Hindawi
International Journal of Aerospace Engineering
Volume 2022, Article ID 8857584, 14 pages
https://doi.org/10.1155/2022/8857584

Research Article

Application of the Improved Grey Wolf Algorithm in Spacecraft Maneuvering Path Planning

Changqing Wu, Xiaodong Han, Weiyu An, Jianglei Gong, and Nan Xu

Institute of Telecommunication Satellite, China Academy of Space Technology, Beijing 100094, China

Correspondence should be addressed to Xiaodong Han; willingdong@163.com

Received 25 July 2020; Revised 26 December 2020; Accepted 9 December 2021; Published 10 January 2022

Academic Editor: Giovanni Palmerini

Copyright © 2022 Changqing Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In many space missions, spacecraft are required to have the ability to avoid various obstacles and finally reach the target point. In this paper, the path planning of spacecraft attitude maneuver under boundary constraints and pointing constraints is studied. The boundary constraints and orientation constraints are constructed as finite functions of path evaluation. From the point of view of optimal time and shortest path, the constrained attitude maneuver problem is reduced to optimal time and path solving problem. To address this problem, a metaheuristic maneuver path planning method is proposed (cross-mutation grey wolf algorithm (CMGWO)). In the CMGWO method, we use angular velocity and control torque coding to model attitude maneuver, which increases the difficulty of solving the problem. In order to deal with this problem, the grey wolf algorithm is used for mutation and evolution, so as to reduce the difficulty of solving the problem and shorten the convergence time. Finally, simulation analysis is carried out under different conditions, and the feasibility and effectiveness of the method are verified by numerical simulation.

1. Introduction

As a low-cost spacecraft, small satellites have the characteristics of light, agility, and large coverage. They have become an ideal spacecraft for studying space, which has put forward new requirements for the movement of satellites [1]. With the development of space exploration technology and space commercial activities, the number of satellites in space has increased dramatically, which will pose a great threat to the safety of existing satellites and the next generation of space missions. There were 19,404 large objects and millions of pieces of debris in Earth orbit, which leads to huge challenges for space resources and the environment [2]. Therefore, owing to ensuring the safety of the spacecraft, it is necessary to avoid the debris when the spacecraft is flying. With the continuous development of space missions, it is extremely urgent to require the ability of rapid and large-angle attitude change, especially for the application of satellite positioning, disaster warning, scientific exploration, and other related tasks [1, 3].

When the spacecraft is maneuvering, it is necessary to prevent the light of bright celestial bodies (such as the sun) from entering the field of view of some optical sensors (such as infrared sensors or low light sensitive elements). Otherwise, the optical sensors will be temporarily blind or damaged. In the maneuvering, the direction vector of the solar array must be kept within a certain required range, so as to continuously provide electric energy. These directional constraints greatly limit the feasible region of spacecraft attitude maneuver [4]. In addition, the limitation of angular velocity and control torque will also affect the attitude maneuver path [5, 6]. Due to these complex constraints, time-optimized maneuvering path planning based on spacecraft guidance, navigation, and control systems is very challenging.

We transform the satellite orbit planning problem into the optimization problem of finding the best time and shortest path and establish the spacecraft attitude maneuver model by angular velocity-time-control torque coding. Compared with the latest and most advanced research [7, 8], our control model also takes into account more comprehensive constraints. The model proposed in this paper has better performance in solving the subsequent optimal path. In recent years, the metaheuristic algorithm has become more and more popular because of its simplicity advantages. In addition, the rapid development of computing technology makes up for the cost of computing. Many metaheuristic algorithms have been proposed. They can be subdivided into two categories: bionic algorithm and physical algorithm. The first is related to animal biology and behavior, while the second is based on the laws of physics. Among the most commonly used algorithms are the genetic algorithm [9], which is inspired by Darwin's theory of evolution; ant colony algorithm [10], which is inspired by the behavior of ants; particle swarm optimization algorithm [11], based on human social behavior and flock of birds; and differential evolution algorithm [12], based on the concept of mutation, reorganization, and selection applicable to swarms. Secondly, Takahama and Sakai proposed an improved constrained DE (differential evolution) algorithm [13] to generate the optimal feasible path. Furthermore, an artificial bee colony algorithm combined with evolutionary programming [14], fuzzy quadtree framework [15], improved traveling salesman problem algorithm [16], and improved pulse coupled neural network model [17] is also applicable to the path planning of agents. There are other algorithms for solving similar optimization problems [18–22]. Moreover, researchers have implemented many methods to find the optimal path in different environments. Yang et al. [23] proposed a finite angle $A*$ algorithm for path planning on satellite images. The algorithm uses a modified line of sight with a branching factor of 16 to improve the efficiency of the finite angle $A*$ algorithm. Duchon [24] and others proposed an improved $A*$ algorithm based on theta∗, phi∗, rectangular symmetry reduction, and hop search, which was implemented in symmetric and asymmetric environments. Lou and Tang [25] summarized the motion planning algorithms from the perspective of UGV guidance. The algorithm of path planning is discussed from the aspects of complexity and mapping with existing standards. One problem of metaheuristic algorithm is how to set some internal parameters correctly, such as maximum iteration times, population size, and internal coefficients. These parameters may change according to different problems that need to be optimized, because they will significantly affect the results of the optimization process. Therefore, this limits the use of such algorithms, which are only used by expert users in this field.

We have noticed that various improved grey wolf algorithms are being applied in various fields to solve various problems. For example, the novel random walking grey wolf optimizer proposed a new crossover operator-double distribution crossover operator (DDX). The performance of DDX is compared with that of the existing real-coded crossover operator, namely, Laplace crossover operator [26]. The random walk grey wolf optimizer for constrained engineering optimization problems proposes a simple constraint processing technique-constrained version GWO and proposes random walk GWO (RW-GWO) [27] by pointing out some shortcomings of the original GWO leaders in the process of searching prey. A new algorithm OCS-GWO is proposed by the opposition-based chaotic grey wolf optimizer for global optimization task. The algorithm effectively uses the search area by introducing opposition-based learning to approach the search candidate solution closer to the global optimum and chaotic local search, thus improving the performance of the original GWO algorithm [28]. The Cauchy grey wolf optimizer for continuous optimization problems proposes an improved classical GWO algorithm, which is named Cauchy-GWO. In the Cauchy-GWO operator, two new wolves are generated by the Cauchy distribution random number, and then, another new wolf is generated by convex combination of these new wolves [29]. The grey wolf optimizer with enhanced leadership inspiration for global optimization problems proposes an improved leadership-based GWO, namely, GLF-GWO. In GLF-GWO, the leader is updated through the Levy flight search mechanism. The proposed GLF-GWO algorithm improves the search efficiency of leading hunters in GWO and provides better guidance for accelerating the search process of GWO [30]. A hybrid grey wolf optimizer with a mutation operator hybridizes the grey wolf optimizer with differential evolution (DE) mutation and puts forward two versions of DE-GWO and GDE-GWO to avoid the stagnation of solution [31], as well as the reliability-redundancy allocation using the random walking grey wolf optimizer [32], memory-based grey wolf optimizer for global optimization task [33], and over-current relay optimization using the improved leadership-inspired grey wolf optimizer [34].

Because the traditional optimization algorithm has the disadvantages of slow convergence, inspired by predecessors [35, 36], we optimized the grey wolf algorithm [37] and applied it to satellite orbit planning. The grey wolf algorithm is a swarm intelligence optimization algorithm which simulates the predation behavior of grey wolves. The grey wolf algorithm has the characteristics of simple operation, few adjustment parameters, and easy programming. Compared with other swarm intelligence optimization algorithms, it has obvious advantages in function optimization. However, at the same time, it also has some shortcomings such as being easy to fall into the local optimum and low solution accuracy. Inspired by previous studies, this paper proposes a cross-mutation grey wolf algorithm in view of the shortcomings of the current algorithm. The algorithm uses the basic GWO algorithm to calculate the fitness value of each grey wolf. With the update of the wolf position, the individual in the wolf group constantly changes, which makes the population constantly update and effectively solves the problem of entering the local optimum. At the same time, the high efficiency of the grey wolf algorithm allows the time for the algorithm to be controlled. In order to improve the search ability of the algorithm, the fitness values of the grey wolf individuals are compared, and a penalty function is imposed on the grey wolf individuals with poor fitness values. By testing five benchmark functions, the results are compared with the traditional GWO algorithm to verify the superiority of the algorithm.

