

RESEARCH ARTICLE

Three-dimensional truss path planning of cellular robots based on improved sparrow algorithm

Ye Dai , Shikun Li, Xinda Chen, Xinlei Nie, Xukun Rui and Qihao Zhang

Key Laboratory of Advanced Manufacturing Intelligent Technology of Ministry of Education, Harbin University of Science and Technology, Harbin, 150080, China

Corresponding author: Ye Dai; Email: daiye312@163.com

Received: 21 July 2023; **Revised:** 20 September 2023; **Accepted:** 13 October 2023; **First published online:** 10 November 2023

Keywords: modular robots; mobile robots; path planning; space robotics; design

Abstract

As space missions are needed in the future, the assembly volume of the space truss will become larger and larger, and the advancing path of the on-orbit cellular robot to the mission target will become more and more complicated. If the shortest moving path cannot be found in the truss environment, the climbing time of the robot on the truss will be greatly increased. To improve the speed of the cellular robot moving to the target point on the large space truss, this paper designs a cellular robot structure and configuration suitable for climbing on the truss and uses the improved sparrow algorithm to solve the problem of robot motion path planning. By establishing a mathematical model of the space truss, the improved sparrow algorithm is used to find the shortest path between the starting point and the end point in the truss environment. Finally, the data of this algorithm are compared with the data of other algorithms. The data results show that the improved sparrow algorithm is very effective in solving the optimal path of the space truss. The improved sparrow algorithm keeps the same optimal path compared with the standard sparrow algorithm, and the overall reaction time is increased by 51.60%, and the number of effective iterations is increased by about 13.87%.

1. Introduction

In recent years, with the steady development of the space industry, the ability of human beings to explore and develop space has been greatly improved, and human beings have begun to increase the exploration and development of space resources, and the corresponding large-scale space structures have become more and more. These large-scale space structures are generally assembled with trusses. Under the circumstances of a heavy workload, it is difficult to achieve the task of truss assembly only for astronauts manually. Moreover, the space environment has potential safety hazards for astronauts who are working outside the cabin, and the manual installation of space trusses by astronauts is limited. It costs a lot of manpower and is dangerous. Therefore, some space control devices are needed to replace astronauts to complete the task. The existing space manipulators are usually fixed in configuration and single in function. It is necessary to launch a number of space control devices with different functions and configurations, but at the same time, it leads to a lot of money consumption [1]. Moreover, most traditional space robots are developed for the task of determining the scene, which has the disadvantages of high cost, single function, and low redundancy [2]. In this context, a reconfigurable cellular robot working in orbit in space has attracted researchers' attention again. A space cell robot is a kind of robot that uses the multilevel structure between organs, tissues, and cells to reconstruct various configurations according to the requirements of space tasks. It originated from the concept of Cebot [3] put forward by Fukuda T and others. With the rise of the concept of Cellular Satellites [4], the space cell robot has become familiar to more researchers, and it has also become a popular research direction concept and a research hotspot in the aerospace industry.

In the aspect of cellular robot structure, many scholars have done relevant research. White et al. have designed an Xbot module with telescopic ability, which relies on the connection of magnetic units distributed at four vertices to form a dynamic chain structure, which can shrink the overall size according to the working environment and has high flexibility, but the whole telescopic movement is completed by external drive, and its autonomy is greatly weakened [5]. Piranda et al. [6] proposed a spherical module made of programmable substances, which realized the connection and movement between different modules by using human–computer interaction function and material characteristics, which broke through the traditional module design idea and provided a reference for the design of new modules. Liu [7] proposed a cubic reconfigurable modular robot. This module has a cubic structure, and each face has a connection interface. The two connection interfaces on the opposite face are different: one is the active connection interface and the other is the passive connection interface, which is a female head. When the two modules are butted, the internal driving mechanism of the male-butted module drives the male head to rotate and adjust the posture, and when the male head and the female head are coaxial, the female head and the male head are pushed to finish butting. In order to ensure stable posture and easy operation, a special base is designed for the cube module. The base is evenly distributed with female heads, which can connect and support the module to ensure the stability of the module. Wen [8] designed three basic cell units, namely connective cells, swing cells, and claw cells. In order to ensure the guiding function in the process of cell unit reconstruction and docking, both active and passive connecting mechanisms are designed as guide cone structures, which increases the efficiency and tolerance of cell unit docking, and the active and passive docking mechanisms are mechanically locked by locking blocks, which increases the connection reliability of cell unit. The connecting cell mainly undertakes the expansion function of the robot, and its six surfaces can be active or passive connecting surfaces. In order to enable the cellular robot to deform to meet different task requirements, we can consider changing the position of the interface between adjacent cell units. Because the active and passive interfaces of these cell units are all the same and the active and passive connecting mechanisms are all integrated on a panel, we can change the position of the cube cell unit at will.

Many scholars have also studied path planning. In the realm of robot trajectory planning, researchers have introduced various improvements and modifications to existing algorithms to optimize performance. Wu et al. [9] proposed a bidirectional adaptive A* algorithm as an improved version of the traditional A* algorithm with low search efficiency. To reduce the convergence time of this algorithm, Fu et al. [10] added the variable step search node to the A* algorithm. On the other hand, Maurović et al. [11] introduced a D* path planning algorithm that can find the optimal path even when the location is unclear, and Wang et al. [12] further reduced the time of path planning by enhancing the D* algorithm's heuristic function. In another modification, Zhang et al. [13] added a collision factor to the D* algorithm, which contributed to finding the optimal route more effectively. In addition to these modifications, swarm intelligence algorithms like genetic algorithm (GA), particle swarm optimization (PSO), and ant colony optimization (ACO) are also utilized in path planning. Yuan et al. [14] combined GA with the artificial potential field method and successfully obtained the least-cost path in the two-dimensional and three-dimensional grid environment. Pehlivanoglu [15] applied the clustering method to develop a vibration GA that significantly reduced the operation time. Moreover, Shao et al. [16] proposed an optimized PSO algorithm that relies on an improved particle mutation replacement strategy, Han [17] introduced the unmanned aerial vehicle (UAV) cooperative task assignment model based on optimizing PSO, and Reynoso [18] improved the mathematical matching method to solve the problem of optimal trajectory planning and trajectory tracking control of robots. This involved the establishment of a complete nonlinear dynamic model and parameter discretization, which yielded better results in trajectory control accuracy through the adoption of the affine time-varying (ATV) control method. Similarly, Amir et al. [19] proposed an adaptive GA that optimizes robot path planning in a two-dimensional complex environment. This algorithm integrates traditional GAs and the Dijkstra algorithm to generate an initial population, thus preventing premature convergence and ensuring population diversity for enhanced iterative efficiency. Anh et al. [20] proposed a robot path coverage planning method relying on an improved

