drones

Article

Optimization of False Target Jamming against UAV Detection

Zheng-Lian Su [1], Xun-Lin Jiang [2], Ning Li [3], Hai-Feng Ling [1] and Yu-Jun Zheng [3,*]

1 College of Field Engineering, Army Engineering University, Nanjing 210007, China; suzhenglian@compintell.cn (Z.-L.S.); hfling@compintell.cn (H.-F.L.)
2 Department of Engineering Technology and Application, Army Infantry College, Nanchang 330100, China; xunlinjiang@163.com
3 School of Information Science and Technology, Hangzhou Normal University, Hangzhou 311121, China; 2021111011001@stu.hznu.edu.cn
* Correspondence: yujun.zheng@computer.org

Abstract: Unmanned aerial vehicles (UAVs) have been widely used for target detection in modern battlefields. From the viewpoint of the opponents, false target jamming is an effective approach to decrease the UAV detection ability or probability, but currently there are few research efforts devoted to this adversarial problem. This paper formulates an optimization problem of false target jamming based on a counterpart problem of UAV detection, where each false target jamming solution is evaluated according to its adversarial effects on a set of possible UAV detection solutions. To efficiently solve the problem, we propose an evolutionary framework, which is implemented with four popular evolutionary algorithms by designing/adapting their evolutionary operators for false target jamming solutions. Experimental results on 12 test instances with different search regions and numbers of UAVs and false targets demonstrate that the proposed approach can significantly reduce the UAV detection probability, and the water wave optimization (WWO) metaheuristic exhibits the best overall performance among the four evolutionary algorithms. To our knowledge, this is the first study on the optimization of false target jamming against UAV detection, and the proposed framework can be extended to more countermeasures against UAV operations.

Keywords: unmanned aerial vehicle (UAV); UAV detection; false target jamming; optimization; evolutionary algorithm

check for updates

Citation: Su, Z.-L.; Jiang, X.-L.; Li, N.; Ling, H.-F.; Zheng, Y.-J. Optimization of False Target Jamming against UAV Detection. *Drones* **2022**, *6*, 114. https://doi.org/10.3390/drones6050114

Received: 8 April 2022
Accepted: 29 April 2022
Published: 2 May 2022

1. Introduction

Drones, i.e., unmanned aerial vehicles (UAVs) have been increasingly used in both civil and military tasks [1]. Particularly, they have been proven to be successful and efficient for target detection and information collection in modern battlefields, especially in areas that are considered to be inaccessible or dangerous for human pilots [2]. There have been many studies on how to improve UAV detection abilities by means of efficient path planning [3–5], multi-UAV cooperation [6–9], human–UAV collaboration [10–12], and AI-enhanced detection techniques [13–15]. However, there are few studies on how to reduce the abilities of UAVs from the viewpoint of the opponents, which has become an important problem for enhancing safety, security, and privacy against hostile UAVs [16].

False target jamming is a common countermeasure technique that imitates physical shapes or electromagnetic signals to lead hostile reconnaissance equipment to detect false information, so as to significantly reduce their abilities or probabilities of detecting real targets. This technique has been widely used in deceiving pilots and radars on reconnaissance aircrafts [17]. Nevertheless, compared to classical aircrafts, today's UAVs are considerably more autonomous and light-weight, equipped with more advanced optical, thermal, acoustic, and other sensors, and hence allow flexible and high-resolution remote sensing. Moreover, there are various UAV planning and control methods, and different methods often result in quite different UAV operational behaviors. As the opponents, we hardly know exactly which method(s) the UAV holders would adopt. Finding the optimal or sufficiently high-quality false target jamming solutions is known to be a complex computational optimization problem [18], and the above issues significantly increase the difficulty of false target jamming that aims at suppressing or reducing the detection abilities of hostile UAVs as much as possible.

To address the above challenges, in this paper we present an optimization problem of false target jamming against UAV detection, which is formulated based on a counterpart problem that determines the search paths and modes of UAVs to maximize the total detection reward. The problem of false target jamming aims to find an optimal false target placement solution, under which the decrease of UAV detection reward (averaged over a set of possible UAV detection solutions) would be maximized. To efficiently solve the problem, we propose an evolutionary framework, which is implemented with four popular evolutionary algorithms including genetic algorithm (GA) [19], particle swarm optimization (PSO), [20], ecogeography-based optimization (EBO) [21], and water wave optimization WWO [22], whose evolutionary operators are elaborately adapted to evolve false target placement solutions. Experimental results on 12 test instances demonstrate that the evolutionary algorithms can reduce the UAV detection reward much more significantly than simple methods, including randomly or evenly distributing false targets, and WWO exhibits the best overall performance among the four evolutionary algorithms. The main contributions of this paper can be summarized as follows:

- We present a false target jamming problem for minimizing the expected total detection reward of UAVs as an adversarial problem of UAV detection;
- We propose an evolutionary framework implemented with four evolutionary algorithms to efficiently solve the false target jamming problem.
- We demonstrate the performance of the proposed framework on the test instances with different UAV detection and false target placement settings.

In the rest of this paper, Section 2 presents the basic UAV detection problem and the adversarial problem of false target jamming, Section 3 proposes the evolutionary framework and its implementations, Section 4 presents the experimental results, and Section 5 concludes with a discussion.