2. Spacecraft Constraints

In this section, how to formulate the satellite's maneuvering path into a behavior schedule optimization problem is presented.

Let us imagine a spacecraft working on the space station, some of those parameters are shown in Table 1.

Table 1: Spacecraft operating parameters.

The angular velocity	w
Control torque	u
Attitude (quaternion)	q
The moment of inertia	J
Spacecraft to the object attitude cosine matrix	R
The angle between some instruments and the sun	α
The orientation angle of the instrument itself	β

Only when some constraints [38, 39] are met can the spacecraft work better in space. That is, only under these conditions can the mission be carried out. Therefore, constraints are a prerequisite for planning trajectories.

2.1. Boundary Constraint. In the implementation process, in order to prevent the control torque from exceeding the boundary and causing adverse effects on the satellite, we define that the control torque is bounded. Furthermore, the following constraints should be satisfied: $|u| \leq \gamma_u, i = 1, 2, 3$.

During attitude maneuver, the angle of some sensors should not rotate too much; otherwise, it will produce excessive force. Therefore, the angular velocity needs to be limited within a certain range and the following constraints should be satisfied: $|w| \leq \gamma_w, i = 1, 2, 3$.

Boundary constraints are the prerequisite to ensure that it works properly, so when the model is initialized, boundary constraints should be added, such as setting a range of angular velocity: $-w$ to w, and keeping the angular velocity behind in this range.

2.2. Pose Kinematic and Dynamic Constraints. When it comes to the attitude of rigid spacecraft, if the four-dimensional space is considered, the unit quaternion is needed to represent the maneuver value of the parameters, and the purpose of avoiding singular value can be achieved. According to the relationship between moment of inertia, angular velocity, and torque, the constraint is

$$J = \operatorname{diag}(J_1, J_2, J_3), \tag{1}$$

$$w = [w_1, w_2, w_3]^T, \tag{2}$$

$$w^\times = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix}, \tag{3}$$

$$u = [u_1, u_2, u_3]^T, \tag{4}$$

where J is the moment of inertia, defined as the diagonal matrix of (J_1, J_2, J_3); w is the angular velocity, $w^\times$ prime is defined as the cross-product matrix of w, and u is the control torque.

After the calculation, the spacecraft should satisfy

$$J\dot{w} = u - w^\times Jw. \tag{5}$$

And $q = 1/2Lw$ and $\dot{q} = 1/2Mq$.

$$L = \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix}. \tag{6}$$

$$M = \begin{bmatrix} 0 & -w_1 & -w_2 & -w_3 \\ w_1 & 0 & w_3 & -w_2 \\ w_2 & -w_3 & 0 & w_1 \\ w_3 & w_2 & -w_1 & 0 \end{bmatrix}. \tag{7}$$

In conclusion, the attitude constraint is

$$\begin{cases} J\dot{w} = u - w^\times Jw, \\ \dot{q} = \dfrac{1}{2}Mq. \end{cases} \tag{8}$$

2.3. Pointing to the Constraint. The attitude orientation constraint is the constraint that the angle relation of a certain angle relative to a certain attitude must satisfy. It depends not only on the angle of its own devices but also on the external environment, such as the position of the sun. If the limit is exceeded, the spacecraft will be affected. It is generally classified into two types: forbidden constraint and mandatory constraint.

2.3.1. Forbidden Constraint. Some of the instruments in the spacecraft during the mission must avoid bright light; otherwise, it will damage the device. In mathematical terms, the instrument and the sun should be greater than the angle of view of the instrument itself. That is the forbidden constraint. m represents the direction vector of an optical sensitive element in the body coordinate system, and $n = [n_1, n_2, n_3]^T$ represents the direction vector of a bright celestial body in the inertial system. α is the field of view of the sensitive instrument. Constraints should satisfy $m^T(C_B n) \leq \cos\alpha$, among them:

$$m = [m_1, m_2, m_3]^T. \tag{9}$$

R is the cosine matrix from the spacecraft inertial coordinate system to the object attitude. C_B is the attitude cosine matrix of spacecraft from its own system to the inertial system. After simplification, the above inequality can be expressed as $q^T Hq \leq 0$. After calculation,

$$H = \begin{bmatrix} m^T n - \cos\alpha & \left(m^\times n^T\right)^T \\ m^\times n^T & nm^T + mn^T - \left(n^T m + \cos\alpha\right) I_3 \end{bmatrix}. \tag{10}$$

Among them, $m^\times$ is the cross-product of m. Therefore, the forbidden constraint is $q^T H q \le 0$.

2.3.2. Mandatory Constraint. On the other hand, some instruments may need to point at a directional angle when operating. For example, taking the solar panel of spacecraft as an example, v_B represents the direction vector of the solar panel in the body coordinate system, and the angle between n and v_B should be less than β. Namely, $v_B^T (C_B n) \ge \cos\beta$.

This is called a mandatory constraint. The constraints are as follows: $q^T K q \ge 0$:

$$K = \begin{bmatrix} n^T v_B - \cos\beta\left(v_B^\times n^T\right)^T \\ v_B^\times n^T n v_B^T + v_B n^T - \left(n^T v_B + \cos\beta\right) I_3 \end{bmatrix}. \tag{11}$$

All in all, the spacecraft needs to satisfy motion constraints, boundary constraints, forbidden constraints, and mandatory constraints during the mission. This is also a prerequisite for the spacecraft to complete its mission.

To sum up, the constraint to be satisfied is

$$\begin{cases} J\dot{w} = u - w^\times J w, \\ \dot{q} = \dfrac{1}{2} M q, \\ |u_i| \le \gamma_u, \quad i = 1, 2, 3, \\ |w_i| \le \gamma_w, \quad i = 1, 2, 3, \\ q^T H q \le 0, \\ q^T K q \ge 0. \end{cases} \tag{12}$$

3. Problem Statement

After studying the mathematical expression of the constraint, the improved grey wolf optimizer can be applied to solve the optimal control problem, and the optimal path can be screened through the improved crossover mutation. The evaluation function is set to the minimum value type, and the optimal path is obtained by searching for the minimum value of the evaluation function. The evaluation function is mainly divided into four parts: maneuver time, path direction constraint, energy bounded constraint, and terminal constraint.

$$Y_1 = t, \tag{13}$$

$$\begin{aligned} Y_2 = \exp&\left(-q(n)^T H_f q(n)\right) \\ &- 1\left\{\text{in case } q(n)^T H_f q(n) < 0, \text{otherwise it is } 0\right\}, \end{aligned} \tag{14}$$

$$Y = \sum_{n=1}^{n_{\max}} \left(Y_1 + Y_2\right). \tag{15}$$

The total function is simplified to $y = Y + Y_1$, a function to be minimized.

Therefore, we transformed the time-constrained attitude maneuver problem into the optimization problem of finding the minimum value vector m to minimize the total evaluation function and then adopted the improved GWO algorithm for optimization iteration in order to find the optimal solution.

In this paper, an improved GWO algorithm is proposed. Fitness function based on motion constraints and boundary constraints is established, and the optimal path is screened through cross-variation, which is conducive to solving the problem of the local optimum in the previous trajectory planning. Last but not least, cross-variation can provide a possibility for the spacecraft to jump out of the local optimum.

4. Cross-Mutation Grey Wolf Optimizer (CMGWO) Method

Since the grey wolf algorithm was proposed in 2014, it has been widely used in various engineering fields due to its advantages of simple parameter setting and strong optimization ability. As a young intelligent optimization algorithm, the grey wolf algorithm inevitably has similar defects. However, as far as the current research status is concerned, there is still room for further improvement in both the convergence accuracy and the convergence speed of the grey wolf algorithm. Therefore, it is necessary to improve the grey wolf algorithm in order to make it obtain more ideal optimization results.