A* algorithm to realize maximum path coverage of the target area while minimizing revisiting the covered area. This method generates path points to cover narrow spaces, adopts the zigzag approach to address the limitations of the traditional algorithm, and improves the working efficiency of robot path coverage. Nouri Rahmat Abadi et al. [21] use the neural network method to plan the motion of the manipulator in the new class, and a neural network based on multilayer perceptron is used for training data. Dharmawan et al. [22] introduce the sequential expanded Lagrangian homotopy (SELH) approach, a method that offers the ability to sequentially determine the globally optimal robot's motion while satisfying the task constraints. Notably, SELH demonstrates superior planning execution speed compared to alternative methodologies.

Many scholars have done research in the field related to the sparrow algorithm. Zhu [23] improved the existing problems of the sparrow search algorithm (SSA) in the research of robot obstacle avoidance and modeled the planning environment by a three-layer neural network. Halton sequence is introduced to get the distribution of the first-generation population, and the more widely distributed and ergodic individual positions are obtained, which improves the speed and efficiency of later optimization, and the indexes of the improved algorithm are obviously optimized. Wang et al. [24] proposed the improved sparrow search algorithm (ISSA) that leverages multi-strategy fusion to improve the quality and diversity of the initial population through high-dimensional sine chaotic mapping while minimizing premature convergence. Additionally, the attenuation factor is introduced in the algorithm to act on the discoverer group to improve the local search ability while increasing the likelihood of jumping out of local optima. Bai et al. [25] put forward an improved search strategy SSA*, which has a larger search range and faster convergence speed and can generate an effective path within a specified number of iterations. ACO-SSA* algorithm is also proposed. The quality and stability of the initial paths play a crucial role in ensuring the accuracy of pathfinding. An ACO algorithm is used by the individuals in the population to generate effective initial paths, leading to high-quality solutions.

To complete truss handling and assembly tasks, space cellular robots need to navigate and climb to the target point with speed and accuracy being paramount. Consequently, there is a pressing need to explore advanced methodologies that can achieve this goal. Researching the path planning challenge for cellular robots on trusses, this paper introduces a novel structure and configuration concept for cellular robots that is specifically tailored to facilitate truss climbing. The paper also presents a truss path planning algorithm based on an enhanced sparrow algorithm, which greatly minimizes the time and power consumption of robots traversing large space trusses. The proposed technique is designed to work seamlessly with various levels of complex truss environments, providing a path toward optimal mode of robot operation and utilization. Furthermore, the successful implementation of this approach can significantly enhance the overall efficiency of space truss handling and assembly tasks, with potential applications in diverse settings such as construction, rescue, mining, and inspection. The extensive scope of this research presents an opportunity for further exploration and insights into the challenges faced by space cellular robots on trusses.

2. Structure and configuration design of cellular robot

2.1. Structural design of robot cell module

Aiming at the problem that the traditional manipulator has a single function and limited operating range, which cannot better match the future on-orbit assembly tasks, this section designs four kinds of cell modules based on the self-reconfigurable modular theory. The truss cell robot adopts the idea of bionics, and the most basic unit of the robot is called the cell. In the microgravity space environment, the cellular robot is connected to a robot that can realize complex actions by combining different cellular modules, which is called reconfiguration. Aiming at the on-orbit assembly task of space trusses with different requirements and complex environments, researchers can use different reconstruction strategies to flexibly complete the corresponding work objectives. A truss cell robot is a complex machine consisting of diverse cell modules that carry out specific and unique functions. These cell modules work together in harmony to achieve the overall goal of the robot, namely to navigate and operate effectively on trusses

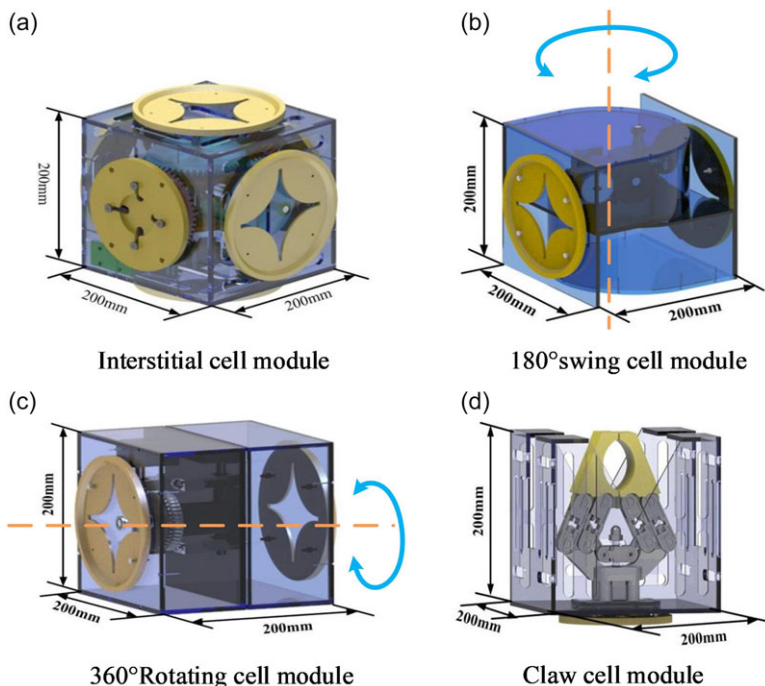


Figure 1. Structure diagram of cell unit module.

and other complex structures. In this paper, cell modules are divided into four categories: one is the interstitial cell module that plays the role of connection and support, and the other three are the functional cell modules that can perform the most basic actions, namely, the 180 swing cell module, the 360 rotation cell module, and the gripper cell module. The shapes of these four cell modules are all cubes with a side length of 200 mm.