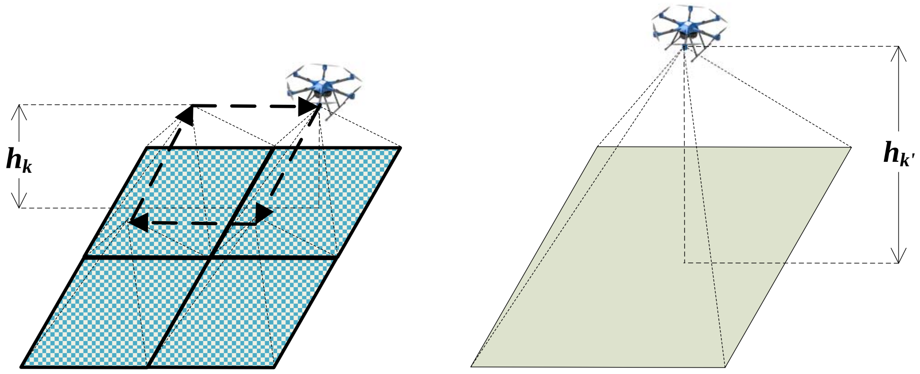
2. Problem Formulation

2.1. Problem of UAV Detection

First consider a basic UAV detection problem, where a swarm of n UAVs are used to search for one or multiple targets in a large region. The whole search region is divided into m sub-regions based on topographic feature, such that different sub-regions have different topographic features, while there is no significant topographic change within one sub-region. Each i-th sub-region is assigned with a target existence probability $p(i)$ ($1 \leq i \leq m$), assuming that the holder of UAVs has a prior probability distribution function that describes the initial belief of the target location. Information about the search region and targets are typically collected by means such as satellite remote sensing and early manual exploration, and target existence probabilities can be estimated using empirical methods as proposed in [4].

According to different flight altitudes and other operational settings, a UAV can use K different search modes that affect its detection ability. For example, when a UAV flies at a low height and uses a high-resolution camera, the detection ability is high, but the detection time period is relatively long; when it flies at a high height and uses a wide-range sensing equipment, the detection ability is relatively low, but the time period is short, as illustrated in Figure 1. We assume that a UAV can use only one mode to search in one sub-region; when it uses the k-th mode to search in the i-th sub-region, the required time period is $t_s(i,k)$, and the posterior probability of detection is $p_d(i,k)$ subject to the precondition of target existence. We are also given the time period $t_f(i,k,i',k')$ for a UAV to fly from the i-th sub-region with the k-th search mode to the i'-th sub-region with the k'-th search mode

($1 \le i, i' \le m; 1 \le k \le K$). Specifically, we use $t_f(0, 0, i, k)$ to denote the time period for a UAV to fly from its starting position to the i-th sub-region with the k-th search mode.



(**a**) At a low height (**b**) At a high height

Figure 1. Different search modes of UAV: (**a**) the UAV searches the sub-region at four hovering points of a low height; (**b**) the UAV searches the sub-region at one hovering point of a high height.

The UAV detection problem needs to plan a search path for each j-th UAV ($1 \le j \le n$), which consists of two parts: the first is the sequence $\mathbf{x}_j = (x_{j,1}, x_{j,2}, \ldots, x_{j,m_j})$ of sub-regions, where m_j denotes the number of sub-regions to be searched by the j-th UAV; the second is the associated search modes $\mathbf{y}_j = (y_{j,1}, y_{j,2}, \ldots, y_{j,m_j})$, where $y_{j,i}$ denotes the search mode of the j-th UAV in the sub-region $x_{j,i}$ ($1 \le i \le m_j$).

Each j-th UAV takes off at time $t = 0$ and arrives at its first sub-region $x_{j,1}$ at time:

$$t(j, 1) = t_f(0, 0, x_{j,1}, y_{j,1}) \tag{1}$$

and it finishes the search on $x_{j,1}$ at time:

$$t'(j, 1) = t(j, 1) + t_s(x_{j,1}, y_{j,1}) \tag{2}$$

By analogy, the time at which the j-th UAV arrives and finishes the search at its i-th sub-region can be iteratively calculated as follows ($2 \le i \le m_j$):

$$\begin{aligned} t(j,i) &= t'(j, i-1) + t_f(x_{j,i-1}, y_{j,i-1}, x_{j,i}, y_{j,i}) \\ t'(j,i) &= t(j,i) + t_s(x_{j,i}, y_{j,i}) \end{aligned} \tag{3}$$

Let T be the predefined time limit of the operation. Whenever the j-th UAV completes the search of the ith sub-region in its sequence, it obtains a reward calculated based on the completion time and detection probability in the sub-region as follows:

$$R(j,i) = \big(T - t'(j, x_{j,i})\big)\, p(x_{j,i})\, p_d(x_{j,i}, y_{j,i}) \tag{4}$$

where the item $\big(T - t'(j, x_{i,j})\big)$ encourages to detect the targets as early as possible.

The objective of the problem is to maximize the total reward (i.e., the total time-weighted detection probability) of the UAVs, and the problem is formulated as follows:

$$\begin{aligned} \max \quad & f(\mathbf{x},\mathbf{y}) = \frac{1}{T}\sum_{j=1}^{n}\sum_{i'=1}^{m_j} R(j,i) && (5)\\ \text{s.t.} \quad & \text{Equations (1)–(4)} \\ & t'(j,x_{j,m_j}) \le T, \quad j=1,2,\ldots,n && (6)\\ & 1 \le y_{j,i} \le K, \quad j=1,2,\ldots,n; i=1,2,\ldots,m_j && (7) \end{aligned}$$

where the denominator T in (5) is used to normalize the objective value in the range of [0, 1].

2.2. Adversarial Problem of False Target Jamming

The opponent wants to jam UAV detection by placing a set of Q false targets in the search region. Each sub-region can have at most one false target, and false targets can only be placed in sub-regions without real targets. The placement of false targets can have one or more of the following effects on UAV detection:

- It will interfere with the observation of the prior probability in the sub-region by means such as satellite remote sensing and early manual exploration.
- For a sub-region without a real target, placing a false target will generate a *false* posterior probability detected by UAV.
- For a sub-region with a real target, placing false targets in its neighboring sub-regions will affect (typically decrease) the *true* posterior probability.