There is a common problem in swarm intelligence optimization algorithms: the adjustment of local search performance and global search performance. An excellent swarm intelligence algorithm must have an excellent mechanism to adjust the global search performance and the local search performance. As a new member of the swarm intelligence algorithm, the grey wolf algorithm also has similar problems. Therefore, a good balance between global search performance and local search performance is indispensable for the grey wolf algorithm. Inspired by the differential evolution optimization algorithm, we merge the cross-mutation part of the algorithm into the iterative update process of the grey wolf algorithm. With the update of the position of the wolves, the individuals in the wolves are constantly mutating, so that the population is constantly updated, which can effectively solve the problem of adjusting the local search performance and global search performance of the original algorithm.

4.1. Improved Cross-Mutation Grey Wolf Algorithm. After comparing the optimization results of different algorithms, how to avoid falling into the local optimization and get the optimization results faster is only the main problem concerned. In this paper, a new grey wolf algorithm called the grey wolf genetic algorithm is proposed. This strategy is

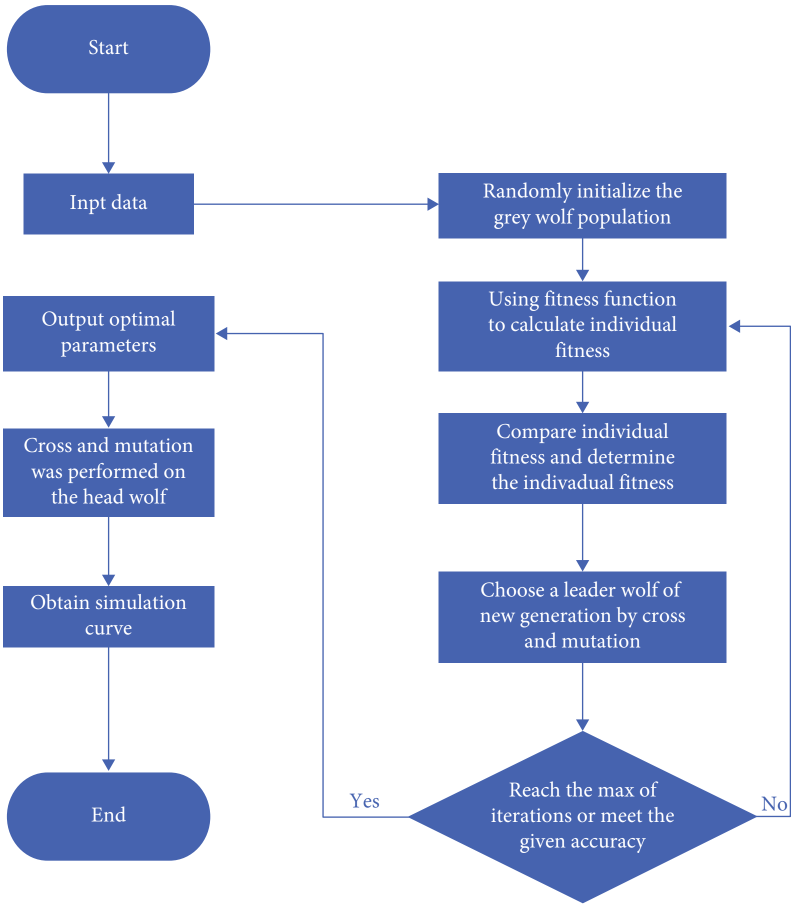
Start
Inpt data
Randomly initialize the grey wolf population
Using fitness function to calculate individual fitness
Compare individual fitness and determine the indivadual fitness
Choose a leader wolf of new generation by cross and mutation
Reach the max of iterations or meet the given accuracy
No
Yes
Output optimal parameters
Cross and mutation was performed on the head wolf
Obtain simulation curve
End

Figure 1: Algorithm flow diagram.

enlightened from two aspects of genetic mutation and grey wolf algorithm. With the updating of the wolf position, the individuals in the wolf group continue to mutate, so that the population keeps updating and the problem of getting into the local optimum can be solved effectively. Meanwhile, the high efficiency of the grey wolf algorithm makes the time of algorithm implementation controlled, and the improved grey wolf algorithm flow is shown in Figure 1.

4.1.1. Grey Wolf Algorithm. According to the previous idea, the common grey wolf is described as follows: at the top of the grey wolf hierarchy is the supreme leader of the grey wolf tribe, known as alpha (α) wolf, who is responsible for overall planning and decisions on hunting. The second layer of the grey wolf's social hierarchy is the second leader of the wolf pack, which is called beta (β) wolf. It mainly plays an assisting role in the overall planning and decision-making of alpha (α) wolf and is second only to alpha (α) wolf in the domination of the wolf population. At the third level of the grey wolf social hierarchy is delta (δ) wolf, also known as scouts, who are responsible for reconnaissance, watchkeeping, and guarding. The fourth level of the grey wolf social rank is the lowest in the grey wolf tribe, known as the omega (ω) wolf, which is essential for balancing the internal relations of the grey wolf tribe. The most suitable solution is considered to be the alpha (α) wolf. Therefore, the second and third best solutions were named beta (β) and omega (ω) wolves. The remaining candidate solutions were assumed to be omega wolves. And the three groups instructed the other wolves (omega) to search for the target. During the optimization process, the wolves update the positions of alpha, beta, delta, and omega; the grey wolf social hierarchy in the grey wolf algorithm is shown in Figure 2.

When the grey wolf population surrounds its prey, its predation behavior is defined as follows:

$$\vec{D} = \left| \vec{C} \cdot \vec{X}_p(t) - \vec{X}(t) \right|, \tag{16}$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D}. \tag{17}$$

Among them, $\vec{D}$ represents the distance between the individual wolf and the prey. The second formula is the updated formula of the individual wolf's position. $\vec{X}_p(t)$ is the position vector of the prey in generation t. $\vec{X}(t)$ is the position vector of the individuals in the t generation wolf tribe. $\vec{A}, \vec{C}$ are coefficient vectors, and their calculation formula is as follows:

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a}. \tag{18}$$

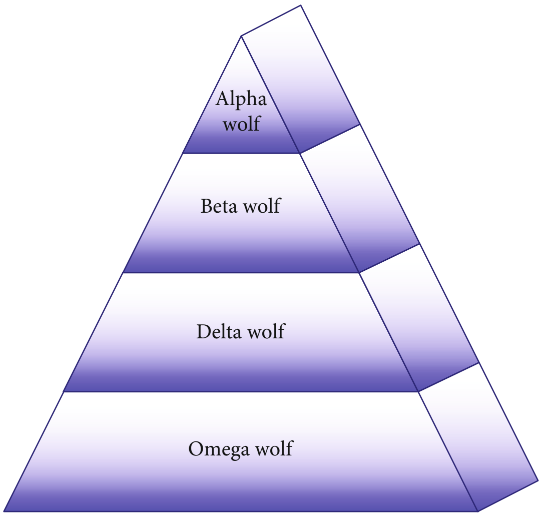

Figure 2: Grey wolf social hierarchy.

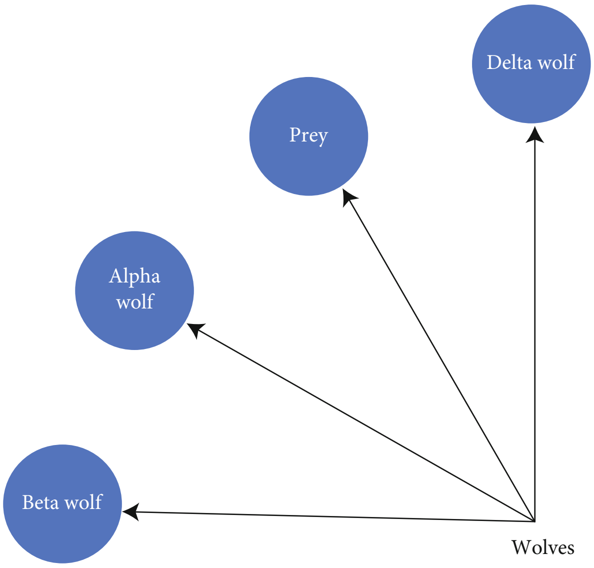

Figure 3: Schematic diagram of grey wolf location update.

$$\vec{C} = 2 \cdot \vec{r}_2. \tag{19}$$

Among them, $\vec{a}$ is the convergence factor. With the number of iterations decreasing linearly from 2 to 0, $\vec{r}_1$, $\vec{r}_2$ express the random number between 0 and 1.

