



Article

# An Efficient Evolution-Based Technique for Moving Target Search with Unmanned Aircraft Vehicle: Analysis and Validation

Mohamed Abdel-Basset<sup>1</sup>, Reda Mohamed<sup>1</sup>, Ibrahim M. Hezam<sup>2,\*</sup>, Ahmad M. Alshamrani<sup>2</sup>  
and Karam M. Sallam<sup>3</sup>

<sup>1</sup> Faculty of Computers and Informatics, Zagazig University, Zagazig 44519, Egypt; mohamedbasset@ieee.org (M.A.-B.); redamoh@zu.edu.eg (R.M.)

<sup>2</sup> Department of Statistics & Operations Research, College of Sciences, King Saud University, Riyadh 11451, Saudi Arabia; ahmadm@ksu.edu.sa

<sup>3</sup> Faculty of Science and Technology, School of IT and Systems, University of Canberra, Canberra, ACT 2601, Australia; karam.sallam@canberra.edu.au

\* Correspondence: ialmishnanah@ksu.edu.sa

**Abstract:** Recent advances in technology have led to a surge in interest in unmanned aerial vehicles (UAVs), which are remote-controlled aircraft that rely on cameras or sensors to gather information about their surroundings during flight. A UAV requires a path-planning technique that can swiftly recalculate a viable and quasi-optimal path in flight if a new obstacle or hazard is recognized or if the target is moved during the mission. In brief, the planning of UAV routes might optimize a specific problem determined by the application, such as the moving target problem (MTP), flight time and threats, or multiobjective navigation. The complexity of MTP ranges from NP-hard to NEXP-complete because there are so many probabilistic variables involved. Therefore, it is hard to detect a high-quality solution for this problem using traditional techniques such as differential calculus. Therefore, this paper hybridizes differential evolution (DE) with two newly proposed updating schemes to present a new evolution-based technique named hybrid differential evolution (HDE) for accurately tackling the MTP in a reasonable amount of time. Using Bayesian theory, the MTP can be transformed into an optimization problem by employing the target detection probability as the fitness function. The proposed HDE encodes the search trajectory as a sequence of UAV motion pathways that evolve with increasing the current iteration for finding the near-optimal solution, which could maximize this fitness function. The HDE is extensively compared to the classical DE and several rival optimizers in terms of several performance metrics across four different scenarios with varying degrees of difficulty. This comparison demonstrates the proposal's superiority in terms of the majority of used performance metrics.

**Keywords:** evolutionary algorithms; differential evolution; UAV; unmanned aerial vehicles; exploration; target search

**MSC:** 68W25; 68W50; 97P80



**Citation:** Abdel-Basset, M.; Mohamed, R.; Hezam, I.M.; Alshamrani, A.M.; Sallam, K.M. An Efficient Evolution-Based Technique for Moving Target Search with Unmanned Aircraft Vehicle: Analysis and Validation. *Mathematics* **2023**, *11*, 2606. <https://doi.org/10.3390/math11122606>

Academic Editors: Matevž Kunaver and Árpád Bűrmen

Received: 11 May 2023

Revised: 5 June 2023

Accepted: 5 June 2023

Published: 7 June 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Unmanned aerial vehicles (UAVs), also known as pilotless aircraft, are remote-controlled aircraft that have recently become popular due to their capability to operate in harsh environments using recent monitoring technology such as cameras or sensors to gather information about these environments during flight [1,2]. Several practical fields have widely used UAVs to carry out various tasks, especially those that put the pilot at risk, such as search and rescue assistance operations, mapping, geology, agriculture, target acquisition, hazardous environmental monitoring, and several others [3]. In these tasks, pilots on the ground can

take turns controlling UAVs in the air, which enables the missions to last for extended periods of time [4]. Recently, the need for unmanned vehicles has significantly increased, and scientists have been asked to provide solutions to numerous challenges that are caused by the utilization of unmanned systems.

The UAV route planning challenge has been receiving a lot of attention, and several methods have been recently developed for three different optimization problems in this challenge, including (1) searching for the optimal path within known locations; (2) looking for lost targets (moving target problem (MTP)); and (3) searching for previously unknown targets that are subsequently followed [4,5]. Over the last few years, the majority of research studies have concentrated on locating the most secure and efficient route to a known target within environments that are fraught with danger [6,7]. These studies are concerned with locating the most efficient route that is risk-free while at the same time minimizing the distance traveled or the amount of time required to complete the mission [8,9].

When utilizing UAVs to search for a missing target, there is frequently a crucial window of time that is referred to as the “golden time”, during which the likelihood of the target being detected should be at its highest [10]. This probability might substantially decrease as time goes on due to the degradation of initial information as well as the effect caused by outside factors, including terrain features, weather conditions, and target movement. The primary goal of employing UAVs to search for a missing target is to estimate a route that can maximize the target’s detection likelihood within a particular flight time, given basic information on the search conditions and target position [10,11]. Probabilistic functions are commonly used to express this problem in the literature, which allows for appropriate accounting of uncertainties in the underlying assumptions, search parameters, and sensor models [10]. As a result, several approaches for maximizing the probability of detecting the lost target using UAV flight routes have been developed over the last few decades. For example, a Bayesian technique was used to design the objective functions used to evaluate the detection likelihood of the generated UAV paths and maximize this probability to detect the lost target [10]. In the same context, the initial search map that forms the likelihood of detecting the target was simulated as a multivariate normal distribution with variance and mean that were estimated according to the known information about the lost target [10]. In addition, a stochastic Markov process was used to describe the target dynamic, which, depending on the search situation, can either be deterministic or not [10]. Furthermore, most of the time, the sensor is modeled as either a binary variable that can take on one of two states, namely identified or not identified, or as a continuous Gaussian variable [12].

The complexity of MTP ranges from NP-hardness to NEXP-completeness because of the numerous probabilistic variables involved [10]. So, finding the exact solution to this problem using conventional techniques such as differential calculus becomes problematic. Several techniques based on ad hoc heuristics or approximation methods have been recently developed to find near-optimal or optimal UAV trajectories in a reasonable time [12]. On the one hand, these ad hoc heuristics are tailored to building high-level UAV trajectories that take advantage of the search problem’s inherent features. On the other hand, good UAV trajectories can be achieved using approximation optimization techniques by either:

1. Sampling the distribution learned from the best solutions estimated in prior iterations of the approach, such as the Bayesian optimization algorithm (BOA) and cross-entropy optimization (CEO), these solutions are evaluated using the objective functions to determine the quality of each solution.
2. or exploring the probability distribution map using evolutionary and metaheuristic algorithms employed with the objective functions to find the best trajectory that will maximize the detection probability.

In general, the existing methods for dealing with MTP make use of information specific to the problem, either directly using certain heuristics or indirectly utilizing the objective function. The metaheuristic methods have gained significant interest among the other methods for tackling this problem due to their strong operators, which have enabled them

to find high-quality solutions for several optimization problems at a lower computational cost. These operators are in exploration and exploitation. The exploration aims at covering as many search regions as possible to reach the most promising solutions. The exploitation operator thoroughly investigates those promising regions to identify nearly ideal solutions to the optimization problems. The tradeoff between those two operators is considered the main challenge that still needs a solution for achieving equilibrium between them during the optimization process. Several existing metaheuristic algorithms tried to relate those operators in different ways in the hope of finding the most effective way that might achieve the equilibrium state. For example, some algorithms have employed the exploration operator in the optimization process's first half to cover as much of the search space as possible for extracting the most promising regions that are extensively exploited in the second half for achieving better outcomes. However, in this way, the algorithm might give more time to one operator on account of the other when solving specific optimization problems. For example, some optimization problems might need more time for exploration operators to avoid stagnation into local optima, while others might need more time for exploitation operators to improve the convergence speed to reach the near-optimal solution in fewer iterations. Some of the authors designed the metaheuristic algorithms to execute each operator in sequential order. Furthermore, by using this method, we could not achieve a balance between those two operators. From that, it is concluded that the balance between exploration and exploitation in the classical metaheuristic is considered a challenge that still needs a solution in the future.

In addition, in the literature, the exploration and exploitation operators of the classical metaheuristic algorithms have been improved using several effective mechanisms. For example, some researchers employed the chaotic maps in the initialization process to distribute the initial solutions within the search boundary as accurately as possible to avoid stagnation into local optima, while others replaced the random values generated by the uniform distribution with the chaotic maps to avoid stagnation into local minima in addition to accelerating the convergence speed towards the near-optimal solution. There are several other mechanisms to improve those two operators, such as quantum computing, ranking-based strategy, and levy flight.

Back to the MTP: this problem has been tackled in the literature using several metaheuristic algorithms, some of which will be reviewed in the rest of this paragraph. In [11], the electrically charged optimization was employed with the motion-encoded mechanism to find the near-optimal path that could detect the moving target using the UAVs. This algorithm, called ECPO-ME, used Bayesian theory to model the MTP as an optimization problem with a maximized objective function based on the target detection probability. In the same context, in [13], the genetic algorithm (GA) was developed to search for moving targets using multiple UAVs. This algorithm used the target detection likelihood based on Bayesian theory as an objective function to determine the probability that the target could be detected using each solution estimated by GA. In [14], the convex combination multiple populations competitive swarm optimization algorithm (CDCSO) is a new metaheuristic-based technique presented for the near-optimal path that may maximize the detection probability of a moving target. To improve the algorithm's search capability and keep the population from getting stuck in a local optimum, CDCSO implemented a novel convex combination updating technique and a multiple population strategy. The experimental findings show the CDCSO's effectiveness in comparison to several rival optimizers. In [15], a parallel multiobjective multiverse optimizer (PMOMVO) was devised and effectively implemented to solve the increasing computing time of the UAV path planning problem in dynamic 3D environments.

