applied sciences

MDPI

Article

Multitarget Search Algorithm Using Swarm Robots in an Unknown 3D Mountain Environment

You Zhou [1], Shaowu Zhou [2,*], Mao Wang [2] and Anhua Chen [3]

[1] College of Electrical and Information Engineering, Hunan Institute of Engineering, Xiangtan 411104, China
[2] College of Information and Electrical Engineering, Hunan University of Science and Technology, Xiangtan 411201, China
[3] College of Mechanical and Electrical Engineering, Hunan University of Science and Technology, Xiangtan 411201, China
* Correspondence: swzhou@hnust.edu.cn; Tel.: +86-165-0732-0111

Abstract: A multitarget search algorithm for swarm robot in an unknown 3D mountain environment is proposed. Most existing 3D environment obstacle avoidance algorithms are potential field methods, which need to consider the location information of all obstacles around the robot, and they easily fall into local optima, and their calculation is complex. Furthermore, they cannot well meet the requirements of real-time obstacle avoidance characteristics of swarm robots in multiobject searches. This paper first focuses on solving the obstacle avoidance problem of swarm robots in mountain environments. A new 3D curved obstacle tracking algorithm (3D-COTA) is designed by discretizing the mountains within the detection range of robot obstacles. Then, a task assignment model and virtual force model in 2D space are extended to 3D, and a particle swarm search model with kinematic constraints is constructed, which considers the kinematic constraints and the limitations of the communication ability of the robots. Finally, a new multitarget search algorithm for swarm robot in an unknown 3D mountain environment is proposed by means of the designed 3D surface obstacle tracking algorithm. Numerical simulation results demonstrate the effectiveness of the proposed algorithm.

Keywords: swarm robot; unknown complex environment; multitarget cooperative search; simplified virtual force model; particle swarm optimization

Citation: Zhou, Y.; Zhou, S.; Wang, M.; Chen, A. Multitarget Search Algorithm Using Swarm Robots in an Unknown 3D Mountain Environment. *Appl. Sci.* **2023**, *13*, 1969. https://doi.org/10.3390/app13031969

Academic Editor: Dimitris Mourtzis

Received: 4 November 2022
Revised: 22 January 2023
Accepted: 28 January 2023
Published: 2 February 2023

1. Introduction

A large number of studies are devoted to swarm robot multitarget search, which is widely used in postdisaster search and rescue, natural resources exploration, enemy position detection, underwater fishing, and other application fields [1]. ZENG et al. mapped particle swarm optimization (PSO) well with the target search process [2]. ZHANG et al. deployed the cooperation and competition to solve the spatial conflicts of swarm robots [3]. LI et al. introduced a probability-constrained finite state machine to effectively resolve individual resource conflicts and improve the efficiency of target search [4]. Taking UAV as the carrier, HE proposed a 3D adaptive inertia weight extended particle swarm optimization (IAEPSO) to realize the search of air targets in a mountain environment [5]. In order to search for the lost object, PHUNG et al. transformed the problem of target search into a probability problem based on the location of the last lost object and the creation of a Bayesian map, and proposed motion-encoded particle swarm optimization (MPSO) [6]. Aiming at the target search of underwater vehicles, CAO et al. proposed a multi-AUV collaborative team integration algorithm, which has the advantages of fewer parameters and no speed jump [7]. In order to reduce the communication pressure of swarm robots, TANG et al. realized the information interaction among swarm robots through indirect communication [8]. Brown et al. assumed that the target was discovered when it was within the detection range of individual UAV, and then proposed an ergodic target search

method; under the background of this method, Brown et al. also proposed an approach to increase or decrease the number of UAV individuals [9,10].

The above research shows that existing research on swarm robot multitarget search is mainly aimed at 2D plane environments or a 3D environment with constant height [11–14]. However, in the practical application of environmental detection and postdisaster rescue, swarm robots may face complex mountain environments. For multitarget searches in 3D environments, many studies have implemented UAVs. For example, Dario [15] proposed a task planning strategy of a UAV swarm in a 3D environment for landslide monitoring and postdisaster search for survivors. Inspired by the gray wolf tracking strategy, Xie Yuxin [16] proposed an adaptive formation tracking control method applied to a UAV swarm system, which improved the system stability and accuracy of formation tracking. Wang [17] customized a UAV interactive decision-making mechanism that could switch the interaction method according to the distance between aircraft during a search for a cooperative UAV swarm search task under the condition of limited communication distance and realized search path planning in a dynamic environment. In view of the realistic environment faced by swarm robots in a targeted search, the premise of their search is to move safely in the task environment, so it is particularly important to consider the obstacle avoidance problem. BinKai Qi [18] proposed UAV path planning based on an improved 1artificial potential field to efficiently solve the UAV obstacle avoidance problem. YuWenqiang [19], in view of the traditional artificial potential field method in a complex environment and the problem of low efficiency of obstacle avoidance, proposed a traditional artificial potential field method as an improved potential field function and improved the traditional spherical potential field for the ellipsoid potential field. The experimental simulation proves that the improved artificial potential field method provides efficient and safe UAV obstacle avoidance path planning in a complex 3D environment.

A UAV has the advantages of information sharing, strong system survivability, and cost performance, which can better meet the needs of a targeted search in 3D space. However, it has some problems for ground targeted search in complex mountain environments. At present, there are few research results on swarm robot targeted ground search in 3D mountain environments. In view of existing 3D environment potential field methods, obstacle avoidance algorithms, the need to consider the obstacle position information around the robot, the complex calculation and ease of falling into local optima, not satisfying swarm robots well in the process of multirobot targeted ground search, and the insufficient real-time obstacle avoidance requirements, this paper proposes a simple 3D curved obstacle tracking algorithm that does not easily fall into local optima.

First, a task assignment model, particle swarm optimization algorithm with kinematic constraints, and a simplified virtual force model in a 2D search environment were extended to 3D space to solve the multiobjective search problem in a 3D scene [20–22]. Then, obstacle tracking was considered in the process of swarm robot completing the task under the condition of different robots according to the kinematic constraint using a particle swarm optimization algorithm, and a virtual force model was simplified to calculate the expected speed after the swarm robot 3D curved obstacle tracking algorithm to realize multitarget search in an unknown complex 3D mountain environment. The simulation results show that the proposed method is an effective method for swarm robots to search for multiple targets in unknown 3D mountain environments.

2. Ground Target Search Modeling in an Unknown Mountain Environment

To better study swarm robot ground target search in an unknown mountain environment, the corresponding environment hypothesis is made, and the corresponding mathematical model is established.

The search task is located in an $a \times b \times c$ mountain environment, which has a horizontal area of $a \times b$ and a height of c. Among them, the mountain height difference is less than c, and the mountain slope changes continuously and is always less than α degrees.

The task object includes the robots, target, and mountain. Robots are represented as set $\boldsymbol{Rob} = \{R_i | i = 1, 2, \ldots, n_u; 30 \leq n_u \leq 100\}$, where R_i represents a robot, and the target is represented as set $\boldsymbol{T} = \{tar_j | j = 1, 2, \ldots, n_T; n_T \geq 1\}$. The mountain is discretized in both the horizontal x- and y-axes with Δl as the unit distance, and the discrete points obtained are called obstacles. Obstacles are represented as set $\boldsymbol{S} = \{obs_k | k = 1, 2, \ldots, n_s; n_s \geq 1\}$. The time t and the locations of R_i, tar_j, and obs_k are represented as $\boldsymbol{R_i}(t)$, $\boldsymbol{tar_j}$, and $\boldsymbol{obs_k}$, respectively, and the speed of R_i is $\boldsymbol{V_i}(t)$.

The Euclidean distance between each individual is expressed as follows: the distance between robots $dr_{i_1,i_2|t} = \|\boldsymbol{R_{i_1}}(t) - \boldsymbol{R_{i_2}}(t)\|$, the distance between a robot and the target $drt_{i,j|t} = \|\boldsymbol{R_i}(t) - \boldsymbol{tar_j}\|$, and the distance between a robot and obstacle $drs_{i,k|t} = \|\boldsymbol{R_i}(t) - \boldsymbol{obs_k}\|$.

Within the task environment, the search task can be described as follows: considering that the target reaches the threshold value d_0, if there is a robot with a target distance $drt_{i,j|t} < d_0$, it indicates that the target is found. When all targets have been found, the target search task is complete.

The robots involved in the search have certain characteristics. Assuming that each robot is isomorphic and the robot velocity $\boldsymbol{V_i}(t)$ satisfies $0 \leq \boldsymbol{V_i}(t) \leq Vm$, each robot can reach any position close to the ground in the task environment. Considering maximum communication distance d_{com}, maximum obstacle sensing distance d_{obs}, and maximum target detection distance d_{tar}, each robot has the following functions: when $drt_{i_1,i_2|t} \leq d_{com}$, it can communicate between robots; when $drt_{i,j|t} \leq d_{tar}$, it can detect the target signal; and when $drs_{i,k|t} \leq d_{obs}$, according to the slope sensor sense obstacles and a robot's relative slope, a robot can climb slopes less than or equal to β and can drive on slopes less than or equal to α without rollover, and $\beta < \alpha$.