As shown in Fig. 1, there are four kinds of cell modules. The whole connecting mechanism of the interstitial cell module can be divided into the external butting surface part and the internal mechanical transmission part, and the internal mechanical transmission part can be divided into translation movement and rotation movement according to the movement mode. Its working principle is that the internal motor controls the transmission part to make the clamping block assembly extend outward from the connection surface through linear translation motion, and at the same time, the clamping block assembly makes radial expansion motion along the bevel gear through rotary motion. The combination of the two motions makes the clamping block assembly firmly stuck in the clamping groove of the external docking surface and finally realizes the bidirectional connection function of interstitial cells. The connecting mechanism used in this design occupies a small space, and each interstitial cell module can have two sets of active connecting mechanisms; that is, the internal space layout of the module is reasonable so that the two sets of active connecting mechanisms can work at the same time and do not interfere with each other, further improving the docking efficiency of interstitial cells. Moreover, the current design utilizes a mechanical slot connection approach, which offers numerous benefits. One of the major advantages of this method is its high connection stability, ensuring a secure and reliable connection. Additionally, the slot connection design features strong connection strength and can withstand high levels of mechanical stress and heavy loads. In general, the mechanical slot connection design employed in this design provides exceptional durability and robustness, making it an ideal choice for a range of applications. The swing cell module consists of two rotating shells. The main components installed in the upper rotating shell are the motor, gear, worm gear, and rotating shaft. The power is output from the motor, transmitted to the worm gear through the gear and finally transmitted to the worm gear shaft. The lower shell is

connected with the upper rotating shaft, and the power of the upper rotating shell can be transmitted to the lower rotating shell through the rotating shaft of the worm wheel, so as to realize the swinging movement of 180° . The worm wheel and worm can make the cell module mechanism realize the self-locking function, which is more effective and stable in the rotation process. Each component is fixed on the rotator, and when the swing cell module rotates, the internal components can operate stably and effectively without being affected by the rotation. The internal structure of the 360° rotating cell module consists of an upper rotating shell and a lower rotating shell, and its working principle is similar to that of the swinging cell module. The worm gear transmission mode is adopted, and the power of the upper rotating body is output through the rotating shaft, and the lower rotating body is connected with the rotating shaft through the connecting disk to realize the rotation of the lower rotating body. The motor shaft in the rotating module is connected with the hole on the worm, and the worm is directly connected with the motor shaft through an interference fit to drive the worm to rotate, thus eliminating the gear transmission part in the swing cell module. Functionally, the advantage of the rotating cell module is that it can rotate 360° , which increases the freedom of rotation. The worm gear structure has stable transmission and good self-locking performance, which can stop the lower rotating shell from rotating at any position. The gripper cell module can be used in many aspects, such as gripping objects, building trusses, transporting trusses, and so on. In the robot design in this paper, the clamping function is to meet the needs of the truss climbing task of an on-orbit truss cell robot. The linear movement of the lead screw nut is caused by the rotational motion of the servo motor, which drives the lead screw. Subsequently, the connecting parts transmit this linear motion to the short arm of the gripper, enabling it to move vertically, so that the long arm can be driven to open and close, so that the whole gripper has two actions of clamping and opening, and the truss can be clamped and loosened. The servo motor can control the output torque more accurately, and it is more conducive to the digital control of the gripper cell module.

2.2. Configuration design of cellular robot

In this section, the serial configuration of cellular robots is designed. The requirements for a reconstructed cellular robot undertaking the on-orbit assembly task of space trusses include various abilities. Firstly, the robot must be capable of performing stable grasping, transporting, and assembling actions on the truss components. Additionally, the robot should be able to move smoothly from its initial position to the designated target location with high precision and reliability. The above-mentioned abilities are imperative to ensure that the robot can execute its tasks effectively in a challenging and complex space environment, where accuracy, stability, and safety are of utmost importance. Achieving these requirements will enhance the robot's performance and flexibility, opening up new possibilities for space truss handling and on-orbit assembly. This requires the reconstructed robot to have enough freedom to support it to complete various grasping and climbing actions in the process of performing tasks.

In the space environment, the cellular robot needs at least 3 degrees of freedom to climb and walk on the truss. Moreover, because the end effector needs one swing degree of freedom to adjust the posture when the cellular robot carries out the truss installation task, the cellular robot needs at least four degrees of freedom to ensure the ability to assemble the truss after reconstruction, including two rotational degrees of freedom at both ends and two swing degrees of freedom in the middle. The number of degrees of freedom of the robot should be suitable for its configuration. Too many degrees of freedom, such as a seven-degree-of-freedom redundant cell robot, will waste the resources of cell modules.

Therefore, based on the perspective of human bionics, this paper designs a six-degree-of-freedom cell robot, as shown in Fig. 2. The robot configuration is similar to the human body, and the first joint uses a rotating cell module to output rotating action, which is similar to the human core. The second and third joints use swing cell modules to output swing action, which is similar to human big and small arms. The fourth and fifth joints use swing cell modules, and the sixth joint uses rotating cell modules. These three joints are similar to human wrists. This configuration is not limited by the rotation angle of the joint module and can assemble truss rods in any position and posture in the space truss environment, which ensures that the robot can stably and flexibly accomplish the target task on the truss and has strong on-orbit assembly efficiency.