The location update of grey wolf is shown in Figure 3.

When grey wolf catches prey, alpha (α) wolf at the first level leads other wolves to surround the prey. In the grey wolf population, alpha (α) wolf, beta (β) wolf, and delta (δ) wolf are the closest and most perceptive individuals. The location update of other grey wolf individuals is determined by the location of these three wolves. The schematic diagram and mathematical model of individual grey wolf searching for prey are as follows.

$$\vec{D}_{i,\alpha}(t) = \left|\vec{D}_1 \cdot \vec{X}_\alpha(t) - \vec{X}_i(t)\right|, \tag{20}$$

$$\vec{D}_{i,\beta}(t) = \left|\vec{D}_2 \cdot \vec{X}_\beta(t) - \vec{X}_i(t)\right|, \tag{21}$$

$$\vec{D}_{i,\delta}(t) = \left|\vec{D}_3 \cdot \vec{X}_\delta(t) - \vec{X}_i(t)\right|, \tag{22}$$

$$\vec{X}_{i,\alpha}(t) = \vec{X}_\alpha - \vec{A} \cdot \vec{D}_{i,\alpha}(t), \tag{23}$$

$$\vec{X}_{i,\beta}(t) = \vec{X}_\beta - \vec{A} \cdot \vec{D}_{i,\beta}(t), \tag{24}$$

$$\vec{X}_{i,\delta}(t) = \vec{X}_\delta - \vec{A} \cdot \vec{D}_{i,\delta}(t), \tag{25}$$

$$\vec{X}_i(t+1) = \frac{\left(\vec{X}_{i,\alpha}(t) + \vec{X}_{i.\beta}(t) + \vec{X}_{i,\delta}(t)\right)}{3}. \tag{26}$$

Among them, $\vec{D}_{i,\alpha}(t)$ represents the distance between the first generation of wolves and alpha (α) wolves, $\vec{D}_{i,\beta}(t)$ represents the distance between the second generation of grey wolves and beta (β) wolves, $\vec{D}_{i,\delta}(t)$ represents the distance between the second generation of wolves and delta (δ) wolves, the last three formulas define the step size and direction of omega (ω) wolves moving to the optimal solution alpha (α) wolves, beta (β) wolves, and delta (δ) wolves, and the last expression represents the new generation of grey wolf individuals after the location update.

4.2. *Cross-Mutation Grey Wolf Algorithm.*

Two different grey wolf individual vectors are randomly selected to subtract and generate the difference vector. After the difference vector is given the weight, it is added to the third randomly selected grey wolf individual vector to generate the variation vector. This operation is called variation. The variation vector and the target vector are mixed to generate the test vector, which is called crossover. With the renewal of wolves, new individuals of grey wolves are produced, which make up for the problem of population limit in the optimization process.

4.2.1. *Mutation Operation.*

The most basic variation component is the difference vector of the parent generation. Each vector pair includes two different individuals$(x_{r_1}^t, x_{r_2}^t)$ in the parent generation (generation t). And the difference vector is defined as

$$D_{r_{1,2}} = x_{r_1}^t - x_{r_2}^t, \tag{27}$$

where r_1 and r_2 represent index numbers of two different individuals in the population. The variation vector is generated by adding the difference vector to another randomly selected individual vector. For each target vector x_i^t, the mutation operation is that

$$v_i^{t+1} = x_{r_3}^t + F * \left(x_{r_1}^t - x_{r_2}^t\right). \tag{28}$$

$r_1, r_2, r_3 \in \{2, \cdots \mathrm{NP}\}$ are different integers, and r_1, r_2, r_3 are different from the current target vector index i, so the

```
Initialize the grey wolf population $X_i (i = 1, 2, \cdots, n)$
Initialize a, A and C
Calculate the fitness of each search agent
$X_\alpha$=the best search agent
$X_\beta$=the second-best search agent
$X_\delta$=the third best search agent
while (t < Max number of iterations)
    for each search agent
        Update the position of the current search agent by equation
    end for
    Update a, A, and C
    Calculate the fitness of all search agents
    Update $X_\alpha$,$X_\beta$,and $X_\delta$
    t=t+1
    end while
    return $X_\alpha$
```

Pseudocode 1: Pseudocode of the grey wolf algorithm.

population size $\mathrm{NP} \geq 4$. F is the scale factor, and the value range of F is $[0, 2]$ to control differential vector scaling.

4.2.2. Cross-Operation. For the target vector individuals x_i^t in the population, cross-operation v_i^{t+1} will be carried out with the variation vector to generate the experimental individuals u_i^{t+1}. In order to ensure the evolution of an individual x_i^t, first by random selection, at least one contributor v_i^{t+1} can be selected randomly in u_i^{t+1}. For others, a cross-probability factor CR can be used to determine which bit is contributed by. The equation of cross-operation is

$$u_i^{t+1} = \begin{cases} v_i^{t+1}, & \text{rand } (j) \leq \mathrm{CR} \text{ or } j = \text{rand } n(i), \\ x_i^t, & \text{rand } n(j) > \mathrm{CR} \text{ or } j \neq \text{rand } n(i), \end{cases} \tag{29}$$

where rand $n(j) \in [0, 1]$ is the random number with uniform distribution. j represents the j-th variable (gene). CR is the cross-probability constant, and its value range is $[0, 1]$. The size is determined in advance. rand $n(i) \in [1, 2, \cdots, D]$ is the index of randomly selected dimension variables to ensure that the test vector has at least dimension variables contributed by the variation vector. Otherwise, the test vector may be the same as the target vector and cannot generate new individuals. It can be seen from the formula that the larger the contribution CR, the more contribution v_i^{t+1} to u_i^{t+1}. When $\mathrm{CR} = 1$, $u_i^{t+1} = v_i^{t+1}$. This is conducive to local search and accelerating convergence rate. The smaller the CR, the more contribution x_i^t to u_i^{t+1}. When $\mathrm{CR} = 0$, $u_i^{t+1} = x_i^t$, which is conducive to maintaining the diversity of the population and global search.

4.3. Grey Wolf Genetic Algorithm Flow

(1) Initialization

Gm is the maximum number of iterations. nn is the number of nodes, $N = 5 * \mathrm{nn} + 1$ is the dimension of the problem, NP is the population size, $G = 1$ is the initial algebra, CR is hybrid parameters, u is the control moment, w is the angular velocity, and j is the moment of inertia.

(2) Initialize the three leaders

Calculate the fitness function of each grey wolf, screen out the smallest three fitness values, and use their corresponding individuals as the corresponding alpha wolf, beta wolf, and delta wolf.

(3) Reinitialize particles

Update the lead wolf position:

$$\vec{D}_{i,\alpha}(t) = \left| \vec{D}_1 \cdot \vec{X}_\alpha(t) - \vec{X}_i(t) \right|, \tag{30}$$

$$\vec{D}_{i,\beta}(t) = \left| \vec{D}_2 \cdot \vec{X}_\beta(t) - \vec{X}_i(t) \right|, \tag{31}$$

$$\vec{D}_{i,\delta}(t) = \left| \vec{D}_3 \cdot \vec{X}_\delta(t) - \vec{X}_i(t) \right|, \tag{32}$$

$$\vec{X}_{i,\alpha}(t) = \vec{X}_\alpha - \vec{A} \cdot \vec{D}_{i,\alpha}(t), \tag{33}$$

$$\vec{X}_{i,\beta}(t) = \vec{X}_\beta - \vec{A} \cdot \vec{D}_{i,\beta}(t), \tag{34}$$

$$\vec{X}_{i,\delta}(t) = \vec{X}_\delta - \vec{A} \cdot \vec{D}_{i,\delta}(t), \tag{35}$$

$$\vec{X}_i(t+1) = \frac{\left(\vec{X}_{i,\alpha}(t) + \vec{X}_{i.\beta}(t) + \vec{X}_{i,\delta}(t) \right)}{3}. \tag{36}$$

