

Article

A Novel Bat Algorithm with Asymmetrical Weighed Variational Method in the Path Planning of UAVs

Xin Cao ^{1,*}, Chenyi Wang ¹ and Weiping Li ²¹ School of Information Engineering, Southwest University of Science and Technology, Mianyang 621010, China² School of Information Engineering, East China Jiaotong University, Nanchang 330013, China

* Correspondence: caoxin@swust.edu.cn

Abstract: In this paper, a novel bat algorithm with an asymmetrical weighed variational method (AWVM-BA) is proposed. The proposed algorithm employs the BA with a point-to-point modified asymmetrical variation above the three-dimensional flying region, which treats the space as sets of geodesics in a second order Euclidean weighed warped space. Mutation and the local selection procedure can be avoided at the same time, which solves the problem of a local optimum in concave regions. As shown in the results, the proposed algorithm does not have much impact on the calculation complexity and time in convex regions. It can greatly reduce the calculation time and avoid local optimization in concave regions. The disadvantage of the proposed algorithm is that the iteration number increases comparatively faster with the increase in the deviation of the wind speed. Therefore, it requires a higher hardware calculation ability.

Keywords: bat algorithm; asymmetrical weighed variation; UAV; path planning; forest fire; concave region; aerial surveillance



Citation: Cao, X.; Wang, C.; Li, W. A Novel Bat Algorithm with Asymmetrical Weighed Variational Method in the Path Planning of UAVs. *Symmetry* **2023**, *15*, 1265. <https://doi.org/10.3390/sym15061265>

Academic Editor: Aviv Gibali

Received: 31 May 2023

Revised: 13 June 2023

Accepted: 14 June 2023

Published: 15 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

Forests are of great importance to the sustainable development of humankind. Forest fires cause great damage to the forest environment and human society. In recent years, forest fires posed great dangers to the world and seriously threatened the ecological environment and human safety [1–8]. Forest fires often occur in high temperatures and dry weather, especially in oleaginous plants. When they occur in windy weather, fires tend to be severe and spread in multiple lines and locations with the changing wind direction. In addition, forest fires often occur in mountainous regions with high altitudes and complex terrain, making it difficult for firefighters and equipment to extinguish the fires [9–14]. Forest protection has made great progress, and unmanned aerial vehicles (UAVs) are being used in forest fire detection. UAVs have unique advantages in forest fire prevention and suppression, such as high temperature adaptation to the environment, detection of multiple physical features, high movement speed, monitoring of a large area, higher safety, and faster response to a task [15–20]. Therefore, a highly accurate navigation algorithm is one of the most important components of UAV deployment, where trajectory planning has become the core issue. An efficient and reliable trajectory planning algorithm can enable UAVs to achieve safety while performing their tasks with the shortest flight time and trajectory and the least energy loss. When performing firefighting tasks, the UAV should quickly plan a trajectory from the starting point to the source of the fire while avoiding obstacles.

In recent years, researchers worked extensively on this topic. Many algorithms, such as particle swarm algorithms [21–26], ant colony algorithms [27–32], genetic algorithms [33–38], and bat algorithms [39–44], have made great developments and attracted more and more attention, especially in the field of solving path planning problems in obstacle environments. UAVs perform firefighting tasks in forest fire areas, and the actual trajectory of UAVs in forest firefighting must be processed based on the appropriate trajectory generation

algorithms in conjunction with the characteristics of the UAV itself and the environmental characteristics to ensure that the final trajectory matches the dynamics of the UAV [45–48]. For reasons of flight safety, minimum energy consumption, and other requirements, it is often necessary to smooth the trajectory [49–52]. The reason for a smooth UAV trajectory is that the UAV should meet the various constraints of the UAV in the process of trajectory planning [53–56]. On the one hand, this can ensure that the UAV does not suffer any accidents during the flight and meets its physical constraints. On the other hand, the constraints should meet the real-time requirements and minimum energy consumption of high-speed flight. Therefore, intelligent algorithms have attracted the attention of more and more scientists due to their bionic properties and good convergence characteristics. The algorithms often become involved in the phenomenon of a local optimum [57–61]. The obvious disadvantage of the ant colony algorithm is that the algorithm often requires many iterations and is prone to stagnation. The bat colony algorithm has been used in many fields, such as image processing, data processing, continuous optimization problems, path planning of unmanned vehicles, pattern recognition, etc. [62–68]. Compared to many other intelligent algorithms, the bat algorithm has the advantages of simple modelling, fewer parameters, and fast convergence speed. However, the disadvantage is that it can easily fall into a local optimum during development.

It can be seen from the above that the bat algorithm has a wide applicability; its research and application have attracted more and more attention, and certain theoretical achievements have been reached. However, as a new type of meta-heuristic algorithm, the bat algorithm has many key problems that need to be solved. The bat algorithm easily falls into local optimization and stagnation. Avoiding the precocious convergence of the algorithm is one of the problems that cannot be ignored. Considering the complexity of the problem and the uncertainty of the objective function, each optimization algorithm has its inherent advantages and disadvantages, but it cannot only use a single optimization calculation to solve the current optimization problem, let alone find a satisfactory solution. In recent years, more and more scholars have focused their research on discrete and multi-objective bat algorithms and achieved the purpose of improving the feasible solution of the algorithm by introducing chaos, Levy flight, or a combination with other algorithms in terms of initializing populations, habitat selection, and control parameters [69–76]. Therefore, it is a development trend to combine the bat algorithm with other methods to solve the disadvantages of the algorithm, and its specific improvement strategy can be combined with the application problems in different fields to flexibly solve the problem according to the actual situation so as to improve the convergence speed and accuracy of the algorithm, the overall performance of the algorithm, and its ability to conduct a local and global search.

To overcome this disadvantage, this paper proposes a novel AWVM-BA algorithm. The proposed algorithm integrates the asymmetrical modified mathematical variation on concave manifolds. Through integration, the local optimization and convergence speed were improved in gradient regions, especially for concave land shapes. In consideration of the real-world application, we added the wind condition as a continuous Brownian random process. Additionally, fire spread under the influence of wind is also modeled to emulate the real-world situation. In windy conditions, the Monte Carlo method with 10,000 independent experiments was carried out, and arithmetic averages in terms of the flight length, iteration number, flight cost, and target function value (TFV) were calculated. As shown in the results, the proposed algorithm has a better trajectory planning ability with less convergence iterations. With the increase in the standard deviation of the wind speed, the iteration number increases more sharply compared with some other algorithms. The limitation of the proposed algorithm is that the iteration number increases faster compared with some other algorithms due to the sub-dimension surface variation process.

2. Theory and Analysis

2.1. The Framework, Advantages, and Drawbacks of BA in Forest Fire Detection

The bat algorithm is named after the living habits of bats. Bats can fly freely at night. The main reason for this is that bats send out ultrasonic signals through their mouths during flight. When the ultrasonic signals are reflected by obstacles or prey, they produce reflected signals. The reflected signals are transmitted through the air. The ears of bats can capture these signals and determine the distance between the obstacle or the prey, so the current position of the target is determined. Additionally, the mechanism for bats is echolocation. Bats use echolocation to avoid obstacles and hunt quickly and accurately in the dark. When the return frequency of the ultrasonic wave emitted by the bat is relatively suitable for the distance between the bat and the target point, the bat makes corresponding actions toward the target. Based on this mechanism, the BA can easily fall into a local optimum in mathematically concave regions because the reflected ultrasonic signal can concentrate to a small area like a concave mirror. The concentration of reflected waves gives the bat the illusion that it is in the global optimum when it falls within one of the local optimums.

In target detection, UAVs can fly in groups in close cooperation when the targets already exist, and the speed of detection is the first priority instead of the saving cost. This is often the case in military detection or object destruction. Many researchers have focused on the study of UAV cooperation during known target detection. However, for forest fire detection, one UAV often surveils a particular area of land since the possibility of a forest fire is much lower compared with a known target. Sending too many UAVs will increase the cost dramatically and is not necessary. Reducing the cost is the first priority instead of the speed of detection in forest fire detection. For example, for a small area of land, the area can be divided into four parts, as shown in Figure 1. Each part is dispatched with a UAV. They work independently in their respective areas and do not cooperate with each other. Many papers on UAV forest fire detection treat UAVs as independent individuals. The real-world factors that affect UAVs are wind, rain, and snow. In rainy or snowy weather, UAVs are not often used, since water or snow may cause mechanical and electrical damage to the machine. The only remaining factor is wind. We performed the simulation under different deviations of wind speed. Additionally, we treated the wind as a three-dimensional vector instead of a scalar. In Section 3, the experiment environment is very close to real-world settings.

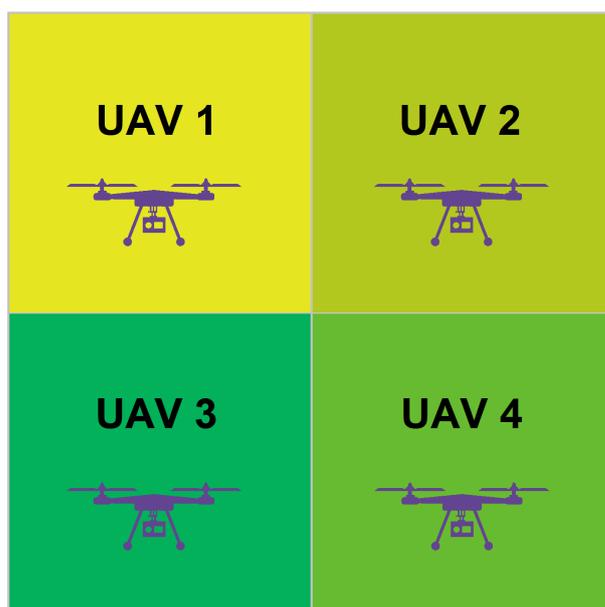


Figure 1. An example of mission division of multiple UAVs in forest fire surveillance.