For the solution of the path planning problem, Ait-Saadi [16] proposed a hybrid optimization technique based on integrating the chaotic Aquila algorithm with simulated annealing. The purpose of SA was to strike a compromise between exploration and exploitation operators. The experimental results indicate that this algorithm could perform better in some situations. There are several other metaheuristic-based techniques for

tackling the path planning problem of UAV, including the particle swarm optimization algorithm [17,18], a metaheuristic-based imitative learning optimization technique [19], the tabu search-based optimization technique [20], and the memetic algorithm based on two levels [21], bat algorithm improved using fruit fly optimization algorithm [22], dynamic group based cooperation algorithm [7], enhanced ant colony optimization [23], grey wolf optimization algorithm [24], immune plasma algorithm based on multi-population [25], improved symbiotic organisms search [26], GWO improved using both the local memory and the fittest's survival principle [27], genetic algorithm [28], reverse glowworm swarm optimization [29], modified honey badger algorithm [30], immune plasma algorithm [31], multi-objective non-dominated sorting genetic algorithm (NSGA-II) [32], search and rescue optimization algorithm [33], improved chimp optimization algorithm [2], spider monkey optimization [1], and metaheuristic-based hybrid algorithm [34].

However, the majority of these methods may suffer from at least one of the following drawbacks: stagnation in local minima, slow convergence speed, or high costs of computation. Those drawbacks prevent them from achieving an acceptable solution in a reasonable amount of time. Therefore, in this study, a new hybrid method, referred to as hybrid differential evolution (HDE), is proposed to accurately handle the MTP in an acceptable time. This algorithm hybridizes differential evolution (DE) with two newly proposed updating methods to improve the traditional DE's exploration and exploitation mechanisms. By utilizing Bayesian theory, the MTP may be recast as an optimization problem, with the likelihood of detecting the target serving as the objective function. This objective function needs to be maximized to increase the target detection probability. HDE encodes the search trajectory as a collection of UAV motion pathways that arise during the process of optimization. It is thoroughly assessed using four different scenarios that range in difficulty level and compared to the classical DE and several metaheuristic-based optimization techniques, such as the artificial gorilla troop optimizer (GTO), gradient-based optimizer (GBO), classical DE, marine predator algorithm (MPA), and motion-encoded particle swarm optimization (MPSO). This comparison is based on several performance indicators, such as the average fitness, standard deviation, amount of CPU time, convergence curve, and search path of HDE. The experimental results disclose that HDE is superior in terms of all performance metrics except CPU time, which is better reduced by MPSO. Our main conclusions regarding the proposed HDE are listed as follows:

- Has better exploration and exploitation operators than the classical DE.
- Requires less computational cost than the classical DE with several other rival optimizers.
- Has better convergence speed.
- Able to find more accurate paths that could increase the detection probability of moving targets.

In brief, this study's main contributions are listed as follows:

- Proposal of the differential evolution (DE) algorithm based on two newly proposed updating schemes to present a new search algorithm named hybrid differential evolution (HDE) for accurately tackling the MTP in a reasonable amount of time.
- Investigating the performance of the classical DE when integrated with the motion encoding mechanism for tackling the MTP.
- Improving the performance of the classical DE using two newly proposed updating mechanisms to improve its exploration and exploitation capabilities.
- Assessing HDE using four different scenarios with different difficulty levels:
- Scenario 1: This scenario features two adjacent high-likelihood zones. They differ slightly in position and value, which may make it difficult to determine where to search for the target.
- Scenario 2: The two high-probability regions in this scenario are evenly distanced from the UAV's initial location. While the target is heading southwest, the algorithm has to swiftly choose the higher-probability zone in which to search and track.

- Scenario 3: In this scenario, there is a single concentrated probability region that is rapidly relocating to the southeast. The algorithm is therefore put to a test of its capacity for exploration and adaptation.
- Scenario 4: This scenario is set up with two static high-probability zones, one on either side of a UAV location, with the right region having a little greater probability than the left part. The algorithm must determine the proper target region as the target moves north.
- Comparing its performance to several rival metaheuristic optimizers for the average fitness, standard deviation, amount of CPU time, and convergence curve.
- Experimental results indicate that HDE outperforms all competitors in all performance metrics except CPU time, where MPSO demonstrates a more effective reduction.

This manuscript is structured as shown next: Section 2 describes the mathematical formulation of MTP; Section 3 overviews the classical differential evolution; Section 4 describes the proposed HDE based on integrating the classical differential evolution with a novel mutation scheme to achieve better UAV trajectories; Section 5 presents the results and discussion for four scenarios with various characteristics; and Section 6 presents the conclusion and future perspectives.

**2. Problem Formulation**

The probability model of the moving target search problem (MTP) using UAV is presented here based on the sensor model, target model, and belief map, which will be discussed in greater depth in the following sections. Additionally, this section will describe the objective function employed with the proposed technique to find the search route that may maximize the detection likelihood of the moved target.

*2.1. Target Model*

In the MTP, the location of the target is denoted by an unidentified variable  $x \in X$ . Prior to beginning the search process, a probability distribution function (PDF) is utilized to simulate the target location according to the information that is currently available. For example, the PDF can take into consideration the target’s most recent known location before missing. This PDF might be a normal distribution with its center at the most recent known position, but it might alternatively be a uniform distribution in the case that no information is available regarding the position of the target. This PDF is depicted as a grid map in the search space and is referred to as the belief map, as depicted in Figure 1. In this figure, the search area  $S$  is divided into a  $4 \times 4$  grid, symbolized as  $S_r \times S_c$ , where the center of each cell in this grid is assigned a probability to indicate the detection probability of a moving target; this initial grid is known as the initial belief map  $b(v_0)$ . The sum of the probabilities in this belief map has to be 1, as formulated in the following equation:

$$\sum_{x_0 \in S} b(v_0) = 1 \tag{1}$$

1 <b>0.06</b>	2 <b>0.05</b>	3 <b>0.2</b>	4 <b>0.03</b>
5 <b>0.04</b>	6 <b>0.11</b>	7 <b>0.08</b>	8 <b>0.15</b>
9 <b>0.07</b>	10 <b>0.1</b>	11 <b>0.06</b>	12 <b>0.05</b>

**Figure 1.** Depiction of an initial belief map  $b(v_0)$ .

During the process of searching, the target might not remain in one place but rather move in a predetermined pattern. A Markov process can be used to mimic this pattern. This particular pattern is only dependent on the initial location  $v_0$  of the moving target in the unique circumstance of a conditionally deterministic target, which is herein considered. In such a scenario, the transition function, denoted by the symbol  $p(v_t|v_{t-1})$  and indicating the likelihood that the target moves from the cell  $v_{t-1}$  to cell  $v_t$ , is identified for all cells  $x_t$  that fall within the range denoted by  $S$ . Therefore, if the initial position of the target is known, the full trajectory that it will take can be anticipated. This assumption is made rather frequently in relation to search problems in general, as discussed in [12].

### 2.2. Sensor Model

Installing a sensor on the UAV to conduct an observation  $z_t$  at each time step  $t$  allows for the searching and locating of a target. The observations are separate to the extent that the occurrence of one observation does not shed any light on the other observations' occurrence. A detection technique is implemented to produce feedback for each observation. This algorithm has only two possible outputs: either target detection, denoted by the expression  $z_t = T_t$ , or no detection, denoted by the expression  $z_t = \bar{T}_t$ . Here,  $T_t$  stands for the detection event of the target that occurred at time  $t$ . An observation of the detected target, denoted by  $z_t = T_t$ , does not necessarily confirm that the target is present at  $v_t$  because the sensor and the detection algorithm are not flawless. This is conveyed through the observation likelihood, which is denoted by the symbol  $p(z_t|v_t)$ , granted that the sensor model has been considered. After that, the probability of there being no detection, given a target position  $x_t$ , is calculated by using the following:

$$p(\bar{T}_t|v_t) = 1 - p(T_t|v_t) \tag{2}$$

### 2.3. Belief Map Update

The Bayesian technique and the sensor's series of observations,  $z_{1:t}$ , are used to update the target's belief map at time  $t$ ,  $b(v_t)$ . This technique, called recursive Bayesian estimation, is carried out recursively via two steps, which are update and prediction. Throughout the prediction process, the belief map is iterated through time in a manner that is consistent with the goal motion model and predicted according to the following formula:

$$\hat{b}(v_t) = \sum_{v_{t-1} \in S} p(v_t|v_{t-1}) \cdot b(v_{t-1}) \tag{3}$$

From Equation (3), it is clear that  $b(v_{t-1})$  is the target's conditional probability to be found at  $v_{t-1}$  based on the observations up to  $t - 1$ ;  $b(v_{t-1}) = p(v_{t-1} | z_{1:t-1})$ . The update is carried out when the observation  $z_t$  is available by merely multiplying the anticipated belief map by the updated conditional observation probability, as shown below:

$$b(v_t) = \eta_t p(z_t|v_t) \cdot \hat{b}(v_t) \tag{4}$$

where  $\eta_t$  represents the normalization factor computed according to the following formula to scale the target's likelihood of appearing within the search area to one:

$$\eta_t = \frac{1}{\sum_{x_t \in S} p(z_t|x_t) \cdot \hat{b}(x_t)} \tag{5}$$

### 2.4. Objective Function

Searching for a moving target could be expressed as an optimization problem with the UAVs' search trajectories as its solutions. The UAVs' search trajectories are evaluated by the cumulative probability, an objective function that measures the probability of each trajectory locating the target. This function is widely used in the literature, so we employ it in this research. The cumulative probability  $P_t$  has a limit and tends to approach 1 as the value of  $t$  goes to infinity. With a limited amount of time to search, the objective function for

MTP may now be constructed based on the probability of detecting the target. Assuming that the search time is  $1, 2, 3, \dots, n$ . Then, the search strategy's objective is to find a search path  $\vec{O} = (o_1, \dots, o_n)$  that may maximize the symbol  $P_t$  (representing the cumulative probability). As a consequence of this, the objective function will ultimately be expressed as follows:

$$f(\vec{O}) = P_t = \sum_{k=1}^n p_k \tag{6}$$

where  $P_t$  stands for the cumulative probability to detect the target in  $t$  steps, which could be computed by the sum of  $p_t$  over those steps  $t$ , as follows:

$$P_t = \sum_{k=1}^t p_k = P_{t-1} + p_t \tag{7}$$

where  $p_k$  stands for the likelihood that the target will be spotted for the first time at time  $t$ , which could be estimated according to the next formula:

$$p_t = \prod_{k=1}^{t-1} r_k = R_{t-1}(1 - r_t) \tag{8}$$

where  $r_t$  indicates the probability that the target could not be identified at time  $t$ , and  $R_t$  stands for the probability of failure in determining the target from time 1 to time  $t$  and is calculated according to the next mathematical formula:

$$R_t = \prod_{k=1}^t r_k = R_{t-1}r_t \tag{9}$$

The Bayesian theory states that the no detection likelihood and the updated belief map from the prediction phase determine the likelihood that the target is not identified at time  $t$  during an observation, where  $r_t = p(\bar{T}_t | z_{1:t-1})$ . The following formulas provide the likelihood that the target is not detected for the entire search area:

$$r_t = \sum_{v_t \in S} p(\bar{T}_t | v_t) \cdot \hat{b}(v_t) \tag{10}$$

### 3. Differential Evolution

Differential evolution [35] is a simple evolution-based optimization algorithm that was proposed for tackling optimization problems. It is analogous to GAs in terms of selection, mutation, and crossover operators. Before beginning the optimization process, DE initializes  $N$  individuals within the upper and lower bounds of each dimension in the optimization problem with  $n$  dimensions. After that, the mutation and crossover operators were utilized so that the search space could be explored to locate superior solutions, as will be detailed further down.

