomly altering the positions of some UAVs to explore new areas of the search space:

$$\text{mutant}_i = \text{offspring}_i + \text{random\_perturbation} \tag{14}$$

The mutated solutions' positions are checked to ensure they are within the defined boundaries of the UAV space, preventing the UAVs from being placed outside the operational area. The fitness of the offspring is then evaluated, and if the offspring have a lower mutual coherence than the current solutions, they replace the worst-performing solutions in the population. This ensures that only the best solutions are retained, driving the population towards optimal placements.

Throughout the process, the best solution found across all generations is tracked to ensure the algorithm converges towards the optimal placement. This involves continuously updating the best-known solution based on the fitness evaluations.

The pseudo-code for the GA algorithm is as follows:

```
Initialize population
Set initial fitness values to a very high value
**For** gen = 1 to max_generations do
    Evaluate fitness of current population
```

```
        Select top-performing solutions for mating pool
        Perform crossover to generate offspring
        Introduce mutations to offspring
        Ensure offspring are within boundaries
        Evaluate fitness of offspring
        Replace worst-performing solutions with offspring
        Update best solution found so far
    End For
    Return best_mu, best_p_tx, best_p_rx
```

4.5. Particle Swarm Optimization (PSO) Placement Optimization

In the PSO-based placement algorithm, the iterative structure is designed to mimic the social behavior of birds flocking or fish schooling. This method effectively explores the search space by leveraging both individual and collective experiences of the particles.

Initially, positions for transmitters and receivers are generated randomly within the defined UAV space, ensuring a broad initial search area. Velocities for the particles are also initialized to control their movement through the search space. The initial personal best positions (p_best) and global best position (g_best) are set.

The quality of each particle's position is assessed by calculating the mutual coherence, which involves computing the dictionary matrix (Ψ) based on the UAV positions and evaluating the coherence matrix ($\Psi'\Psi$). The goal is to minimize the maximum off-diagonal element of this matrix, representing the mutual coherence.

For each particle, the current fitness is compared with its personal best fitness. If the current fitness is better, the personal best position and fitness are updated. The best fitness of all particles is compared with the global best fitness, and if any particle's best fitness is better than the global best fitness, the global best position and fitness are updated.

The velocity of each particle is updated based on its current velocity, the distance to its personal best position, and the distance to the global best position. The velocity update formula is:

$$\boldsymbol{v}_i = w\boldsymbol{v}_i + c1 \cdot r1 \cdot (\boldsymbol{p\_best}_i - \boldsymbol{x}_i) + c2 \cdot r2 \cdot (\boldsymbol{g\_best} - \boldsymbol{x}_i) \tag{15}$$

where (w) is the inertia weight, ($c1$) and ($c2$) are cognitive and social coefficients, and ($r1$) and ($r2$) are random numbers between 0 and 1. The position of each particle is then updated based on its updated velocity:

$$\boldsymbol{x}_i = \boldsymbol{x}_i + \boldsymbol{v}_i \tag{16}$$

The updated positions are checked to ensure they are within the defined boundaries of the UAV space, preventing the UAVs from being placed outside the operational area:

$$\boldsymbol{x}_i = max(min(\boldsymbol{x}_i, [xlength, ylength, zlength]), [0,0,0]) \tag{17}$$

Throughout the process, the best solution found across all iterations is tracked to ensure the algorithm converges towards the optimal placement. This involves continuously updating the best-known solution based on the fitness evaluations.

The pseudo-code for the PSO algorithm is as follows:

```
Initialize positions and velocities
Evaluate initial solutions
Set personal and global bests
For iter = 1 to max_iter do
    For each particle do
        Evaluate fitness of current position
        Update personal best if current fitness is better
        Update global best if any personal best is better
    End For
    For each particle do
        Update velocity
        Update position
        Ensure position is within boundaries
    End For
    Update best solution found so far
End For
Return best_mu, best_p_tx, best_p_rx
```

4.6. The Algorithm Combined HS and GD Algorithm

Based on the heuristic algorithm and gradient descent algorithm proposed in the literature [Distributed UAV Swarm Placement Optimization for Compressive Sensing based Target Localization]. We found that, in the deployment of UAV localization, the heuristic algorithm has a high localization accuracy, but the system overhead is very large, once the number of UAVs or the number of target points increases, or the terrain of the space to be deployed is complex, the total amount of computation will increase dramatically, which is not conducive to the long-term work of the UAV; and the gradient descent algorithm, although the consumption of system resources is not high, but the localization accuracy is not high.