The target being searched for is stationary on the mountain surface within the mission mountain environment. When searching for a target, each robot can continuously detect the target signal using a sensor. The target signal and $drt_{i,j|t}$ meet an environmental interference function and should describe the function of the target as a response function [23]. The function can be set as Equation (1):

$$I_{i,j}(t) = \begin{cases} \frac{mQ}{drt_{i,|t}^2} + \eta & drt_{i,j|t} \leq d_{tar} \\ 0 & drt_{i,j|t} > d_{tar} \end{cases} \tag{1}$$

where $I_{i,j}(t)$ represents the target signal detected by R_i at time t; η is the environmental disturbance satisfying the standard normal distribution; $drt_{i,j|t}$ is the objective existence, which is unknown to the robots; m is the attenuation coefficient of the environment $(0 < m < 1)$; and Q is the constant signal power of the target.

The mountain surface is separated into obstacle points with spacing Δl, which are static, and the position of each obstacle point can be specifically expressed as $\boldsymbol{obs_k} = [xs_k, ys_k, zs_k]$.

In a 3D search task environment, each robot can locate itself through its own sensor position and speed information, can communicate through a communication device within the scope of communication with other robots, and can sense obstacle slope information in its detection scope. The robot pose and location information followed within the search environment is as Equation (2):

$$\begin{cases} \boldsymbol{R_i}(t) = \left[xu_{i|t}, yu_{i|t}, zu_{i|t}\right] \\ \boldsymbol{V_i}(t) = \left[\dot{xu}_{i|t}, \dot{yu}_{i|t}, \dot{zu}_{i|t}\right] \\ \dot{xu}_{i|t} = \frac{d(xu_{i|t})}{d(t)} = \|\boldsymbol{V_i}(t)\| \cos\theta \sin\phi \\ \dot{yu}_{i|t} = \frac{d(yu_{i|t})}{d(t)} = \|\boldsymbol{V_i}(t)\| \cos\theta \cos\phi \\ \dot{zu}_{i|t} = \frac{d(zu_{i|t})}{d(t)} = \|\boldsymbol{V_i}(t)\| \sin\theta \end{cases} . \tag{2}$$

where $xu_{i|t}$, $yu_{i|t}$ and $zu_{i|t}$ are the coordinate positions of the robot at time t in the Cartesian global coordinate system $xoyz$. $\boldsymbol{V_i(t)}$ is the movement speed of the robot at time t, ϕ is the angle between the projection vector of $V_i(t)$ in the xoy plane and the forward direction of the x-axis, and θ is the angle between $\boldsymbol{V_i(t)}$ and the forward direction of the z-axis. If the time change Δt is small enough, the relationship between the robot's position and speed in Equation (2) can be expressed as Equation (3):

$$\boldsymbol{R_i}(t + \Delta t) = \boldsymbol{R_i}(t) + \boldsymbol{V_i}(t + \Delta t) \tag{3}$$

To facilitate the planning of the trajectory of the robot, this study takes Δt as unit time, that is, $\Delta t = 1$, and the iterative relationship between the position and velocity of the group robot satisfies the following as Equation (4):

$$\boldsymbol{R_i}(t + 1) = \boldsymbol{R_i}(t) + \boldsymbol{V_i}(t + 1) \tag{4}$$

3. Robot Search Task Assignment Mechanism

3.1. Three Robot States

To make the robot swarm target search more coordinated and efficient, robots are divided into three states as shown in Figure 1: roaming search state, cooperative search state, and task completion state.

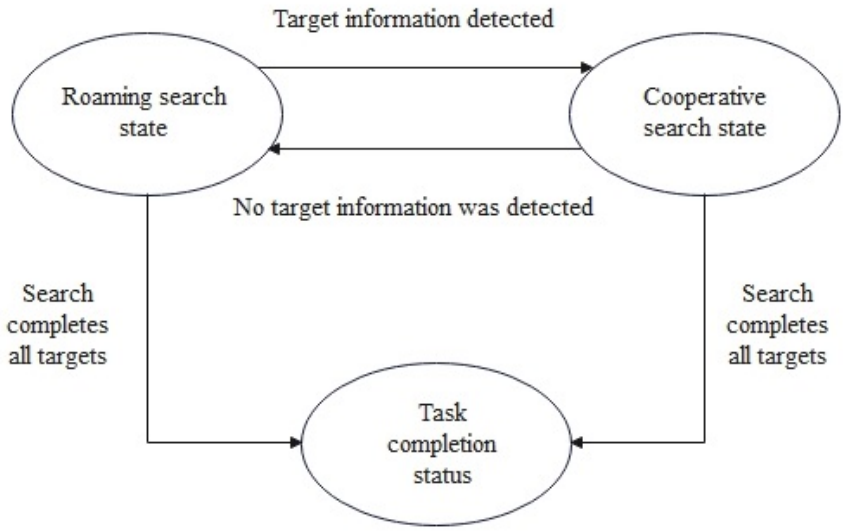


Figure 1. Three robot states relationship.

When the robots do not detect the target information, they are in a roaming search state; that is, the robots repel each other at the maximum speed to rapidly search the global environment [24,25]. When a robot detects the target signal, a multitarget task allocation model based on the response threshold is used to construct a suballiance. The robot members in the same suballiance search for the target corresponding to the suballiance, and the state of the robots forming the suballiance changes to the cooperative search state. When the robot and a target distance are less than the target reached threshold, the robot and the distance of a target $drt_{i,j|t} < d_0$, the target is regarded as a search success, and the robot changes to the task completed state. When all targets are successfully found, all robots change to the mission completed state.

3.2. Robot Task Assignment Model

In the process of task search, each robot participates in a task search process through self-organization and decides whether to choose task tar_1 or task tar_2 and whether to change the task between task selection and task completion. The process is as follows: First,

the sensor detects the target response value of a robot in the detection range. If the robot detects multiple target signals in the detection range, the response probability of the robot to each target is calculated according to a response probability evaluation model, and then a roulette algorithm is used to make a decision regarding which target to search for [26]. The response probability assessment is expressed as Equation (5):

$$p(i,j) = \frac{I_{i,j}^2(t)}{\sum_{k=1}^{m} I_{i,k|t}^2}, \forall j, k = \{1, 2, 3, \cdots, m\} \tag{5}$$

where $I_{i,j}^2(t)$ is the target tar_j signal detected by robot R_i at time t. If the robot detects m target signals within its detection range, the probability of robot R_i responding to target tar_j excitation is $P(i,j)$, as Equation (6):

$$P(i,j) = \sum_{k=1}^{j} p_{ik}, j = 1, 2, \cdots, m \tag{6}$$

When $P(i,j-1) < r_a < P(i,j)$, robot R_i selects target tar_j as the target of collaborative search, and $r_a \in (0,1)$.

Robots can obtain target information in two ways during driving. On the one hand, robots can directly detect target signals through their own sensors, which is called a class I robot. On the other hand, a robot fails to detect a target signal within its detection range but indirectly obtains the signal information of a target through communication with other robots. This kind of robot is called a class II robot [27]. If a target signal detected by two robots is the same target, they can participate in the target search process task together. When multiple robots participate in the same search task, these robots can form a suballiance to carry out a cooperative search for the target.

In the process of searching a 3D task environment, multiple robots will search for the same target, or only a few robots will search for the same target; that is, in the process of forming a suballiance, there will be an uneven distribution of robot resources. To avoid this situation, closed-loop regulation is introduced; that is, the resource allocation of each suballiance is re-evaluated after the first subtask assignment. When the number of members of a suballiance reaches an upper limit N_m, the suballiance preferentially selects N_m robots, and the remaining robots not selected will quit the suballiance and reselect other targets as their search tasks or switch to the roaming search state. When the number of members of a suballiance does not reach the upper limit N_m, suballiance members can be recruited from the surrounding robots to participate in the target task search. The priority principle of suballiance member selection is as follows: the priority of class I targets is greater than that of class II targets; if the priority is the same, the robot is evaluated according to the intensity of the target excitation signal; namely, the higher the intensity of the target signal is, the higher the dominant position. If the number of class I targets is less than N_m, a robot close to the class II communicating robot will be preferentially selected. If there is a robot with the same distance as the communicating robot, the robot with a strong signal will be preferentially selected. See Table 1 for details.

Table 1. Group drones rank the suballiance U_1 members at t = 36.