When the bat initially searches for prey, it emits ultrasonic waves with a high sound wave volume and a low sound wave frequency, so that it can find out in which small area of the whole environment the prey exists, and then reduce the sound wave volume. It then increases the frequency of the sound waves, searches further in this small area, and finds the prey accurately. Using the bat algorithm to plan the UAV's trajectory can approximate the UAV as a bat, allowing the UAV to simulate the process of bats looking for prey and avoiding obstacles during flight. In this process, idealized rules can be set as follows. First, the bat perceives the location of the obstacles and target points through echolocation to distinguish the obstacles and target points. Then, the bat flies at the position \vec{p} at the speed of \vec{v} , where \vec{p} and \vec{v} are three-dimensional vectors. The minimum frequency is f_{min} , and the wavelength and pulse volume can be adjusted. At the same time, the bat can autonomously adjust the sound wave frequency according to the position of the bat itself, the obstacles, and the target point. Assuming that the impulse frequency changes from the maximum f_{max} to the minimum f_{min} , the change interval is related to the real-time problem. The specific rules of the bat algorithm can be described as a multi-dimensional search area to obtain the optimal solution through continuous iterations. In the iterative process, the position $\vec{p}[n]$ and velocity $\vec{v}[n]$ of the n -th generation bat are updated according to the rules described as follows:

$$f[n] = K[n](f_{max} - f_{min}) + f_{min} \quad (1)$$

$$\vec{v}[n] = \vec{v}[n-1] + f[n](\vec{p}_{new}[n-1] - \vec{p}_o) \quad (2)$$

$$\vec{p}_{old}[n] = \vec{p}_{new}[n-1] + \vec{v}[n]t_g \quad (3)$$

where $f[n]$ represents the sound frequency at n -th iteration; f_{min} and f_{max} are the minimum and the maximum for the initial setting of the frequency; $K[n]$ is a uniformly distributed random variable in the range of $[0, 1]$; \vec{p}_{old} and \vec{p}_{new} are the individual position before and after the update, and the update function is given in (4); \vec{p}_o is the current optimal solution; and t_g is the quantized time of each step. When the bat finds the approximate range of its prey, it indicates that the local optimal strategy can be used to find the global optimal solution near the current optimal individual as

$$\vec{p}_{new}[n] = \vec{p}_{old}[n] + (\vec{e}_x\alpha_x[n] + \vec{e}_y\alpha_y[n] + \vec{e}_z\alpha_z[n])u_1[n] \quad (4)$$

where $u_1[n] \in U(-1, 1)$, $\alpha_x[n]$, $\alpha_y[n]$, and $\alpha_z[n]$ are the volumes of the sound wave in the x , y , and z directions, respectively. The search can be carried out in a local range. The rules for the adjustment of the volumes $\alpha_x[n]$, $\alpha_y[n]$, and $\alpha_z[n]$ and the pulse indicator $r[n]$ can be described as

$$\begin{cases} \alpha_x[n] = u_2\alpha_x[n-1] \\ \alpha_y[n] = u_2\alpha_y[n-1] \\ \alpha_z[n] = u_2\alpha_z[n-1] \end{cases} \quad (5)$$

$$r[n] = r[0](1 - e^{-\beta n}) \quad (6)$$

where β is the wave propagation constant and $u_2 \in U(0, 1)$. When the iteration is close to infinity, the pulse indicator gets closer to $r[0]$.

It can be observed from Formulas (1) to (6) that this algorithm is sensitive to four major parameters. If u_2 is too small, the algorithm converges too quickly, which leads to premature problems, and the obtained solution is not optimal. If u_2 is too large, the search time increases and the search result is close to the optimal solution, but the algorithm's calculation time becomes longer. If β is large, the current position of the bat is updated according to the formula. Otherwise, the search continues near the current optimal position. Since it is affected by the current optimal position, the search easily falls into the local optimal position. Therefore, together, u_2 and β determine the search accuracy and convergence

speed of the bat algorithm. The search process of the bat algorithm includes a global search and a local search. Although it has a good global optimization performance, it also easily falls into local optimization, which greatly affects its convergence speed and accuracy.

During the flight path, the UAVs are required to find a path from the starting point to the end point with certain constraints. In [77], the consumption cost is the energy that is consumed to keep the UAV at certain height. Then, the target function can be given as

$$C[n] = \gamma C_L[n] \quad (7)$$

where γ is a positive scaling factor. Since it is different for UAVs that move in the xy directions and the z direction, we modified the distance function as

$$C_L[n] = \int_{(x_s[n], y_s[n], z_s[n])}^{(x_e[n], y_e[n], z_e[n])} \sqrt{dx^2 + dy^2 + kdz^2} \quad (8)$$

where k is the coefficient to characterize the extra efforts when the UAV moves in the z direction. $(x_s[n], y_s[n], z_s[n])$ and $(x_e[n], y_e[n], z_e[n])$ are the coordinates of the starting point and ending point, respectively. The target function employed is the Euclidian distance function, which can be expressed as follows :

$$f_t[n] = C[n] - \gamma \left\{ \begin{array}{l} (x_e[n] - x_s[n])^2 + (y_e[n] - y_s[n])^2 \\ + k(z_e[n] - z_s[n])^2 \end{array} \right\}^{\frac{1}{2}} \quad (9)$$

Due to the small number of parameters and the cooperation mechanism, the bat algorithm has strong flexibility and is easy to combine with other technologies so as to improve the performance of the original algorithm and, at the same time, solve complex continuous optimization problems and discrete optimization problems, which shows that the bat algorithm has a wide applicability. The framework of the bat algorithm is as follows. The first step is to initialize the parameters of the algorithm, that is, the population of the bats, the iteration number, the target function, the positions of the bats, the flying speed, the frequency, and the volume. Then, through iteration, the optimum positions of the bats are selected. Additionally, the velocities of the bats are updated. A random number with Gaussian distribution u_1 is generated at this point. If $u_1 > r[n]$, then a global search is performed to select the bat in the optimum position at the given search area. If this condition is not satisfied, then the position is updated based on the principle. At this time, a second random number u_2 is produced. If $u_2 < r[n]$ and the target function is smaller than the threshold f_{th} , then the position is acceptable at this stage, or $r[n]$ should be adjusted. This is the step of one recursive procedure in the whole process. If the convergence condition is reached, then the program stops, and the optimum path can be obtained. If the convergence cannot be achieved at the maximum iteration step N_{max} , then the program also stops and produces a path. After many steps of iteration, the optimum solution often cannot be obtained. The algorithm itself also has some unavoidable defects and easily falls into local optimization and low optimization accuracy in the later stage, which affect the search ability of the bat algorithm and restrict the ability and scope of the algorithm to solve forest fire detection problems. The flow of MSC-BA algorithm is shown in Algorithm 1.

Generally, the global optimization method is used to optimize the current bat population, which enhances the local search performance of the bat algorithm. Some improved algorithms are more suitable for solving high-dimensional problems, and some algorithms are only suitable for certain specific problems and assumptions, and cannot effectively solve all optimization problems. Therefore, it is very necessary to continue to improve the existing bat algorithm or propose new improvement methods to solve more optimization problems and management problems, and it also has certain practical significance. Compared with other intelligent algorithms, the BA has many advantages. The algorithm has fewer parameters and a fast convergence. The algorithm also has significant disadvantages. Since the bat algorithm does not have a mutation mechanism in the population, the lack

of diversity in the population causes the bats to think that the local optimal solution is the global optimal solution when they find the local optimal solution, so they fall into premature areas. So, the problem for the BA is that the global convergence is based on probability and a strong convergence.

Algorithm 1 The MSC-BA algorithm

Begin
 Initialize $f_{min}, f_{max}, \vec{p}[0], \vec{v}[0], t_g, K[0], r[0], N_{max}, f_{th}, \alpha_x[n], \alpha_y[n], \alpha_z[n], \beta$
for $i = 1:m$ **do**
 for $n = 1:N_{max}$ **do**
 compute frequency form (1)
 compute solutions from (2) and (3)
 generate u_1
 if ($u_1 > r[n]$) **then**
 calculate $\vec{v}[n+1]$ and $r[n+1]$ from (5) and (6)
 calculate cost from (7)
 produce $f_t[n]$ from (9)
 generate u_2
 end if
 if ($u_2 < r[n]$ and $f_t[n] < f_{th}$) **then**
 accept the current position as the optimal position
 end if
 end for
end for
End

2.2. Improvement of BA through Weighted Variation

The improvement of the bat algorithm itself is mainly achieved by adjusting its own parameters so that the algorithm can jump out of the local optimum. Therefore, the algorithm can solve the problem faster and more accurately. It assumes that the bats are randomly distributed and widely spaced. In practical problems, most bats do not meet this requirement. Second, the locally optimal solution may be missed if the step size is too large. When the algorithm is used for each cycle of the algorithm, it is necessary to judge the gap between the fitness and the optimal solution, which increases the time and complexity of the algorithm.

Topological manifold refers to the topological space of the local isomorphism in the Euclidean space, and in general, in order to adapt to the needs, it is often required that this topological space satisfies the separation property and has countable topological bases. This refers to the existence of a neighborhood of any point, which is isomorphic to the Euclidean space. The topological manifold considered here is boundless, and for some marginal manifolds, it is not possible to require that their boundary points have neighborhood homology in the Euclidean space, so in this case, the object of the same morphism will replace the entire Euclidean space with the half space of the Euclidean space. On a curved surface, a geodesic has similar properties. As shown in Figure 2, the variation of the curve C is defined above the surface S , where s is the arc length parameters of the curve C . p is the differential function along the curve C during variation. This can be used to model both concave and convex landscapes. The differential function defined on the area can be given as [78]

$$p^\theta = p^\theta(s, t), \theta = 1, 2, 3 \quad (10)$$

where s represents the geodesic distance in each direction and t represents time. The curves of each planning in the variation of the UAVs are dynamic. The initial condition should be satisfied as

$$p^\theta(s, 0) = p^\theta(s), \quad (11)$$

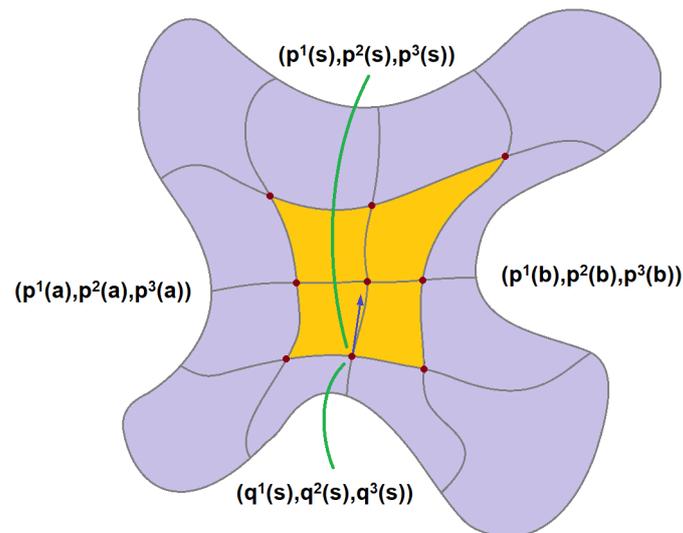


Figure 2. Path variation of a curve above a concave surface altitude.