#### 3.1. Mutation Operator

The DE has made use of this operator to construct a mutant vector, denoted by the notation  $\vec{v}_i^l$ , for each individual  $\vec{x}_i^t$  in the current population. This mutant vector could be generated using several mutation schemes, such as "DE/rand/1", "DE/target-to-best/1", and several others. In this study, the first scheme is employed to solve MTP due to its ability to preserve population diversity as well as improve the convergence speed for reaching better outcomes in a few iterations; its mathematical model is described as follows:

$$\vec{v}_i^l = \vec{x}_a^l + F \cdot (\vec{x}_k^l - \vec{x}_j^l) \tag{11}$$

where  $a, k$ , and  $j$  are three integers to represent three individuals' indices selected randomly from the individuals in the current population.  $l$  represents the current iteration.  $F$  represents a scaling factor with a positive numerical value, ranging between 0 and 1.

### 3.2. Crossover Operator

This operator has been utilized to recombine the  $i$ th individual's position and the mutant vector based on a predefined crossover probability ( $Cr$ ) to generate a trial vector  $\vec{u}_i^l$  that will represent the new solution of this individual for the optimization problem. This trial vector could be created using the next mathematical formula:

$$u_{i,j}^l = \begin{cases} v_{i,j}^l & \text{if } (r_1 \leq Cr) \\ x_{i,j}^l & \text{otherwise} \end{cases} \quad (j = j_r) \tag{12}$$

where  $j_r$  is an integer selected at random between 1 and  $n$ ,  $j$  stands for the current dimension, and  $Cr$  is a fixed value estimated within the experiments.

### 3.3. Selection Operator

Lastly, this operator is utilized to compare the trial vector  $\vec{u}_i^l$  to the current vector  $\vec{x}_i^l$ , with the former being carried forward to the next generation if it proves to be the fittest. This is mathematically expressed in the following formula for a minimization problem:

$$\vec{x}_i^l = \begin{cases} \vec{u}_i^l & \text{if } (f(\vec{u}_i^l) < f(\vec{x}_i^l)) \\ \vec{x}_i^l & \text{otherwise} \end{cases} \tag{13}$$

Finally, the classical DE's pseudocode is listed in Algorithm 1.

---

**Algorithm 1:** Standard differential evolution.  $l$ : represents the current iteration;  $T_m$  represents the maximum iteration.

---

1.     Initializes  $N$  individuals,  $\vec{x}_i, i = 1, 2, \dots, N$
  2.     Evaluation and extraction of the best solution achieved yet  $\vec{x}_l^*$
  3.      $l = 0$
  4.     Set  $F$  and  $Cr$
  5.     **while** ( $l < T_m$ )
  6.     **for**  $i = 0$  to  $N$
  7.      $\% \% \% \text{ Mutation Operator } \% \% \%$
  8.     Compute  $\vec{v}_i^l$  using Equation (11)
  9.      $\% \% \% \text{ Crossover Operator } \% \% \%$
  10.    **for**  $j = 0$  to  $n$
  11.     $r_1$  :a numeric value selected randomly in the range (0, 1).
  12.    **If** ( $r_1 < Cr \mid \mid j = j_r$ )
  13.     $u_{i,j}^l = v_{i,j}^l$
  14.    **Else**
  15.     $u_{i,j}^l = x_{i,j}^l$
  16.    **End if**
  17.    **end for**
  18.     $\% \% \% \text{ Selection Operator } \% \% \%$
  19.    **if** ( $f(\vec{u}_i^l) < f(\vec{x}_i^l)$ )
  20.     $\vec{x}_i^l = \vec{u}_i^l$
  21.    **end if**
  22.    Replace  $\vec{x}_l^*$  with  $\vec{u}_i^l$  if better
  23.    **end for**
-



---

**Algorithm 1.** *Cont.*

---

24.  $l = l + 1$   
 25. **end while**  
**Return**  $\vec{x}_l^*$

---

**4. The Proposed Algorithm**

In [10], UAV motion has been presented as a means of encoding the positions of the solutions in metaheuristics. In this idea, each search path is viewed as a sequence of UAV emotional segments, where each segment represents the UAV’s transition from one cell on the flight plane to the next. This motion can be fully characterized by the vector  $U_t = (\rho_t, \alpha_t)$ , where  $\rho_t$  and  $\alpha_t$  are the amplitude and direction of the motion at time t, respectively. In this way, the search path is designed as a vector of n segments, where each segment consists of the vector  $U_t$  to determine the amplitude and direction of motion at time t. In the proposed algorithm, before starting the optimization process, N individuals, representing the search paths used to solve the MTP, with n dimensions for each one, where each dimension involves  $U_t$ , are randomly initialized within the search range of the optimization problems. This search range includes an upper bound  $x_u$  of 4 and a lower bound  $x_l$  of  $-4$  for all dimensions to represent the motion range as used in [10]. Generally, the following formulas are used to randomly initialize these solutions:

$$\vec{\rho} = r_1 \cdot (x_u - x_l) + x_l \tag{14}$$

$$\vec{\alpha} = r_1 \cdot (x_u - x_l) + x_l \tag{15}$$

$$\vec{U}_t = [\vec{\rho}_t; \vec{\alpha}_t], \vec{x}_i^t = \vec{U}_t \tag{16}$$

where  $r_1$  is a numerical vector selected randomly in the range (0, 1) according to the uniform distribution. After that, each initialized solution is converted into an  $\vec{O}_i$  the direct path that can be used to solve the MTP. As stated in [10], the mapping stage can be performed by initially limiting UAV movement to one of eight neighbors at each time step. After that, the magnitude of the motion, denoted by  $\rho_t$ , may be normalized, and the angle of the motion, denoted by  $\alpha_t$ , can be quantized as follows:

$$\rho_t^* = 1 \tag{17}$$

$$\alpha_t^* = 45^\circ \lfloor \alpha_t / 45^\circ \rfloor \tag{18}$$

where  $\lfloor \cdot \rfloor$  stands for the rounding-to-the-nearest-integer operator. Then, the Cartesian node  $o_{i,t}$  corresponding to the UAV’s position is computed as follows:

$$o_{i,t+1} = o_{i,t} + U_{i,t}^* \tag{19}$$

$$U_{i,t}^* = (\lfloor \cos(\alpha_t^*) \rfloor, \lfloor \sin(\alpha_t^*) \rfloor) \tag{20}$$

Afterward, the decoded path of the  $i$ th solution is assessed according to the fitness function described previously to determine its quality. After repeating the same process for  $N$  solutions that represent the population, the fittest solution achieved yet is set to the variable  $\vec{x}_l^*$  and then the optimization process of the classical DE is fired to update the current positions of the individual in the population to new positions for finding better solutions in the MTP. In the same way, each updated solution is mapped using the mapping stage described above to generate the decoded path, which represents the solution to this problem.

However, the classical DE has two challenges: falling into local optima and slow convergence speed, which need to be effectively solved to improve its performance in tackling the MTP. Therefore, in this study, two effective updating schemes have been proposed to be integrated with the classical DE to present a new robust variant, namely hybrid differential evolution (HDE). Due to these updated schemes, this variant, which is listed in Algorithm 2 and depicted in Figure 2, enjoys better exploitation and exploration mechanisms that enable it to achieve better convergence speed and avoid getting stuck in local optima. The first scheme is based on updating  $M$  solutions picked randomly from the individuals in the population according to the following formula:

$$\vec{u}_i^l = \vec{x}_a^l + A \cdot (\vec{x}_b^l - \vec{x}_c^l) \cdot (r_2 < \delta) + |r_n| \cdot (\vec{x}_i^l - \vec{x}_i^{*l}) \cdot (r_2 \geq \delta) \tag{21}$$

$$A = (r_1 \cdot (1 - \mu) + \mu) \tag{22}$$

where  $r_n$  is a number generated randomly according to the normal distribution,  $\vec{r}_2$  is a vector assigned at random between 0 and 1,  $\mu$  is a random value chosen randomly in the interval (0, 1) and fixed for all solutions at iteration  $l$ . This scheme is based on two states: the first one encourages the exploration operator if the vector  $\vec{r}_2$  is smaller than the constant value  $\delta$ ; otherwise, the exploitation operator is preferred by moving the current solution in the direction of the best solution obtained so far. To avoid the local optimal caused by Equation (21), it is applied to a number of solutions, denoted as  $M$ , randomly selected from the current populations. To further improve both exploitation and exploration capabilities for covering all the directions that might involve the lost targets, the second scheme is herein proposed to generate the mutant vector  $\vec{v}_i^l$  as formulated in Equation (23). This scheme is based on relating the exploration and exploitation capabilities with the current iteration to encourage the exploration in the optimization process's first half and the exploitation in the second half.

$$\vec{v}_i^l = \left( \left( \frac{l}{T_m} \right) \cdot \vec{x}_i^{*l} + \left( 1 - \frac{l}{T_m} \right) \cdot \vec{x}_a^l \right) + r_n \cdot (\vec{x}_b^l - \vec{x}_c^l) \cdot (r_2 < 0.5) \tag{23}$$

where  $\vec{r}_n$  is a vector generated according to the normal distribution. The expression  $(r_2 < 0.5)$  in the previous equation is used to maintain individual diversity throughout the optimization process, returning either 1 to indicate adding a step to the current position to avoid stagnation in local optima or 0 to skip this step to accelerate the convergence speed; therefore, the value 0.5 is used to tradeoff equally between those two options. This mutant vector is recombined with the  $i$ th individual's position using an adaptive crossover probability ( $Cr'$ ) to generate the trial vector  $\vec{u}_i^l$  in the same way, as formulated in Equation (12) by replacing  $Cr$  with  $Cr'$ .  $Cr'$  could be computed according to the next mathematical equation:

$$Cr' = \gamma + \left( 1 - \frac{l}{T_m} \right) * (1 - \gamma) \tag{24}$$

where  $\gamma$  is the smallest crossover probability used to recombine the mutant vector with the current one; the best value for this probability is determined in experiments.