Let $\mathbf{z} = (z_1, z_2, \ldots, z_m)$ be a solution of false target placement, where $z_i = 1$ denotes that a false target is placed in the i-th sub-region and $z_i = 0$; otherwise, we use $\widetilde{p}(i,\mathbf{z})$ and $\widetilde{p}_d(i,k,\mathbf{z})$ to denote the updated prior probability and posterior probability after placing false targets according to $\mathbf{z}$, respectively. Initially, the UAVs do not have knowledge of the false targets; therefore, from the viewpoint of the holder of UAVs, the objective function (5) of the UAV detection problem will be updated as:

$$\widetilde{f}_{\mathbf{z}}(\mathbf{x},\mathbf{y}) = \frac{1}{T}\sum_{j=1}^{n}\sum_{i'=1}^{m_j}\left(T - t'(j,x_{j,i})\right)\widetilde{p}(x_{j,i},\mathbf{z})\widetilde{p}_d(x_{j,i},y_{j,i},\mathbf{z}) \quad (8)$$

Ideally, suppose that the holder of UAVs can find the optimal solution $(\mathbf{x}^*_{\mathbf{z}}, \mathbf{y}^*_{\mathbf{z}})$ that maximizes the updated objective function (8) while satisfying all constraints of the original UAV detection problem. From the viewpoint of the opponent, the adversarial problem of false target jamming aims to find a false target placement solution $\mathbf{z}^*$, such that, among all possible false target placement solutions, the corresponding UAV detection solution $(\mathbf{x}^*_{\mathbf{z}^*}, \mathbf{y}^*_{\mathbf{z}^*})$ will result in the minimum value of the original objective function (5). Consequently, the adversarial problem can be formulated as follows:

$$\begin{aligned} \min \quad & g(\mathbf{z}) = f\left(\arg\max_{\mathbf{x},\mathbf{y}} \widetilde{f}_{\mathbf{z}}(\mathbf{x},\mathbf{y})\right) && (9)\\ \text{s.t.} \quad & z_i \in \{0,1\}, \quad i=1,2,\ldots,m && (10) \end{aligned}$$

However, in many practical situations, the holder of UAVs cannot guarantee finding the optimal solution: for a difficult detection problem instance, it is more possible for the holder to obtain one or more near-optimal solutions. Therefore, we consider a set $L(\mathbf{z})$ of possible solutions to the updated UAV detection problem subject to a given false target placement solution $\mathbf{z}$, where each detection solution $\mathbf{l} = (\mathbf{x}(\mathbf{l}), \mathbf{y}(\mathbf{l})) \in L(\mathbf{z})$ is associated a probability $p_a(\mathbf{l})$ of being adopted by the holder of UAVs. In this way, the objective function (9) of the false target jamming problem will be updated as:

$$\min g(\mathbf{z}) = \frac{1}{|L(\mathbf{z})|} \sum_{\mathbf{l} \in p_a(\mathbf{l})} \widetilde{f}_{\mathbf{z}}(\mathbf{x}(\mathbf{l}), \mathbf{y}(\mathbf{l})) \tag{11}$$

3. Evolutionary Algorithms for the Adversarial Problem of False Target Jamming

To solve an instance of the above adversarial problem of false target jamming against UAV detection, we search in its solution space; for each solution $\mathbf{z}$ encountered, we construct an instance of the corresponding UAV detection problem with the objective function (8), and solve it using one or more heuristic and metaheuristic algorithms [23,24], to produce the set $L(\mathbf{z})$ of possible solutions to the UAV detection problem, so as to evaluate the solution $\mathbf{z}$ according to the objective function g defined in (11).

We propose an evolutionary algorithm framework for solving the adversarial problem of false target jamming, which consists of the following steps:

(1) Select a set A_u of heuristic/metaheuristic algorithms for the UAV detection problem;
(2) Initialize a population of solutions to the adversarial problem of false target jamming;
(3) For each solution $\mathbf{z}$ in the population:
    (3.1) Derive an instance of the UAV detection problem according to (8);
    (3.2) Use algorithms from A_u to solve the instance to produce the set $L(\mathbf{z})$ of possible solutions to the UAV detection problem;
    (3.3) Evaluate the fitness $g(\mathbf{z})$ according to (11);
(4) Based on the fitness, evolve the solutions using evolutionary operations;
(5) If the stopping condition is not satisfied, go to Step (3); otherwise, return the best solution $\mathbf{z}^*$ found so far.

In this section, we use four typical evolutionary algorithms, including GA [19], PSO [20], EBO [21], and WWO [22], to implement the framework. The following subsections describe the evolutionary operations of the algorithms.

3.1. Genetic Algorithm

The GA uses crossover and mutation operators to evolve the solutions. The crossover operates on two solutions, denoted by $\mathbf{z}^{\mathrm{a}} = (z_1^{\mathrm{a}}, z_2^{\mathrm{a}}, \ldots, z_m^{\mathrm{a}})$ and $\mathbf{z}^{\mathrm{b}} = (z_1^{\mathrm{b}}, z_2^{\mathrm{b}}, \ldots, z_m^{\mathrm{b}})$, at a time: it generates a random integer m' between $(1, m)$ to divide the two solutions as $\mathbf{z}^{\mathrm{a}} = (z_1^{\mathrm{a}}, \ldots, z_{m'}^{\mathrm{a}}) \oplus (z_{m'+1}^{\mathrm{a}}, \ldots, z_m^{\mathrm{a}})$ and $\mathbf{z}^{\mathrm{b}} = (z_1^{\mathrm{b}}, \ldots, z_{m'}^{\mathrm{b}}) \oplus (z_{m'+1}^{\mathrm{b}}, \ldots, z_m^{\mathrm{b}})$, and then recombines them into two offspring solutions $\mathbf{z}^{\mathrm{c}} = (z_1^{\mathrm{a}}, \ldots, z_{m'}^{\mathrm{a}}) \oplus (z_{m'+1}^{\mathrm{b}}, \ldots, z_m^{\mathrm{b}})$ and $\mathbf{z}^{\mathrm{d}} = (z_1^{\mathrm{b}}, \ldots, z_{m'}^{\mathrm{b}}) \oplus (z_{m'+1}^{\mathrm{a}}, \ldots, z_m^{\mathrm{a}})$, where $\oplus$ denotes concatenation of sequences. After recombination, the number Q_1 of false targets (i.e., the number of one components) in an offspring solution may be not equal to Q: if $Q_1 > Q$, we randomly select $(Q_1 - Q)$ one components and change them to zero; otherwise, if $Q_1 < Q$, we randomly select $(Q - Q_1)$ zero components and change them to one. Figure 2 shows an example of the GA crossover operation.