This is a variation of the curve C . It is a continuous function of time and can evolve itself after each step. If the variation has fixed end points, then

$$p^\theta(a, t) = p^\theta(a) \quad (12)$$

$$p^\theta(b, t) = p^\theta(b) \quad (13)$$

For each parameter, the function group provides a curve, and its parameters can be restricted as

$$p^\theta = p_i^\theta(s) = p^\theta(s, t) \quad a \leq s \leq b. \quad (14)$$

It can be shown that the curve L is a member of this family of curves with the condition of $L = L_0$. Therefore, the variation of the curve C is to embed it in a curve family L_t that changes around it. This can be seen as the sets of time varying lines in an enclosed sprawling manifold. Outer differential forms can also be defined on differential manifolds. The p external differential form is a linear combination of the exterior products of some differential that are asymmetrical as p covariant tensors. It also implicates that the variational curve L_t of the curve L and the curve L have the same starting point and end point. Therefore, curve L_t is a subset of curve C . It should be pointed out that although the parameter s can be assumed to be the arc length parameter of the curve L , s is not necessarily the arc length parameter of the variational curve L_t . Then, the cost of the curve L can be calculated as

$$C_1 = \int_a^b \sqrt{S_{xyz}(k_1 p^1(s), k_1 p^2(s), k_2 p^3(s)) \frac{dp^x(s)}{ds} \frac{dp^y(s)}{ds} \frac{dp^z(s)}{ds}} ds \quad (15)$$

where $S_{xyz}[n]$ is a set of weighted surfaces above the landscape which contains the starting point and the end point. The flight routes of the UAVs are the subspaces of $S_{xyz}[n]$. During the numerical analysis, the starting surface $S_{xyz}[0]$ can be generated as an arbitrary curvilinear plane above the landscape. Through each iteration, $S_{xyz}[n]$ may change after each step. During each step, the cost changes with the evolution of the line sets. If the curve L has the lowest cost connecting its two end points, its cost should not be larger than the cost of the other curves connecting the end points of the curve L . In particular, if the curve L is embedded in any line set that has the same end point, the variational curve family L_t could satisfy

$$\left. \frac{d}{dt} \right|_{t=0} C(L_t) = 0 \quad (16)$$

When the curve L above the surface S satisfies this condition with respect to its variational L_t , the arc length of the curve L reaches its variational curve L_t . The external characteristics of the surface S are only related to the first basic form of the surface S , which indicates that the above considerations belong to the category of the intrinsic geometry of the surface S . The partial derivative q of the curve L can be expressed as

$$q^\theta(s) = \left. \frac{\partial}{\partial t} \right|_{t=0} p^\theta(s, t) \quad (17)$$

$$q(s) = q^\theta(s)r_\theta(k_1u^1(s), k_1u^2(s), k_2u^3(s)) \quad (18)$$

The partial derivative q of curve L is a tangent vector field defined along the curve L on the surface S as the variational vector field of the variational. The variational vector is the variational in $s = s_0$ and the tangent vector at $t = 0$. For the variation of the curve L with fixed endpoints, its variational vector field at the endpoints is zero. Conversely, if a tangent vector field is given along the curve L on the surface S , then a variation of the curve L can be defined with the variational vector field as

$$p^\theta(s, t) = p^\theta(s) + tq^\theta(s) \quad (19)$$

The first basic form of the surface S is

$$I[n] = S_{xyz}[n](k_1p^1, k_1p^2, k_2p^3)dp^x dp^y dp^z \quad (20)$$

where the function $S_{xyz}[n]$ is the first fundamental form of the surface S from differential geometry in the n -th step. Then, the cost of the variational curve L_t is

$$C[n](L_t) = \int_a^b \sqrt{S_{xyz}[n](k_1p^1(s, t), k_1p^2(s, t), k_2p^3(s, t)) \cdot \frac{\partial p^x(s, t)}{\partial s} \frac{\partial p^y(s, t)}{\partial s} \frac{\partial p^z(s, t)}{\partial s}} ds \quad (21)$$

The specific process of using the AWVM-BA is described as follows. First, the appropriate fitness function is selected, and the relevant parameters of the algorithm are initialized, including the population size, the scaling factor, and the probability of crossing. Then, a population is randomly generated based on the initial parameters. Then, the fitness function of each individual in the current population is calculated. At this stage, it is determined whether the algorithm can be terminated at the present time. If it can be terminated, the result can be output; otherwise, the algorithm continues to run. Mutation and crossover operations are performed to create new individuals. In the mutation operation, the difference between the two individuals is multiplied by the scaling factor and added to the third individual to obtain a new individual. In the crossover operation, a hybridization probability is set. If the conditions are met, a new individual is created according to the hybridization probability. Then, the selection operation is performed to select individuals from the current population and create a new population. Finally, the number of iterations is increased by one, and the execution continues with the previous steps.

2.3. Other Improvements and Modifications of the Proposed AWVM-BA

The improved artificial potential field method (APF) can be carried out to analyze the force exerted on the UAV [79], as shown in Figure 3. $\vec{F}_g[n]$ and $\vec{F}_o[n]$ stand for the attractive force and repulsive force, respectively. ρ_o represents the distance to the barrier and ρ_g represents the distance to the target point. First, the attractive potential $U_G[n]$ and the repulsive potential $U_o[n]$ can be given as

$$U_G[n] = \begin{cases} \frac{1}{2}\varepsilon\rho_G^2, & \text{for } \rho_G \leq d_{goal} \\ \varepsilon d_{goal}\rho_G - \frac{1}{2}\varepsilon d_{goal}^2, & \text{for } \rho_G > d_{goal} \end{cases} \quad (22)$$

$$U_O[n] = \begin{cases} \frac{1}{2}\sigma\rho_O^2\left(\frac{1}{\rho_O} - \frac{1}{d_{ob}}\right)^2, & \text{for } \rho_O \leq d_{ob} \\ 0, & \text{for } \rho_O > d_{ob} \end{cases} \quad (23)$$

where ε is defined as the scale factor. d_{goal} is the threshold for the distance between the UAV and the target point and d_{ob} is the threshold for the distance between the UAV and the barrier. Then, in combination with the variation method, the cost can be modified as

$$C[n](L_t) = \int_a^b \sqrt{\frac{S_{xyz}[n](k_1p^1(s,t), k_1p^2(s,t), k_2p^3(s,t))}{\frac{\partial p^x(s,t)}{\partial s} \frac{\partial p^y(s,t)}{\partial s} \frac{\partial p^z(s,t)}{\partial s}} (\nabla U_G[n] \cdot \nabla U_O[n])} ds \quad (24)$$

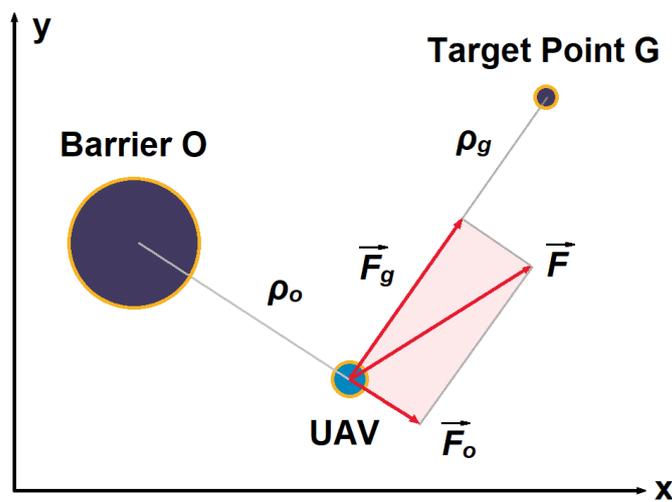


Figure 3. The force analysis of a UAV based on the APF method.

The mountain peaks in the forest that are higher than the flight range of the drone can be considered as obstacles. The flight path of the drone for firefighting should be planned in an obstacle environment. The distance between the obstacle and the drone is divided into six parts, and five points are selected in the flight path, namely, points $T_1, T_2, T_3, T_4,$ and T_5 . When flying on this trajectory, the threat at each division point is different, as shown in Formula (25). The threat from the obstacles at the site is shown in Figure 4. The designed division is based on the mathematical golden ratio, which can avoid threat at the optimal distance.

$$T_k = \begin{cases} T_1 = \frac{\sqrt{5}-1}{2}T \approx 0.608T \\ T_2 = \left(\frac{\sqrt{5}-1}{2}\right)^2 T \approx 0.382T \\ T_3 = \left(\frac{\sqrt{5}-1}{2}\right)^3 T \approx 0.236T \\ T_4 = \frac{2}{3}\left(\frac{\sqrt{5}-1}{2}\right)^3 T \approx 0.157T \\ T_5 = \frac{1}{3}\left(\frac{\sqrt{5}-1}{2}\right)^3 T \approx 0.079T \end{cases} \quad (25)$$

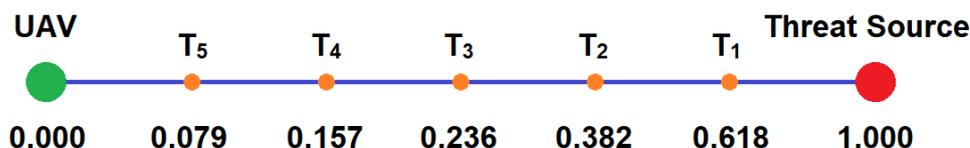


Figure 4. The evaluation of the threat cost.

Then, by considering the threat cost value, the final variational cost function can be modified as

$$C[n](L_t) = \int_a^b \sqrt{\frac{S_{xyz}[n](k_1 p^1(s, t), k_1 p^2(s, t), k_2 p^3(s, t))}{\frac{\partial p^x(s, t)}{\partial s} \frac{\partial p^y(s, t)}{\partial s} \frac{\partial p^z(s, t)}{\partial s}} (1 - T_k)(\nabla U_G[n] \cdot \nabla U_O[n])} ds \quad (26)$$

2.4. The Implementation of the Proposed AWVM-BA

The implementation steps of the hybrid bat algorithm based on differential evolutions are as follows. First, the algorithm parameters and population size are initialized. In the forest fire area, an initial trajectory from the starting point to the fire location is randomly generated with the initial solution. Next, the algorithm parameters, the size of the population, the initial position, the speed of each bat, the sound wave frequency, and the sound wave loudness of each bat are set. Then, each individual is evaluated, and the best individual in the population is selected at the moment. For each track point at the initial moment, the distance between each track point and the threat of obstacles in the environment to the current track point are calculated. The fitness value is obtained with the minimum track point of the function value. Next, the speed and position of the individual bat are updated. Each track point is renewed. After that, a local disturbance operation is performed on the current optimal bat individual. Additionally, the location of the local solution is used to replace the current optimal solution. The track point with the best fitness value in the track then is offset to a certain extent. This operation is mainly used to prevent the track generated by the algorithm from being the local optimal one. Otherwise, no offset is performed with respect to the operation of the track points. The next step is to accept the renewed solution. If the previous condition is satisfied, then the new solution is accepted, and the loudness of the sound wave is then updated. At this point, the global optimal solution is updated. The obtained bat populations are sorted to find the position of the optimal population. The next step is to determine whether the algorithm is iteratively terminated or whether the algorithm has reached the maximum number of iterations. If the maximum number is reached, the algorithm execution is terminated. Otherwise, the algorithm returns to the previous steps, and the iteration continues. The final step is to output the optimal solution of the UAV to perform the task.

The differential evolution uses a differential strategy for individual mutation. Its idea is to sum the difference between the current individual and two randomly selected individuals in the population to obtain a new individual as

$$x[n + 1] = Fx[n](1 - x[n - 1]) \quad (27)$$

where F is the logistic factor that decides the position variable $x[n]$ between each time step. If F is below 4, the x increases at a linear step without random distribution between each step. When F reaches above 4 and below 5, the distribution changes at large intervals with the property of gradual randomness. We performed the logistic mapping, and the results are shown in Figure 5. In the case where F reaches 5, the distribution is in the full range between 0 and 1. Then, in the solution space, the proposed algorithm can avoid the local minimum with an enhanced solving speed using this chaos strategy, which is also integrated in our proposed algorithm. The drawback of the proposed algorithm is that the amount of computation can increase faster with the complication of the outside environment, such as the deviation of wind speed. Additionally, it is shown in the final experiment that the iteration number of the proposed algorithm increases faster compared with some other intelligent algorithms. This is the main disadvantage of the proposed AWVM-BA. The flow of the proposed algorithm is given in Algorithm 2.