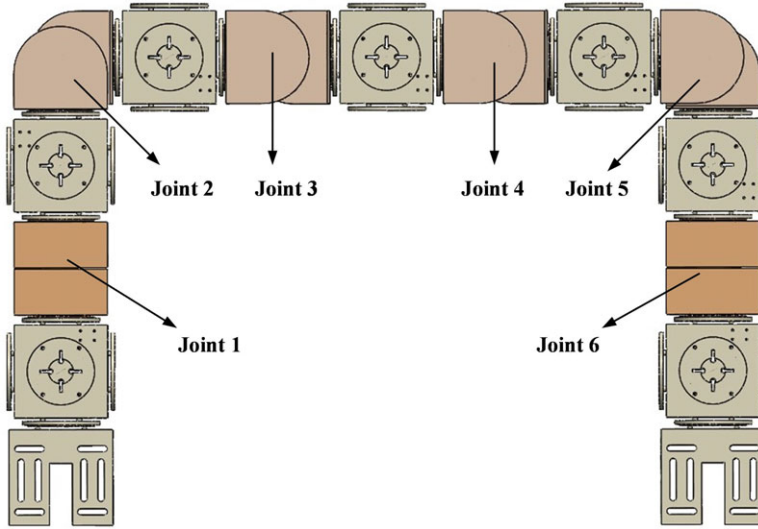


Figure 2. Six-DOF in-orbit truss cell robot.

3. Improved sparrow search algorithm

The SSA is a meta-heuristic algorithm that draws inspiration from sparrows' foraging and anti-predation behaviors [26]. Sparrow individuals are categorized as discoverers and entrants, working together to determine the foraging direction and search range of the population. Additionally, certain individuals within the population exhibit warning behavior, simulating the natural dangers that sparrows face and ensuring the safety of the foraging process. SSA is characterized by simple parameters and high solving efficiency. However, when tackling complex optimization problems, issues arise such as reduced population diversity and local optimal solutions. To address these challenges, this study proposes an enhanced version of the standard SSA algorithm.

3.1. Improved sparrow search algorithm design

Within the SSA algorithm, the discoverer plays a crucial role in determining the foraging direction for the entire population. Consequently, it is essential for the discoverer to possess a higher fitness value and a broader search range. Additionally, the initial position of the individual population significantly impacts the optimization performance of the algorithm. To enhance global exploration capabilities during the update of the discoverer's position and mitigate the negative impact of local optimal solutions in later iterations, it introduces chaotic mapping for the initialization of the sparrow population. The update process for the discoverer's location is described as follows:

$$X_{ij}^{t+1} = \begin{cases} X_{ij}^t \cdot \exp\left(-\frac{i}{\alpha \cdot \text{iter}_{\max}}\right) & \text{if } R_2 < ST \\ X_{ij}^t + Q \cdot L & \text{if } R_2 \geq ST \end{cases} \quad (1)$$

In the given context, the variables and notations are described as follows: t denotes the current iteration number, and j takes integer values from 1 to d representing the dimension information, where d signifies the dimensionality. iter_{\max} is a constant that signifies the maximum number of iterations. X_{ij} denotes the position information of the i th sparrow in the j th dimension. α represents a random number sampled from the uniform distribution in the interval $(0, 1]$. R_2 ($R_2 \in [0, 1]$) and ST ($ST \in [0.5, 1]$) represent the early warning value and safety value, respectively. Q is a random number following a normal distribution. L denotes a $1 \times d$ matrix, where all elements in the matrix are equal to 1.

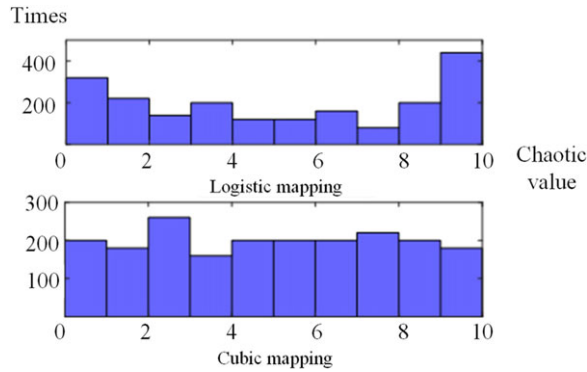


Figure 3. Distribution histogram of chaotic mapping.

To ensure a random and regular distribution of the sparrow population, it is necessary to apply a chaos operator that maps the initial sparrow population. This ensures that the sparrow individuals traverse all positions within the search range. The comparative analysis shown in Fig. 3 demonstrates the superior distribution characteristics of the cubic map compared to the classical logistic map. Notably, the cubic map significantly enhances the diversity of the sparrow algorithm, especially in the later stages. Based on this observation, we include the chaotic operator from the cubic map in the SSA, aiming to maximize global search efficiency.

Chaos operator has the advantages of randomness and regularity, and the cubic chaotic map has better uniform distribution performance between [0,1]. Chaos operator has the advantages of randomness and regularity, and the cubic chaotic map has better uniform distribution performance between [0,1]. The formula for the cubic mapping is defined as follows:

$$\begin{aligned}
 y(i+1) &= 4y(i)^3 - 3y(i) \\
 -1 < y(i) < 1, y(i) \neq 0, i = 0, 1, \dots, n
 \end{aligned}
 \tag{2}$$

According to formula (1), $M \times d$ sparrows form an initial population, and the concrete steps of initializing the sparrow population by using the cubic mapping chaos operator are as follows: To initialize the sparrow population, each sparrow undergoes iterative computations using the formula (2) to generate novel variable values within each dimension. Subsequently, these values are mapped to the sparrow population employing the function defined by formula (3). This procedure ensures the effective initialization of the sparrow population:

$$X_i = X_{lb} + (X_{ub} - X_{lb}) \times (y_i + 1) \times 0.5
 \tag{3}$$

In the context provided, X_{lb} and X_{ub} represent the lower and upper boundaries, respectively, of the search dimension for each sparrow individual. The variable X_i denotes the resulting value of the individual variable after undergoing chaotic mapping. y_i represents cubic sequences, a set of chaotic operators, and the cubic sequence can be mapped to sparrow individuals according to formula (3).