---

**Algorithm 2:** Hybrid differential evolution (HDE).

---

1. Input the required data, such as UAV location, etc.
  2. Initialize Belief map
  3. Initializes  $N$  individuals,  $\vec{x}_i, i = 1, 2, \dots, N$
  4. Decoding and evaluation
  5. Extraction of the fittest solution achieved yet  $\vec{x}_i^*$
-

**Algorithm 2.** *Cont.*


---

```

6.    $l = 0$ 
7.   Set  $F$  and  $Cr$ 
8.   while ( $l < T_m$ )
9.   for  $i = 0$  to  $N$ 
10.  %%% Mutation Operator %%%
11.  Compute  $\vec{v}_i^l$  using Equation (11)
12.  %%% Crossover Operator %%%
13.  for  $j = 0$  to  $n$ 
14.   $r_1$  : a numeric value selected randomly in the range (0, 1).
15.  If ( $r_1 < Cr$  ||  $j = j_r$ )
16.   $u_{i,j}^l = v_{i,j}^l$ 
17.  Else
18.   $u_{i,j}^l = x_{i,j}^l$ 
19.  End
20.  end
21.  %%% Mapping process %%%
22.  Decode the solution  $\vec{u}_i^l$  to  $O_i^{\rightarrow l+1}$ 
23.  %%% Selection Operator %%%
24.  if ( $f(O_i^{\rightarrow l+1}) < f(O_i^{\rightarrow l})$ )
25.   $x_i^{\rightarrow t} = u_i^{\rightarrow t}$ 
26.  end
27.  Replace  $x_i^*$  with  $\vec{u}_i^l$  if better
28.  end
29.   $l = l + 1$  %%% Increment the current iteration %%%
30.   $SL$ : a set including  $M$  solutions chosen at random
31.  for  $i = 0$  to  $N$ 
32.  If  $i \in SL$  %%% The first updating scheme %%%
33.  Update  $\vec{u}_i^l$  using Equation (21)
34.  Else %%% The second updating scheme %%%
35.  Create  $\vec{v}_i^l$  using Equation (23)
36.  Update  $Cr'$ 
37.  for  $j = 0$  to  $n$ 
38.   $r_1$  : a numerical value chosen at random in the range (0, 1)
39.  If ( $r_1 < Cr'$  ||  $j = j_r$ )
40.   $u_{i,j}^l = v_{i,j}^l$ 
41.  Else
42.   $u_{i,j}^l = x_{i,j}^l$ 
43.  End
44.  End If
45.  End If
46.  %%% Mapping process %%%
47.  Decode the solution  $\vec{u}_i^l$  to  $O_i^{\rightarrow l+1}$ 
48.  %%% Selection Operator %%%
49.  if ( $f(O_i^{\rightarrow l+1}) < f(O_i^{\rightarrow l})$ )
50.   $x_i^{\rightarrow t} = u_i^{\rightarrow t}$ 
51.  end
52.  Replace  $x_i^*$  with  $\vec{u}_i^l$  if better
53.  end
54.   $l = l + 1$ 
55.  end while
Return  $x_i^*$ 

```

---

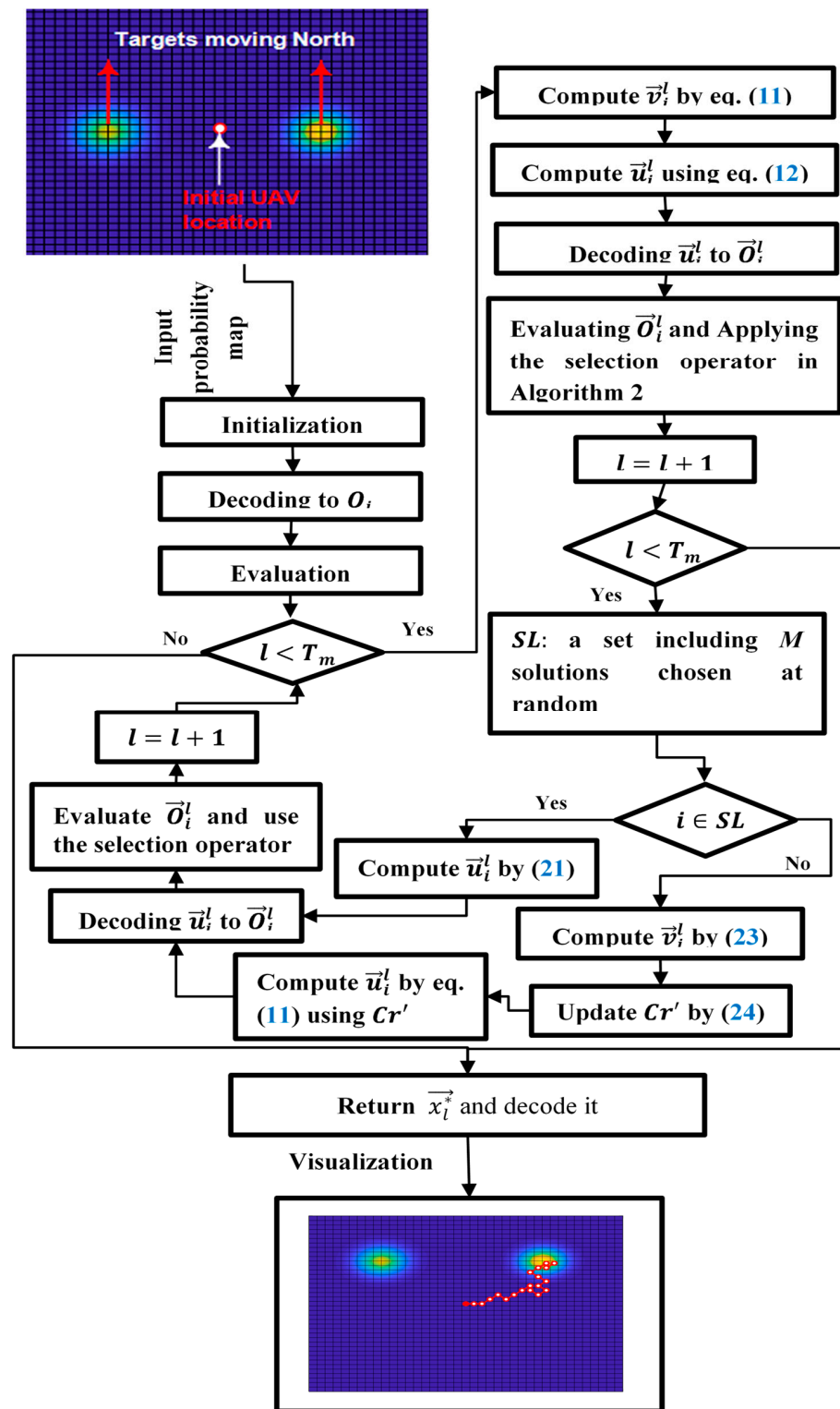


Figure 2. The proposed HDE’s flowchart.

### 5. Results and Discussion

Herein, we will examine the stability and effectiveness of HDE for the MTP by conducting several experiments under four different scenarios of varying degrees of complexity. The effectiveness of the proposed HDE is disclosed by contrasting it to five well-established optimizers, such as GTO [36], GBO [37], classical DE, MPA [38], and MPSO [10], in terms of numerous performance measures, including convergence curve, standard deviation (SD),

average fitness value, and CPU time. The regulating parameters of these algorithms have been set by the recommendations made in the cited paper, except  $T_m$  and  $N$ , which have been restricted to the values of 300 and 30, respectively, to guarantee an accurate comparison. It is important to point out that MATLAB R2019a is used to carry out the implementation of each method on a device that possesses the following capabilities: 32 gigabytes of random access memory (RAM), a 2.40 GHz Intel Core i7-4700MQ CPU, and the 64-bit professional edition of Windows 10.

5.1. Scenario Setup

In this study, the proposed and compared algorithms are extensively assessed using four distinct search scenarios taken from [10]. The map size of each scenario is defined to be the same with a value of 40 ( $S_r = S_c = 40$ ); however, the target motion model  $P(v_t|v_{t-1})$ , the initial locations of the UAV, and the initial belief map  $b(v_0)$  are different. As can be seen in Figure 3, the characteristics of these scenarios are as follows: The target dynamics are represented by a red arrow, and the initial location of the UAV is depicted by a white arrow. Other characteristics of these scenarios are as follows:

Scenario 1: This scenario is taken from [10] and features two adjacent high-likelihood zones. They differ slightly in position and value, which may make it difficult to determine where to search for the target.

Scenario 2: The two high-probability regions in this scenario are evenly distanced from the UAV's initial location. While the target is heading southwest, the algorithm has to swiftly choose the higher-probability zone in which to search and track.

Scenario 3: In this scenario, there is a single concentrated probability region that is rapidly relocating to the southeast. The algorithm is therefore put to a test of its capacity for exploration and adaptation.

Scenario 4: This scenario is set up with two static high-probability zones, one on either side of a UAV location, with the right region having a little greater probability than the left part. The algorithm has to determine the proper target region as the target moves north.