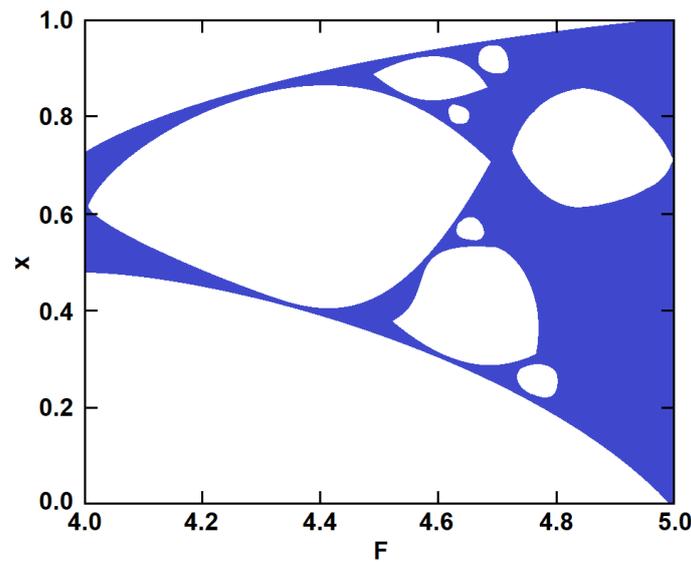


Figure 5. The logistic mapping of the distribution.

Algorithm 2 The proposed bat algorithm using asymmetrical variational method

Begin

Initialize $f_{min}, f_{max}, \vec{p}[0], \vec{v}[0], t_g, K[0], r[0], N_{max}, f_{th}, F, x[0], S_{xyz}[0], \beta, k_1, k_2$

for $i = 1:m$ **do**

for $n = 1:N_{max}$ **do**

compute frequency form (1)

compute solutions from (2) and (3)

calculate the path variation cost from (26)

calculate individual mutation from (27)

generate

if $(\frac{d}{dt}C[n](L_t) < 0)$ **then**

accept the current cost as the minimum

end if

calculate $\vec{v}[n+1]$ and $r[n+1]$ from (5) and (6)

produce $f_t[n]$ from (9)

if $(x[n] < r[n]$ and $f_t[n] < f_{th})$ **then**

accept the current position as the optimal position

end if

end for

end for

End

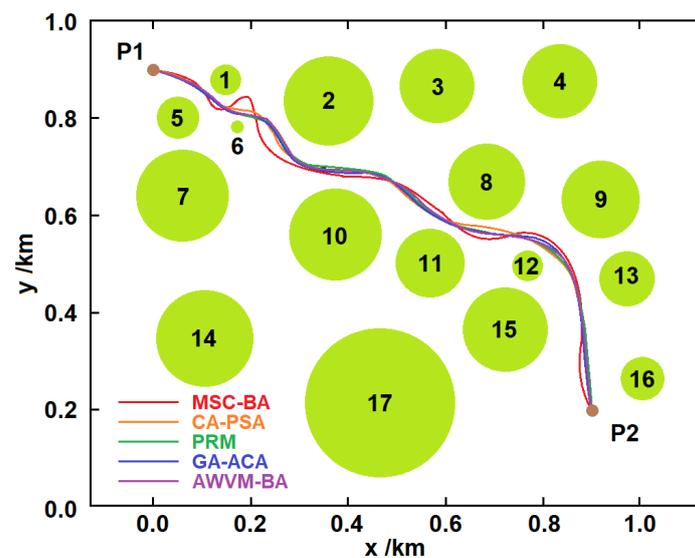
3. Calculation and Simulation

Several algorithms are compared using the simulation experiment to test the optimization performance of the proposed improved bat algorithm in different environments. The simulation is performed through programming. In this experiment, the size of the bat population is set to 70, and the maximum number of iterations of the algorithm is 50. In the optimal solution update formula, the update cost is set to 0.5, and the maximum threat cost of obstacles is 0.9. The differential evolution algorithm is used. When performing the bat population mutation, the scaling factor of the differential evolution algorithm is set to 0.7. The size of the simulation environment is $1 \text{ km} \times 1 \text{ km}$, including the horizontal and vertical coordinates of the center point of the obstacle and the obstacle radius, as shown in Table 1. The obstacles are expanded, and the radius of the obstacles is increased by 2 m to ensure safety during the flight. The maximum allowed flight height is set at 120 m to ensure flight safety.

Table 1. Performance comparison of different algorithms in convex obstacle regions.

Algorithms	Flight Length/m	Iteration Number	Flight Cost	TFV
MSC-BA	1697	27	8655	38
CA-PSA	1664	21	8487	35
PRM	1661	19	8471	35
GA-ACA	1667	19	8502	35
AWVM-BA	1666	21	8494	35

In this obstacle environment, the AWVM-BA algorithm and some other algorithms are used for simulation experiments. The simulation results are shown in Figures 6 and 7. In Figure 6, the UAV is designed to fly from P1 to P2 with 17 obstacles in the region. These obstacles are denoted as green circles where the UAV should not be able to fly across. Figure 7 shows the convergence curves of the algorithms. The slope of the convergence curve represents the convergence speed of the corresponding algorithm. It can be seen that the PRM has the best convergence performance in the convex obstacle region. Through continuous iterations, the flight target function of the UAV has a relatively fixed value in the convergence range, which can be used to monitor the convergence accuracy of the corresponding algorithm. The fitness function is the threat of the obstacle to the UAV. The data in Table 1 are evaluation indicators for different algorithms for trajectory planning. By combining the simulation results and the trajectory evaluation indicators, it can be concluded that the proposed AWVM-BA shows comparable performances to some other recently proposed algorithms.

**Figure 6.** The planned flying trajectories of UAV under different algorithms in circular convex obstacle regions.

Since the obstacles are round and convex, it saves the trouble of local optimization and expedites the convergence speed. In the next experiment, we changed the obstacles with rugged outlines. Additionally, in this environment, we compared the performances of the AWVM-BA algorithm with some other algorithms. The results are shown in Figures 8 and 9. It can be seen that the AWVM-BA has the best convergence performance in the convex obstacle region, with the shortest path length and the lowest cost. Through continuous iterations, the flight target function of the UAV is relatively fixed in the convergence range, which can be used to monitor the convergence accuracy of the corresponding algorithm. For the concave surfaces, the reflected detection parameter can easily focus on a local optimum just as a ray of light being reflected by a concave mirror to form a focus point. With the increase in the iteration, the situation is unbalanced by the free parameters in the

algorithm. This process can be regarded as changing the curvature of the mirror. Then, the planned path from P1 to P2 is renewed after each iteration. The target function does not always decrease with the increase in the iteration number for some algorithms. In our experiment, we chose some recent intelligent algorithms for performance comparison. They are stable and converge fast without irregularities during iterations. So, the target function decreases with each iteration. The data in Table 2 are evaluation indicators for different algorithms for trajectory planning. The concave parts along the obstacles can lure algorithms into local optimization more easily. By combining the simulation results and the trajectory evaluation indicators, it can be concluded that the proposed AWVM-BA shows the best performance.

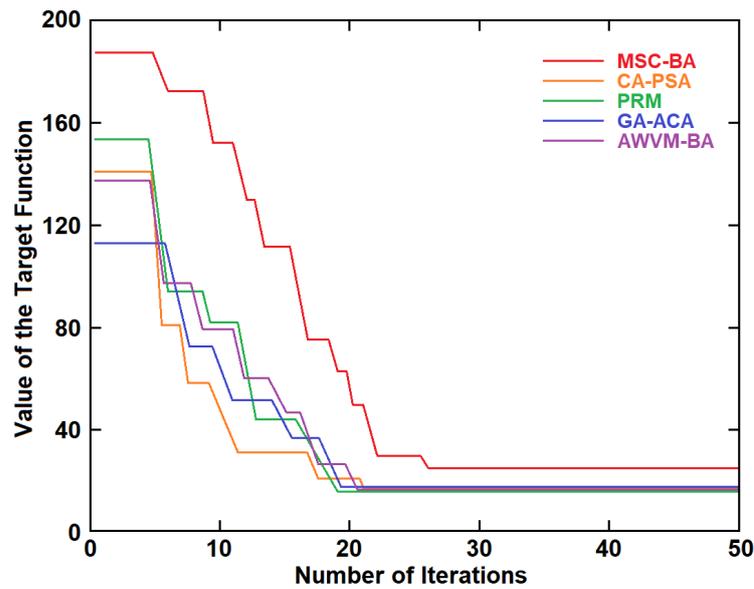


Figure 7. The target function value with the iteration of different algorithms in circular convex obstacle regions.

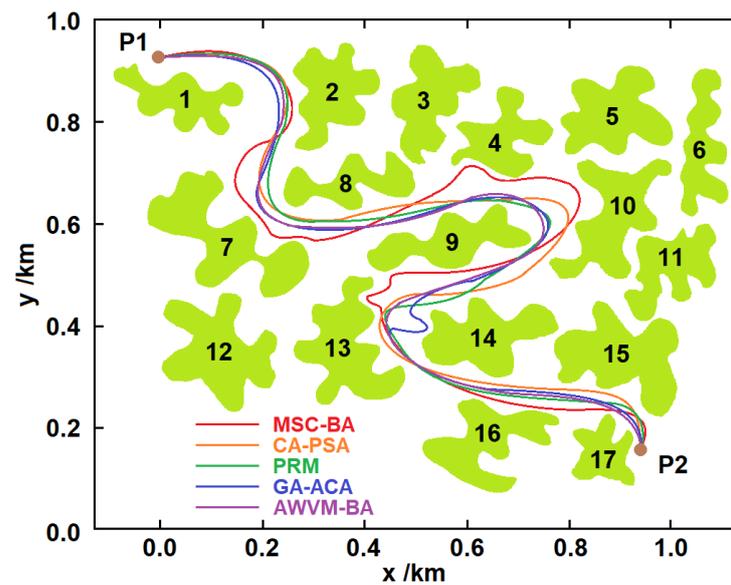


Figure 8. The planned flying trajectories of UAV under different algorithms in concave obstacle regions.

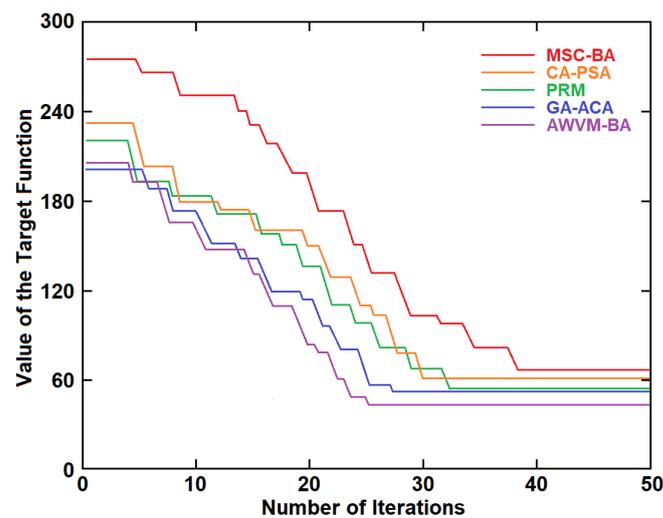


Figure 9. The target function value with the iteration of different algorithms in concave obstacle regions.

Table 2. Performance comparison of different algorithms in convex obstacle regions.