(4) Cross

```
Initialize the grey wolf population $X_i (i = 1, 2, \cdots, n)$
Initialize a, A and C
Calculate the fitness of each search agent
population[t] = initialize Population (population Size)
evaluate Population(population[t]
$X_\alpha$=the best search agent
$X_\beta$=the second-best search agent
$X_\delta$=the third best search agent
while (t < Max number of iterations)
    for each search agent
        Update the position of the current search agent by equation
    end for
    population[t+1] = crossover(population[t])
    population[t+1] = mutate(population[t+1])
    evaluate Population(population[t])
    Update a, A, and C
    Calculate the fitness of all search agents
    Update $X_\alpha$,$X_\beta$,and $X_\delta$
    t=t+1
    end while
    return $X_\alpha$
```

Pseudocode 2: Pseudocode of the cross-mutation grey wolf algorithm.

$$u_i^{t+1} = \begin{cases} v_i^{t+1}, & \text{rand}\ (j) \le \text{CR or } j = \text{rand}\ n(i), \\ x_i^t, & \text{rand}\ n(j) > \text{CR or } j \ne \text{rand}\ n(i). \end{cases} \tag{37}$$

(5) Mutation

$$D_{r_{1,2}} = x_{r_1}^t - x_{r_2}^t. \tag{38}$$

$$v_i^{t+1} = x_{r_3}^t + F * \left(x_{r_1}^t - x_{r_2}^t \right). \tag{39}$$

(6) Competition

The particles are brought into the constraint function to calculate their fitness function values, and the minimum three are filtered out, which is the minimum value of this generation.

The constraints to be satisfied are

$$\begin{cases} J\dot{w} = u - w^\times J w, \\ q = \dfrac{1}{2} M q, \\ u_i \le u \le u_j, \\ w_i \le w \le w_j, \\ q^T H q \le 0, \\ q^T K q \ge 0. \end{cases} \tag{40}$$

Iteratively update and calculate the minimum value of each generation. Then, bring the minimum value of each generation into the fitness function to calculate the minimum value, which is the required value.

(7) End

5. Simulation and Analysis

Basing on the research of previous scientists, we found some questions. In the first segment of this chapter, we will introduce the cross-mutation grey wolf optimizer algorithm and conduct simulation analysis. Later in this section, we will show the simulation results to get the advantages of this algorithm. Finally, we will compare the cross-mutation grey wolf Optimizer with the traditional GWO to verify the validity of the cross-mutation grey wolf optimizer.

5.1. Parameter Initialization. In the process of spacecraft, the vector components of four strong light celestial bodies in the inertial system are r_A, r_B, r_C, and r_D and position vectors are r_1 and r_2. The initial time of spacecraft is at position r. The minimum angle between r and r_1 is required to be α_1, and the minimum angle between r and r_2 is required to be α_2. The included angle of self-instrument is β. The initial attitude and angular velocity of the spacecraft are, respectively, q and ω, as shown in Table 2.

Besides, after data deduction, the condition of forbidden constraint is that the range of ω is from -0.512 to 0.512 and the range of u is from -0.114 to 0.114.

Table 2: Numerical simulation of spacecraft operating parameters.

Variable	Simulation value
w	[0, 0, 0] rad/s
q	[-0.383, 0, 0.926, 0]
J	diag(1500,1500,1500)kg·m^2
r_1	$[0.56, 0.31, -0.18]^T$
r_2	$[0.65, -0.19, 0.54]^T$
α_1	15°
α_2	30°
β	15°
r_A	$[0,0.2,0.5]$
r_B	$[0.8,0.3,-0.4]$
r_C	$[0.5,0.3,-0.1]$
r_D	$[0.8,0.6,0.2]$
r	$[0, 0, 1]^T$
γ_w	0.512
γ_u	0.114

5.2. Drawing Results and Result Analysis. The following is the simulation results of the cross-mutation grey wolf optimizer:

Using the cross-mutation grey wolf optimizer algorithm to solve, when $\triangle T$ is 1 s, the total time of the maneuvering process is 210 s. The number of path node is 210. Figure 4 shows the attitude maneuver path of spacecraft in the spherical coordinate system of spacecraft, in which two red stars are the starting point and the ending point. We can see that in the process of movement, spacecraft is out of strong light and avoid being directly at the beam of strong light celestial bodies. So, the optimization result is wonderful; the trajectory of the infrared telescope vector in the celestial coordinate system of the cross-mutation grey wolf optimizer is shown in Figure 4, where blue dots indicate obstacles.

Figures 5 and 6 are the change of w, q, and u of the satellite in the application of the cross-mutation GWO algorithm.

According to the boundary constraint, when the spacecraft is working, w should be within a constraint. The attitude model can predict the change of w. Experimental data confirm this, and we find that the range of change is small and is in the boundary constraint, as shown in Figure 5.

Control torque is also within the constraint range and finally tends to be stable, which is a good thing for the spacecraft and is conducive to stable execution, as shown in Figure 6.

We found that both angular velocity and control torque are within their range; that is to say, boundary constraints are satisfied. It is a good proof of feasibility. Secondly, the curves of angular velocity and moment are also very good, which conform to the normal working parameters of spacecraft and will not cause failure to spacecraft.

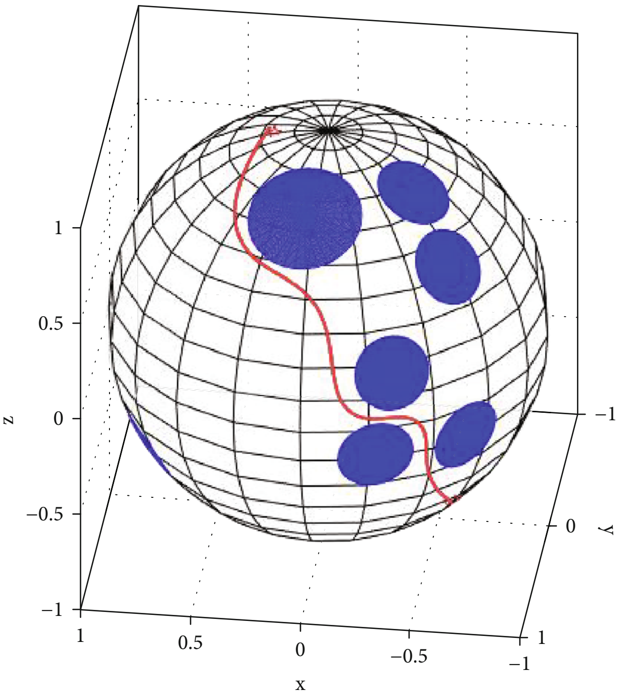

Figure 4: Trajectory of the infrared telescope vector in the celestial coordinate system of the cross-mutation grey wolf optimizer.

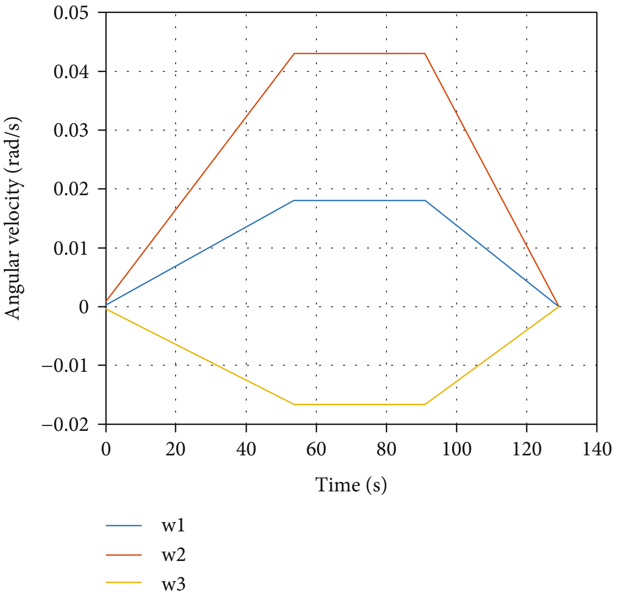

Figure 5: Angular velocity time history based on the cross-mutation grey wolf optimizer.

By boundary constraint and attitude constraint, the above two results are obtained, which strongly supports the key assumption that the cross-mutation grey wolf optimizer can optimize the flight stability of spacecraft.

The change of quaternion is very smooth, and the range is large, so it is easy to face all kinds of situations. And it is helpful to avoid singular value, as shown in Figure 7.