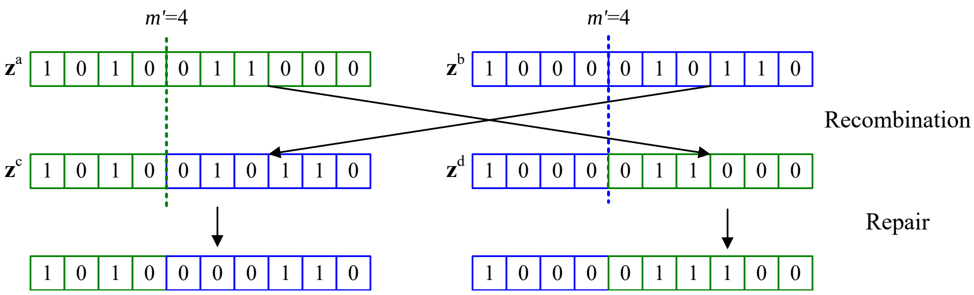


Figure 2. Illustration of a crossover operation on solutions, where $m = 10$, $Q = 4$.

The mutation operates on one solution at a time by generating a random number Q' between $[1, Q/2]$ and randomly exchanging a one component and a zero component for Q' times.

Algorithm 1 presents the pseudo-code of an iteration of the GA that creates a new offspring population P' from the current population P, where r_c denotes the crossover rate and $r_m(\mathbf{z})$ denotes the variable mutation rate for solution $\mathbf{z}$ that is inversely proportional to its fitness [19].

Algorithm 1: GA procedure for creating an offspring population P' from the current population P at each iteration

Create an empty offspring population P';
while $|P'| < |P|$ **do**
  Use roulette wheel selection to randomly select two solutions $\mathbf{z}^{\mathrm{a}}$ and $\mathbf{z}^{\mathrm{b}}$ from P;
  Perform possible reparation on X';
  if $rand(0,1) < r_c$ **then**
    Perform crossover on $\mathbf{z}^{\mathrm{a}}$ and $\mathbf{z}^{\mathrm{b}}$ to produce two offsprings $\mathbf{z}^{\mathrm{c}}$ and $\mathbf{z}^{\mathrm{d}}$;
    if $rand(0,1) < r_m(\mathbf{z}^{\mathrm{c}})$ **then**
      Perform mutation on $\mathbf{z}^{\mathrm{c}}$;
      Repair $\mathbf{z}^{\mathrm{c}}$ if needed;
    if $rand(0,1) < r_m(\mathbf{z}^{\mathrm{d}})$ **then**
      Perform mutation on $\mathbf{z}^{\mathrm{d}}$;
      Repair $\mathbf{z}^{\mathrm{d}}$ if needed;
    Add $\mathbf{z}^{\mathrm{a}}$ and $\mathbf{z}^{\mathrm{b}}$ to P';

3.2. Particle Swarm Optimization

In the PSO algorithm, each solution $\mathbf{z}$ is associated with an m-dimensional real-valued velocity vector $\mathbf{v_z}$, each component of which is randomly initialized between $[-1, 1]$. The algorithm also records the current best known solution ***gbest*** of the whole swarm (population) and the best history $\boldsymbol{pbest^{\mathbf{z}}}$ for each individual solution $\mathbf{z}$. At each iteration, each solution $\mathbf{z}$ learns from both the global best ***gbest*** and its personal best $\boldsymbol{pbest^{\mathbf{z}}}$. At each dimension of $\mathbf{z}$, the learning operator generates three random numbers r, r_1, and r_2 between [0, 1], and updates the dimension z_i as follows ($1 \le i \le m$):

$$z_i = \begin{cases} z_i, & r < w/(w + c_1 r_1 + c_2 r_2) \\ pbest_i^{\mathbf{z}}, & w/(w + c_1 r_1 + c_2 r_2) \le r < (w + c_1 r_1)/(w + c_1 r_1 + c_2 r_2) \\ gbest_i, & r \ge (w + c_1 r_1)/(w + c_1 r_1 + c_2 r_2) \end{cases} \tag{12}$$

where w is the inertial weight and c_1 and c_2 are two learning coefficients, as used in the basic PSO algorithm [25,26].

After updating all dimensions of the solution, if the number of false targets (i.e., the number of 1 components) is not equal to Q, the repair policy described in Section 3.1 is also applied.

Algorithm 2 presents the pseudo-code of an iteration of the PSO algorithm.