Therefore, this paper tries to combine the advantages of the gradient descent method with low computing power and the heuristic algorithm with high accuracy. In the selection of the combination scheme, considering that the combination scheme of "external HS and internal GD" will lead to running the HS algorithm and the GD algorithm once in each iteration, which will make the system consume a lot of resources to carry out meaningless calculations, this paper decides to adopt the combination strategy of "HS first and GD later". ". That is, the HS algorithm is first used to determine the approximate candidate range of UAV deployment locations, and then the GD algorithm is used to further calculate the more accurate locations.

In the simulation of this algorithm, the space where the UAVs are to be deployed is an unobstructed cubic space, the target points are all located on the ground, i.e., a plane of height 0, and the UAVs are deployed in a plane of height 10. The pseudo-code of the combination algorithm corresponding to the optimization corresponding to this combination strategy is as follows:

```
    Initialization parameters
    For iteration loop starts
    If first HS iteration
            initialize temporary correlations
    Else
            calculate the approximate ranges of the transmitter and receiver UAV positions and the
corresponding mu values, respectively.
    End For
    For enter GD iteration
    If first GD iteration
                Import the inter-correlation matrix calculated by HS iteration.
    Else
    calculate more accurate drone candidate positions and their corresponding mu values in the candidate
range.
    output the mu value, the position of the transmitting UAV, and the position of the receiving UAV
    End For
```

However, simulation results show that the RMSE performance of this combined algorithm is not excellent. Compared with simply using the HS algorithm or the GD algorithm, even if the relevant parameters are improved, this combination scheme does not show higher superiority. Therefore, in this paper, we will try to use other intelligent optimization algorithms to deploy the UAV's position.

5. Results and Discussion

In our model, the Region of Interest (ROI), denoted as S, is a 1 m × 1 m 2D plane that contains two targets. The ROI is discretized into a 10 × 10 grid, with each grid point separated by 0.1 m. The number of UAVs (U) is chosen from a range of 4 to 12. These UAVs operate within a 3D space (S_{UAV}) with dimensions of 3 m × 3 m × 3 m, located 4 m above the ROI. For the Linear Frequency Modulated Continuous Wave (LFMCW) radar system, the carrier frequency (f_c) is 60 GHz, with a frequency modulation slope of 60 MHz/μs and a bandwidth of 1.5 GHz. A total of 256 samples are used for signal processing. Monte Carlo simulation, denoted as M_c, with 10^4 simulation rounds, is applied to assess the localization performance under various placement optimization algorithms. Localization performance is evaluated using the root mean square error (MSE) of the estimation, defined as:

$$MSE = \frac{\sum_{i,k} \|\hat{t}_k^i - t_k^i\|^2}{K \cdot M_c} \tag{18}$$

where ($\hat{t}_k^i$) is the estimated position of the kth target in the ith Monte Carlo round, and (t_k^i) is the true position of the kth target in the ith Monte Carlo trial. K represents the number of targets. Figure 1 illustrates the MSE performance of the CS-based localization for two fixed targets under various optimized UAV placement strategies. These strategies include random placement (RD), heuristic search placement (he), gradient descent placement (GD), Optimized UAV (OPT), Ant Colony Optimization (ACO), Differential Evolution (DE), Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and Simulated Annealing (SA). In this scenario, four UAVs are considered, with two serving as transmitters and two as receivers. The targets are located at [0.2 m, 0.2 m] and [0.8 m, 0.8 m], which correspond exactly to the grid points in our dictionary. Therefore, it is possible that the two targets are precisely localized by the UAV swarm without errors if optimization algorithm and SNR condition allows. The ends of the curves represent the capability of achieving such none error performance, and are noted as precise estimation points (PEP). Consequently, several well-performing placements, including HE, GD, ACO, DE, and PSO, can accurately estimate the target locations once a certain SNR threshold is achieved. In contrast, the RA, GA, OPT, and SA placements do not reach this level of accuracy. Among all proposed algorithms, ACO reaches its PEP firstly where SNR is about 7.5dB, regardless its poorer accuracy at lower SNR. On the contrary, SA displays the best accuracy where SNR is below 0dB, though it has no PEP under 20 dB. The result leads to a possible strategy that UAV swarm can choose different algorithms to optimize their relative position for better localization performance according to the SNR of its surroundings.