For the entrants, their foraging behavior mainly depends on the optimal position of the discoverer, and some entrants will always monitor the latter. Once the predation conditions are met, sparrows promptly displace themselves from their current positions to engage in competitive foraging for sustenance. The process of updating the location for an entrant is detailed as follows:

$$X_{ij}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{X_{\text{worst}} - X_{ij}^t}{i^2}\right) & \text{if } i > n/2 \\ X_p^{t+1} + |X_{ij}^t - X_p^{t+1}| \cdot A^+ \cdot L & \text{otherwise} \end{cases}
 \tag{4}$$

Within the given context, the notation and variables are described as follows: X_p^t represents the position of an individual with the best fitness value in the t -th iteration, while X_{worst} corresponds to the

position with the lowest global fitness value at the present moment. A denotes a $1 \times d$ matrix, and A^+ is the transpose of A multiplied by the inverse of the transpose, denoted as $A^T(A^T)^{-1}$. When $i > n/2$, it means that the participant has a low fitness value and has no right to fight for food. Consequently, they are obliged to explore new locations in search of foraging opportunities. Conversely, participants with an index i less than or equal to $n/2$ pursue foraging opportunities in proximity to the optimal position.

During the simulation experiment, a subset of sparrows, commonly known as “watchmen,” typically comprise approximately 10% to 20% of the total sparrow population. These watchmen fulfill the ongoing responsibility of diligently observing the sparrow population, promptly conveying early warning signals whenever perilous situations are detected. The primary objective behind these cautionary signals is to guide the entire population toward adopting anti-predation behaviors. The convergence rate of the standard vigilante position updating formula is reduced in the later period, and there is a local optimal constraint. Therefore, the number of vigilantes is dynamically adjusted by using an adaptive factor, and the search range is expanded by a larger number of vigilantes at the beginning of the algorithm. Toward the conclusion of the algorithm, reducing the number of watchmen can have advantageous effects, promoting algorithmic accuracy and facilitating enhanced local exploration capabilities. The dynamic decreasing formula of the number of watchmen is as follows:

$$\text{Num} = \text{int} \left(\left(1 - \frac{t}{t_{\max}} \right) \times \text{Num}_{\text{ini}} \right) + 1 \tag{5}$$

where Num is the number of current watchmen, the symbol “ t ” represents the ongoing iteration count, “ t_{\max} ” denotes the maximum allowable number of iterations, and “ Num_{ini} ” represents the initial count of watchmen.

The location update of the vigilante is described as follows:

$$X_{ij}^{t+1} = \begin{cases} X_{\text{best}}^t + \beta \cdot |X_{ij}^t - X_{\text{best}}^t| & \text{if } f_i > f_g \\ X_{ij}^t + K \cdot \left(\frac{|X_{ij}^t - X_{\text{worst}}^t|}{(f_i - f_w) + \epsilon} \right) & \text{if } f_i = f_g \end{cases} \tag{6}$$

Given the context, the notation and variables are defined as follows: X_{best} represents the current global optimal position. β is employed as the step control parameter. K is a random number sampled from the interval $[-1, 1]$, and f_i represents the fitness value of the current sparrow. Additionally, f_g and f_w are utilized to denote the best and worst fitness values, respectively. ϵ denotes a minimal constant value.

Upon reaching a stagnation point in the algorithm, a Gaussian random walk strategy is employed to generate fresh individuals. The generation formula for determining the position is as follows:

$$X_i^{t+1} = \text{Gaussian}(X_i^t, \sigma) \tag{7}$$

$$\sigma = \cos \left(\pi \times \frac{t}{2t_{\max}} \right) \times (X_i^t - X_r^*(t)) \tag{8}$$

where X_r^* is a random individual in the discoverer population, and the step size is adjusted by a cosine function. As the number of iterations increases, there is a gradual reduction in disturbance, leading to an enhanced ability to adjust the optimization direction of the algorithm. Consequently, this results in an improvement in the search efficiency of sparrows.

The schematic representation of the enhanced SSA is illustrated in Fig. 4.

3.2. Example experiment and result analysis

To validate the efficacy of the enhanced chaotic sparrow search algorithm (CSSA), comparative experiments were conducted, involving the ACO algorithm [27], PSO algorithm [28], and SSA algorithm [29]. To maintain fairness and consistency across the algorithms, a population size $N = 30$ and a maximum iteration count $T = 500$ were set. Table I provides a comprehensive overview of the specific algorithm parameters.

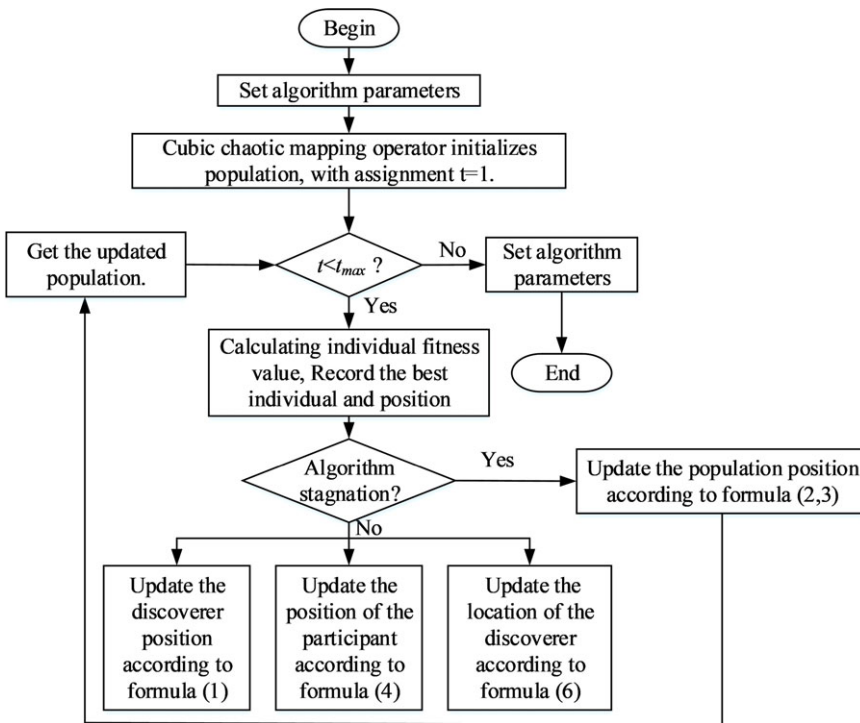


Figure 4. Flowchart of the algorithm.

Table II provides a comprehensive overview of the essential details of each test function used.