The calculation time curve of the cross-mutation grey wolf optimizer is shown in Figure 8.

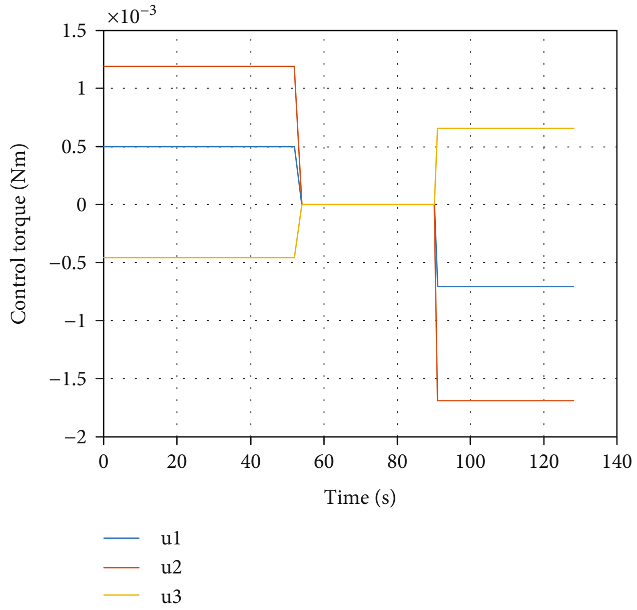

Figure 6: Control torque time history based on the cross-mutation grey wolf optimizer.

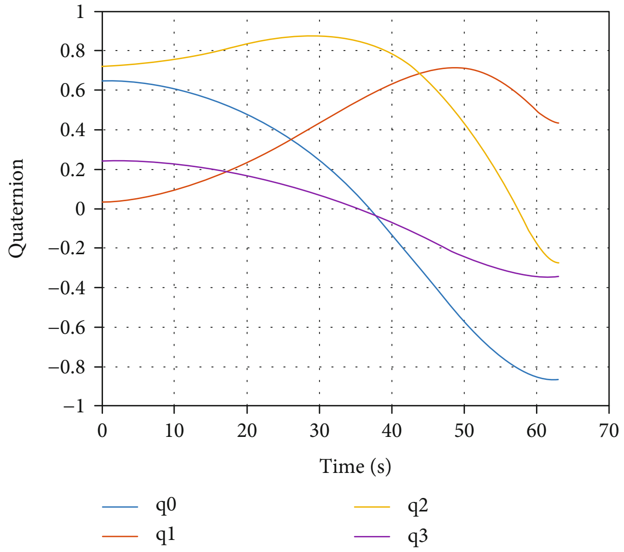

Figure 7: Attitude quaternion time history based on the cross-mutation grey wolf optimizer.

The cross-mutation grey wolf optimizer algorithm solves some shortcomings of the previous algorithm. For example, the previous algorithm is unable to cope with wide range of parameters and the influence of nonlinear factors on the system. Besides, the cross-mutation grey wolf optimizer improved computing and motion speed. The improved curve is very stable, and it can obviously improve the steady and dynamic performance of the system and has better control effect.

The control effect of the cross-mutation grey wolf optimizer algorithm is verified by numerical simulation. From the result chart, the effect is very useful, and it can effectively control the spacecraft to work. At the same time, the cross-mutation operation is good for jumping out of the local optimal value. In trajectory planning, the optimal route can be planned faster and better.

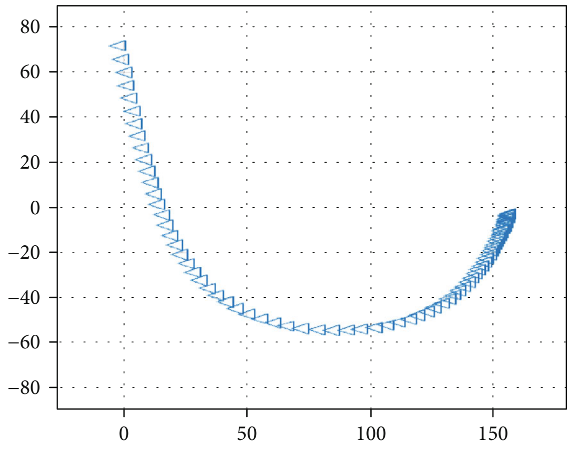

Figure 8: The computation time curve of the cross-mutation grey wolf optimizer.

Table 3: Algorithm simulation experiment data comparison.

Times maneuver time algorithm	GWO	Cross-mutation grey wolf optimizer
1	258.38	212.20
2	261.24	209.35
3	266.15	214.34
4	256.36	213.22
5	260.53	214.97
6	255.87	216.75
7	259.61	208.67
8	261.98	209.66
9	264.62	207.48
10	259.41	209.11
Average time	260.415	211.575

It is not difficult to verify the optimization stability of the cross-mutation grey wolf optimizer algorithm through pictures. The curves are smooth and the straight-line transition is very smooth, which proves the stability of the cross-mutation grey wolf optimizer algorithm.

Besides, we choose to simulate spacecraft flights many times to be samples to draw the pictures, which can predict most situations. By doing that, we can adjust parameters to ensure the comprehensiveness of information collection. This will improve the stability of the spacecraft.

5.3. The Comparison of the Cross-Mutation Grey Wolf Optimizer and Traditional GWO. In order to verify the effectiveness of the cross-mutation grey wolf optimizer algorithm, we choose the traditional GWO algorithm to be applied to the model and compare it with the cross-mutation algorithm.

Table 4: Evaluating indicator.

Algorithm	Boundary constraints	Forbidden constraint	Computing time	Movement time	The change of w and u
Traditional GWO	×	√	105.2 s	265 s	×
Cross-mutation grey wolf optimizer	√	√	95 s	210 s	√

Under the same simulation conditions, we also make experiment with the traditional GWO algorithm which takes 265 s. With the increase of the number of experiments, the traditional GWO is easy to fall to the local optimum. And it can only avoid most celestial bodies. Trajectory cannot avoid all objects.

Because of the randomness of the algorithm, we have carried out several experiments and recorded the time, as shown in Table 3.

In addition, we conducted 100 more experiments. We used the GWO and cross-mutation grey wolf optimizer algorithm to calculate the satellite's maneuver time. We can draw the conclusion that the maneuvering time of GWO with cross-variation is about 210 s. The same GWO maneuver time is between 260 s and 270 s, as shown in Table 4.

5.3.1. Interpretations

(1) According to the image of angular velocity of the traditional GWO, we find that the maximum value of w exceeds the boundary constraint 5.12; that is, boundary constraints are not satisfied. Spacecraft is prone to failure. And the cross-mutation grey wolf optimizer solves this problem very well, as shown in Figure 4

(2) In terms of time, no matter it is computing time or maneuvering time, the cross-mutation grey wolf optimizer is shorter than the traditional GWO showing that the cross-mutation grey wolf optimizer algorithm is more efficient. It is also very stable in execution and not prone to errors

(3) Another possibility is that the traditional GWO algorithm produces a local optimal value in operation. Therefore, it is not possible to quickly iterate the head wolf and filter out the minimum value, thus increasing the calculation time. However, the cross-mutation grey wolf optimizer solves this problem well

(4) For the change of ω and u, the change of the cross-mutation grey wolf optimizer is smoother than that of common GWO range which is also more reasonable, so it has better application effect

In addition, we also explore the local optimization of the two algorithms, because the cross-mutation grey wolf optimizer can effectively improve the local optimum, so we did 100 experiments and recorded the times of falling into the local optimum, as shown in Figure 9.

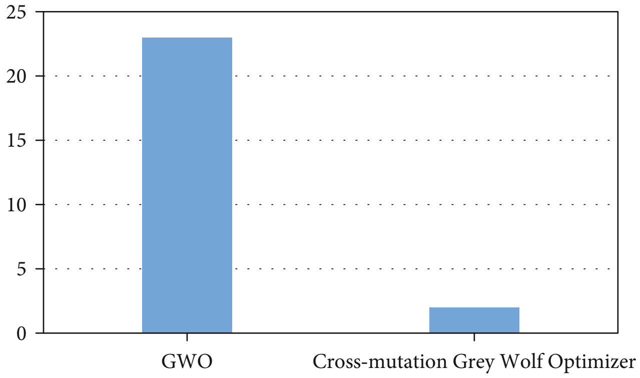

Figure 9: Discussion on the situation of the algorithm falling into the local optimum.