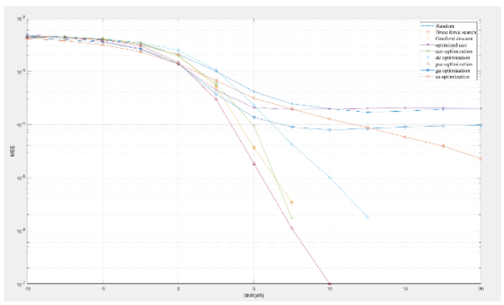

Figure 1. The MSE performance of 2 fixed targets.

Figure 2 and Figure 3 compare the computational time of various optimization algorithms as they handle different numbers of UAVs. In Fig 2, the horizontal axis represents the number of UAVs, ranging from 0 to 12. The curves show the average consuming time of calculation for different algorithms with different UAVs initial positions. The performance of these algorithms can be categorized into four levels. The first level includes OPT and SA, which complete their` calculations in just 2 seconds. The second level is represented by GD. The third level comprises ACO, GA, and PSO, with their computational time reaching up to 6 seconds when processing 12 drones. The fourth level is occupied by de, which takes the longest time to complete calculations. While Fig 3 exhibits the all the distribution of calculation times of each algorithm under 4 drones (2 transmitters, 2 receivers) with different UAVs initial position, with the average time points of each algorithm correspond to the point in Fig 2. All algorithms have relatively stable calculation times except GD and OPT, which prove the robustness of these algorithms.

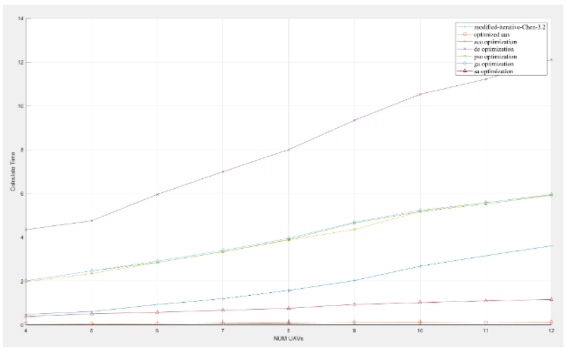

Figure 2. Computational complexity comparison, in terms of the required calculation time.

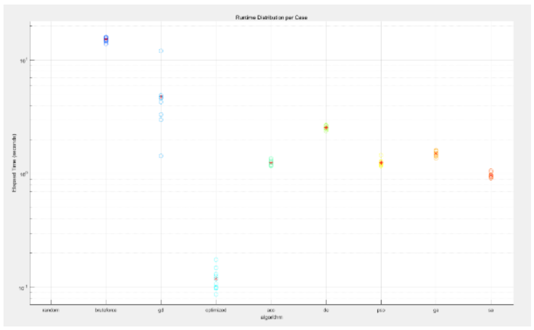

Figure 3. The scatter plot of calculation time of different algorithm.

Figure 4 illustrates the impact of UAV count on mutual coherence and MSE performance. The results demonstrate that increasing the number of UAVs enhances localization accuracy. This improvement is attributed to the fact that adding more UAVs increases the number of rows in the dictionary matrix (Ψ_{grid}), which facilitates achieving lower mutual coherence. As a result, when UAV placements are optimized using our algorithms, the mutual coherence of (Ψ_{grid}) is further reduced, leading to improved MSE performance.