Serial Number	Robot	Target Type	Intensity of the Response	Nearest Communication Robot	Distance from Communication Robot	Priority Sorting
1	R_2	II	-	R_{14}	213.2349341	11
2	R_3	II	-	R_{19}	209.3224293	9
3	R_5	I	2.099988287	-	-	2
4	R_9	II	-	R_{14}	33.53008801	5
5	R_{11}	II	-	R_5	44.66655953	6
6	R_{14}	I	2.024188002	-	-	3
7	R_{17}	II	-	R_{19}	171.3542868	8
8	R_{18}	II	-	R_{14}	232.4477832	12
9	R_{19}	I	6.13611151	-	-	1
10	R_{21}	II	-	R_5	212.6859702	10
11	R_{22}	II	-	R_{14}	30.39406231	4
12	R_{23}	II	-	R_{19}	142.4618399	7

Swarm robots should not only avoid all obstacles but also complete all target searches in the process of movement. To complete all target search tasks quickly and effectively, the robots can form suballiances to search for the same target together according to the detected target signals and communicate with the surrounding robots. As presented in Table 1, the members of suballiances U_1 are sorted. Robots R_5, R_{14}, and R_{19} detect the signal of target tar_1 during their driving. Robots R_5, R_{14}, and R_{19} are class I robots. At this time, the number of class I robots is less than N_m, and a class I robot recruits the robots within its communication range as a communication robot. R_2, R_3, R_9, R_{11}, R_{17}, R_{18}, R_{21}, R_{22}, and R_{23} receive the recruitment information of class I robots and join one by one in the target tar_1 search task and form suballiance U_1 for this target. According to the principle of selecting members of suballiances, the priority of class I is higher than that of class II. Class I is sorted according to the corresponding intensity of the target.

The higher the target response intensity is, the higher the priority is. The class II robots are sorted according to the distance between them and communication robots. The closer the distance is, the higher the priority is. Therefore, suballiance U_1 is sorted by priority as R_{19}, R_5, R_{14}, R_{22}, R_9, R_{11}, R_{23}, R_{17}, R_3, R_{21}, R_2, and R_{18}. According to the priority order, R_{23}, R_{17}, R_3, R_{21}, R_2, and R_{18} quit the suballiance, and finally, R_{19}, R_5, R_{14}, R_{22}, R_9, and R_{11} form a suballiance and participate in the target tar_1 search task.

4. Multitarget Ground Search Algorithm for Swarm Robots in a 3D Mountain Environment

4.1. 3D Virtual Force Model Roaming Search

When no target signal is obtained, each robot conducts roaming search to quickly detect more areas. Here, a virtual force model is adopted [28]. When the distance between robots is less than $\min(d_{com}, d_{tar})$, the robots repel each other, making the robots spread quickly to quickly evaluate the search area. To simplify the calculation, a robot is repulsed only by the nearest two robots.

Assuming that the robot nearest to robot R_i is R_{i_1} as shown in Figure 2, the positions of the two robots are $\boldsymbol{R_i}(\boldsymbol{t}) = \left[xu_{i|t}, yu_{i|t}, zu_{i|t}\right]$ and $\boldsymbol{R_{i_1}}(\boldsymbol{t}) = \left[xu_{i_1|t}, yu_{i_1|t}, zu_{i_1|t}\right]$. In addition, $dr_{i,i_2|t} \leq \min(d_{com}, d_{tar})$. The repulsive force of R_{i_1} on R_i is calculated using Equation (7), and the repulsive force direction is that the former points to the latter:

$$\boldsymbol{f_{i,i_1}}(\boldsymbol{t}) = \frac{c \cdot l_m^2}{dv_{i,i_1|t}{}^3}\left[\left(xu_{i|t} - xu_{i_1|t}\right), \left(yu_{i|t} - yu_{i_1|t}\right), \left(zu_{i|t} - zu_{i_1|t}\right)\right] \tag{7}$$

where $\boldsymbol{f_{i,i_1}}(\boldsymbol{t})$ is the repulsion of R_{i_1} on R_i at time t. l_m strengthens the obstacle avoidance distance, and c optimizes the robot movement path.

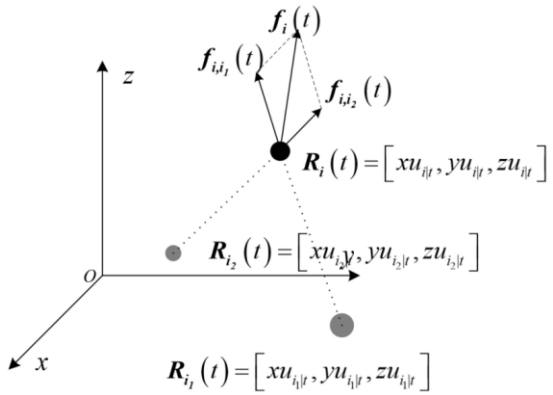


Figure 2. Virtual force model.

If the two robots closest to R_i satisfy $dv_{i,|t} \leq min(d_{com}, 2d_{tar})$, then the virtual force on R_i is as shown in Figure 2. In the figure, $xu_{i_1|t} > xu_{i|t} > xu_{i_2|t}$, $yu_{i_1|t} > yu_{i|t} > yu_{i_2|t}$, $zu_{i|t} > zu_{i_2|t} > zu_{i_1|t}$, and the virtual forces $\boldsymbol{f_{i,i_1}}(\boldsymbol{t})$ and $\boldsymbol{f_{i,i_2}}(\boldsymbol{t})$ satisfy Equation (7). The virtual forces applied to the robot satisfy the vector sum $\boldsymbol{f_i}(\boldsymbol{t}) = \boldsymbol{f_{i,i_1}}(\boldsymbol{t}) + \boldsymbol{f_{i,i_2}}(\boldsymbol{t})$. The speed of the robot in the roaming search state is the direction indicated by the virtual force; that is, the next expected speed of the roaming search is as Equation (8):

$$\boldsymbol{Ve_i}(\boldsymbol{t+1}) = Vm\frac{\boldsymbol{f_i}(\boldsymbol{t})}{\|\boldsymbol{f_i}(\boldsymbol{t})\|} \tag{8}$$

4.2. 3D Particle Swarm Cooperative Search Optimization with Motion Constraints

Swarm robot system is a typical distributed system. Comparing swarm robots with particle swarm optimization [29–31], a mapping relationship is found between the two. The particle swarm optimization algorithm can be applied to robot movement. Considering the movement constraints of a robot and the limitation of its communication ability, a particle swarm optimization model with kinematic constraints can be constructed to calculate the next expected velocity $\boldsymbol{Ve_i}(\boldsymbol{t+1})$. The specific description is as Equation (9):

$$\begin{cases} \boldsymbol{Vp_i}(\boldsymbol{t+1}) = \omega \boldsymbol{V_i}(\boldsymbol{t}) + c_1 r_1(\boldsymbol{R_i}^*(\boldsymbol{t}) - \boldsymbol{R_i}(\boldsymbol{t})) + c_2 r_2(\boldsymbol{g_i}^*(\boldsymbol{t}) - \boldsymbol{R_i}(\boldsymbol{t})) \\ \boldsymbol{Ve_i}(\boldsymbol{t+1}) = \boldsymbol{V_i}(\boldsymbol{t}) + (\boldsymbol{Vp_i}(\boldsymbol{t+1}) - \boldsymbol{V_i}(\boldsymbol{t}))\cdot\lambda \end{cases} \tag{9}$$

where $R_i(\boldsymbol{t})$ and $\boldsymbol{V_i}(\boldsymbol{t})$ represent the velocity and position vectors of robot R_i at time t, respectively; $\boldsymbol{Vp_i}(\boldsymbol{t+1})$ is the velocity obtained by direct particle swarm iteration; $\boldsymbol{Ve_i}(\boldsymbol{t+1})$ is the expected velocity vector of robot R_i at the next moment; the introduction of λ is to consider that the movement of the robot has a certain inertia; c_1 and c_2 are the individual cognitive coefficient and social cognitive coefficient of the robot, respectively; r_1 and r_2 are random variables in the interval (0,1); ω is the inertial weight; $\boldsymbol{R_i}^*(\boldsymbol{t})$ represents the optimal position experienced by robot R_i thus far after joining the current suballiance; and $\boldsymbol{g_i}^*(\boldsymbol{t})$ is the optimal position traversed by the suballiance cutoff time t.

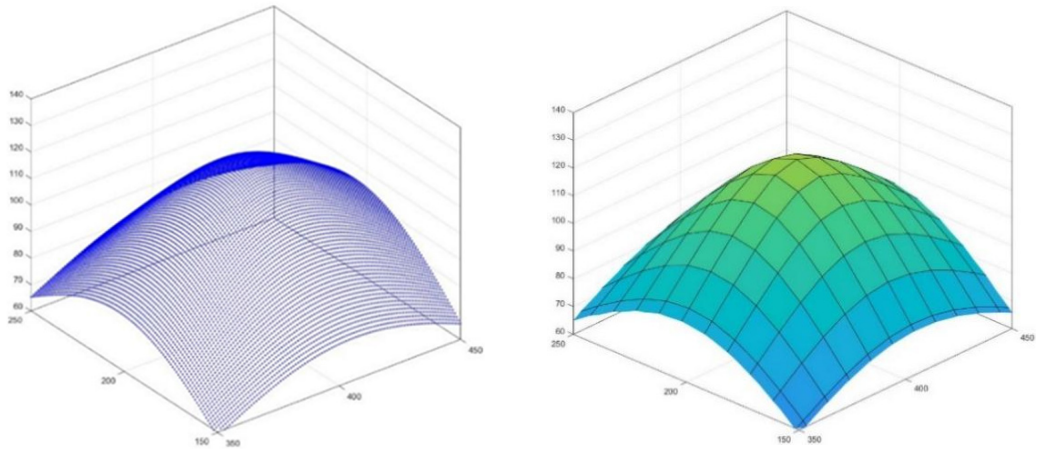
4.3. 3D Curved Obstacle Tracking Algorithm (3D-COTA)

The search for ground targets in a 3D mountain environment is similar to that in a 2D environment. Curving a 2D search environment can obtain a mountain surface. When searching for a target, a robot needs to move along the mountain surface. The mountain surface is curved; therefore, the velocity direction of the robot at any time is the tangent direction of the surface corresponding to its position. Due to the limited climbing ability of the robot, it is necessary to avoid areas with high slopes. After the velocity of the robot in the roaming state or collaborative search is calculated according to the corresponding method, the velocity direction may not meet the speed requirements in the search environment. Therefore, further calculation is required after the expected velocity is obtained through the calculation of robots in different states. To ensure that the next velocity direction is

the tangent direction of the curve, the mountain slope in the velocity direction must meet the requirements.