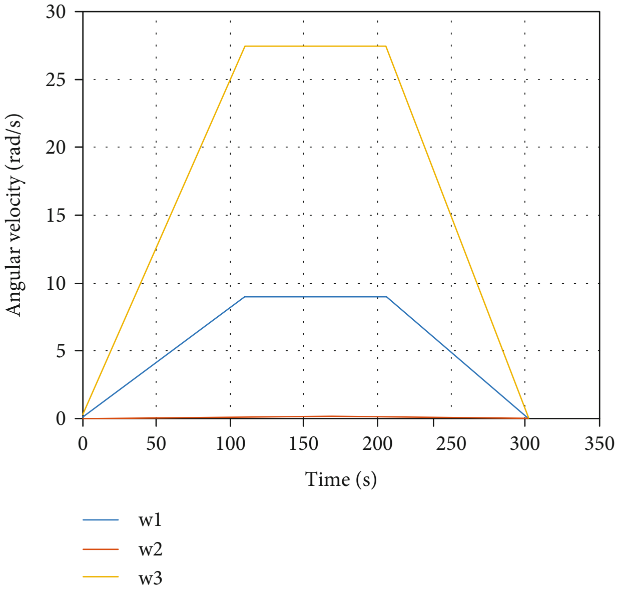

Figure 10: Angular velocity time history based on the grey wolf optimizer.

It can be seen from the chart that cross and mutation can significantly reduce the number of times falling into the local optimum, which is effective. In practical application, it also performs well.

The following figures are the simulation diagram under the grey wolf optimizer algorithm.

The time history analysis of angular velocity based on the grey wolf optimization algorithm is shown in Figure 10.

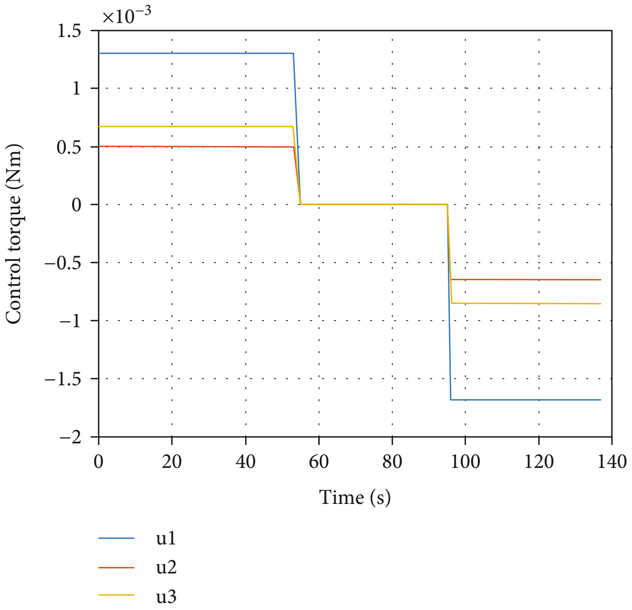

Figure 11: Control torque time history based on the grey wolf optimizer.

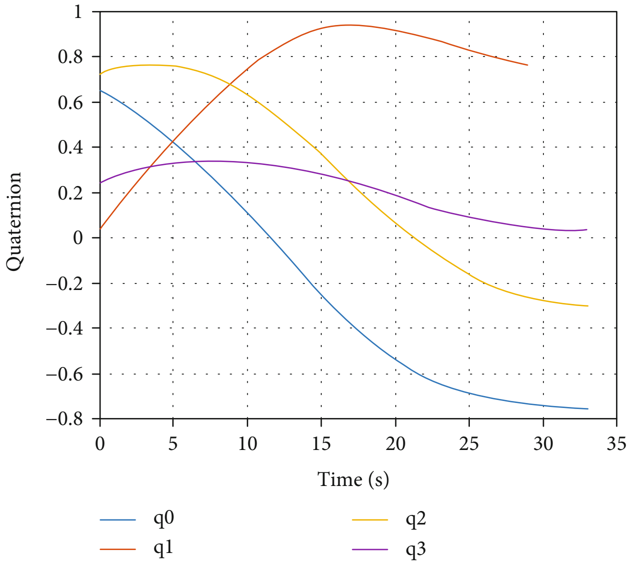

Figure 12: Attitude quaternion time history based on the grey wolf optimizer.

The control torque time history based on the grey wolf optimizer is shown in Figure 11.

The attitude quaternion time history based on the grey wolf optimizer is shown in Figure 12.

The computation time curve of the grey wolf optimizer is shown in Figure 13.

The trajectory of the infrared telescope vector in the celestial coordinate system of the grey wolf optimizer is shown in Figure 14.

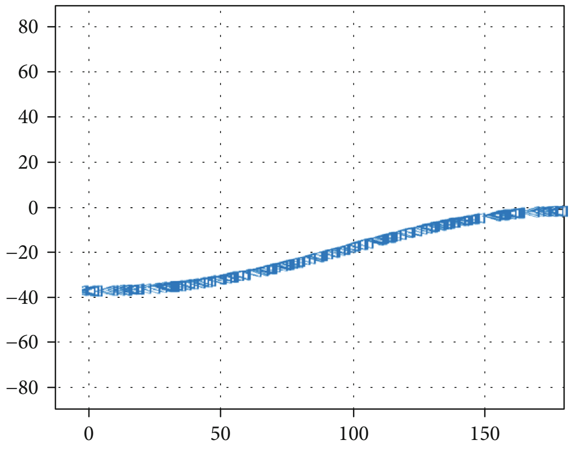

Figure 13: The computation time curve of the grey wolf optimizer.

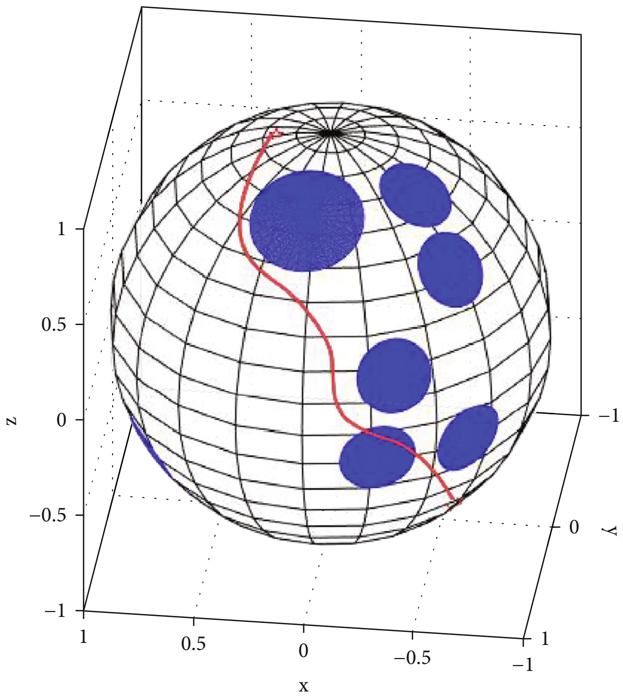

Figure 14: Trajectory of the infrared telescope vector in the celestial coordinate system of the grey wolf optimizer.

6. Conclusions

This paper describes and analyzes the boundary and pointing constraints of the spacecraft trajectory planning problem based on time and path optimization. Firstly, a mathematical model based on boundary constraints and pointing constraints is established, and then, they are constructed as a finite function of path evaluation. In order to solve the path planning problem under the condition of satisfying the constraints, we propose a new attitude maneuver path planning method—cross-mutation grey wolf algorithm, which improves the diversity of the wolves' offspring through cross-mutation and obtains the nearest optimal path based on iterative rules. Simulation results show that other forms of the GWO algorithm have good applicability in the process of spacecraft path planning, and the simulation results prove the feasibility of applying grey wolf to the spacecraft control model to solve the shortest path, although there are some problems such as unsatisfying constraints or taking too long to optimize. The CMGWO method performs well in simulation experiments, and it fits perfectly with the control model and not only satisfies the boundary constraints and direction constraints but also greatly shortens the attitude maneuver time and significantly reduces the times of falling into the local optimum. In addition, the angular velocity and control torque curves obtained by this method are smoother and gentler, which has a good application prospect in the future. In the future research, a single satellite may not be able to complete the task accurately and quickly in some cases, which indicates that the difficulty of solving the problem will continue to increase, and it will take more time to plan the attitude maneuver path. The CMGWO method also has the potential to be applied to the trajectory planning task of multiple satellites. Finally, the proposed numerical test is aimed at a specific situation. For some reasons, the simulation cannot be extended, but its universality is worthy of recognition

Data Availability

The raw data required to reproduce these findings cannot be shared at this time as the data also forms part of an ongoing study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (No. 61972398).