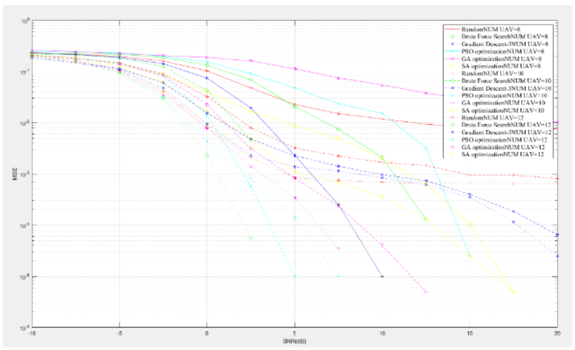

Figure 4. The relationship between the localization performance (in terms of MSE performance) and the number of UAVs.

6. Conclusion

This paper demonstrates the effectiveness of various UAV placement optimization algorithms in enhancing the localization accuracy of CS-based radar systems in a 3D environment. Through Monte Carlo simulations, the performance of these algorithms is evaluated by MSE and computational time. The results highlight that certain algorithms—Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), and Differential Evolution (DE)—re particularly effective in achieving precise target localization under specific SNR conditions. Furthermore, the computational efficiency of the algorithms varies, with some, like the Optimized UAV (OPT) and Simulated Annealing (SA). The findings also indicate that increasing the number of UAVs enhances localization performance by reducing mutual coherence in the dictionary matrix Ψ_{grid}, thus lowering the MSE. Overall, the choice of UAV placement strategy should be adaptive, depending on the surrounding SNR environment and the balance between accuracy and calculation complexity. By selecting the most appropriate algorithm, UAV swarms can optimize their positions more effectively, leading to better localization outcomes in practical applications.

Acknowledgment

Du Zeyu, Liu Luoyu, Zhou Hangyu, Chen Ziyi and Yusen Yao contributed equally to this work and should be considered co-first authors.

References

[1] Y. -C. Wang and D. Cabric, "Distributed UAV Swarm Placement Optimization for Compressive Sensing based Target Localization," 2023 International Conference on Computing, Networking and Communications (ICNC), Honolulu, HI, USA, 2023, pp. 547-551, doi: 10.1109/ICNC57223.2023.10074263.

[2] D. S. Lakew, A. Masood and S. Cho, "3D UAV Placement and Trajectory Optimization in UAV Assisted Wireless Networks," 2020 International Conference on Information Networking (ICOIN), Barcelona, Spain, 2020, pp. 80-82, doi: 10.1109/ICOIN48656.2020.9016553.

[3] J. Won, D. -Y. Kim and J. -W. Lee, "Joint Optimization of Placement, Beamwidth, and Power Allocation in the UAV-Enabled Network," 2022 13th International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea, Republic of, 2022, pp. 236-238, doi: 10.1109/ICTC55196.2022.9952708.

[4] B. I. -D. Ghomri, M. Y. Bendimerad and F. T. Bendimerad, "Utilizing Deep Reinforcement Learning for Optimal UAV 3D Placement in NOMA-UAV Networks," 2024 2nd International Conference on Electrical Engineering and Automatic Control (ICEEAC), Setif, Algeria, 2024, pp. 1-5, doi: 10.1109/ICEEAC61226.2024.10576211.

[5] D. S. Lakew, A. Masood and S. Cho, "3D UAV Placement and Trajectory Optimization in UAV Assisted Wireless Networks," 2020 International Conference on Information Networking (ICOIN), Barcelona, Spain, 2020, pp. 80-82, doi: 10.1109/ICOIN48656.2020.9016553.

[6] L. Yongjiang, B. Luhao and Z. Dong, "Adaptive digital self-interference cancellation based on fractional order LMS in LFMCW radar," in Journal of Systems Engineering and Electronics, vol. 32, no. 3, pp. 573-583, June 2021, doi: 10.23919/JSEE.2021.000049.

[7] Y. Yu, S. Sun, R. N. Madan and A. Petropulu, "Power allocation and waveform design for the compressive sensing based MIMO radar," in IEEE Transactions on Aerospace and Electronic Systems, vol. 50, no. 2, pp. 898-909, April 2014, doi: 10.1109/TAES.2014.130088.