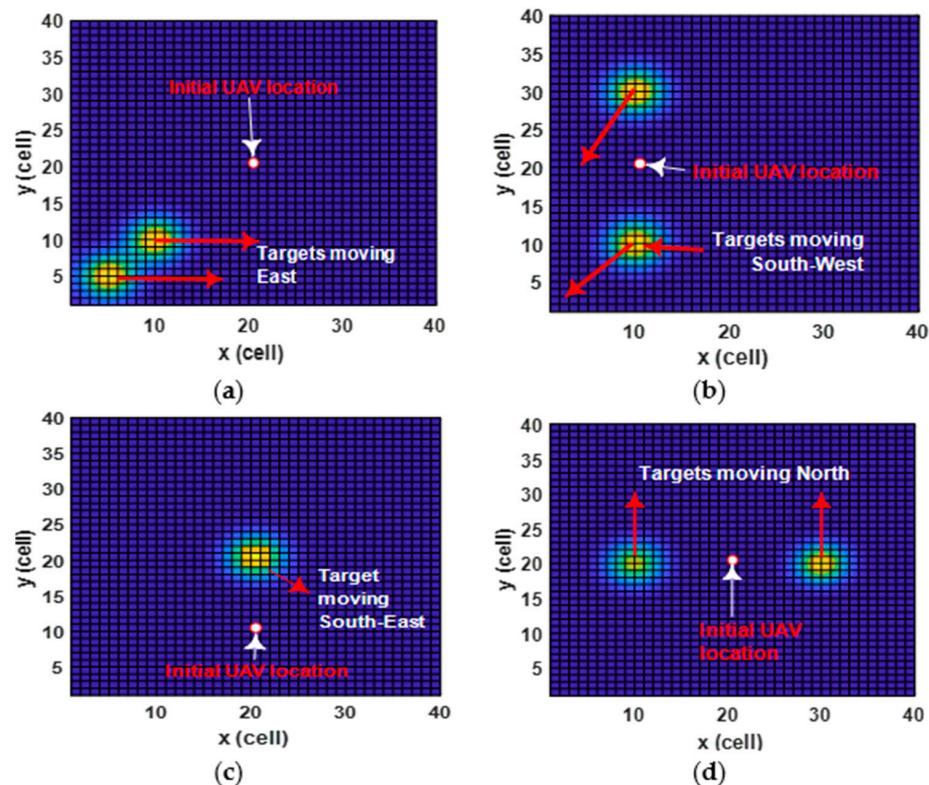


Figure 3. Investigated search scenarios; (a) Scenario 1; (b) Scenario 2; (c) Scenario 3; (d) Scenario 4.

5.2. Parameter Adjustment

The proposed algorithm has three newly designed controlling parameters:  $\delta$ ,  $\gamma$ , and  $M$ , in addition to those of the classical DE, namely  $Cr$  and  $F$ . These parameters need to be accurately estimated to maximize their performance for the MTP. Therefore, several experiments based on different values for each parameter are investigated to find the best for HDE. For example, the optimal value for the parameter  $Cr$  is herein estimated by conducting several experiments under various numeric values, including 0.01, 0.1, 0.3, 0.5, 0.8, 0.9, and 0.99, and reporting the average fitness in Figure 4. Based on this figure, the best value for this parameter is 0.8. Similarly, the best-performing values for the other parameters, according to the results reported in Figures 4–7, have been estimated.

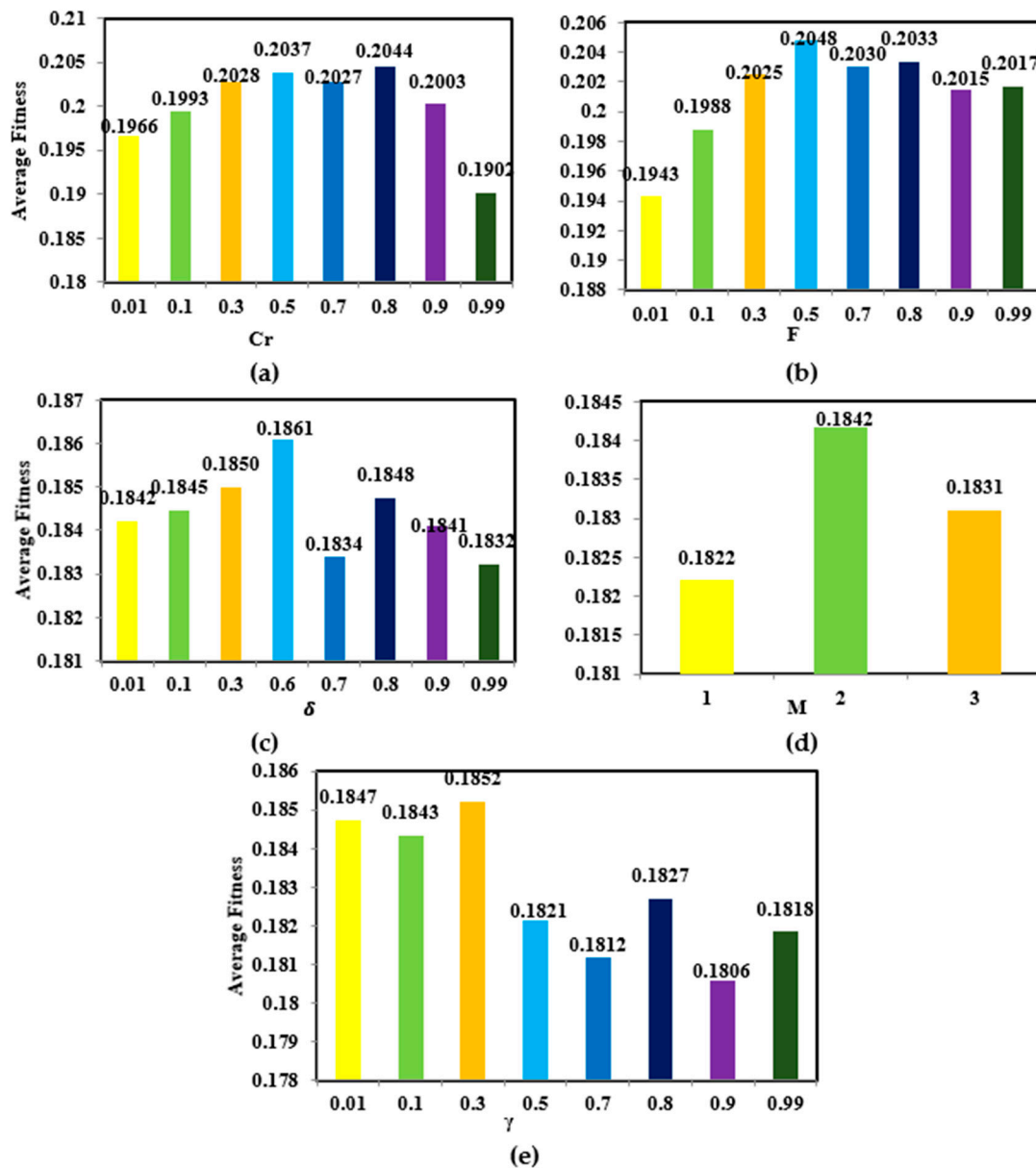
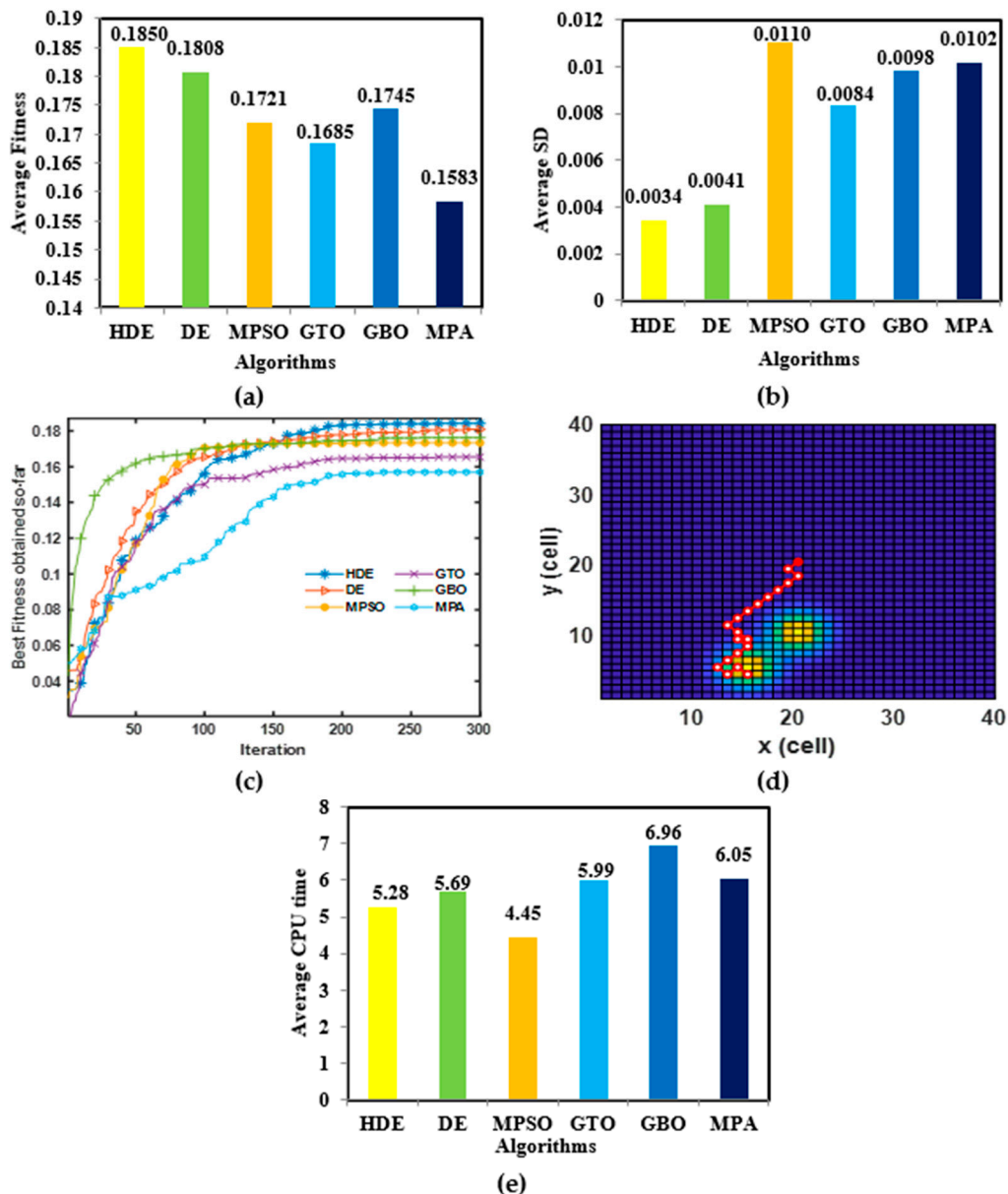


Figure 4. HDE’s parameter adjustment; (a) Tuning the parameter  $Cr$ ; (b) Tuning the parameter  $F$ ; (c) Tuning the parameter  $\delta$ ; (d) Tuning the parameter  $M$ ; (e) Tuning the parameter  $\gamma$ .