The robot discretizes the mountains within the detection range of obstacles and considers the discrete point obstacles. For example, the mountain detected by a robot shown in Figure 3a is discretized to obtain Figure 3b. The point set in the figure can be expressed as the obstacle set. The Euclidean distance between the robot and the obstacle can be expressed as Equation (10):

$$drs_{i,k|t} = \|\boldsymbol{R_i(t)} - \boldsymbol{obs_k}\| = \sqrt{\left(xu_{i|t} - xs_k\right)^2 + \left(yu_{i|t} - ys_k\right)^2 + \left(zu_{i|t} - zs_k\right)^2} \tag{10}$$



(**a**) (**b**)

Figure 3. Discretization of a mountain. (**a**) Mountain, (**b**) mountain dispersion.

The slope of the obstacle relative to the robot can be expressed as Equation (11):

$$grd_{i,k|t} = \arctan \frac{zs_k - zu_{i|t}}{\sqrt{\left(xu_{i|t} - xs_k\right)^2 + \left(yu_{i|t} - ys_k\right)^2}} \tag{11}$$

4.3.1. Initial Obstacle Tracking

For robots in the roaming or collaborative search state, the expected velocity $\boldsymbol{Ve_i(t+1)} = \left[\dot{xe}_{i|(t+1)}, \dot{ye}_{i|(t+1)}, \dot{ze}_{i|(t+1)}\right]$ is calculated according to the corresponding method. However, the expected velocity direction may not be tangent to the ground but may point to the air or the ground. Therefore, it is necessary to further calculate the velocity tangent to the ground.

Consider the nearest obstacle and two obstacle points not collinear to the nearest obstacle. In Equation (12), the nearest obstacle point to robot R_i is described as obs_{k_0}.

$$drs_{i,k_0|t} = \min_{obs_k \in \boldsymbol{S}} \left(drs_{i,k|t}\right) \tag{12}$$

Based on obs_{k_0}, two other obstacle points, obs_{k_1} and obs_{k_2}, are found to satisfy the conditions described in Equation (13). According to Equation (13), points $\boldsymbol{obs_{k_0}}$, $\boldsymbol{obs_{k_1}}$, and $\boldsymbol{obs_{k_2}}$ are not collinear, so these three points can determine plane $fl_{i|t}$. For obstacle tracking to search for ground targets, the robot will tend to move parallel to plane $fl_{i|t}$.

$$\begin{cases} \boldsymbol{obs_{k_0}} = \left[xs_{k_0}, ys_{k_0}, zs_{k_0}\right] \\ \boldsymbol{obs_{k_1}} = \left[xs_{k_1}, ys_{k_1}, zs_{k_1}\right] \\ \boldsymbol{obs_{k_2}} = \left[xs_{k_2}, ys_{k_2}, zs_{k_2}\right] \\ xs_{k_1} = xs_{k_0} - \Delta l.\,\mathrm{sign}\left(xs_{k_0} - xu_{i|t}\right) \\ ys_{k_1} = ys_{k_0} \\ xs_{k_2} = xs_{k_0} \\ ys_{k_2} = ys_{k_0} - \Delta l.\,\mathrm{sign}\left(ys_{k_0} - yu_{i|t}\right) \end{cases} \tag{13}$$

At this time, plane $fl_{i|t}$ is shifted so that the resulting plane $fl_{i|t}{}'$ passes through $\boldsymbol{R_i(t)}$, and the coordinate system $\boldsymbol{R_{it}}xyz$ is established with $\boldsymbol{R_i(t)}$ as the origin. In this coordinate system, $\boldsymbol{R_i(t)} = [0, 0, 0]$, and $\boldsymbol{obs_{k_0}}$, $\boldsymbol{obs_{k_1}}$, and $\boldsymbol{obs_{k_2}}$ are expressed as Equation (14):

$$\begin{cases} \boldsymbol{obs'_{k_0}} = \left[\left(xs_{k_0} - xu_{i|t}\right), \left(ys_{k_0} - yu_{i|t}\right), \left(zs_{k_0} - zu_{i|t}\right)\right] \\ \boldsymbol{obs'_{k_1}} = \left[\left(xs_{k_1} - xu_{i|t}\right), \left(ys_{k_1} - yu_{i|t}\right), \left(zs_{k_1} - zu_{i|t}\right)\right] \\ \boldsymbol{obs'_{k_2}} = \left[\left(xs_{k_2} - xu_{i|t}\right), \left(ys_{k_2} - yu_{i|t}\right), \left(zs_{k_2} - zu_{i|t}\right)\right] \end{cases} \tag{14}$$

Under the $\boldsymbol{R_{it}}xyz$ coordinate system, plane $fl_{i|t}{}'$ is determined. Let Equation (15) of the plane be:

$$ax + by + cz = 0 \tag{15}$$

The vector normal to the plane for $\boldsymbol{nl_{i|t}} = [a, b, c]$ is a plane of two known vectors as Equation (16):

$$\begin{cases} \boldsymbol{p_{1,i|t}} = \left[\left(xs_{k_1} - xs_{k_0}\right), \left(ys_{k_1} - ys_{k_0}\right), \left(zs_{k_1} - zs_{k_0}\right)\right] \\ \boldsymbol{p_{2,i|t}} = \left[\left(xs_{k_2} - xs_{k_0}\right), \left(ys_{k_2} - ys_{k_0}\right), \left(zs_{k_2} - zs_{k_0}\right)\right] \end{cases} \tag{16}$$

The normal vector can be obtained as Equations (17) and (18):

$$\boldsymbol{nl_{i|t}} = [a, b, c] = \begin{bmatrix} [1, 0, 0] & [0, 1, 0] & [0, 0, 1] \\ \left(xs_{k_1} - xs_{k_0}\right) & \left(ys_{k_1} - ys_{k_0}\right) & \left(zs_{k_1} - zs_{k_0}\right) \\ \left(xs_{k_2} - xs_{k_0}\right) & \left(ys_{k_2} - ys_{k_0}\right) & \left(zs_{k_2} - zs_{k_0}\right) \end{bmatrix} \tag{17}$$

$$\begin{cases} a = \left(ys_{k_1} - ys_{k_0}\right)\left(zs_{k_2} - zs_{k_0}\right) - \left(ys_{k_2} - ys_{k_0}\right)\left(zs_{k_1} - zs_{k_0}\right) \\ b = \left(zs_{k_1} - zs_{k_0}\right)\left(xs_{k_2} - xs_{k_0}\right) - \left(zs_{k_2} - zs_{k_0}\right)\left(xs_{k_1} - xs_{k_0}\right) \\ c = \left(xs_{k_1} - xs_{k_0}\right)\left(ys_{k_2} - ys_{k_0}\right) - \left(xs_{k_2} - xs_{k_0}\right)\left(zs_{k_1} - zs_{k_0}\right) \end{cases} \tag{18}$$

After the normal vector $\boldsymbol{nl_{i|t}} = [a, b, c]$ of the plane is calculated, the robot begins to move in the direction parallel to plane $fl_{i|t}$, that is, motion tangential to the mountain ground at $\boldsymbol{R_i(t)}$. Considering the obstacle tracking velocity $\boldsymbol{Vo_i}(t+1) = \left[\dot{xo}_{i|(t+1)}, \dot{yo}_{i|(t+1)}, \dot{zo}_{i|(t+1)}\right]$ and expected velocity $\boldsymbol{Ve_i(t+1)} = \left[\dot{xe}_{i|(t+1)}, \dot{ye}_{i|(t+1)}, \dot{ze}_{i|(t+1)}\right]$ of the 3D curved obstacle tracking algorithm, the calculation process of the first obstacle tracking velocity $\boldsymbol{Vo_i}(t+1)$ of the 3D curved obstacle tracking algorithm is as Equation (19):

$$\begin{cases} \dot{x} = \dot{xe}_{i|(t+1)} \\ \dot{y} = \dot{ye}_{i|(t+1)} \\ \dot{z} = -\dfrac{a\dot{xe}_{\,i|(t+1)} + b\,\dot{ye}_{i|(t+1)}}{c} \end{cases} \tag{19}$$

(1) If the robot is in the roaming search state, it can be calculated as Equation (20):

$$\boldsymbol{Vo_i}(t+1) = [\dot{x}, \dot{y}, \dot{z}] \cdot \frac{V_m}{\sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}} \tag{20}$$

(2) If the robot is in the cooperative search state, it can be calculated as Equation (21):

$$\boldsymbol{Vo_i}(t+1) = \begin{cases} [\dot{x}, \dot{y}, \dot{z}] \cdot \dfrac{V_m}{\sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}}, & \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} > Vm \\ [\dot{x}, \dot{y}, \dot{z}], & \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} \leq Vm \end{cases} \tag{21}$$

4.3.2. Second-Obstacle Tracking

After $\boldsymbol{Vo_i}(t+1)$ is calculated using the corresponding method for the robot in the roaming or cooperative search state, the speed direction is adjusted according to the slope of the surrounding mountains, avoiding the movement direction of the mountain slope beyond the robot climbing ability range.