Algorithm 2: PSO procedure for evolving the population P at each iteration

```
foreach z ∈ P do
    for i = 1 to m do
        Update z_i according to Equation (12);
    Repair z if needed;
    if g(z) < g(pbest^z) then
        pbest^z ← z;
        if g(z) < g(gbest) then
            gbest ← z;
```

3.3. Ecogeography-Based Optimization

EBO is an extension of the BBO algorithm [27], which evolves solutions by "migrating" features from other solutions. Each solution $\mathbf{z}$ is associated with an immigration rate and an emigration rate as follows:

$$r_\mu(\mathbf{z}) = \frac{g(\mathbf{z}) - g_{\min} + \epsilon}{g_{\max} - g_{\min} + \epsilon} \tag{13}$$

$$r_\nu(\mathbf{z}) = \frac{g_{\max} - g(\mathbf{z}) + \epsilon}{g_{\max} - g_{\min} + \epsilon} \tag{14}$$

where $g_{\max}$ and $g_{\min}$ are the maximum and minimum objective function values in the population, respectively, and ϵ is a very small number to avoid division by zero.

The migration operation of BBO on a solution $\mathbf{z}$ is performed by, at each dimension of $\mathbf{z}$, with a probability of $r_\mu(\mathbf{z})$, migrating the corresponding component of another solution $\mathbf{z}'$, which is selected with a probability of $r_\nu(\mathbf{z}')$:

$$z_i = z'_i \tag{15}$$

EBO enhances BBO by using a local neighborhood structure where each solution is connected to a set of neighboring solutions [28]. For example, Figure 3 shows a ring neighborhood structure where each solution has just two neighbors. Based on the neighborhood structure, EBO defines two migration operators. The first is local migration, which uses Equation (15), but the emigrating solution $\mathbf{z}'$ is selected from the neighbors rather than from the whole population. The second is global migration, which uses two emigrating solutions $\mathbf{z}'$ and $\mathbf{z}''$, one selected from the neighbors and the other selected from non-neighbors; the current dimension of $\mathbf{z}$ will be set to the corresponding dimension of the better one between $\mathbf{z}'$ and $\mathbf{z}''$:

$$z_i = \begin{cases} z'_i, & g(\mathbf{z}') \le g(\mathbf{z}'') \\ z''_i, & g(\mathbf{z}') > g(\mathbf{z}'') \end{cases} \tag{16}$$

EBO uses a parameter η to control the probability of global migration. The parameter value linearly decreases from an upper limit $\eta_{\max}$ to a lower limit $\eta_{\min}$, such that global migration is preferred in early stages to facilitate global exploration and local migration is encouraged in later stages to enhance local exploitation.

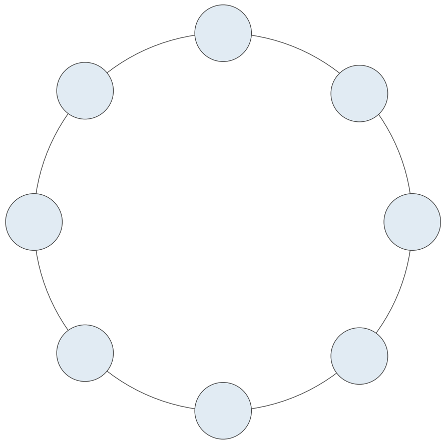


Figure 3. Ring neighborhood structure for EBO population.

Similarly, after updating all dimensions of the solution, if the number of false targets is not equal to Q, the repair policy described in Section 3.1 is applied.

Algorithm 3 presents the pseudo-code of an iteration of the BBO algorithm.

Algorithm 3: EBO procedure for evolving the population P at each iteration

```
foreach z ∈ P do
    for i = 1 to m do
        if rand(0,1) < η then
            Select an emigrating solution z′ according to the emigration rate;
            Perform local migration according to Equation (15);
        else
            Select a neighboring solution z′ and a non-neighboring solution z″
              according to the emigration rates;
            Perform global migration according to Equation (16);
    Repair z if needed;
    if the updated solution is better then
        Replace the original solution with updated solution in P;
Update the parameter η;
```

3.4. Water Wave Optimization

WWO is an evolutionary algorithm inspired by the motion of shallow water waves for optimization problems. It assigns each solution $\mathbf{z}$ with a "wave length" $\lambda(\mathbf{z})$ that is inversely proportional to solution fitness, such that low-fitness solutions explore large ranges while high-fitness solutions explore small ranges. For the adversarial problem of false target jamming, we design the following wave length calculation method:

$$\lambda(\mathbf{z}) = 1 + (\frac{m}{2} - 1)\frac{g(\mathbf{z}) - g_{\min} + \epsilon}{g_{\max} - g_{\min} + \epsilon} \tag{17}$$

At each iteration, WWO generates an offspring for each solution $\mathbf{z}$ by randomly exchanging a one-component and a zero-component of $\mathbf{z}$ for Q' times, while Q' is a random number between $[1, \lambda(\mathbf{z})]$. If the offspring is better than the original $\mathbf{z}$, it replaces $\mathbf{z}$ in the population. Moreover, if the new solution is the new best known solution, WWO performs K_N steps of one-zero exchange on the solution, checking whether the solution is improved after each step (where K_N is a control parameter set to $\min(6, Q/2)$).

As for the variable-size WWO proposed in [29], we use a variable population for WWO by linearly decreasing the population size from an upper limit $NP_{\max}$ to a lower limit

$NP_{\min}$. Whenever the population size is reduced by one, the worst solution is removed from the population.

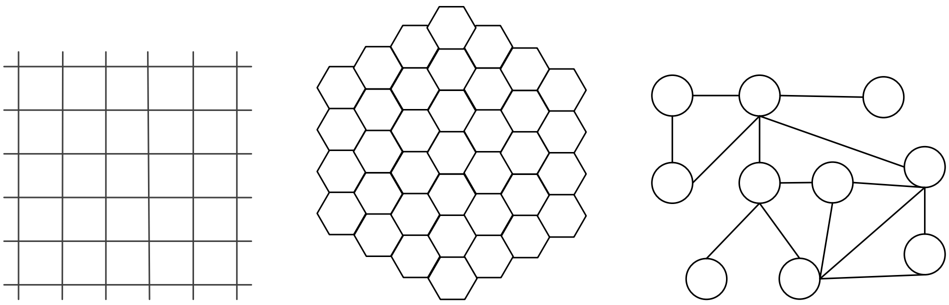
Algorithm 4 presents the pseudo-code of an iteration of the WWO algorithm.