Algorithms	Flight Length/m	Iteration Number	Flight Cost	TFV
MSC-BA	2687	38	13,704	64
CA-PSA	2346	30	11,965	60
PRM	2383	32	12,153	58
GA-ACA	2423	26	12,357	57
AWVM-BA	2318	25	11,822	53

The influence of uncertain factors can easily lead to the inaccurate execution of fire-fighting tasks. When the drone's flying altitude is too low, there is a large number of obstacles, such as mountains and trees in the environment, and flames will also cause safety issues to the drone itself. Therefore, the flying height of the drone cannot be too low or too high, and the flying height should be adjusted at any time along with the ups and downs of the undulating mountains in the environment. The drone's flight should consider the altitude of the current flight position at this time, and then the drone can be on the further planned trajectory in the forest fire-fighting mission.

The AWVM-BA algorithm, with some other newly proposed algorithms, are simulated in a continuous gradient environment for the trajectory planning. The size of the task space is $1 \text{ km} \times 1 \text{ km}$, as shown in Figure 10. In order to ensure the safe flight of the UAV, the obstacles are expanded, and the radius of the obstacles is increased by 1 m.

The simulation results are simulated in circular convex regions with a continuous gradient. The results show that these algorithms can plan a safe flight path for UAVs in mountainous areas, meeting the requirements of UAVs for fire-fighting in mountainous areas, and proving that the algorithms are feasible and correct. Figure 11 shows the change curve of the target function value that the algorithm runs when used for trajectory planning. Through the comparison of the iterative curves, it can be seen that the target function value of the trajectory planning is better for the PRM algorithm. Still, the AWVM-BA shows relatively comparable results with the other algorithms. The smaller the obstacles threatened by the drone during the flight are, the safer the planned trajectory becomes. The data in Table 3 are the evaluation comparison of the five different algorithms for trajectory planning. By combining the simulation results and the trajectory evaluation indicators, it can be concluded that the UAV trajectory planned by the AWVM-BA has a shorter length and poses a smaller cost to the UAV. It is also more conducive to the UAV flight.

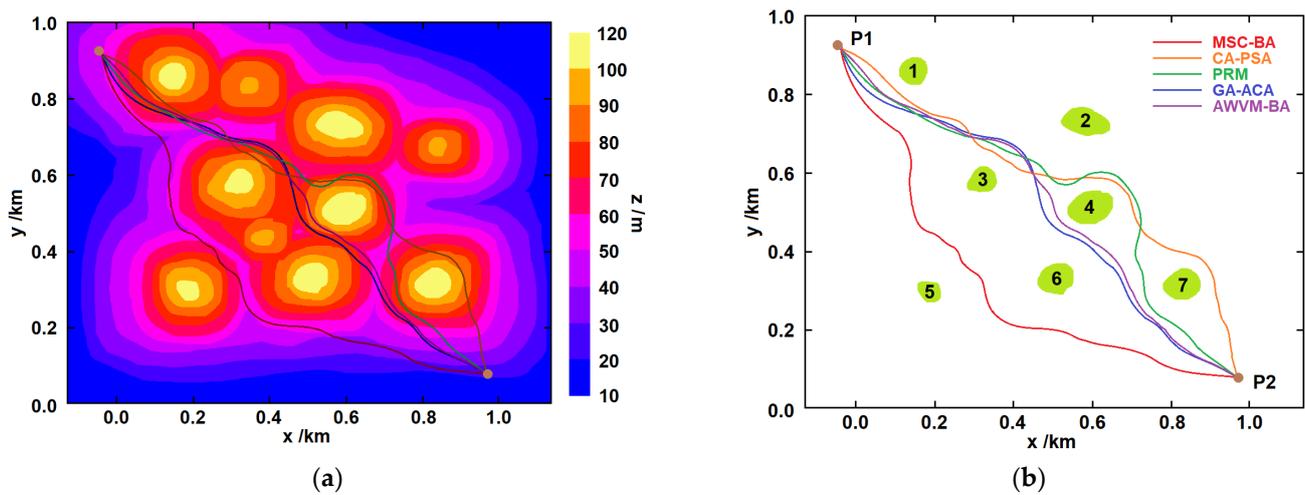


Figure 10. The planned flying trajectories of UAV under different algorithms in circular convex regions with continuous gradient.

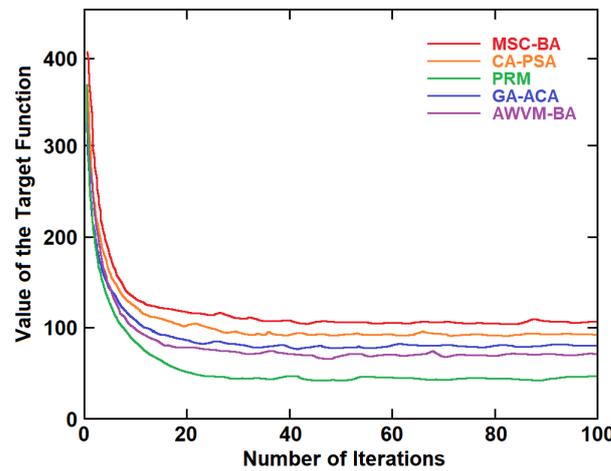


Figure 11. The target function value with the iteration of different algorithms in circular convex regions with continuous gradient.

Table 3. Performance comparison of different algorithms in convex obstacle regions.

Algorithms	Flight Length/m	Iteration Number	Flight Cost	TFV
MSC-BA	1921	43	12,487	133
CA-PSA	1746	40	10,825	98
PRM	1721	24	10,670	63
GA-ACA	1548	34	9289	92
AWVM-BA	1523	33	8986	87

Next, we simulated the results in concave regions with a continuous gradient, which are shown in Figure 12. These results also show that these algorithms can plan a safe flight path for UAVs in mountainous areas, meeting the requirements of UAVs for firefighting in mountainous areas, and proving that the algorithms are feasible and correct in concave regions with a continuous gradient. Figure 13 shows the change curve of the target function value that the algorithm runs when used for trajectory planning. By comparison, the target function value of the trajectory planning is better when the bat algorithm is based on differential evolution. The data in Table 4 give the performance comparison of the five different algorithms in concave regions with a continuous gradient. By combining the simulation results and the trajectory evaluation indicators, it can be concluded that the

UAV trajectory planned by the AWVM-BA has a shorter length and a smaller cost to the UAV compared with the other four algorithms.

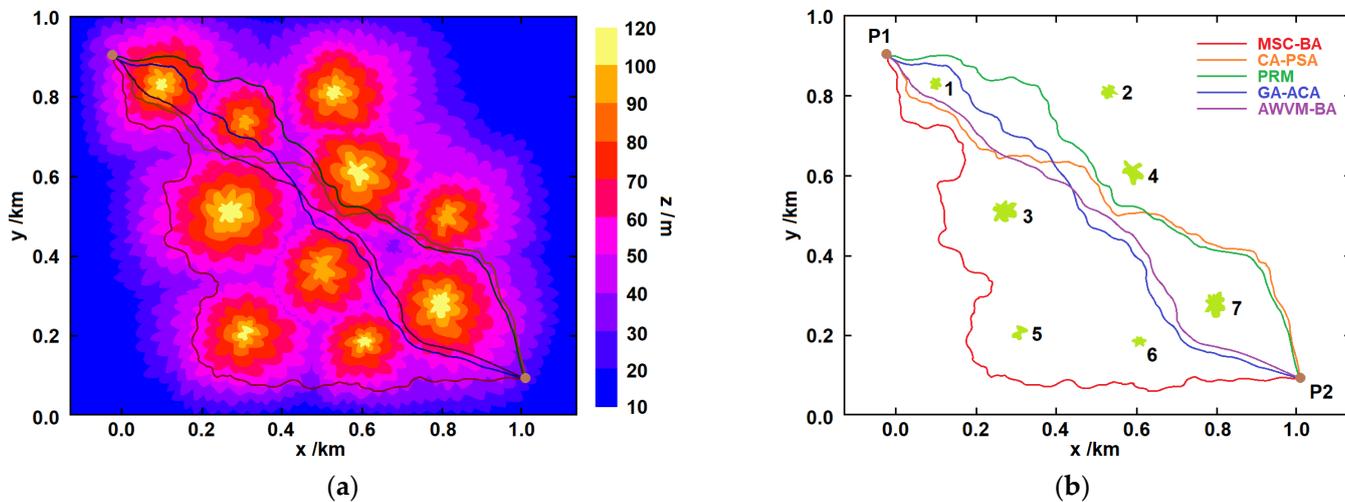


Figure 12. The planned flying trajectories of UAV under different algorithms in circular concave regions with continuous gradient.

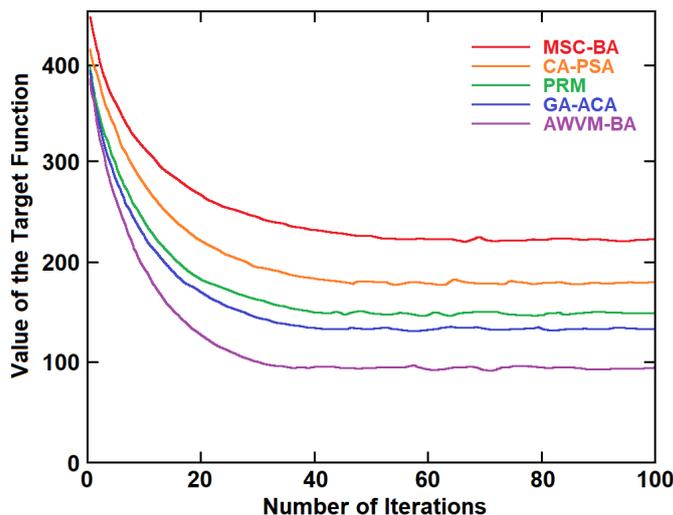


Figure 13. The target function value with the iteration of different algorithms in circular convex regions with continuous gradient.

Table 4. Performance comparison of different algorithms in convex obstacle regions.

Algorithms	Flight Length/m	Iteration Number	Flight Cost	TFV
MSC-BA	1943	62	12,435	234
CA-PSA	1601	44	9766	198
PRM	1594	42	9723	172
GA-ACA	1521	41	9126	136
AWVM-BA	1492	37	8654	98

In the actual flight route planning, it is necessary to consider the wind condition from the viewpoint of fuel consumption. Additionally, it is necessary to consider that the fire area also expands and changes depending on the wind conditions. Since the wind speed is not a known variable in a local region, we modeled the wind behavior as a three-dimensional continuous Brownian stochastic process. Due to the effects of the landscapes, the wind

moving in the xy direction is much stronger than that moving in the z direction. Therefore, we introduced a damping factor D_p . D_p is in the range of $[0, 1]$. If D_p is 0, then there is no wind flow in the z direction. Additionally, if D_p is 1, then the strength of the wind flow in the vertical direction is the same as that in the horizontal direction. Therefore, the wind behavior can be modeled as

$$\vec{B} = \vec{e}_x B_x + \vec{e}_y B_y + \vec{e}_z D_p B_z \quad (28)$$

where \vec{e}_x , \vec{e}_y , and \vec{e}_z are the three basic unit vectors in the xyz coordinate systems. The components B_x , B_y , and B_z are independent to each other. In general conditions, the probability of wind directions is equally likely. Therefore, we took the expectation of each component μ to be zero. The standard variance σ characterizes the disturbance and deviation of the strength of the wind with respect to time. If σ is small, then the wind level is relatively weak around the expectation. However, with the increase in σ , the wind strength can become stronger during the Brownian stochastic process. As for the special condition, if $\sigma = 0$, this means that the UAVs are flying in the ideal windless environment. Figure 14 shows the wind speed with respect to time in the xyz directions as a continuous Brownian stochastic process at the condition of $\mu = 0$, $\sigma = 5$, and $D_p = 0.2$. The deviation in the z direction is much smaller in amplitude due to the damping factor. If the wind speed takes a negative sign, it means the wind is blowing in the opposite direction. The wind effect can have some influences on the evaluation parameters, such as the flight length, iteration number, flight cost, and TFV. Then, the wind effects can be regarded as the added random noise in the flight of the UAVs.