5.3. Performance Evaluation Using the First Scenario

Figure 5 portrays the average and SD values of the cumulative detection probability, representing the objective function, as well as the average convergence curve and average

execution time, attained by all methods after 30 replications over the first scenario. Furthermore, this figure includes the best search trajectory obtained by the proposed HDE from the initial UAV location to the highest probability regions, which might include the lost target. This figure discloses that HDE could provide the best outcomes for all performance indicators employed in comparison, except CPU time. In this metric, MPSO performs slightly better than HDE with a value of 4, followed by HDE in the second rank with a value of 5. Additionally, HDE could make an accurate search path to the area with the highest likelihood, which might be where the target has moved.



**Figure 5.** Comparison of the first scenario; (a) Average fitness; (b) Average SD; (c) Convergence curve; (d) Search Track of HDE; (e) Execution time.

5.4. Performance Evaluation Using the Second Scenario

In this section, the proposed HDE is further assessed using the second scenario to see its ability to explore the search map and reach the regions with the highest probability. Figure 6 portrays the average and SD of the fitness values that were achieved by all of the algorithms after being applied to this scenario 30 times independently. In addition to that,

the best search trajectory that the proposed HDE was able to find has been included in this figure. This figure makes it quite clear that HDE delivers the greatest results for all performance parameters that were utilized for comparison, except CPU time. With a score of 1.5, MPSO performs somewhat better than HDE according to this criterion, which places HDE in the second rank with a score of 3. Moreover, HDE can build an accurate search path to the region with the highest likelihood, which may be the location where the target has migrated.

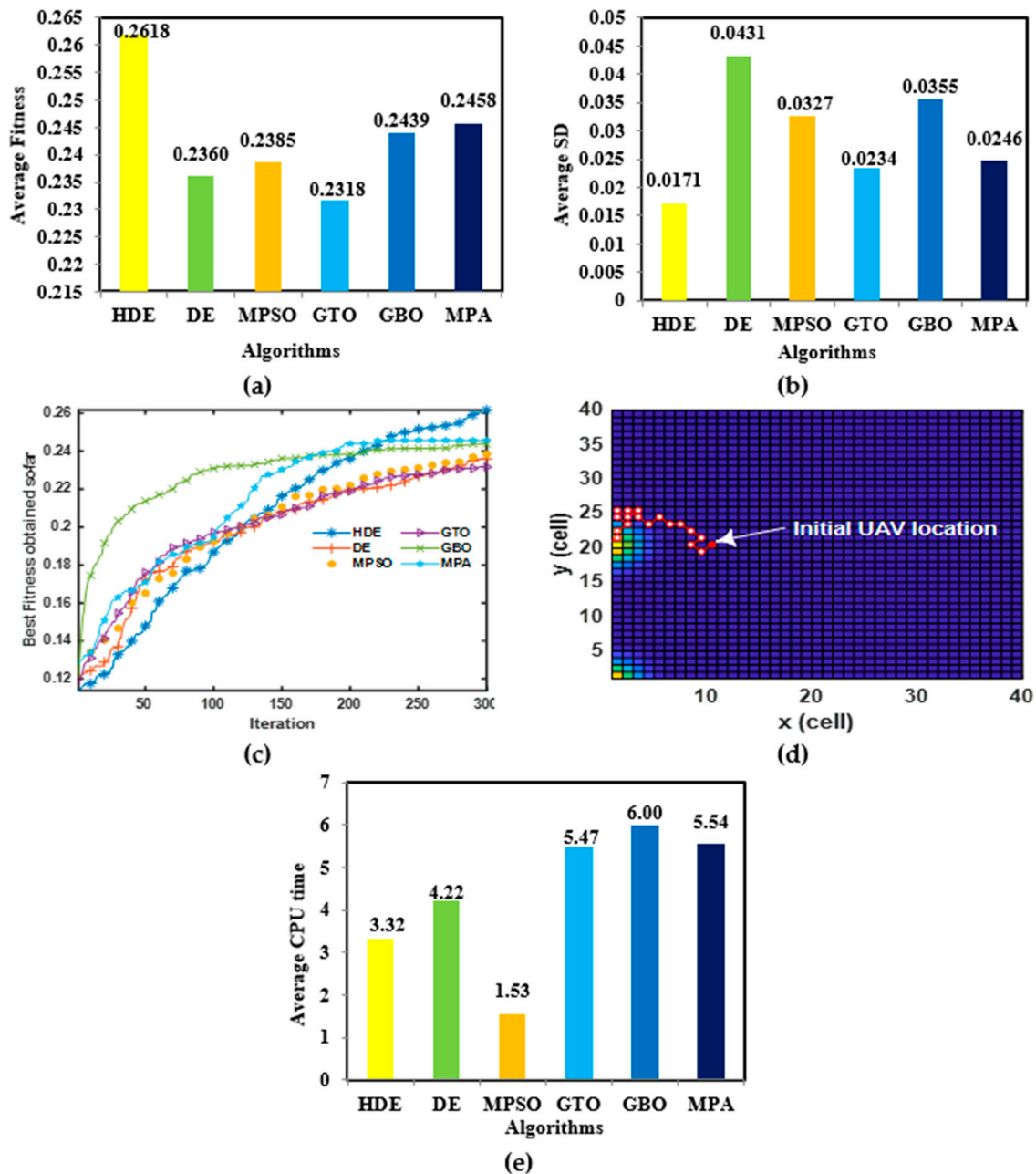


Figure 6. Comparison over the second scenario; (a) Average fitness; (b) Average SD; (c) Convergence curve; (d) Search Track of HDE; (e) Execution time.

### 5.5. Performance Evaluation Using the Third Scenario

This section is presented to further assess HDE by making use of the third scenario to determine how well it can explore the search map to get to the lost target. Figure 7 portrays the average and SD of the fitness values that were achieved by all of the algorithms after



30 independent runs over this scenario. In addition to that, the best search trajectory that the suggested HDE was able to find has been included in this figure. The comparison findings shown in this figure make it abundantly evident that HDE achieves the best results for all performance criteria that were used, except CPU time. According to this criterion, MPSO performs slightly better than HDE, earning a score of 2.5, which ranks HDE in second order behind HDE, which earned a score of 4.5. Moreover, HDE can construct an accurate search path to the region with the highest likelihood, which may be the location where the lost target has moved.

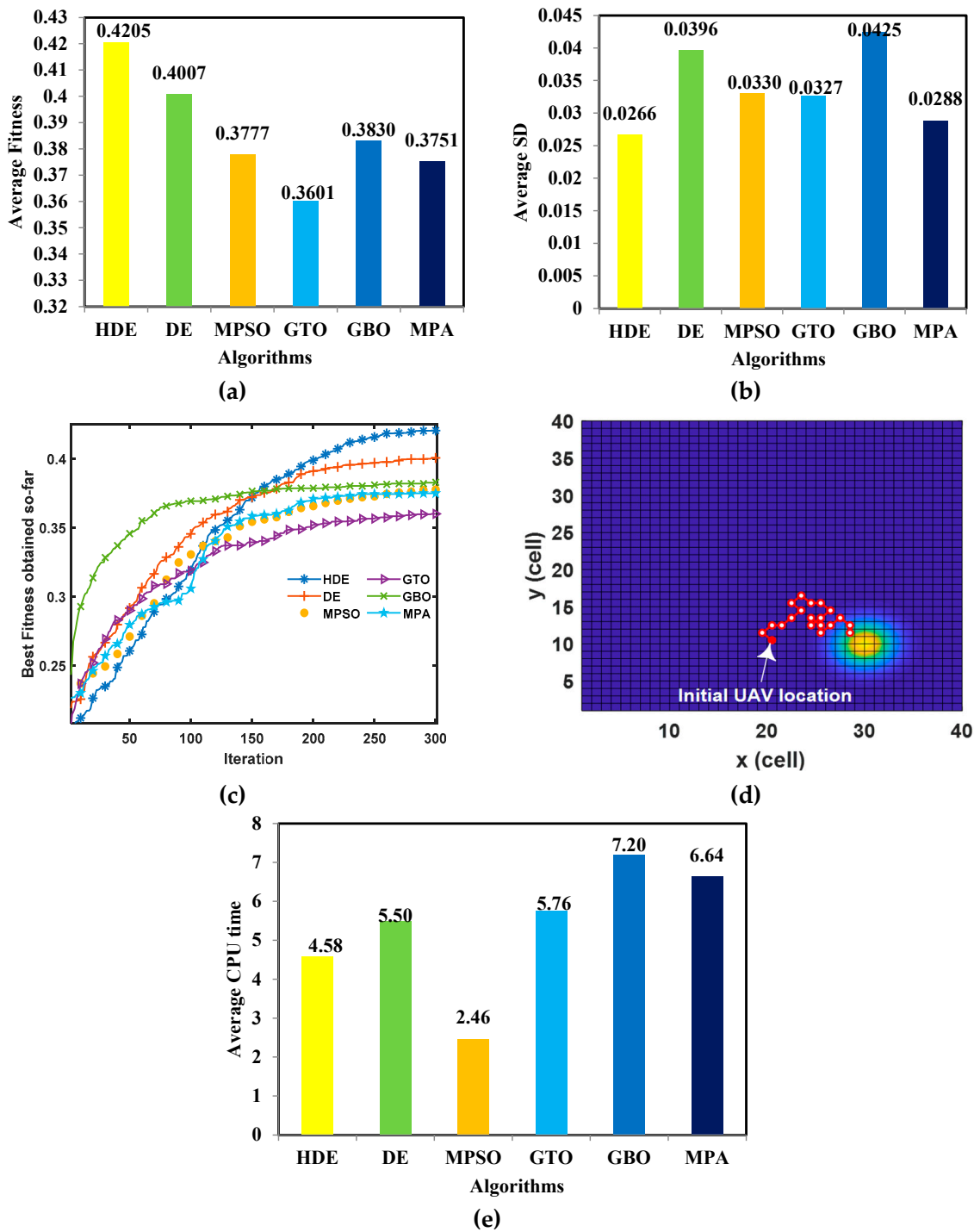


Figure 7. Comparison of the third scenario; (a) Average fitness; (b) Average SD; (c) Convergence curve; (d) Search Track of HDE; (e) Execution time.