It is assumed that, according to the perception of the slope sensor, the distance near the robot is less than $\|\boldsymbol{Vo_i}(t+1)\|$, and in the direction of angle set $\boldsymbol{\Theta}$, the slope exceeds the climbing ability of the robot; that is, the slope is greater than β.

Within the set $\boldsymbol{\Phi} = \left\{ \frac{2\pi}{n_\phi} n \middle| n \in \left[-\frac{n_\phi}{2}, \frac{n_\phi}{2} \right] \bigcap \boldsymbol{Z} \right\}$, the sensor can identify mountain slopes in the n_ϕ angle directions, where $\left[-\frac{n_\phi}{2}, \frac{n_\phi}{2} \right]$ represents the set of numbers between $-\frac{n_\phi}{2}$ and $\frac{n_\phi}{2}$, and $\boldsymbol{Z}$ is the set of integers.

Suppose that the function $F(\varphi)$ has the following expression as Equation (22):

$$F(\varphi) = \varphi - 2\pi \cdot \mathrm{sgn}(\varphi) \cdot \delta(|\varphi| - \pi) \quad -2\pi < \varphi < 2\pi \tag{22}$$

Among them:

$$\mathrm{sgn}(\varphi) = \begin{cases} -1 & \varphi < 0 \\ 0 & \varphi = 0 \\ 1 & \varphi > 0 \end{cases} \tag{23}$$

$$\delta(\varphi) = \begin{cases} 0 & \varphi \leq 0 \\ 1 & \varphi > 0 \end{cases} \tag{24}$$

The second obstacle tracking velocity is denoted as $\boldsymbol{Vt_i}(\boldsymbol{t+1}) = \left[\dot{xt}_{i|(t+1)}, \dot{yt}_{i|(t+1)}, \dot{zt}_{i|(t+1)} \right]$, and $\boldsymbol{Vo_i}(t+1) = \left[\dot{xo}_{i|(t+1)}, \dot{yo}_{i|(t+1)}, \dot{zo}_{i|(t+1)} \right]$. Subsequently, $\boldsymbol{Vt_i}(\boldsymbol{t+1})$ is calculated as:

(1) If $\arctan\left(\frac{\dot{yo}_{i|(t+1)}}{\dot{xo}_{i|(t+1)}} \right) + \delta\left(-\dot{xo}_{i|(t+1)} \right) \cdot \mathrm{sgn}\left(\dot{yo}_{i|(t+1)} \right) \cdot \pi \notin \boldsymbol{\Theta}$

$$\boldsymbol{Vt_i}(\boldsymbol{t+1}) = \boldsymbol{Vo_i}(\boldsymbol{t+1}) \tag{25}$$

(2) If $\arctan\left(\frac{\dot{yo}_{i|(t+1)}}{\dot{xo}_{i|(t+1)}} \right) + \delta\left(-\dot{xo}_{i|(t+1)} \right) \cdot \mathrm{sgn}\left(\dot{yo}_{i|(t+1)} \right) \cdot \pi \in \boldsymbol{\Theta}$

Calculation angle:

$$\varphi_0 = \min_{\varphi \in \boldsymbol{\Phi}, \varphi_n \in (\boldsymbol{\Phi} - \boldsymbol{\Theta})} |F(\varphi - \varphi_n)| \tag{26}$$

To calculate $\boldsymbol{Vt_i}(\boldsymbol{t+1})$:

$$\begin{cases} \dot{xt}_{i|(t+1)} = \left(\sqrt{\dot{xo}_{i|(t+1)}^2 + \dot{xo}_{i|(t+1)}^2} \right) cos(\varphi_0) \\ \dot{yt}_{i|(t+1)} = \left(\sqrt{\dot{xo}_{i|(t+1)}^2 + \dot{xo}_{i|(t+1)}^2} \right) sin(\varphi_0) \\ \dot{zt}_{i|(t+1)} = -\dfrac{a\ \dot{xt}_{i|(t+1)} + b\ \dot{yt}_{i|(t+1)}}{c} \end{cases} \tag{27}$$

a, b, and c are shown in Equation (18).

4.4. Robot Velocity and Position Iteration

When the robot moves on the mountain ground, the speed of the robot is along the tangent direction of the mountain surface at all times. When planning the path of swarm robots, there is a time interval between each iteration, and at the time between the two iterations, the velocity is also along the tangent direction of the mountain surface. Therefore, the position of the robot needs to be modified when updating its position. According to Equation (4), the velocity is corrected as the average velocity vector before the position is corrected.

It is assumed that the mapping relationship between coordinates $obs_k = (xs_k, ys_k, zs_k)$, zs_k, xs_k, and ys_k of the mountain surface is expressed as $zs_k = \mathrm{Fs}(xs_k, ys_k)$. The robot calculates velocity $\boldsymbol{V_i(t+1)}$ according to $\boldsymbol{Vt_i(t+1)} = \left[\dot{xt}_{i|(t+1)}, \dot{yt}_{i|(t+1)}, \dot{zt}_{i|(t+1)}\right]$. For robot R_i, the next velocity $\boldsymbol{V_i(t+1)}$ is calculated as follows:

$$\boldsymbol{V_i(t+1)} = \left[\dot{xt}_{i|(t+1)}, \dot{yt}_{i|(t+1)}, \mathrm{Fs}\left(\dot{xt}_{i|(t+1)} + xu_{i|t}, \dot{yt}_{i|(t+1)} + yu_{i|t}\right) - zu_{i|t}\right] \tag{28}$$

After calculating $\boldsymbol{V_i(t+1)}$, the robot position is updated as follows:

$$\boldsymbol{R_i(t+1)} = \boldsymbol{R_i(t)} + \boldsymbol{V_i(t+1)} \tag{29}$$

In summary, the multitarget ground search process in an unknown mountain environment is shown in Figure 4.

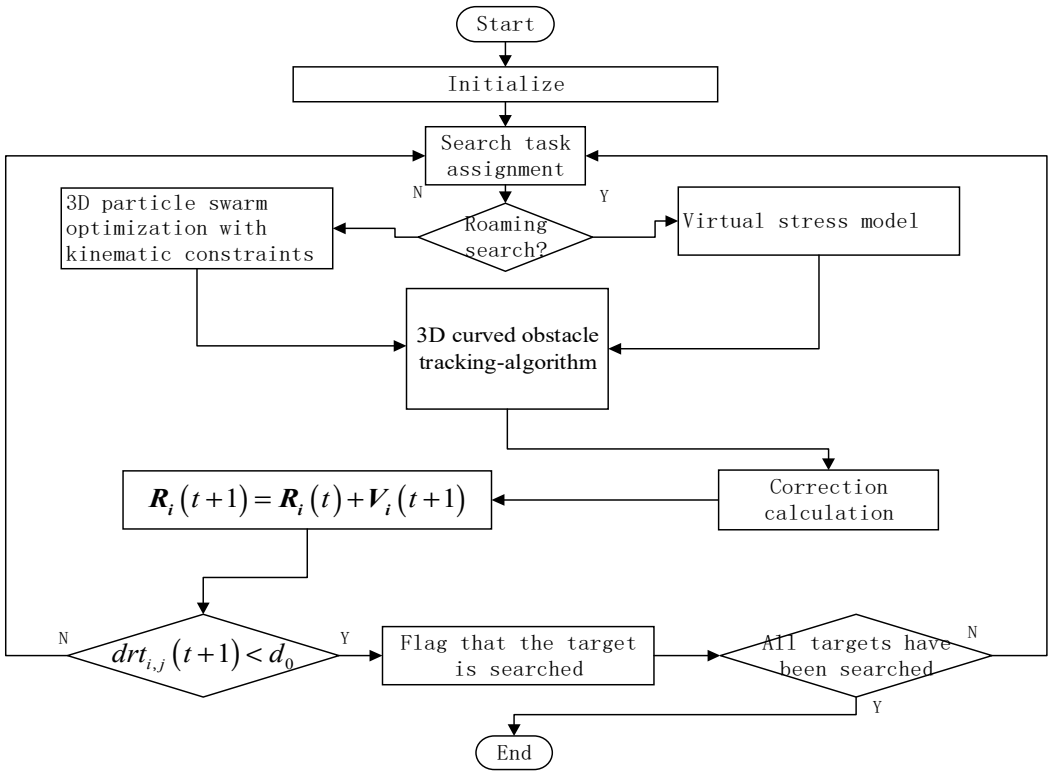


Figure 4. Target search process.