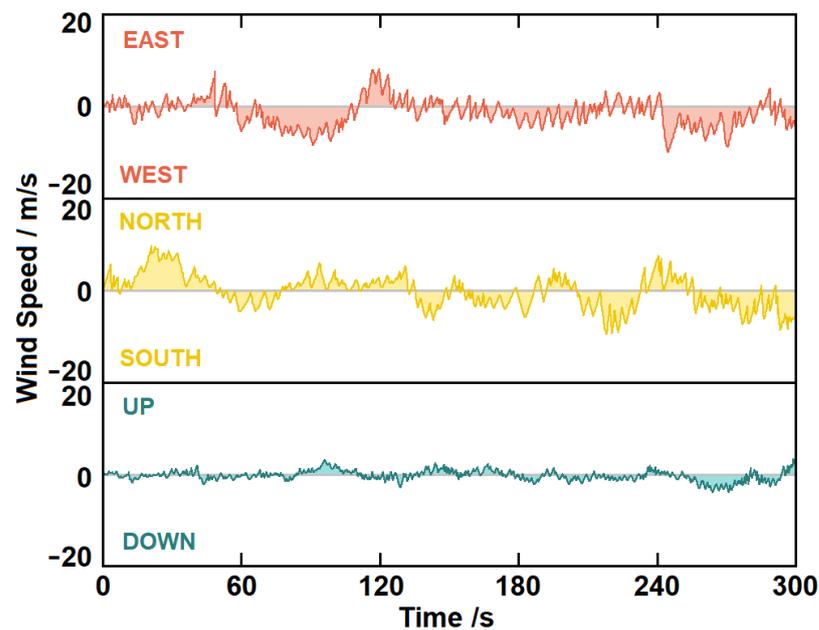


Figure 14. The wind speed with respect to time in the xyz directions as a continuous Brownian stochastic process ($\mu = 0$, $\sigma = 5$, and $D_p = 0.2$).

Additionally, it is necessary to consider that the fire area also expands and changes depending on the wind conditions, as shown in Figure 15a. In a windless condition, the fire spread speed \vec{V}_{sp} is determined by the contour of its geometric topology, and can be modeled as

$$\vec{V}_{sp} = k_p \nabla G(x, y, z) \quad (29)$$

where ∇ is the del operator, and $G(x, y, z)$ is the contour of the fire region. k_p is the fire spread factor and is affected by the types of plants in the area, since the wind affects the

fire spread in the xy direction. Therefore, if the wind effect is taken into consideration, then the modified fire spread speed can be given as

$$\vec{V}_{sp} = k_p \nabla G(x, y, z) + \vec{e}_x k_w B_x + \vec{e}_y k_w B_y \quad (30)$$

where k_w is defined as the wind effect factor. Figure 15b shows the flight path of a UAV under different route planning algorithms with the effects of wind and fire spread. It can be seen that the flight paths become perturbed compared with those in a windless condition. Additionally, some segments of the paths were slightly altered. P2 stands for the fire spread region. The fire spread effect is not that significant compared with the wind effect since the speed of the fire spread is very small compared with the speed of UAVs. In our simulation, we still considered the fire spread effect in the P2 region. The path deviation is mainly affected by the deviation of the wind speed. The exertion of wind has effects on the performance, especially with the increase in the deviation of the wind speed. The wind can be modeled as an additive noise along the flying path. In our experiment, we used the Monte Carlo method with 10,000 independent trials.

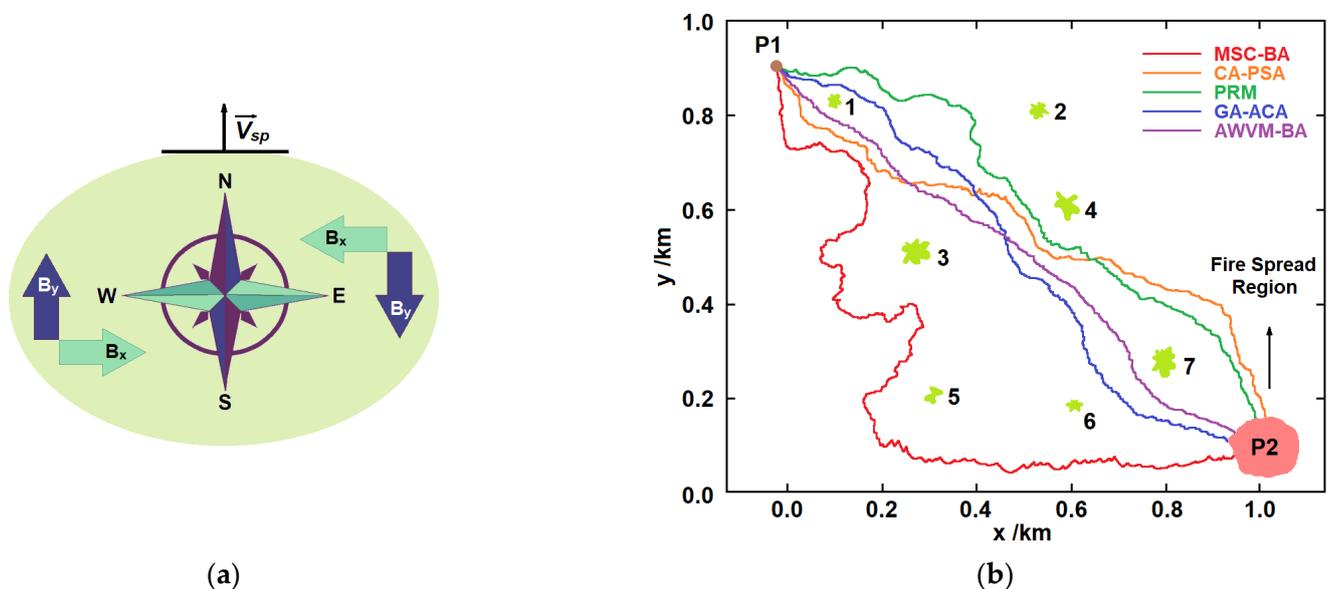


Figure 15. The path planning problem in consideration of (a) fire spread speed and (b) wind effects.

Figure 16 shows the evaluation parameters in consideration of the wind effect and fire spread effect. The proposed AWVM-BA is again tested in comparison with some other recently proposed algorithms. The simulation is based on the Monte Carlo method with 10,000 independent trials. The arithmetic averages were calculated as the final results. It can be seen that the evaluation parameters increase nonlinearly with the increase in σ . Compared with the MSC-BA, CA-PSA, PRM, and GA-ACA, the proposed algorithm achieved better results in the flight length, flight cost, and TFV. The limitation of the proposed algorithm is that the variation calculation is more sensitive with the added wind effects. Therefore, the iteration number increases more quickly than the other algorithms when σ increases. This is because the deviation can be regarded as the complication of the sub-dimension surface variation. This can also have an impact on the convergence time. Therefore, a higher calculation ability is required for hardware setup. Still, the proposed algorithm shows a great advantage in convex landscapes in windy conditions.

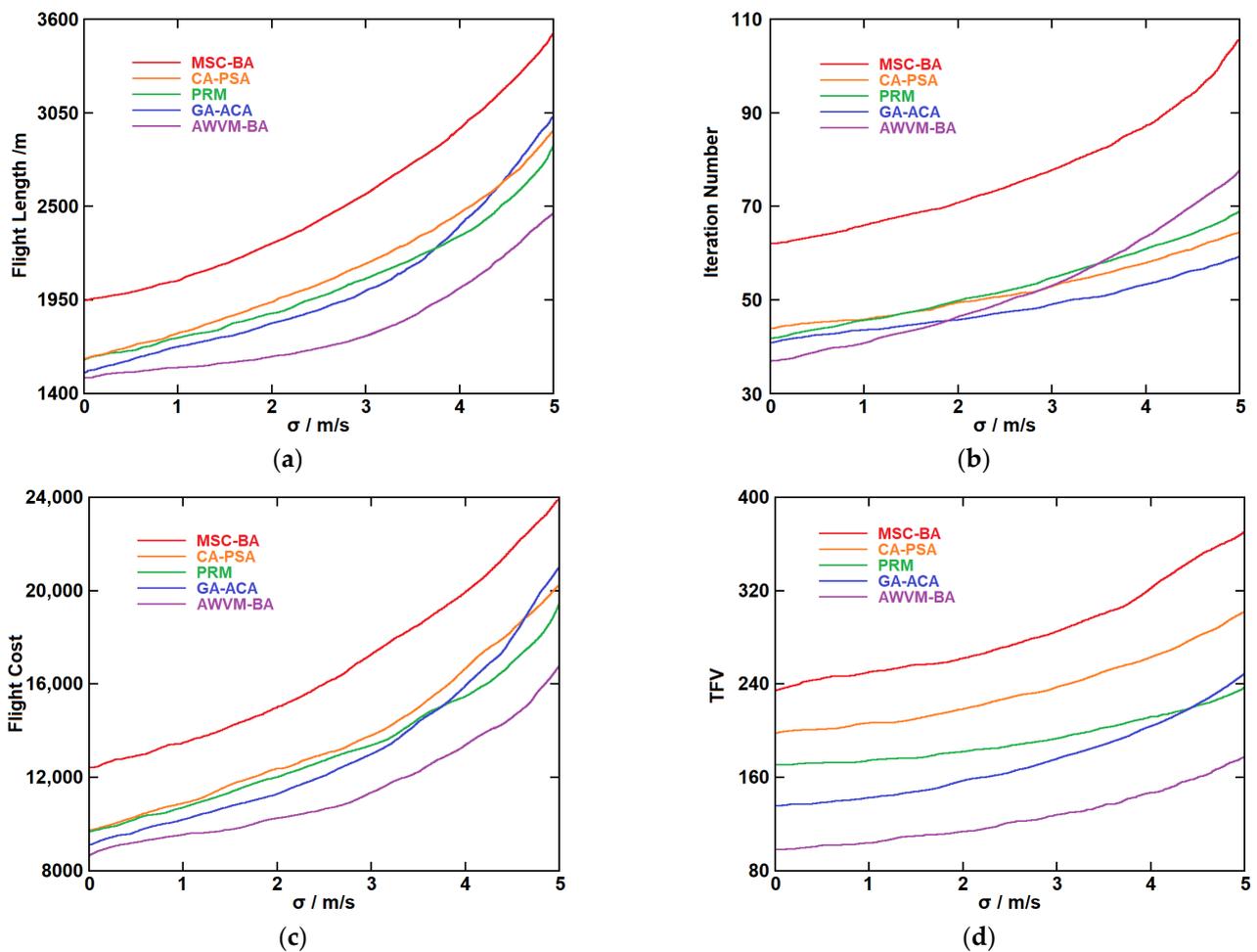


Figure 16. The evaluation parameters in consideration of wind effect and fire spread effect with the increase in the standard deviation; (a) flight length, (b) iteration number, (c) flight cost, and (d) TFV.