(1) Optimization and comparison of the unimodal test function.

F1 and F2 are unimodal test functions that have the ability to test the local development of the algorithm. To evaluate the performance of each algorithm, independent runs were conducted 50 times, with the average value and standard deviation serving as evaluation metrics. The optimization outcomes are presented in Table III, and the search space is shown in Fig. 5. It can be seen that, compared with the classical algorithm, CSSA is greatly improved, and the global optimal solution can be found for F1, and it performs better than other algorithms. The result obtained by CSSA for F2 is closer to the theoretical optimal solution, and the optimization result is more stable. Obviously, CSSA has stable performance and a better convergence effect, ranking first among these algorithms, and the improvement is of great significance. CSSA can enhance the quality and distribution uniformity of the initial population by introducing chaotic sequence mapping, which is helpful for the algorithm to conduct a more comprehensive search in the search space and achieve the purpose of improving the convergence accuracy and optimization performance of the algorithm. Chaos operators can usually better explore the search space and help to jump out of the local optimal solution by introducing randomness and nonlinear characteristics, which is very important for unimodal test functions. This balance of exploration and development ability is very important for finding the global optimal value in unimodal function.

(2) Optimize the comparison on multimodal test function.

In addition to that, F3 and F4 represent multimodal test functions, characterized by a significant number of local extreme points. These test functions serve as suitable benchmarks for evaluating the algorithm's optimization capability. Each algorithm was independently executed 50 times, and the optimization results are detailed in Table IV, while the convergence behavior is visually represented in Fig. 6.

Table I. Algorithm parameters.

Algorithm	Parameter setting
ACO	$\alpha = 1, \beta = 5, \tau_0 = 0.55, \rho = 0.3$
PSO	$c_1 = c_2 = 1.49445, \omega = 0.729$
SSA	SD = 0.2, ST = 0.8, PD = 0.2
CSSA	SD = 0.2, ST = 0.8, PD = 0.2

Table II. Test function information table.

Serial number	Testing function	Dimension d	Hunting zone	Optimal value
1	$F_1(x) = \sum_{i=1}^n x_i^2$	30	[-100,100]	0
2	$F_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	[-10,10]	0
3	$F_3(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	30	[-500,500]	$-418.9829 \times d$
4	$F_4(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	[-5.12,5.12]	0
5	$F_5(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_i(b^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	[-5,5]	0.0003
6	$F_6(x) = \sum_{i=1}^4 -c_i \exp\left(-\sum_{j=1}^3 a_{ij}(x_j - p_{ij})^2\right)$	3	[0,1]	-3.86

Table III. The optimization results of the single-peak test function.

Single-peak test function		ACO	PSO	SSA	CSSA
F1	Average value	1.34E-05	7.41E-44	1.89E-257	0.00E+00
	Standard deviation	1.19E-20	1.32E-43	0.00E+00	0.00E+00
F2	Average value	1.42E-05	1.24E-23	1.63E-93	9.54E-129
	Standard deviation	8.55E-21	7.66E-24	8.49E-93	5.02E-128

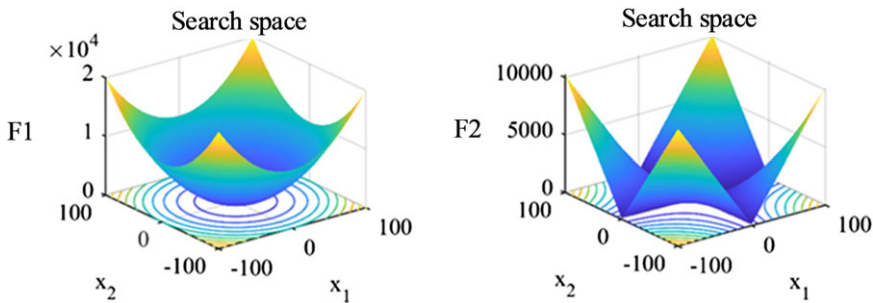


Figure 5. Search space of F1 and F2.

A noteworthy observation is that CSSA consistently converges toward the global optimal solution when dealing with F3, demonstrating superior stability compared to other algorithms. Notably, ACO exhibits the highest level of instability. Moreover, CSSA showcases improved stability and convergence speed in addressing the challenges posed by F4.

Table IV. The optimization results of the multi-peak test function.

Multi-peak test function		ACO	PSO	SSA	CSSA
F3	Average value	-5.65E+02	-9.87E+02	-1.02E+03	-1.26E+03
	Standard deviation	2.75E-12	3.09E+02	2.36E+01	8.86E-08
F4	Average value	1.41E+01	3.31E-02	1.01E-04	0.00E+00
	Standard deviation	0.00E+00	1.81E-02	0.00E+00	0.00E+00

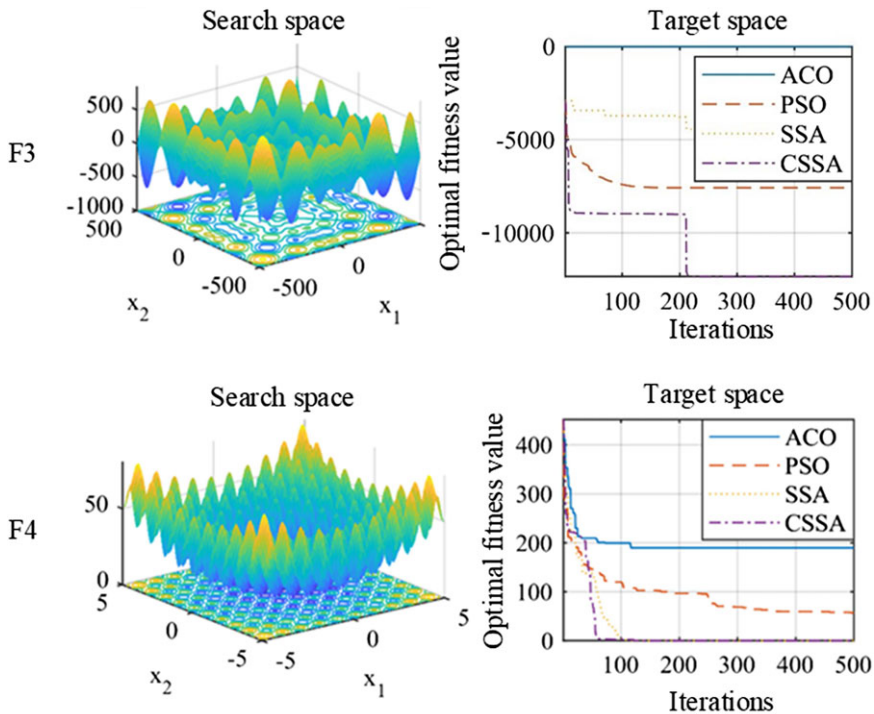


Figure 6. Optimization curve of F3 and F4.