5. Simulation Experiment and Results

The parameters are set based on actual search requirements, as shown in Table 2.

Table 2. Parameter values.

Parameter	Value	Parameter	Value
α	40°	β	30°
n_u	30~60	N_m	6
n_T	10	m	0.1
n_ϕ	360	Δl	1
Vm	10	Q	10^5
r_{com}	300	c_1	1
r_{obs}	100	c_2	1.2
r_{tar}	100	ω	0.5
d_0	10	λ	0.1

As an example, when $n_u = 40$, assuming that slopes in the mountain environment are all less than or equal to β, a schematic of the target search process is shown in Figure 5. Figure 5a is a topographic map of the mountain area for target search. Figure 5b is the top view of the search area when t = 1, the robot is in a 100 × 100 area in the lower left corner, and the target is in an 800 × 800 area in the middle of the horizontal direction.

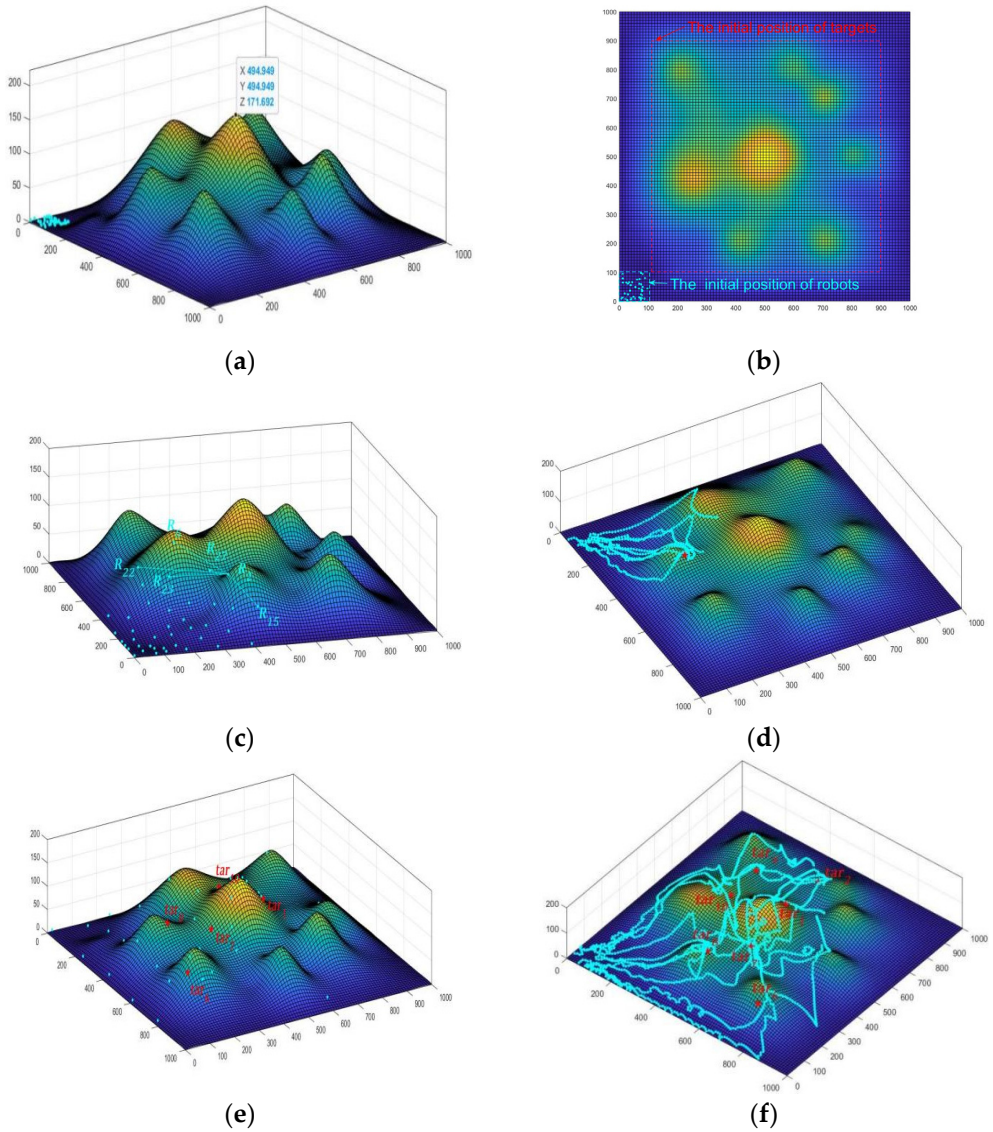
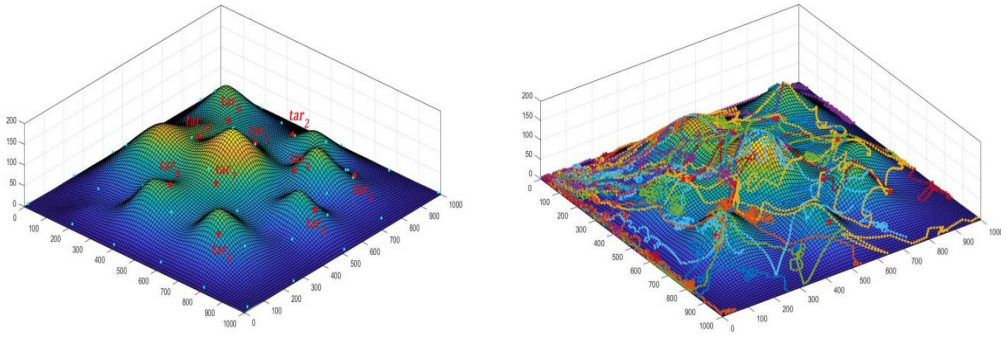


Figure 5. *Cont.*



(**g**) (**h**)

Figure 5. 3D search simulation diagram. (**a**) Target search terrain, (**b**) When t = 1, the robot and the target are located, (**c**) When t = 37, a robot detects a target signal and forms a suballiance. (**d**) When t = 1~58, a robot searches the trajectory of target tar_8. (**e**) When t = 127, the target is found. (**f**) When t = 185, the trajectory of the robot. (**g**) When t = 255, all targets have been found. (**h**) When t = 1–255, the trajectories of all the robots.

In Figure 5c, R_2 detects the signal of target tar_8, and robots R_6, R_{15}, R_{22}, R_{23}, and R_{27} join the suballiance. In Figure 5d, from $t = 1 \sim 58$, R_2, R_6, R_{15}, R_{22}, R_{23}, R_{27}, R_8, R_{10}, R_{11}, and R_{17} in the suballiance participate in the search for tar_8, and R_{10} searches for R_{10} at t = 58.

In Figure 5e, when t = 127, the robots successfully complete the search for target tar_1. By this time, 5 targets (tar_1, tar_7, tar_6 and tar_8) have been found.

In Figure 5f, t = 1~185, 11 robots including, R_5, R_7, R_{11}, R_{17}, R_{18}, R_{19}, R_{20}, R_{22}, R_{24}, R_{28}, and R_{32}, participate in the collaborative search for target tar_2, and before this, multiple robots participated in the cooperative search for other targets.

In Figure 5g, t = 255, the robot swarm finally found all 10 targets. In Figure 5h, all robot movement tracks of swarm robots in the search for targets are shown, and the robots successfully found all ground targets.

Taking the number of robots $n_u = 30,\ 40,\ 50,$ or 60 and the number of targets $n_T = 10$, the experiment was repeated 30 times, and the following data as shown in Table 3.

After verifying the effectiveness of the swarm robot target search in a mountain environment with a slope less than or equal to β, the existence of an environment with a slope greater than β in a mountain environment is verified. Assuming $n_u = 40$, the mountain slope is less than or equal to α. A diagram of the target search process is shown in Figure 6.

Figure 6a shows a mountain topographic map for the target search and the positions of the targets and robots when $t = 1$. Figure 6b is a top view of the search area. The areas marked in red indicate that each location within the region has a slope greater than β in one direction.

Figure 6c shows the process of searching for target tar_6. When $t = 48$, robot R_{37} detects the target signal of tar_6 and forms a suballiance with robots R_{27}, R_{30}, R_{40}, R_{23}, and R_{17}. The suballiance starts to search for target tar_6. At $t = 53$, R_{21} also detects the target signal of tar_6, joins the suballiance, and pushes R_{23} out of the suballiance. Finally, when $t = 62$, the target is found, and the suballiance is dissolved. It can be seen from the figure that when a robot is searching for a target, it can move in a direction with a smaller slope according to the 3D curved obstacle tracking algorithm and then smoothly search for a target in a region with a higher slope.