4. Conclusions

This paper focuses on the trajectory planning of UAV forest firefighting and plans the mission area according to the altitude difference in mountainous areas. We analyzed how to plan the UAV trajectory in the corresponding environment. The bat algorithm with an asymmetrical weighed variational method is proposed for the trajectory planning in continuous gradient mountainous regions. The algorithm is designed to include obstacle threat weight and trajectory length. The fitness function of the bat and the variational method are used to solve the premature problem of the bat algorithm. Finally, it can be seen from the results that the proposed algorithm can quickly generate a flight track for the UAV with a low cost, especially in concave landscape regions. The limitation of the proposed algorithm is that it is more sensitive to added noise, and the iteration number increases comparatively faster with the increase in the deviation of the wind speed. Therefore, it requires a higher hardware calculation ability. The proposed algorithm provides a reference for the practical application of the UAV in forest firefighting in both windless and windy conditions, especially for concave landscape regions.

Author Contributions: Methodology, software, and writing—original draft, X.C., C.W. and W.L.; writing—review and editing, project administration, and funding acquisition, X.C. and C.W. All authors have read and agreed to the published version of the manuscript.

Funding: This paper was funded by the Industry-University Cooperation Collaborative Education Project of the Ministry of Education, grant number 202101097010, and the Natural Science Foundation of Sichuan Province, grant number 2022NSFSC0953.

Data Availability Statement: The data that support the findings of this study are available from the corresponding authors upon reasonable request.

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

References

1. Keeley, J.E.; Mantgem, P.V.; Falk, D.A. Fire, climate and changing forests. *Nat. Plants* **2019**, *5*, 774–775. [[CrossRef](#)] [[PubMed](#)]
2. Hart, S.J.; Henkelman, J.; McLoughlin, P.D.; Nielsen, S.E.; Truchon-Savard, A.; Johnstone, J.F. Examining forest resilience to changing fire frequency in a fire-prone region of boreal forest. *Glob. Chang. Biol.* **2019**, *25*, 869–884. [[CrossRef](#)] [[PubMed](#)]
3. Galván, L.; Magaña, V. Forest fires in Mexico: An approach to estimate fire probabilities. *Int. J. Wildland Fire* **2020**, *29*, 753–763. [[CrossRef](#)]
4. Mourao, P.R.; Martinho, V.D. Forest fire legislation: Reactive or proactive? *Ecol. Indic.* **2019**, *104*, 137–144. [[CrossRef](#)]
5. Liu, X.; Jing, T.; Hou, L. An FW–GA Hybrid Algorithm Combined with Clustering for UAV Forest Fire Reconnaissance Task Assignment. *Mathematics* **2023**, *11*, 2400. [[CrossRef](#)]
6. Lin, K.; Zhang, L.; Huang, L.; Feng, Z.; Chen, T. Improved Particle Swarm Path Planning Algorithm with Multi-Factor Coupling in Forest Fire Spread Scenarios. *Fire* **2023**, *6*, 202. [[CrossRef](#)]
7. Zheng, S.; Gao, P.; Zhou, Y.; Wu, Z.; Wan, L.; Hu, F.; Wang, W.; Zou, X.; Chen, S. An Accurate Forest Fire Recognition Method Based on Improved BPNN and IoT. *Remote Sens.* **2023**, *15*, 2365. [[CrossRef](#)]
8. Sikuzani, Y.U.; Mukenza, M.M.; Malaisse, F.; Kaseya, P.K.; Bogaert, J. The Spatiotemporal Changing Dynamics of Miombo Deforestation and Illegal Human Activities for Forest Fire in Kundelungu National Park, Democratic Republic of the Congo. *Fire* **2023**, *6*, 174. [[CrossRef](#)]
9. Laneve, G.; Fusilli, L.; Bernini, G.; Suarez Beltran, J. Preventing Forest Fires through Remote Sensing: Achievements of the Prevention and Recovery of Forest Fires Emergency in the Mediterranean Area Project. *IEEE Geosci. Remote Sens. Mag.* **2020**, *8*, 17–49. [[CrossRef](#)]
10. Mazzella, M.N.; Koprowski, J.L. Response to fire by a forest specialist in isolated montane forest. *For. Ecol. Manag.* **2020**, *462*, 117996. [[CrossRef](#)]
11. Xu, X.; Jia, G.; Zhang, X.; Riley, W.J.; Xue, Y. Climate regime shift and forest loss amplify fire in Amazonian forests. *Glob. Chang. Biol.* **2020**, *26*, 5874–5885. [[CrossRef](#)] [[PubMed](#)]
12. Aleksandrov, A.A.; Ksenofontov, B.S.; Kozodaev, A.S.; Taranov, R.A.; Vyazova, V.D.; Ivanov, M.V. Development of an Algorithm for Calculating the Moisture Content and Time of Forest Fire Maturation of Forest Combustible Materials for Determining Forest Fire Hazards. *Electronics* **2023**, *12*, 1937. [[CrossRef](#)]
13. Jiao, Q.; Fan, M.; Tao, J.; Wang, W.; Liu, D.; Wang, P. Forest Fire Patterns and Lightning-Caused Forest Fire Detection in Heilongjiang Province of China Using Satellite Data. *Fire* **2023**, *6*, 166. [[CrossRef](#)]
14. Lin, X.; Li, Z.; Chen, W.; Sun, X.; Gao, D. Forest Fire Prediction Based on Long- and Short-Term Time-Series Network. *Forests* **2023**, *14*, 778. [[CrossRef](#)]
15. Pan, H.; Badawi, D.; Zhang, X.; Cetin, A.E. Additive neural network for forest fire detection. *Signal Image Video Process.* **2020**, *14*, 675–682. [[CrossRef](#)]
16. Scholten, R.C.; Jandt, R.; Miller, E.A.; Rogers, B.M.; Veraverbeke, S. Overwintering fires in boreal forests. *Nature* **2021**, *593*, 399–404. [[CrossRef](#)]
17. Xie, Y.; Peng, M. Forest fire forecasting using ensemble learning approaches. *Neural Comput. Appl.* **2019**, *31*, 4541–4550. [[CrossRef](#)]
18. Barmpoutis, P.; Kastridis, A.; Stathaki, T.; Yuan, J.; Shi, M.; Grammalidis, N. Suburban Forest Fire Risk Assessment and Forest Surveillance Using 360-Degree Cameras and a Multiscale Deformable Transformer. *Remote Sens.* **2023**, *15*, 1995. [[CrossRef](#)]
19. Zhang, L.; Wang, M.; Ding, Y.; Bu, X. MS-FRCNN: A Multi-Scale Faster RCNN Model for Small Target Forest Fire Detection. *Forests* **2023**, *14*, 616. [[CrossRef](#)]
20. Maraş, E.E.; Dönmez, K.; Emecen, Y. GIS-Based Determination of the Optimal Heliport and Water Source Locations for Forest Fire Suppression Using Multi-Objective Programming. *Aerospace* **2023**, *10*, 305. [[CrossRef](#)]
21. Tang, Y.; Li, X.; Zhang, X.; Xia, X.; Gui, L. Dynamic Multi-swarm Global Particle Swarm Optimization. In Proceedings of the 2019 IEEE Congress on Evolutionary Computation (CEC), Wellington, New Zealand, 10–13 June 2019; pp. 1030–1037.
22. Shi, H.; Liu, S.; Wu, H.; Li, R.; Liu, S.; Kwok, N.; Peng, Y. Oscillatory Particle Swarm Optimizer. *Appl. Soft Comput.* **2018**, *73*, 316–327. [[CrossRef](#)]
23. Fernandes, C.M.; Fachada, N.; Merelo, J.J.; Rosa, A.C. Steady state particle swarm. *PeerJ. Comput. Sci.* **2019**, *5*, e202. [[CrossRef](#)] [[PubMed](#)]
24. Halimu, Y.; Zhou, C.; You, Q.; Sun, J. A Quantum-Behaved Particle Swarm Optimization Algorithm on Riemannian Manifolds. *Mathematics* **2022**, *10*, 4168. [[CrossRef](#)]
25. Muslimov, T. Particle Swarm Optimization for Target Encirclement by a UAV Formation. *Eng. Proc.* **2023**, *33*, 15.
26. Zhou, L.; Wang, M.; Zhang, X.; Qin, P.; He, B. Adaptive SLAM Methodology Based on Simulated Annealing Particle Swarm Optimization for AUV Navigation. *Electronics* **2023**, *12*, 2372. [[CrossRef](#)]
27. Greenwald, E.; Eckmann, J.P.; Feinerman, O. Colony entropy-Allocation of goods in ant colonies. *PLoS Comput. Biol.* **2019**, *15*, e1006925. [[CrossRef](#)]