5.6. Performance Evaluation Using the Fourth Scenario

Finally, the third scenario is used to test the proposed HDE’s search map exploration to find the missing target. Figure 8 shows the average and SD fitness values obtained by all algorithms after 30 runs over this scenario. Those runs are independently conducted due to the stochastic nature of the proposed HDE. This figure also shows the average execution time and average convergence curve, in addition to the best search trajectory found by the proposed HDE. This figure shows that HDE outperforms all performance metrics except CPU time. HDE can also create an accurate search path to the region with the highest probability of being where the missing target may be.

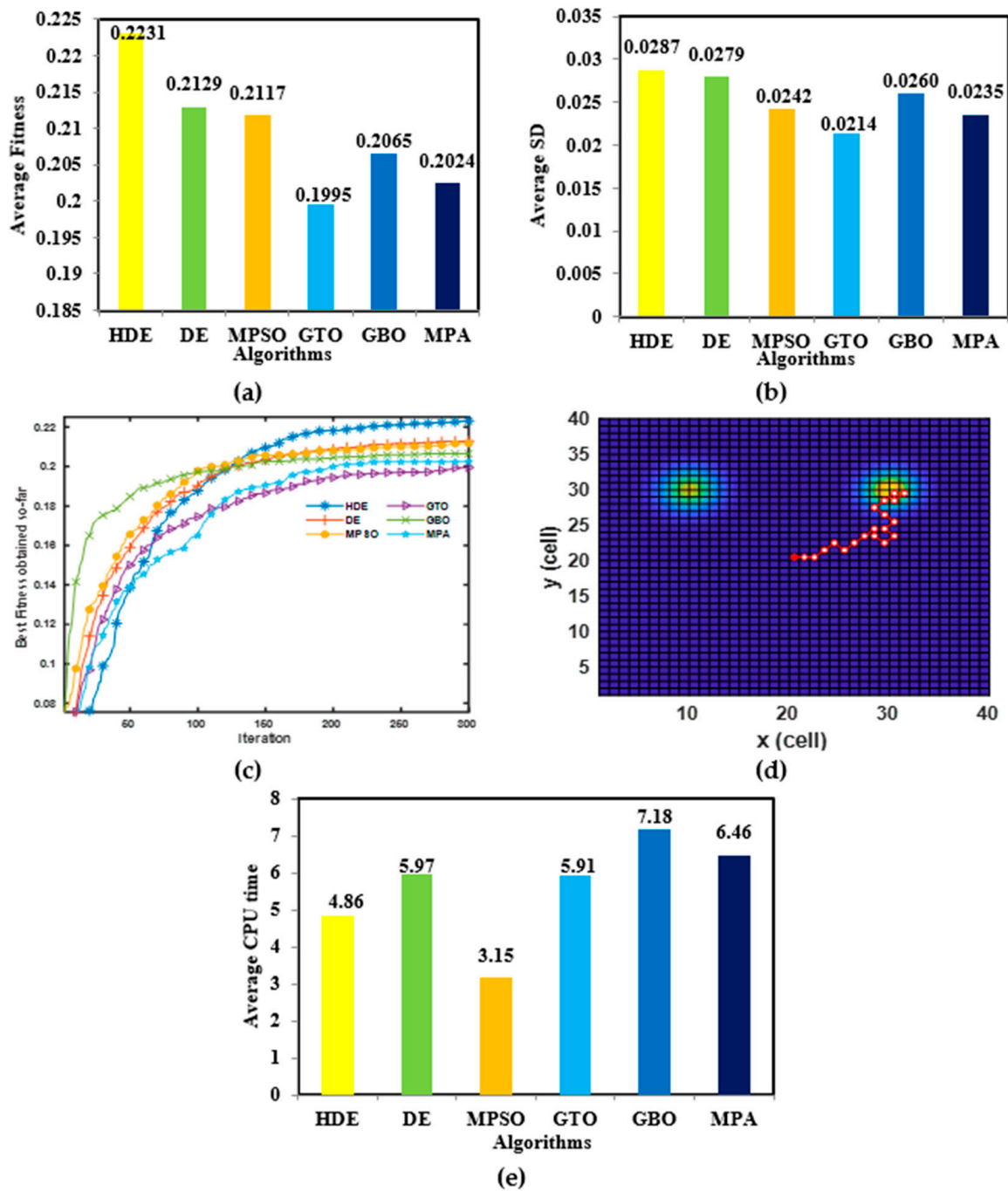


Figure 8. Comparison of the fourth scenario; (a) Average fitness; (b) Average SD; (c) Convergence curve; (d) Search Track of HDE; (e) Execution time.

## 6. Conclusions and Future Perspectives

Unmanned aerial vehicles (UAVs) require a path-planning technique that can promptly recalculate a viable and quasi-optimal path in flight if the target is moved during the mission. The complexity of the moving target problem (MTP) ranges from NP-hard to NEXP-complete because there are so many probabilistic variables involved. Therefore, it is challenging to detect a high-quality solution for this problem using traditional techniques. To address this problem, over the last few decades, researchers have turned to metaheuristic algorithms. However, those algorithms still suffer from some drawbacks, such as local minima, slow convergence speeds, or expensive computation costs. Therefore, this study designs a new hybrid algorithm named HDE for accurately tackling the MTP in a reasonable amount of time. This algorithm integrates differential evolution (DE) with two newly proposed updating schemes to improve its exploitation and exploration capabilities. The MTP can be recast as an optimization problem by applying Bayesian theory, with the detection likelihood of the target serving as the objective function. The HDE encodes the search path as a series of UAV motion pathways that develop during the optimization process. Four different scenarios with different difficulty levels are used to assess the HDE's effectiveness in comparison to several rival optimizers, including DE, GTO, MPA, GBO, and MPSO, in terms of several performance metrics. Those metrics include average fitness, SD, CPU time, convergence curve, and search path of HDE. The comparison reveals that the proposed HDE is superior concerning the vast majority of the performance criteria, except for the CPU time metric, where MPSO demonstrates a more effective reduction. As a result, HDE is considered a strong alternative to finding more accurate paths that could increase the detection probability of moving targets.

Our future perspectives include applying HDE to several other complex optimization problems to further show its exploration and exploitation operator, in addition to presenting a novel technique considering the optimization of other UAV criteria such as reducing the flight time in some applications such as surveillance and rescue and applying the Pareto optimality for multi-objective navigation in a dynamic environment. Furthermore, the uniform distribution will be used instead of the normal distribution to investigate its effect on solving the MTP.

**Author Contributions:** Conceptualization, M.A.-B., R.M., I.M.H., A.M.A. and K.M.S.; Methodology, M.A.-B., R.M., A.M.A. and K.M.S.; Software, M.A.-B. and R.M.; Validation, M.A.-B., R.M., I.M.H., A.M.A. and K.M.S.; Formal analysis, M.A.-B., R.M., I.M.H., A.M.A. and K.M.S.; Investigation, M.A.-B., I.M.H., A.M.A. and K.M.S.; Data curation, I.M.H.; Writing—original draft, M.A.-B. and R.M.; Writing—review & editing, R.M., M.A.-B., I.M.H. and A.M.A.; Visualization, R.M., M.A.-B. and I.M.H.; Supervision, M.A.-B. Project administration, I.M.H.; Funding acquisition, I.M.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** The authors extend their appreciation to the Deputyship for Research and Innovation, Ministry of Education in Saudi Arabia, for funding this research work through project no. (IFKSUOR3-037-2).

**Data Availability Statement:** The datasets generated during and/or analyzed during the current study are not publicly available due to the privacy-preserving nature of the data but are available from the corresponding author upon reasonable request.

**Acknowledgments:** The authors would like to express their gratitude to the anonymous referees, Chief Editor, and support editors for their helpful feedback and propositions that helped to enhance the quality of this research. The authors extend their appreciation to the Deputyship for Research and Innovation, Ministry of Education in Saudi Arabia, for funding this research work through project no. (IFKSUOR3-037-2).