References

[1] L. C. Lai, C. C. Yang, and C. J. Wu, "Time-optimal maneuvering control of a rigid spacecraft," *Acta Astronautica*, vol. 60, no. 10-11, pp. 791–800, 2007.

[2] P. D. Anz-Meador, "Orbital Debris Quarterly News," NASA, Washington, DC, USA, 2019.

[3] H. Zhou, D. Wang, B. Wu, and E. K. Poh, "Time-optimal reorientation for rigid satellite with reaction wheels," *International Journal of Control*, vol. 85, no. 10, pp. 1452–1463, 2012.

[4] N. Sueoka, "Directional mutation pressure, selective constraints, and genetic equilibria," *Journal of Molecular Evolution*, vol. 34, no. 2, pp. 95–114, 1992.

[5] G. Mengali and A. A. Quarta, "Spacecraft control with constrained fast reorientation and accurate pointing," *The Aeronautical Journal*, vol. 108, no. 1080, pp. 85–91, 2004.

[6] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.

[7] F. Celani and D. Lucarelli, "Spacecraft attitude motion planning using gradient-based optimization," *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 6, pp. 1–6, 2019.

[8] J. D. Biggs and L. Colley, "Geometric attitude motion planning for spacecraft with pointing and actuator constraints," *Journal of Guidance Control & Dynamics*, vol. 39, no. 7, pp. 1672–1677, 2016.

[9] D. E. Goldberg, *Genetic Algorithm in Search Optimization and Machine Learning*, Addison Wesley, 1989.

[10] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[11] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, pp. 1942–1948, Piscataway, 1995.

[12] R. Storn and K. V. Price, "Differential evolution–a simple and efficient adaptive scheme for global optimization over continuous spaces," Tech. Rep. TR-95-012, International Computational Science Institute, Berkley, Mich, USA, 1995.

[13] T. Takahama and S. Sakai, "Constrained optimization by the ε constrained differential evolution with gradient-based mutation and feasible elites," in *2006 IEEE International Conference on Evolutionary Computation*, Vancouver, BC, Canada, 2006.

[14] M. A. Contreras-Cruz, V. Ayala-Ramirez, and U. H. Hernandez-Belmonte, "Mobile robot path planning using artificial bee colony and evolutionary programming," *Applied Soft Computing Journal*, vol. 30, pp. 319–328, 2015.

[15] S. Ghosh, A. Halder, and M. Sinha, "Micro air vehicle path planning in fuzzy quadtree framework," *Applied Soft Computing*, vol. 11, no. 8, pp. 4859–4865, 2011.

[16] L.-N. Xing, Y.-W. Chen, K.-W. Yang, F. Hou, X.-S. Shen, and H.-P. Cai, "A hybrid approach combining an improved genetic algorithm and optimization strategies for the asymmetric traveling salesman problem," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 8, pp. 1370–1380, 2008.

[17] H. Qu, S. X. Yang, A. R. Willms, and Z. Yi, "Real-time robot path planning based on a modified pulse-coupled neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 11, pp. 1724–1739, 2009.

[18] V. Roberge, M. Tarbouchi, and G. Labonte, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 132–141, 2013.

[19] R. el Sehiemy, A. Abou el-Ela, and A. A. Shaheen, "Multi-objective fuzzy-based procedure for enhancing reactive power management," *Transmission and Distribution*, vol. 7, no. 12, pp. 1453–1460, 2013.

[20] R. E. Precup, R. C. David, E. M. Petriu, S. Preitl, and M. B. Rădac, "Fuzzy logic-based adaptive gravitational search algorithm for optimal tuning of fuzzy-controlled servo systems," *IET Control Theory and Applications*, vol. 7, no. 1, pp. 99–107, 2013.

[21] M. J. Gacto, M. Galende, R. Alcalá, and F. Herrera, "METSK-HD^e: a multiobjective evolutionary algorithm to learn accurate TSK-fuzzy systems in high-dimensional and large-scale regression problems," *Information Sciences*, vol. 276, pp. 63–79, 2014.

[22] H. Niu, Y. Lu, A. Savvaris, and A. Tsourdos, "Efficient path planning algorithms for unmanned surface vehicle," *IFAC-Papers OnLine*, vol. 49, no. 23, pp. 121–126, 2016.

[23] J. M. Yang, C. M. Tseng, and P. S. Tseng, “Path planning on satellite images for unmanned surface vehicles,” *International Journal of Naval Architecture and Ocean Engineering*, vol. 7, no. 1, pp. 87–99, 2015.

[24] F. Duchon, “Path planning with modified a star algorithm for a mobile robot,” *Procedia Engineering*, vol. 96, pp. 59–69, 2014.

[25] L. Jia and T. Bin, “Application of new grey wolf optimization algorithm in function optimization,” *Journal of Lanzhou University of Technology*, vol. 6, no. 3, pp. 97–101, 2016.

[26] S. Gupta and K. Deep, “A novel random walk grey wolf optimizer,” *Swarm and Evolutionary Computation*, vol. 44, pp. 101–112, 2019.

[27] S. Gupta and K. Deep, “Random walk grey wolf optimizer for constrained engineering optimization problems,” *Computational Intelligence*, vol. 34, no. 4, pp. 1025–1045, 2018.

[28] S. Gupta and K. Deep, “An opposition-based chaotic grey wolf optimizer for global optimisation tasks,” *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 31, no. 5, pp. 751–779, 2019.

[29] S. Gupta and K. Deep, “Cauchy grey wolf optimiser for continuous optimisation problems,” *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 30, no. 6, pp. 1051–1075, 2018.

[30] S. Gupta and K. Deep, “Enhanced leadership-inspired grey wolf optimizer for global optimization problems,” *Engineering with Computers*, vol. 36, no. 4, pp. 1777–1800, 2020.

[31] S. Gupta and K. Deep, “Hybrid grey wolf optimizer with mutation operator,” in *Soft Computing For Problem Solving*, pp. 961–968, Springer, Singapore, 2019.

[32] S. Gupta, K. Deep, and A. Assad, “Reliability–redundancy allocation using random walk gray wolf optimizer,” in *Soft Computing for Problem Solving*, pp. 941–959, Springer, Singapore, 2020.

[33] S. Gupta and K. Deep, “A memory-based grey wolf optimizer for global optimization tasks,” *Applied Soft Computing*, vol. 93, 2020.

[34] S. Gupta and K. Deep, “Optimal coordination of overcurrent relays using improved leadership-based grey wolf optimizer,” *Arabian Journal for Science and Engineering*, vol. 45, no. 3, pp. 2081–2091, 2020.

[35] Z. Wei, H. Zhao, H. Huang, X. Wang, and R. Zhou, “Multi-UCAV cooperative attack target decision based on adaptive GWO,” *Computer Engineering and Applications*, vol. 25, no. 18, pp. 97–101, 2016.

[36] J. C. Yang and W. Long, “Improved grey wolf optimization algorithm for constrained mechanical design problems,” *Applied Mechanics and Materials*, vol. 851, pp. 553–558, 2016.

[37] S. Mirjalili, S. Mirjalili, and A. Lewis, “Grey wolf optimizer,” *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[38] C. Wu and X. Han, “Energy-optimal spacecraft attitude maneuver path-planning under complex constraints,” *Acta Astronautica*, vol. 157, pp. 415–424, 2019.

[39] R. Xu, C. Wu, S. Zhu et al., “A rapid maneuver path planning method with complex sensor pointing constraints in the attitude space,” *Information Systems Frontiers*, vol. 19, no. 4, pp. 945–953, 2017.