Figure 6d shows the trajectory of robot R_{21} during the period from $t = 1$ to the robot swarm finding all targets. As seen from the trajectory shown in the figure, when the upward slope of the robot's movement direction is too high for it to climb, the robot will adjust its movement direction to the climbing slope according to the 3D curved obstacle tracking algorithm and move as close to the original direction as possible.

Figure 6e shows the positions of the robots and targets at $t = 329$. All targets have been found by swarm robots at this point.

Figure 6f shows the trajectories of all robots in the swarm robot target search process. The figure shows that according to the proposed method, the robots can successfully find all targets in the task area. The robots will be more inclined to move in the region with a slower slope, but they can also move in the direction with a lower slope in a region with a higher slope.

Table 3. Target n_T = 10 and mountain slope less than or equal to β: the number of steps and energy consumption required for the robots to complete the task search.

n_u	Step				Energy Consumption ($\times 10^4$)			
	30	**40**	**50**	**60**	**30**	**40**	**50**	**60**
Data from 30 experiments	481	250	226	217	11.494	8.191	9.109	10.761
	370	283	220	215	8.857	9.197	8.868	10.483
	332	343	271	211	8.277	10.653	10.668	10.429
	455	249	232	219	11.385	8.128	9.474	10.784
	354	247	234	230	8.729	7.799	9.483	11.259
	287	249	242	205	6.956	7.951	9.887	10.260
	356	267	253	216	8.492	8.515	10.152	10.685
	286	230	245	190	7.085	7.466	9.694	9.456
	282	237	260	217	7.008	7.487	10.235	10.699
	311	215	207	227	7.611	7.024	8.480	10.972
	367	297	235	209	9.088	9.579	9.434	10.453
	282	232	235	194	6.962	7.501	9.244	9.474
	316	260	222	230	7.767	8.350	8.954	11.185
	343	295	244	206	8.513	9.532	9.731	10.289
	272	277	225	200	6.736	8.732	8.992	9.941
	240	248	227	220	5.883	8.009	9.217	10.916
	360	262	207	227	9.011	8.231	8.460	11.091
	294	280	244	195	7.068	8.923	9.709	9.728
	379	299	230	216	9.444	9.568	9.310	10.643
	355	269	239	225	8.679	8.590	9.537	10.901
	336	218	236	196	8.364	6.932	9.666	9.756
	253	352	275	221	6.225	10.924	10.894	10.800
	358	255	203	175	8.912	8.152	8.373	8.802
	259	239	227	229	6.288	7.729	9.191	11.157
	336	244	219	217	8.295	7.646	8.863	10.754
	457	251	260	216	11.264	8.030	10.370	10.591
	328	259	243	190	8.016	8.419	9.658	9.565
	261	329	234	203	6.615	10.298	9.465	9.996
	305	263	218	222	7.376	8.497	8.877	11.051
	340	280	227	206	8.399	8.946	9.098	10.251
Mean	**331.833**	**265.967**	**234.667**	**211.467**	**8.160**	**8.500**	**9.436**	**10.438**

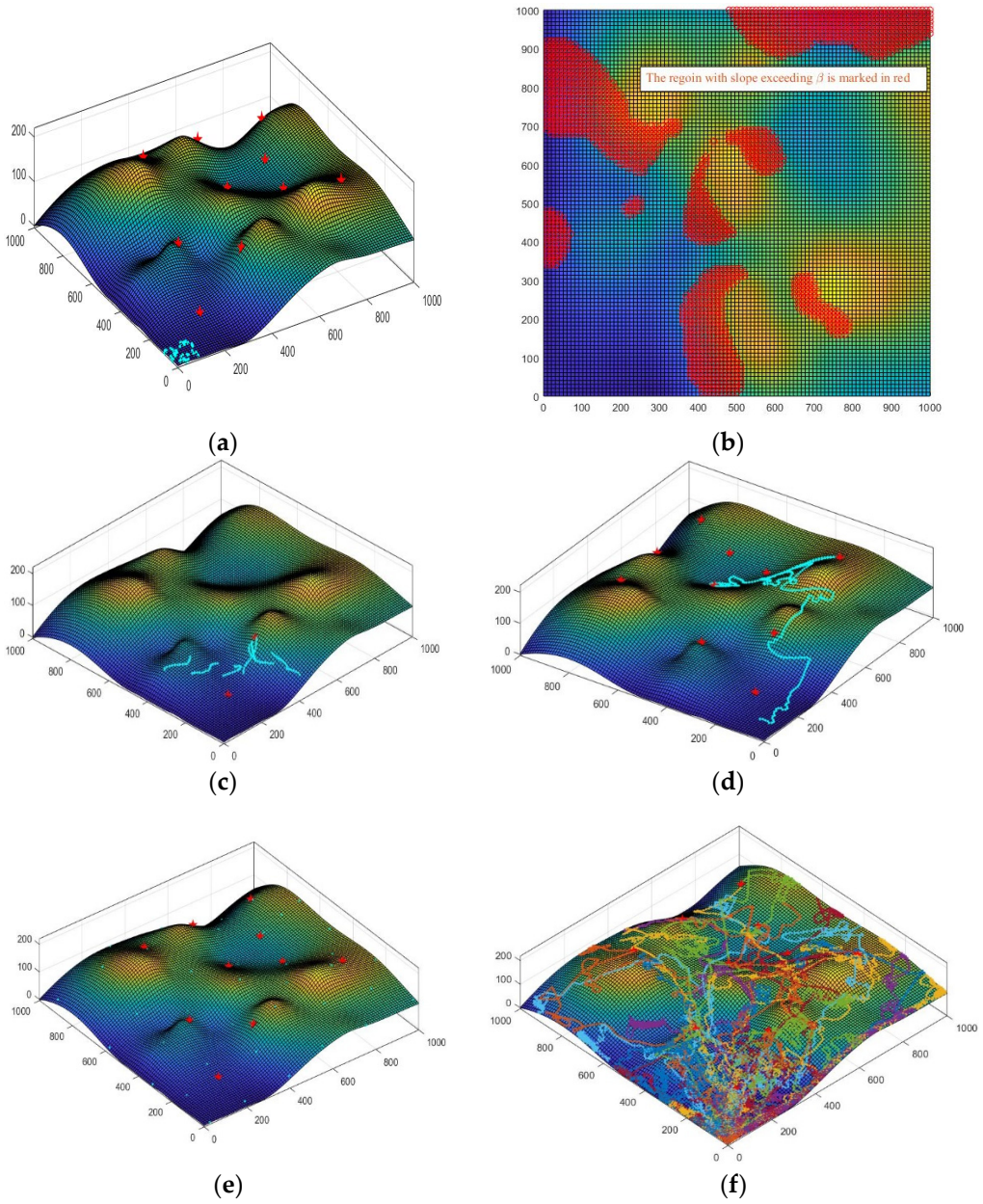


Figure 6. 3D simulation of a target search when the slope is greater than β. (**a**) Mountainous terrain, initial target positions. (**b**) Diagram of the slope over the β zone. (**c**) The robots search for tar_6. (**d**) Trajectory of robot R_{21} (**e**) When $t = 329$, the swarm robots have found all targets. (**f**) The trajectories of all the robots.

Taking the number of robots $n_u = 30, 40, 50,$ or 60 and the number of targets $n_T = 10$, the experiment was repeated 30 times, and the following data as shown in Table 4.

Table 4. Target n_T = 10 and mountain slope greater than β: the number of steps and energy consumption required for the robots to complete the task search.

n_u	Step				Energy Consumption ($\times 10^4$)			
	30	**40**	**50**	**60**	**30**	**40**	**50**	**60**
Data from 30 experiments	598	328	452	289	16.480	12.548	21.803	16.916
	545	327	402	533	15.366	12.534	19.240	31.185
	464	401	397	370	12.544	15.245	19.237	21.520
	426	424	292	296	12.164	15.746	14.237	17.282
	496	311	461	544	13.248	11.843	22.182	31.659
	354	400	286	348	10.087	15.227	13.567	20.232
	430	334	456	336	12.090	12.779	21.769	19.447
	368	339	478	248	10.418	12.907	22.795	14.466
	425	508	385	311	11.934	19.194	18.640	18.117
	477	521	342	269	13.494	19.431	16.561	15.709
	436	376	366	310	12.268	14.428	17.494	18.032
	428	391	352	293	11.946	14.865	16.995	16.946
	377	515	650	307	10.343	19.732	30.874	17.752
	506	454	316	387	14.005	17.052	15.301	22.504
	528	452	310	313	14.284	17.315	15.036	18.232
	436	334	434	282	12.299	12.849	20.916	16.437
	502	366	306	393	13.941	13.977	14.680	22.798
	513	531	514	360	14.101	20.287	24.730	20.971
	425	353	269	291	11.521	13.482	12.969	16.984
	474	321	484	257	13.410	12.232	23.167	14.847
	520	586	403	379	14.135	21.730	19.310	21.910
	503	344	380	331	13.953	13.084	18.299	19.251
	346	323	421	410	9.752	12.457	20.163	23.883
	296	375	386	313	8.25	14.396	18.335	18.105
	638	376	342	399	17.944	14.457	16.606	23.201
	447	529	465	349	12.583	19.217	21.740	20.366
	428	321	369	383	11.826	12.105	17.919	22.415
	348	341	364	358	9.813	12.914	17.613	20.917
	509	531	308	538	14.502	20.242	14.828	30.778
	564	523	353	409	15.457	19.799	16.802	23.864
Mean	**460.233**	**407.833**	**391.433**	**353.533**	**12.805**	**15.469**	**18.794**	**20.558**