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

## Nomenclature

### List of notations

$v_0$	The initial location of moving target
$b(v_0)$	Initial belief map
$T_t$	Detection event of the target at time $t$
$\eta_t$	Normalization factor
$S$	Search space
$\vec{O}$	Decoded search path
$P_t$	Cumulative probability
$F$	Scaling Factor
$Cr$	Crossover probability
$\vec{v}$	Mutant solution
$\vec{u}$	Trial solution
$\vec{x}$	Current solution
$l$	The current count of function evaluation
$T_m$	Maximum number of function evaluations
$N$	Population size
$\alpha_t$	The direction of the motion at time $t$
$\rho_t$	The amplitude of the motion at time $t$
$Cr'$	Adaptive crossover probability
$n$	Dimension size

### List of acronyms

UAVs	Unmanned aerial vehicles
CEO	Cross-entropy optimization
BOA	Bayesian optimization algorithm
HDE	Hybrid differential evolution
MTP	Moving target problem
DE	Differential evolution
GTO	Gorilla troops optimizer
GBO	Gradient-based optimizer
MPA	Marine Predators Algorithm
GWO	Grey wolf optimizer
MPSO	Motion-encoded particle swarm optimization

## References

- Zhu, H.; Wang, Y.; Li, X. UCAV Path Planning for Avoiding Obstacles using Cooperative Co-evolution Spider Monkey Optimization. *Knowl.-Based Syst.* **2022**, *246*, 108713. [[CrossRef](#)]
- Du, N.; Zhou, Y.; Deng, W.; Luo, Q. Improved chimp optimization algorithm for three-dimensional path planning problem. *Multimed. Tools Appl.* **2022**, *81*, 27397–27422. [[CrossRef](#)]
- Mohsan, S.A.H.; Othman, N.Q.H.; Li, Y.; Alsharif, M.H.; Khan, M.A. Unmanned aerial vehicles (UAVs): Practical aspects, applications, open challenges, security issues, and future trends. *Intell. Serv. Robot.* **2023**, *16*, 109–137. [[CrossRef](#)] [[PubMed](#)]
- Pitre, R.R.; Li, X.R.; Delbalzo, R. UAV Route Planning for Joint Search and Track Missions—An Information-Value Approach. *IEEE Trans. Aerosp. Electron. Syst.* **2012**, *48*, 2551–2565. [[CrossRef](#)]
- Wu, Y.; Liang, T.; Gou, J.; Tao, C.; Wang, H. Heterogeneous Mission Planning for Multiple UAV Formations via Metaheuristic Algorithms. *IEEE Trans. Aerosp. Electron. Syst.* **2023**, 1–16. [[CrossRef](#)]
- Aggarwal, S.; Kumar, N. Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges. *Comput. Commun.* **2020**, *149*, 270–299. [[CrossRef](#)]
- Qadir, Z.; Zafar, M.H.; Moosavi, S.K.R.; Le, K.N.; Mahmud, M.A.P. Autonomous UAV Path-Planning Optimization Using Metaheuristic Approach for Predisaster Assessment. *IEEE Internet Things J.* **2021**, *9*, 12505–12514. [[CrossRef](#)]
- Yang, P.; Tang, K.; Lozano, J.A.; Cao, X. Path Planning for Single Unmanned Aerial Vehicle by Separately Evolving Waypoints. *IEEE Trans. Robot.* **2015**, *31*, 1130–1146. [[CrossRef](#)]
- Roberge, V.; Tarbouchi, M.; Labonte, G. Fast Genetic Algorithm Path Planner for Fixed-Wing Military UAV Using GPU. *IEEE Trans. Aerosp. Electron. Syst.* **2018**, *54*, 2105–2117. [[CrossRef](#)]
- Phung, M.D.; Ha, Q.P. Motion-encoded particle swarm optimization for moving target search using UAVs. *Appl. Soft Comput.* **2020**, *97*, 106705. [[CrossRef](#)]
- Alanezi, M.A.; Bouchekara, H.R.E.H.; Shahriar, M.S.; Sha'aban, Y.A.; Javaid, M.S.; Khodja, M. Motion-Encoded Electric Charged Particles Optimization for Moving Target Search Using Unmanned Aerial Vehicles. *Sensors* **2021**, *21*, 6568. [[CrossRef](#)] [[PubMed](#)]

12. Perez-Carabaza, S.; Besada-Portas, E.; Lopez-Orozco, J.A.; de la Cruz, J.M. Ant colony optimization for multi-UAV minimum time search in uncertain domains. *Appl. Soft Comput.* **2018**, *62*, 789–806. [[CrossRef](#)]
13. Alanezi, M.A.; Bouchekara, H.R.E.H.; Apalara, T.A.-A.; Shahriar, M.S.; Sha'Aban, Y.A.; Javaid, M.S.; Khodja, M.A. Dynamic Target Search Using Multi-UAVs Based on Motion-Encoded Genetic Algorithm with Multiple Parents. *IEEE Access* **2022**, *10*, 77922–77939. [[CrossRef](#)]
14. Ma, T.; Wang, Y.; Li, X. Convex combination multiple populations competitive swarm optimization for moving target search using UAVs. *Inf. Sci.* **2023**, *641*, 119104. [[CrossRef](#)]
15. Jarray, R.; Bouallègue, S.; Rezk, H.; Al-Dhaifallah, M. Parallel Multiobjective Multiverse Optimizer for Path Planning of Unmanned Aerial Vehicles in a Dynamic Environment with Moving Obstacles. *Drones* **2022**, *6*, 385. [[CrossRef](#)]
16. Ait-Saadi, A.; Meraihi, Y.; Soukane, A.; Ramdane-Cherif, A.; Gabis, A.B. A novel hybrid Chaotic Aquila Optimization algorithm with Simulated Annealing for Unmanned Aerial Vehicles path planning. *Comput. Electr. Eng.* **2022**, *104*, 108461. [[CrossRef](#)]
17. Wu, Y.; Low, K.H.; Lv, C. Cooperative Path Planning for Heterogeneous Unmanned Vehicles in a Search-and-Track Mission Aiming at an Underwater Target. *IEEE Trans. Veh. Technol.* **2020**, *69*, 6782–6787. [[CrossRef](#)]
18. Huang, Y.; Wang, H.; Yao, P. Energy-optimal path planning for Solar-powered UAV with tracking moving ground target. *Aerosp. Sci. Technol.* **2016**, *53*, 241–251. [[CrossRef](#)]
19. Wu, Y.; Low, K.H. Route Coordination of UAV Fleet to Track a Ground Moving Target in Search and Lock (SAL) Task Over Urban Airspace. *IEEE Internet Things J.* **2022**, *9*, 20604–20619. [[CrossRef](#)]
20. Ghdiri, O.; Jaafar, W.; Alfattani, S.; Ben Abderrazak, J.; Yanikomeroglu, H. Offline and Online UAV-Enabled Data Collection in Time-Constrained IoT Networks. *IEEE Trans. Green Commun. Netw.* **2021**, *5*, 1918–1933. [[CrossRef](#)]
21. Agrawal, P.; Ganesh, T.; Mohamed, A.W. Chaotic gaining sharing knowledge-based optimization algorithm: An improved metaheuristic algorithm for feature selection. *Soft Comput.* **2021**, *25*, 9505–9528. [[CrossRef](#)]
22. Wang, Y.; Li, K.; Han, Y.; Ge, F.; Xu, W.; Liu, L. Tracking a dynamic invading target by UAV in oilfield inspection via an improved bat algorithm. *Appl. Soft Comput.* **2020**, *90*, 106150. [[CrossRef](#)]
23. Wu, Y.; Low, K.H.; Pang, B.; Tan, Q. Swarm-Based 4D Path Planning for Drone Operations in Urban Environments. *IEEE Trans. Veh. Technol.* **2021**, *70*, 7464–7479. [[CrossRef](#)]
24. Dewangan, R.K.; Shukla, A.; Godfrey, W.W. Three dimensional path planning using Grey wolf optimizer for UAVs. *Appl. Intell.* **2019**, *49*, 2201–2217. [[CrossRef](#)]
25. Aslan, S.; Erkin, T. A multi-population immune plasma algorithm for path planning of unmanned combat aerial vehicle. *Adv. Eng. Inform.* **2023**, *55*, 101829. [[CrossRef](#)]
26. Rajmohan, S.; Ramasubramanian, N. Improved Symbiotic organisms search for path planning of unmanned combat aerial vehicles. *J. Ambient. Intell. Humaniz. Comput.* **2023**, *14*, 4289–4311. [[CrossRef](#)]
27. Yao, P.; Wang, H.; Ji, H. Multi-UAVs tracking target in urban environment by model predictive control and Improved Grey Wolf Optimizer. *Aerosp. Sci. Technol.* **2016**, *55*, 131–143. [[CrossRef](#)]
28. Li, J.; Deng, G.; Luo, C.; Lin, Q.; Yan, Q.; Ming, Z. A Hybrid Path Planning Method in Unmanned Air/Ground Vehicle (UAV/UGV) Cooperative Systems. *IEEE Trans. Veh. Technol.* **2016**, *65*, 9585–9596. [[CrossRef](#)]
29. Chowdhury, A.; De, D. RGSO-UAV: Reverse Glowworm Swarm Optimization inspired UAV path-planning in a 3D dynamic environment. *Ad Hoc Netw.* **2023**, *140*, 103068. [[CrossRef](#)]
30. Hu, G.; Zhong, J.; Wei, G. SaCHBA\_PDN: Modified honey badger algorithm with multi-strategy for UAV path planning. *Expert Syst. Appl.* **2023**, *223*, 119941. [[CrossRef](#)]
31. Aslan, S.; Erkin, T. An immune plasma algorithm based approach for UCAV path planning. *J. King Saud Univ.-Comput. Inf. Sci.* **2023**, *35*, 56–69. [[CrossRef](#)]
32. Singh, M.K.; Choudhary, A.; Gulia, S.; Verma, A. Multi-objective NSGA-II optimization framework for UAV path planning in an UAV-assisted WSN. *J. Supercomput.* **2023**, *79*, 832–866. [[CrossRef](#)]
33. Zhang, C.; Zhou, W.; Qin, W.; Tang, W. A novel UAV path planning approach: Heuristic crossing search and rescue optimization algorithm. *Expert Syst. Appl.* **2023**, *215*, 119941. [[CrossRef](#)]
34. Wang, Y.; Li, K.; Han, Y.; Yan, X. Distributed multi-UAV cooperation for dynamic target tracking optimized by an SAQPSO algorithm. *ISA Trans.* **2022**, *129*, 230–242. [[CrossRef](#)]
35. Storn, R.; Price, K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [[CrossRef](#)]
36. Abdollahzadeh, B.; Gharehchopogh, F.S.; Mirjalili, S. Artificial gorilla troops optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems. *Int. J. Intell. Syst.* **2021**, *36*, 5887–5958. [[CrossRef](#)]
37. Ahmadianfar, I.; Bozorg-Haddad, O.; Chu, X. Gradient-based optimizer: A new metaheuristic optimization algorithm. *Inf. Sci.* **2020**, *540*, 131–159. [[CrossRef](#)]
38. Faramarzi, A.; Heidarinejad, M.; Mirjalili, S.; Gandomi, A.H. Marine Predators Algorithm: A nature-inspired metaheuristic. *Expert Syst. Appl.* **2020**, *152*, 113377. [[CrossRef](#)]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.