(3) Optimize the comparison on the fixed-dimension test function.

F5 and F6 are fixed-dimension test functions, which are suitable for comprehensive verification of global search and local exploration capability. Each algorithm has been independently run 50 times, and the optimization outcomes of the aforementioned algorithm are presented in Table V, while Fig. 7 depicts its convergence behavior. A close analysis of the data reveals that the convergence speed of CSSA on F5 is absolutely superior, and it effectively converges to the global optimal value, which is obviously superior to ACO, PSO, and SSA. On F6, CSSA can quickly converge to the optimal value.

In a single-peak test function, traditional sparrow algorithms may excessively rely on collective intelligence, leading to suboptimal performance in the global search process. In contrast, the improved sparrow algorithm introduces individual exploration behavior, enabling better local search and faster convergence to the extreme value of the unimodal function. Compared to ACO and PSO algorithms, the improved sparrow algorithm can adjust the level of cooperation among sparrows more flexibly, allowing the algorithm to better adapt to problem characteristics. In multimodal functions, there are multiple local optima. The improved sparrow algorithm has the ability to escape local optima to some extent. This is because the improved sparrow algorithm incorporates the cubic mapping chaotic operator into the basic sparrow algorithm, facilitating better initialization of the sparrow population. This ensures that sparrow

Table V. The optimization results of the fixed-dimension test functions.

Fixed-dimension test function		ACO	PSO	SSA	CSSA
F5	Average value	1.36E-03	4.41E-04	3.34E-04	3.23E-04
	Standard deviation	4.39E-19	3.01E-04	2.36E-04	0.97E-06
F6	Average value	-4.64E-01	3.08E-12	-2.89E-01	-3.91E+00
	Standard deviation	2.17E-03	-3.86E+00	2.50E-03	2.64E-04

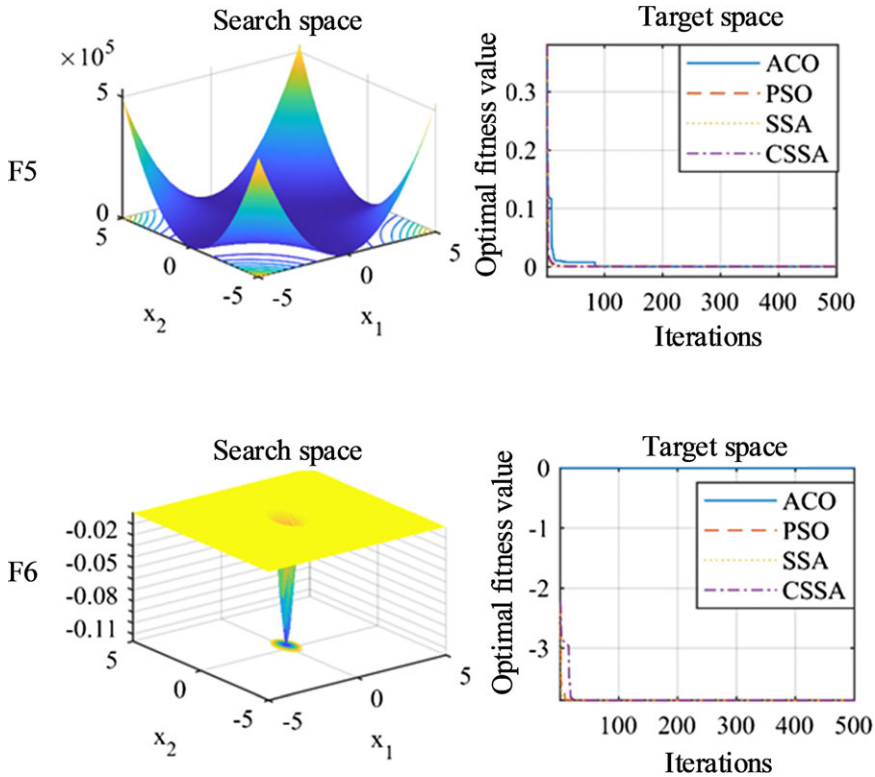


Figure 7. Optimization curve of F5 and F6.

individuals traverse all positions, thereby enhancing the robustness and optimization performance of the algorithm.

To sum up, the improved sparrow algorithm performs better in both unimodal and multimodal functions, primarily due to its diverse search strategies and a balance between collective intelligence and individual exploration. To sum up, CSSA has good optimization performance and obvious competitive advantage regardless of unimodal, multimodal, and fixed-dimension test functions.

4. Path planning based on improved sparrow algorithm

This section focuses on employing the improved CSSA to optimize path planning for in-orbit cellular robots navigating within a space truss environment. Specifically, the objective is to determine the shortest path from the robot's starting point to the target point within a known space coordinate environment.