28. Zhong, J.; Fung, Y.F.; Dai, M. Biologically inspired ant colony simulation. *Int. J. Control Autom. Syst.* **2010**, *8*, 519–526. [[CrossRef](#)]
29. Huang, Z.M.; Chen, W.N.; Li, Q.; Luo, X.N.; Yuan, H.Q.; Zhang, J. Ant Colony Evacuation Planner: An Ant Colony System with Incremental Flow Assignment for Multipath Crowd Evacuation. *IEEE Trans. Cybern.* **2021**, *51*, 5559–5572. [[CrossRef](#)]
30. Phongmoo, S.; Leksakul, K.; Charoenchai, N.; Boonmee, C. Artificial Bee Colony Algorithm with Pareto-Based Approach for Multi-Objective Three-Dimensional Single Container Loading Problems. *Appl. Sci.* **2023**, *13*, 6601. [[CrossRef](#)]
31. Kaya, E. A New Neural Network Training Algorithm Based on Artificial Bee Colony Algorithm for Nonlinear System Identification. *Mathematics* **2022**, *10*, 3487. [[CrossRef](#)]
32. Zhou, H.; Jiang, Z.; Xue, Y.; Li, W.; Cai, F.; Li, Y. Research on Path Planning in 3D Complex Environments Based on Improved Ant Colony Algorithm. *Symmetry* **2022**, *14*, 1917. [[CrossRef](#)]
33. Kazakovtsev, L.; Rozhnov, I.; Shkaberina, G.; Orlov, K. K-Means Genetic Algorithms with Greedy Genetic Operators. *Math. Probl. Eng.* **2020**, *2020*, 16. [[CrossRef](#)]
34. Drezner, Z.; Drezner, T.D. The alpha male genetic algorithm. *IMA J. Manag. Math.* **2019**, *30*, 37–50. [[CrossRef](#)]
35. Cao, Y.; Wu, M. A Novel RPL Algorithm Based on Chaotic Genetic Algorithm. *Sensors* **2018**, *18*, 3647. [[CrossRef](#)]
36. Hou, J.; Du, J.; Chen, Z. Time-Optimal Trajectory Planning for the Manipulator Based on Improved Non-Dominated Sorting Genetic Algorithm II. *Appl. Sci.* **2023**, *13*, 6757. [[CrossRef](#)]
37. Roach, L.; Gao, X. Graphical Local Genetic Algorithm for High-Dimensional Log-Linear Models. *Mathematics* **2023**, *11*, 2514. [[CrossRef](#)]
38. Nazerian, M.; Karimi, J.; Torshizi, H.J.; Papadopoulos, A.N.; Hamed, S.; Vatankhah, E. An Improved Optimization Model to Predict the MOR of Glulam Prepared by UF-Oxidized Starch Adhesive: A Hybrid Artificial Neural Network-Modified Genetic Algorithm Optimization Approach. *Materials* **2022**, *15*, 9074. [[CrossRef](#)]
39. Al-Betar, M.A.; Awadallah, M.A. Island bat algorithm for optimization. *Expert Syst. Appl.* **2018**, *107*, 126–145. [[CrossRef](#)]
40. Xu, T.; Xiang, Z. Modified Constant Modulus Algorithm Based on Bat Algorithm. *Artif. Intell. Adv. Manuf.* **2021**, *41*, 4493–4500. [[CrossRef](#)]
41. Ren, Y.; Sun, Y.; Jing, X.; Cui, Z.; Shi, Z. Adaptive Makeup Transfer via Bat Algorithm. *Mathematics* **2019**, *7*, 273. [[CrossRef](#)]
42. Yuan, X.; Yuan, X.; Wang, X. Path Planning for Mobile Robot Based on Improved Bat Algorithm. *Sensors* **2021**, *21*, 4389. [[CrossRef](#)] [[PubMed](#)]
43. Alharbi, A.; Alosaimi, W.; Alyami, H.; Rauf, H.T.; Damaševičius, R. Botnet Attack Detection Using Local Global Best Bat Algorithm for Industrial Internet of Things. *Electronics* **2021**, *10*, 1341. [[CrossRef](#)]
44. Aalimahmood, N.; Bedon, C.; Hasanzadeh-Inanlou, N.; Hasanzade-Inallu, A.; Nikoo, M. BAT Algorithm-Based ANN to Predict the Compressive Strength of Concrete—A Comparative Study. *Infrastructures* **2021**, *6*, 80. [[CrossRef](#)]
45. Yang, L.; Li, D.; Tan, R. Research on the Shortest Path Solution Method of Interval Valued Neutrosophic Graphs Based on the Ant Colony Algorithm. *IEEE Access* **2020**, *8*, 88717–88728. [[CrossRef](#)]
46. Wei, X.; Xu, J. Distributed Path Planning of Unmanned Aerial Vehicle Communication Chain Based on Dual Decomposition. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 12. [[CrossRef](#)]
47. Wang, Y.; Wang, P.; Zhang, J.; Cui, Z.; Cai, X.; Zhang, W.; Chen, J. A Novel Bat Algorithm with Multiple Strategies Coupling for Numerical Optimization. *Mathematics* **2019**, *7*, 135. [[CrossRef](#)]
48. Yu, M. A Solution of TSP Based on the Ant Colony Algorithm Improved by Particle Swarm Optimization. *Discret. Contin. Dyn. Syst.* **2019**, *12*, 979–987. [[CrossRef](#)]
49. Polat, E. The effects of different weight functions on partial robust M-regression performance: A simulation study. *Commun. Stat. Simul. Comput.* **2020**, *49*, 1089–1104. [[CrossRef](#)]
50. Ia, J.; Li, X. Research on evacuation path planning in single-story building fire based on genetic-ant colony algorithm. *J. Saf. Sci. Technol.* **2020**, *6*, 122–126.
51. Liang, Y.; Wang, H.; Hong, W.-C. Sustainable Development Evaluation of Innovation and Entrepreneurship Education of Clean Energy Major in Colleges and Universities Based on SPA-VFS and GRNN Optimized by Chaos Bat Algorithm. *Sustainability* **2021**, *13*, 5960. [[CrossRef](#)]
52. Dong, J.; Wang, Z.; Mo, J. A Phase Angle-Modulated Bat Algorithm with Application to Antenna Topology Optimization. *Appl. Sci.* **2021**, *11*, 2243. [[CrossRef](#)]
53. Rajalakshmi, M.; Chandramohan, S.; Kannadasan, R.; Alsharif, M.H.; Kim, M.-K.; Nebhen, J. Design and Validation of BAT Algorithm-Based Photovoltaic System Using Simplified High Gain Quasi Boost Inverter. *Energies* **2021**, *14*, 1086. [[CrossRef](#)]
54. Šipoš, M.; Klaić, Z.; Nyarko, E.K.; Fekete, K. Determining the Optimal Location and Number of Voltage Dip Monitoring Devices Using the Binary Bat Algorithm. *Energies* **2021**, *14*, 255. [[CrossRef](#)]
55. Kim, J.; Yoo, Y. Sensor Node Activation Using Bat Algorithm for Connected Target Coverage in WSNs. *Sensors* **2020**, *20*, 3733. [[CrossRef](#)]
56. Slim, M.; Rokbani, N.; Neji, B.; Terres, M.A.; Beyrouthy, T. Inverse Kinematic Solver Based on Bat Algorithm for Robotic Arm Path Planning. *Robotics* **2023**, *12*, 38. [[CrossRef](#)]
57. Liu, J.; Liu, Y.; Zhang, Q. A Gradient-Based Particle-Bat Algorithm for Stochastic Configuration Network. *Appl. Sci.* **2023**, *13*, 2878. [[CrossRef](#)]
58. Alphonse, A.S.; Abinaya, S.; Arikumar, K.S. A Novel Monogenic Sobel Directional Pattern (MSDP) and Enhanced Bat Algorithm-Based Optimization (BAO) with Pearson Mutation (PM) for Facial Emotion Recognition. *Electronics* **2023**, *12*, 836. [[CrossRef](#)]

59. Pang, A.; Liang, H.; Lin, C.; Yao, L. A Surrogate-Assisted Adaptive Bat Algorithm for Large-Scale Economic Dispatch. *Energies* **2023**, *16*, 1011. [[CrossRef](#)]
60. Yu, S.; Zhu, J.; Lv, C. A Quantum Annealing Bat Algorithm for Node Localization in Wireless Sensor Networks. *Sensors* **2023**, *23*, 782. [[CrossRef](#)]
61. Younas, W.; Ali, G.; Ahmad, N.; Abbas, Q.; Masood, M.T.; Munir, A.; ElAffendi, M. Improving Convergence Speed of Bat Algorithm Using Multiple Pulse Emissions along Multiple Directions. *Sensors* **2022**, *22*, 9513. [[CrossRef](#)]
62. Luo, Y.; Wu, C.; Leng, Y.; Huang, N.; Mao, L.; Tang, J. Throughput Optimization for NOMA Cognitive Relay Network with RF Energy Harvesting Based on Improved Bat Algorithm. *Mathematics* **2022**, *10*, 4357. [[CrossRef](#)]
63. Shen, Y.; Zheng, K.; Yang, Y.; Liu, S.; Huang, M. CBA-CLSVE: A Class-Level Soft-Voting Ensemble Based on the Chaos Bat Algorithm for Intrusion Detection. *Appl. Sci.* **2022**, *12*, 11298. [[CrossRef](#)]
64. Adel, A.; Omar, N.; Abdullah, S.; Al-Shabi, A. Co-Operative Binary Bat Optimizer with Rough Set Reducts for Text Feature Selection. *Appl. Sci.* **2022**, *12*, 11296. [[CrossRef](#)]
65. Kumar Mohapatra, P.; Kumar Rout, S.; Kishoro Bisoy, S.; Kautish, S.; Hamzah, M.; Jasser, M.B.; Mohamed, A.W. Application of Bat Algorithm and Its Modified Form Trained with ANN in Channel Equalization. *Symmetry* **2022**, *14*, 2078. [[CrossRef](#)]
66. Duan, M.; Huang, Q.; Xu, R.; Wang, C.; Xu, J. Optimization of Shearer Drum Based on Multi-Objective Bat Algorithm with Grid (MOBA/G). *Machines* **2022**, *10*, 733. [[CrossRef](#)]
67. Feng, J.; Kuang, H.; Zhang, L. EBBA: An Enhanced Binary Bat Algorithm Integrated with Chaos Theory and Lévy Flight for Feature Selection. *Future Internet* **2022**, *14*, 178. [[CrossRef](#)]
68. Ge, D.; Zhang, Z.; Kong, X.; Wan, Z. Extreme Learning Machine Using Bat Optimization Algorithm for Estimating State of Health of Lithium-Ion Batteries. *Appl. Sci.* **2022**, *12*, 1398. [[CrossRef](#)]
69. Damaševičius, R.; Maskeliūnas, R. Agent State Flipping Based Hybridization of Heuristic Optimization Algorithms: A Case of Bat Algorithm and Krill Herd Hybrid Algorithm. *Algorithms* **2021**, *14*, 358. [[CrossRef](#)]
70. Zheng, J.; Wang, Y. A Hybrid Bat Algorithm for Solving the Three-Stage Distributed Assembly Permutation Flowshop Scheduling Problem. *Appl. Sci.* **2021**, *11*, 10102. [[CrossRef](#)]
71. Qi, Y.; Cai, Y. Hybrid Chaotic Discrete Bat Algorithm with Variable Neighborhood Search for Vehicle Routing Problem in Complex Supply Chain. *Appl. Sci.* **2021**, *11*, 10101. [[CrossRef](#)]
72. Zheng, J.; Wang, Y.; Li, S.; Chen, H. The Stock Index Prediction Based on SVR Model with Bat Optimization Algorithm. *Algorithms* **2021**, *14*, 299. [[CrossRef](#)]
73. Zheng, J.; Wang, Y. A Hybrid Multi-Objective Bat Algorithm for Solving Cloud Computing Resource Scheduling Problems. *Sustainability* **2021**, *13*, 7933. [[CrossRef](#)]
74. Su, Y.; Liu, L.; Lei, Y. Structural Damage Identification Using a Modified Directional Bat Algorithm. *Appl. Sci.* **2021**, *11*, 6507. [[CrossRef](#)]
75. Gong, X.; Pei, J.; Wang, W.; Osman, M.K.; Jiang, W.; Zhao, J.; Deng, Q. Nature-Inspired Modified Bat Algorithm for the High-Efficiency Optimization of a Multistage Centrifugal Pump for a Reverse Osmosis Desalination System. *J. Mar. Sci. Eng.* **2021**, *9*, 771. [[CrossRef](#)]
76. Guerraiche, K.; Dekhici, L.; Chatelet, E.; Zebalah, A. Multi-Objective Electrical Power System Design Optimization Using a Modified Bat Algorithm. *Energies* **2021**, *14*, 3956. [[CrossRef](#)]
77. Lin, N.; Tang, J.; Li, X.; Zhao, L. A novel improved bat algorithm in uav path planning. *Comput. Mater. Contin.* **2019**, *61*, 323–344. [[CrossRef](#)]
78. Filali, D.; Dilshad, M.; Akram, M.; Babu, F.; Ahmad, L. Viscosity method for hierarchical variational inequalities and variational inclusions on Hadamard manifolds. *J. Inequalities Appl.* **2021**, *2021*, 66. [[CrossRef](#)]
79. Chen, Y.; Li, L.; Xiao, J.; Yang, Y.; Liang, J.; Li, T. Particle swarm optimizer with crossover operation. *Eng. Appl. Artif. Intell.* **2018**, *70*, 159–169. [[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.