6. Conclusions

Aiming at the problem of robot swarm multitarget ground search in an unknown 3D mountain environment, this paper, based on unknown 2D environment robot swarm multiobject search research, extends the multiobjective task assignment model, particle swarm optimization algorithm, and virtual force model from a 2D environment to a 3D environment. A new multiobject ground search algorithm for swarm robots in a 3D mountain environment is proposed. Aiming at the problems of swarm robot's speed direction being tangent to the ground, each robot avoids a steep slope that cannot be climbed, and a 3D curved obstacle tracking algorithm that can effectively avoid conflict between the swarm robots and the mountain plans the speed based on the direction tangent to the 3D surface so that a robot can find the ground targets in a mountain environment more quickly and effectively. A 3D particle swarm optimization algorithm with kinematic constraints and a multiobjective task assignment model is used to complete multitarget search in the swarm robot system. A virtual force model is used to calculate the expected velocity during roaming search. During collaborative search, a 3D particle swarm optimization algorithm is used to calculate the expected velocity. After the expected velocity is calculated, the final planned velocity is calculated according to the 3D curved obstacle tracking algorithm. Simulation results show that the proposed method can not only find targets quickly but also avoid conflict effectively. The simulation results demonstrate the effectiveness of the proposed algorithm. However, the environment considered in this study is ideal, and prob-

lems such as environmental interference, communication delay, and energy consumption constraints in the swarm robot task execution are not considered. Therefore, in subsequent work, the above problems will be studied.

Author Contributions: Y.Z. and M.W.: conceptualization, methodology, writing—original draft preparation; S.Z.: conceptualization, methodology, writing—reviewing and editing; A.C.: conceptualization, software, data curation, writing—reviewing and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (62271199).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Senanayake, M.; Senthooran, I.; Barca, J.C.; Chung, H.; Kamruzzaman, J.; Murshed, M. Search and tracking algorithms for swarms of robots: A survey. *Robot. Auton. Syst.* **2016**, *75*, 422–434. [CrossRef]
2. Xue, S.; Zhang, J.; Zeng, J. Parallel asynchronous control strategy for target search with swarm robots. *Int. J. Bio-Inspired Comput.* **2009**, *1*, 151–163. [CrossRef]
3. Zhang, Y.; Xue, S.; Zeng, J. Cooperative and Competitive Coordination in Swarm Robotic Search for Multiple Targets. *Robot* **2015**, *37*, 142–151.
4. Li, J.; Tan, Y. A probabilistic finite state machine based strategy for multi-target search using swarm robotics. *Appl. Soft Comput.* **2019**, *77*, 467–483. [CrossRef]
5. He, X.; Zhou, S.; Zhang, H.; Wu, L.; Zhou, Y.; He, Y.; Wang, M. Multiobjective coordinated search algorithm for swarm of UAVs based on 3D-simplified virtual forced model. *Int. J. Syst. Sci.* **2020**, *51*, 2635–2652. [CrossRef]
6. Phung, M.D.; Ha, Q.P. Motion-encoded particle swarm optimization for moving target search using UAVs. *Appl. Soft Comput.* **2020**, *97*, 106705. [CrossRef]
7. Cao, X.; Sun, H.; Jan, G.E. Multi-AUV cooperative target search and tracking in unknown underwater environment. *Ocean Eng.* **2018**, *150*, 1–11. [CrossRef]
8. Tang, Q.; Yu, F.; Zhang, Y.; Zhang, J. A stigmergetic method based on vector pheromone for target search with swarm robots. *J. Exp. Theor. Artif. Intell.* **2020**, *32*, 533–555. [CrossRef]
9. Brown, D.; Sun, L. Exhaustive mobile target search and non-intrusive reconnaissance using cooperative unmanned aerial vehicles. In Proceedings of the 2017 International Conference on Unmanned Aircraft Systems (ICUAS), Miami, FL, USA, 13–16 June 2017; pp. 1425–1431.
10. Brown, D.; Sun, L. Dynamic exhaustive mobile target search using unmanned aerial vehicles. *IEEE Trans. Aerosp. Electron. Syst.* **2019**, *55*, 3413–3423. [CrossRef]
11. Pan, W.T. A new Fruit Fly Optimization Algorithm: Taking the financial distress model as an example. *Knowl. Based Syst.* **2012**, *26*, 69–74. [CrossRef]
12. Yang, X.-S. A new metaheuristic bat-inspired algorithm. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*; González, J.R., Pelta, D.A., Cruz, C., Terrazas, G., Krasnogor, N., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 65–74.
13. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
14. Dorigo, M.; Gambardella, L.M. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Trans. Evol. Comput.* **1997**, *1*, 53–66. [CrossRef]
15. Guastella, D.C.; Cavallaro, N.D.; Melita, C.D.; Savasta, M.; Muscato, G. 3D path planning for UAV swarm missions. In *ICMSCE 2018: Proceedings of the 2018 2nd International Conference on Mechatronics Systems and Control Engineering*; Association for Computing Machinery: New York, NY, USA, 2018.
16. Xie, Y.; Han, L.; Dong, X.; Li, Q.; Ren, Z. Bio-inspired adaptive formation tracking control for swarm systems with application to UAV swarm systems. *Neurocomputing* **2021**, *453*, 272–285. [CrossRef]
17. Wan, N.; Li, Z.; Liang, X.L.; Wang, Y.B. Cooperative region search of UAV swarm with limited communication distance. *J. Systems Engineering and Electronics* **2022**, *44*, 1615–1625.
18. Qi, B.; Li, M.; Yang, Y.; Wang, X. Research on UAV path planning obstacle avoidance algorithm based on improved artificial potential field method. *J. Phys. Conf. Ser.* **2021**, *1948*, 012060. [CrossRef]
19. Yu, W.; Lu, Y. UAV 3D environment obstacle avoidance trajectory planning based on improved artificial potential field method. *J. Phys. Conf. Ser.* **2021**, *1885*, 022020. [CrossRef]
20. Zhou, Y.; Chen, A.; Zhang, H.; Zhang, X.; Zhou, S. Multitarget Search of Swarm Robots in Unknown Complex Environments. *Complexity* **2020**, *2020*, 8643120. [CrossRef]

21. Zhou, S.W.; Zhang, X.; Zhang, H.Q.; Zhou, Y.; Li, C.Y. Coordinated Control of Swarm Robots for Multi-target Search Based on a Simplified Virtual-Force Model. *Robots* **2016**, *11*, 641–650.
22. Zhou, Y.; Chen, A.; He, X.; Bian, X. Multi-Target Coordinated Search Algorithm for Swarm Robotics Considering Practical Constraints. *Front. Neurorobotics* **2021**, *15*, 753052. [CrossRef]
23. Pugh, J.; Martinoli, A. Inspiring and modeling multi-robot search with particle swarm optimization. In Proceedings of the Swarm Intelligence Symposium, 2007, Honolulu, HI, USA, 1–5 April 2007; pp. 332–339.
24. Viswanathan, G.M.; Buldyrev, S.V.; Havlin, S.; Da Luz, M.; Raposo, E.; Stanley, H.E. Optimizing the success of random searches. *Nature* **1999**, *401*, 911–914. [CrossRef]
25. Bénichou, O.; Loverdo, C.; Moreau, M.; Voituriez, R. Two-dimensional intermittent search processes: An alternative to Lévy flight strategies. *Phys. Rev. E Stat. Nonlinear Soft Matter Phys.* **2006**, *74*, 020102. [CrossRef] [PubMed]
26. Huang, T.Y.; Chen, X.B.; Xu, W.B.; Zhou, Z.W. A self-organizing cooperative hunting by swarm robotic systems based on loose-preference rule. *Acta Autom. Sin.* **2013**, *39*, 57–68. [CrossRef]
27. He, X.J.; Zhou, S.W.; Zhang, H.Q.; Zhou, Y. A 3D Parallel Multi-target Search Coordination Control Strategy for Swarm UAVS. *Inf. Control* **2020**, *49*, 605–614.
28. Zhang, H.Q. Research on Self-Organizing Cooperative Hunting by Swarm Robots Based on Simplified Virtual-Force Model. Ph.D. Thesis, Hunan University, Changsha, China, 2015.
29. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
30. Liang, J.J.; Suganthan, P.N. Dynamic multi-swarm particle swarm optimizer with local search. In Proceedings of the 2005 IEEE Congress on Evolutionary Computation, Edinburgh, UK, 2–5 September 2005; pp. 124–129.
31. Sun, W.; Lin, A.; Yu, H.; Liang, Q.; Wu, G. All-dimension neighborhood based particle swarm optimization with randomly selected neighbors. *Inf. Sci.* **2017**, *405*, 141–156. [CrossRef]