Algorithm 4: WWO procedure for evolving the population P at each iteration

```
Calculate the wavelengths of all solutions in the population;
foreach z ∈ P do
    Let Q′ = rand(1, λ(z));
    Produce an offspring z′ by randomly exchanging a one-component and a
      zero-component of z for Q′ times;
    if g(z′) < g(z) then
        z ← z′;
        if z is a new best known solution then
            for k = 1 to k_N do
                Produce a neighbor z′ by performing a one-zero exchange on z;
                if g(z′) < g(z) then
                    z ← z′;
Update the population size;
if the population size is reduced by one then
    Remove the worst solution from P;
```

4. Experiments

We test the performance of the proposed evolutionary algorithms on 12 instances, the search regions of which are based on three types of structures: grid, honeycomb, and general graph, as shown in Figure 4. For each structure type, we generate two search regions with different numbers of sub-regions; for each search region, we generate two instances with different numbers of UAVs and false targets. The basic information of the 12 instances (numbered by #1–#12) is given in Table 1.



(**a**) Grid (**b**) Honeycomb (**c**) General graph

Figure 4. Three types of structures of search regions for constructing the test instances.

Table 1. Summary of the test instances of the UAV detection problem/false target jamming problem.

Structure	m	$n \times Q$
Grid	64	#1 (3×2), #2 (8×4)
	225	#3 (3×3), #4 (5×5)
Honeycomb	52	#5 (3×2), #6 (5×4)
	331	#7 (5×4), #8 (10×8)
General graph	50	#9 (3×2), #10 (5×3)
	201	#11 (5×5), #12 (10×8)

On each test instance, we first run the hyper-heuristic and memetic algorithms [23,24] to solve the UAV detection problem without false targets, and record the top ten solutions in terms of the objective function (5); then, we use the four evolutionary algorithms together with two simple methods to solve the adversarial problem of false target jamming against UAV detection: the first method (denoted by Rand) randomly selects sub-regions without real targets to place false targets; the second method (denoted by Even) always aims to distribute all (real and false) targets throughout the whole search region as evenly as possible. For the four evolutionary algorithms, we tune their control parameters on the whole test set, the results of which are shown in Table 2. To ensure a fair comparison, the evolutionary algorithms use the same stopping condition that the number of objective function evaluations reaches $1000nQ$. Each evolutionary algorithm runs 30 times on each instance, and the median objective function value and standard deviation over the 30 runs are recorded.

Table 2. Parameter setting of the four evolutionary algorithms tuned on the 12 false target jamming problem instances.

Algorithm	Parameters
GA	$r_c = 0.93$, $r_m^{\max} = 0.12$
PSO	$c_1 = c_2 = 2.0$, $w_{\max} = 0.9$, $w_{\min} = 0.4$
EBO	$\eta_{\max} = 0.7$, $\eta_{\min} = 0.4$
WWO	$NP_{\max} = 45$, $NP_{\min} = 6$

Table 3 presents the experimental results, where the "UAV" column gives the median objective function value of the top ten UAV detection solutions without false targets, in order to show to which degree the false target jamming solutions deteriorate the UAV detection solutions. In the last four columns, the values in parentheses are standard deviations of the results of the evolutionary algorithms. For each instance, the best median value among the comparative algorithms is shown in bold. We also conduct nonparametric Wilcoxon rank sum test to compare the statistical difference among the evolutionary algorithms, and use a superscript '+' to denote that the best result is more statistically significantly than the result of the corresponding comparative algorithm (at a confidence level of 95%).

The experimental results show that, regarding total detection reward, using false targets can effectively decrease the UAV detection efficiency in terms of (5); however, the effectiveness heavily depends on the false target placement solutions obtained by the algorithms for the adversarial problem. Obviously, the solutions obtained by randomly placing false targets are of low quality, making the total detection awards of UAVs become around 75%∼90% of those without false targets. The quality of the solutions obtained by evenly distributing false targets is slightly better than those obtained by Rand, but the difference is not significant. Compared to the two simple methods, the four evolutionary algorithms based on the proposed evolutionary framework significant reduce the total detection rewards of UAVs:

- For instances using two or three false targets, the total detection rewards of UAVs become around 30% of those without false targets;

- For instances using four or five false targets, the total detection rewards of UAVs become around 20∼30% of those without false targets;
- For instances #8 and #12, using eight false targets, the total detection rewards of UAVs become around 10∼15% of those without false targets.

Table 3. Comparative results on the 12 test instances.

Instances	UAV	Rand	Even	GA	PSO	EBO	WWO
#1	0.406	0.309	0.270	$^{+}$0.224 (0.043)	**0.196** (0.022)	0.206 (0.031)	0.208 (0.021)
#2	0.432	0.325	0.306	$^{+}$0.198 (0.036)	0.177 (0.032)	$^{+}$0.191 (0.031)	**0.166** (0.024)
#3	0.130	0.118	0.114	$^{+}$0.074 (0.014)	0.070 (0.017)	0.070 (0.015)	**0.068** (0.009)
#4	0.154	0.136	0.120	$^{+}$0.052 (0.015)	0.046 (0.009)	0.044 (0.011)	**0.043** (0.006)
#5	0.268	0.223	0.204	$^{+}$0.120 (0.022)	**0.109** (0.017)	$^{+}$0.125 (0.016)	0.111 (0.008)
#6	0.314	0.277	0.246	$^{+}$0.106 (0.018)	0.093 (0.014)	$^{+}$0.100 (0.018)	**0.091** (0.016)
#7	0.095	0.078	0.067	$^{+}$0.036 (0.006)	0.031 (0.007)	0.030 (0.005)	**0.030** (0.005)
#8	0.138	0.121	0.112	$^{+}$0.023 (0.007)	$^{+}$0.019 (0.004)	$^{+}$0.019 (0.005)	**0.017** (0.002)
#9	0.182	0.157	0.153	0.120 (0.039)	**0.117** (0.027)	$^{+}$0.124 (0.030)	0.118 (0.016)
#10	0.222	0.201	0.203	$^{+}$0.126 (0.035)	0.113 (0.026)	**0.112** (0.018)	0.116 (0.020)
#11	0.067	0.052	0.049	$^{+}$0.016 (0.004)	$^{+}$0.015 (0.003)	0.013 (0.003)	**0.012** (0.003)
#12	0.096	0.080	0.071	$^{+}$0.009 (0.003)	$^{+}$0.011 (0.002)	$^{+}$0.009 (0.002)	**0.007** (0.001)

Among the four evolutionary algorithms, WWO exhibits the best performance, as it obtains the best solutions on eight test instances, which are typically with large search areas and a relatively large number of false targets, demonstrating that WWO is very efficient in searching large-size solution spaces of the adversarial problem of false target jamming. For the remaining four instances, according to the statistical tests, there are no significant differences between the results of WWO and the best results. PSO obtains the best solutions on three test instances, which are with relatively small n and Q values, demonstrating that PSO converges fast in searching small-size solution spaces. Comparatively, the performance of EBO is relatively poor for small-size instances, but is relatively good for large-size instances; in particular, the performance of EBO is the best for instance #10 among the four evolutionary algorithms. The general performance of GA is the worst among the evolutionary algorithms, mainly because its crossover operations easily lead to local optima. A general conclusion of the results is that, when solving small-size instances, PSO is preferred; for medium- and large-size instances, WWO can produce high-quality solutions.

5. Conclusions and Discussion

This paper studies a false target jamming problem from the viewpoint of opponents of UAVs, and aims to optimize the placement of false targets to decrease the total detection reward of UAVs. Currently, studies on this problem and its algorithms are few. We formulate the false target jamming problem based on the counterpart UAV detection problem, and propose an evolutionary framework that evolves a false target placement solution evaluated according to its adversarial effects on a set of possible UAV detection solutions. We implemented the framework using four popular evolutionary algorithms. Experimental results demonstrate that the proposed approach significantly reduces the UAV detection reward; among the four evolutionary algorithms, WWO exhibits the best overall performance, while PSO converges fast in solving small-size instances. We believe that the proposed framework can be extended to more countermeasures (including active defense and passive defense measures) against more UAV operations (including surveillance, search and rescue, attacking, etc.).

The present research aims to optimize the placement of false targets before the invasion of UAVs, which is a static optimization problem. In practice, the number of UAVs and their search strategies can dynamically change during the search and detection process, and we are now studying the dynamic optimization of false target jamming (e.g., using electromagnetic signals to camouflage false targets to deceive UAV airborne radars), using real-time optimization methods such as reinforcement learning [30]. Our future work will study the combination of false target jamming and air-defense weapons against UAV detection, which can be a considerably more effective yet more challenging approach.

Author Contributions: Conceptualization, Z.-L.S. and H.-F.L.; methodology, Y.-J.Z. and X.-L.J.; validation, X.-L.J. and Z.-L.S.; investigation, Z.-L.S. and N.L.; writing—original draft preparation, Z.-L.S. and N.L.; writing—review and editing, X.-L.J. and Y.-J.Z.; project administration, H.-F.L.; funding acquisition, Y.-J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by National Natural Science Foundation of China under Grant 61872123 and Natural Science Foundation of Zhejiang Province (Outstanding Youth Foundation) under Grant LR20F030002.

Conflicts of Interest: The authors declare that they have no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BBO	biogeography-based optimization
EBO	ecogeography-based optimization
GA	genetic algorithm
PSO	particle swarm optimization
UAV	unmanned aerial vehicle
WWO	water wave optimization

References

1. Bai, T.; Wang, D. Cooperative trajectory optimization for unmanned aerial vehicles in a combat environment. *Sci. China Inf. Sci.* **2019**, *62*, 10205. [CrossRef]
2. Xia, Y.; Batta, R.; Nagi, R. Controlling a fleet of unmanned aerial vehicles to collect uncertain information in a threat environment. *Oper. Res.* **2017**, *65*, 674–692. [CrossRef]
3. Hayat, S.; Yanmaz, E.; Brown, T.X.; Bettstetter, C. Multi-objective UAV path planning for search and rescue. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 5569–5574. [CrossRef]
4. Du, Y.; Zhang, M.; Ling, H.; Zheng, Y. Evolutionary planning of multi-UAV search for missing tourists. *IEEE Access* **2019**, *7*, 73480–73492. [CrossRef]
5. Khoufi, I.; Laouiti, A.; Adjih, C. A Survey of Recent Extended Variants of the Traveling Salesman and Vehicle Routing Problems for Unmanned Aerial Vehicles. *Drones* **2019**, *3*, 66. [CrossRef]
6. Altshuler, Y.; Yanovsky, V.; Wagner, I.A.; Bruckstein, A.M. Efficient cooperative search of smart targets using UAV swarms. *Robotica* **2008**, *26*, 551–557. [CrossRef]
7. Zhou, R.; Feng, Y.; Di, B.; Zhao, J.; Hu, Y. Multi-UAV cooperative target tracking with bounded noise for connectivity preservation. *Front. Inform. Technol. Electron. Eng.* **2020**, *21*, 1494–1503. [CrossRef]
8. Kent, T.; Richards, A.; Johnson, A. Homogeneous Agent Behaviours for the Multi-Agent Simultaneous Searching and Routing Problem. *Drones* **2022**, *6*, 51. [CrossRef]
9. Yu, X.; Gao, X.; Wang, L.; Wang, X.; Ding, Y.; Lu, C.; Zhang, S. Cooperative Multi-UAV Task Assignment in Cross-Regional Joint Operations Considering Ammunition Inventory. *Drones* **2022**, *6*, 77. [CrossRef]
10. Bertuccelli, L.F.; Cummings, M.L. Scenario-based robust scheduling for collaborative human-UAV visual search tasks. In Proceedings of the 2011 50th IEEE Conference on Decision and Control and European Control Conference, Orlando, FL, USA, 12–15 December 2011; pp. 5702–5707. [CrossRef]

11. Zheng, Y.J.; Du, Y.C.; Sheng, W.G.; Ling, H.F. Collaborative human-UAV search and rescue for missing tourists in nature reserves. *INFORMS J. Appl. Analy.* **2019**, *49*, 371–383. [CrossRef]
12. Zheng, Y.; Du, Y.; Ling, H.; Sheng, W.; Chen, S. Evolutionary collaborative human-UAV search for escaped criminals. *IEEE Trans. Evol. Comput.* **2020**, *24*, 217–231. [CrossRef]
13. Csillik, O.; Cherbini, J.; Johnson, R.; Lyons, A.; Kelly, M. Identification of Citrus Trees from Unmanned Aerial Vehicle Imagery Using Convolutional Neural Networks. *Drones* **2018**, *2*, 39. [CrossRef]
14. Fysh, M.C.; Bindemann, M. Person Identification from Drones by Humans: Insights from Cognitive Psychology. *Drones* **2018**, *2*, 32. [CrossRef]
15. Martinez-Alpiste, I.; Casaseca-de-la Higuera, P.; Alcaraz-Calero, J.M.; Grecos, C.; Wang, Q. Smartphone-based object recognition with embedded machine learning intelligence for unmanned aerial vehicles. *J. Field Robot.* **2020**, *37*, 404–420. [CrossRef]
16. Castrillo, V.U.; Manco, A.; Pascarella, D.; Gigante, G. A Review of Counter-UAS Technologies for Cooperative Defensive Teams of Drones. *Drones* **2022**, *6*, 65. [CrossRef]
17. Bo, Z.; Zhou, F.; Shi, X.; Wu, Q.; Zheng, B. Multiple Targets Deception Jamming Against ISAR Using Electromagnetic Properties. *IEEE Sens. J.* **2015**, *15*, 2031–2038. [CrossRef]
18. Zhou, C.; Tang, Z.; Dai, Y.; Li, X. Anti-intermittent sampling repeater jamming method based on convex optimization techniques. In Proceedings of the 2016 CIE International Conference on Radar, Guangzhou, China, 10–13 October 2016; pp. 1–5. [CrossRef]
19. Stanhope, S.A.; Daida, J.M. An individually variable mutation-rate strategy for genetic algorithms. In *Proceedings of the Evolutionary Programming VI*; Angeline, P.J., Reynolds, R.G., McDonnell, J.R., Eberhart, R., Eds.; Springer: Berlin/Heidelberg, Germany, 1997. [CrossRef]
20. Bansal, J.C.; Deep, K. A Modified Binary Particle Swarm Optimization for Knapsack Problems. *Appl. Math. Comput.* **2012**, *218*, 11042–11061. [CrossRef]
21. Zheng, Y.J.; Ling, H.F.; Shi, H.H.; Chen, H.S.; Chen, S.Y. Emergency railway wagon scheduling by hybrid biogeography-based optimization. *Comput. Oper. Res.* **2014**, *43*, 1–8. [CrossRef]
22. Zheng, Y.J. Water wave optimization: A new nature-inspired metaheuristic. *Comput. Oper. Res.* **2015**, *55*, 1–11. [CrossRef]
23. Wang, Y.; Zhang, M.X.; Zheng, Y.J. A hyper-heuristic method for UAV search planning. In *Advances in Swarm Intelligence, Part II*; Tan, Y., Takagi, H., Shi, Y., Niu, B., Eds.; Springer: Berlin/Heidelberg, Germany, 2017; pp. 454–464._48. [CrossRef]
24. Hong, L.; Wang, Y.; Du, Y.; Chen, X.; Zheng, Y. UAV search-and-rescue planning using an adaptive memetic algorithm. *Front. Inf. Technol. Electron. Eng.* **2021**, *22*, 1477–1491. [CrossRef]
25. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948. [CrossRef]
26. Shi, Y.; Eberhart, R.C. A modified particle swarm optimizer. In Proceedings of the IEEE Congress on Evolutionary Computation, Anchorage, AK, USA, 4–9 May 1998; pp. 69–73.
27. Simon, D. Biogeography-Based Optimization. *IEEE Trans. Evol. Comput.* **2008**, *12*, 702–713. [CrossRef]
28. Zheng, Y.J.; Ling, H.F.; Wu, X.B.; Xue, J.Y. Localized biogeography-based optimization. *Soft Comput.* **2014**, *18*, 2323–2334. [CrossRef]
29. Zheng, Y.J.; Zhang, B. A simplified water wave optimization algorithm. In Proceedings of the IEEE Congress on Evolutionary Computation, Sendai, Japan, 25–28 May 2015; pp. 807–813. [CrossRef]
30. Bello, I.; Pham, H.; Le, Q.V.; Norouzi, M.; Bengio, S. Neural combinatorial optimization with reinforcement learning. *arXiv* **2016**, arXiv:1611.09940.