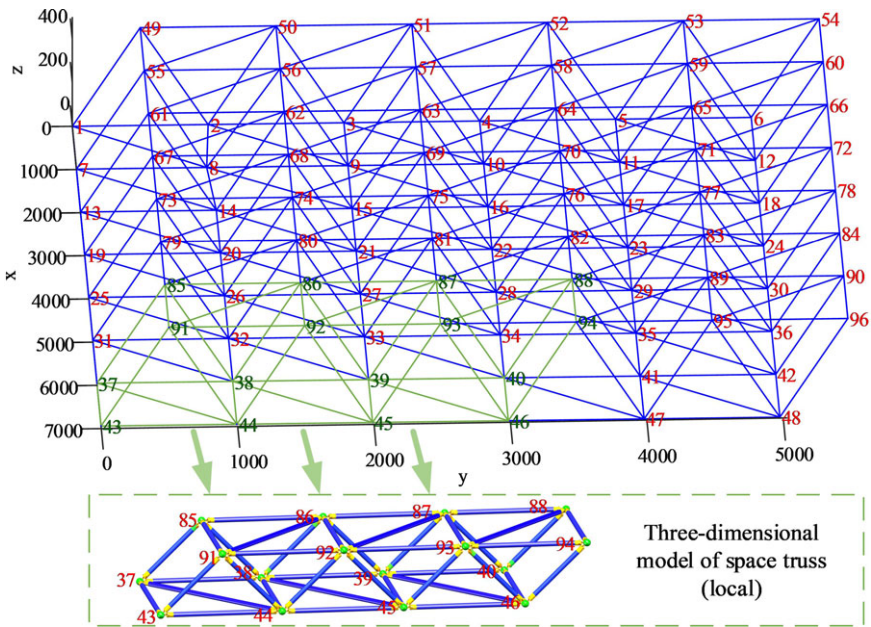


Figure 8. Truss mathematical model.

4.1. Construction of truss space model

The spatial truss is composed of truss rods and truss joints. We take the geometric centers of 96 truss joints as path nodes and input the coordinate data of these nodes into Matlab to generate the mathematical model of the spatial truss and establish the three-dimensional spatial coordinate system of the model, as shown in Fig. 8. The coordinates of some truss joints are shown in Table VI. The starting point of setting path optimization is truss joint No.1, and the target point is truss joint No.96. Then the target task is to find the shortest path from node 1 to node 96 in the three-dimensional space model.

4.2. Path planning algorithm flow

In this section, CSSA is used to solve the three-dimensional path planning problem and find out the shortest path of the climbing movement of the in-orbit cellular robot in a truss environment. The specific algorithm flow steps based on CSSA are as follows:

Step 1: Import the coordinate information of each node and establish the coordinate model of the space truss.

Step 2: Initialize parameters and set the maximum number of iterations of the algorithm, safety threshold, etc.

Step 3: Set the number of population discoverers, watchmen, and participants.

Step 4: Set the starting point and target point.

Step 5: Update the positions of discoverer, vigilante, and entrant to get the target path.

Step 6: Conduct a comparison between the current path and the previous path. In the event that the current path proves to be shorter, update the optimal path accordingly.

Step 7: Evaluate whether the termination condition has been met. If not, proceed to repeat steps 5 and 6. Conversely, if the termination condition is satisfied, the optimal path is to be presented as the final output.

Table VI. Coordinates of some truss joints.

Node	x	y	z	Node	x	y	z
1	0	0	0	79	4957.20	497.61	449.61
2	0	995.21	0	80	4957.20	1492.83	449.61
3	0	1990.42	0	81	4957.20	2488.05	449.61
4	0	2985.63	0	82	4957.20	3483.27	449.61
5	0	3980.84	0	83	4957.20	4478.49	449.61
6	0	4976.05	0	84	4957.20	5473.71	449.61
7	991.44	0	0	85	5948.64	497.61	449.61
8	991.44	995.21	0	86	5948.64	1492.83	449.61
9	991.44	1990.42	0	87	5948.64	2488.05	449.61
10	991.44	2985.63	0	88	5948.64	3483.27	449.61
11	991.44	3980.84	0	89	5948.64	4478.49	449.61
12	991.44	4976.05	0	90	5948.64	5473.71	449.61
13	1990.42	0	0	91	6940.08	497.61	449.61
14	1990.42	995.21	0	92	6940.08	1492.83	449.61
15	1990.42	1990.42	0	93	6940.08	2488.05	449.61
16	1990.42	2985.63	0	94	6940.08	3483.27	449.61
17	1990.42	3980.84	0	95	6940.08	4478.49	449.61
18	1990.42	4976.05	0	96	6940.08	5473.71	449.61

Table VII. Initial parameter settings.

Algorithm name	Initial parameter setting
CSSA	$n = 50, I = 50, ST = 0.8, PD = 0.2, SD = 10$
SSA	$n = 50, I = 50, ST = 0.8, PD = 0.2, SD = 10$
ACO	$\alpha = 1, \beta = 5, Nc = 50, I = 50$
PSO	$c_1 = c_2 = 1.49445, \omega = 0.729$

4.3. Path planning simulation and analysis

In this section, Matlab is used to simulate the path planning of an on-orbit cellular robot. To evaluate the performance of the enhanced SSA, we utilized the ACO algorithm, the PSO algorithm, and the conventional SSA for on-orbit cellular robot path planning.

As shown in Table VII, for CSSA and SSA, the population size was set to $n = 50$, with a maximum of 50 iterations. The safety threshold (ST) was determined as 0.8, and the discoverer accounted for 20% of the population size, while the number of sparrows aware of danger was set to 10. ACO utilized parameters “ $\alpha = 1$ ” and “ $\beta = 5$,” with a population size of $Nc = 50$ and a maximum iteration count of 50. As for PSO, the inertia weight was assigned as 0.729, while the social learning factor ($c1$) and individual learning factor ($c2$) were both set to 1.49445. The starting point of the path planning was designated as truss joint 1, with truss joint 96 serving as the target point. In order to enhance path planning efficiency, this simulation study disregarded the actual volume of the robot, resulting in simulation-obtained paths that align with the truss model.

The optimization outcomes of the ACO path planning are presented in Fig. 9. The simulation results demonstrate that the algorithm yielded the shortest path nodes as follows: “1 → 49 → 55 → 56 → 62 → 63 → 69 → 70 → 76 → 77 → 83 → 84 → 90 → 96,” with an optimal path length of 12,628.74 mm. Figure 10 exhibits the optimization results of PSO path planning, illustrating that the shortest path nodes acquired by this algorithm are “1 → 8 → 15 → 22 → 29 → 36 → 42 → 48 → 96,” with an optimal path length of 9719.42 mm. Moreover, the results of the SSA path planning optimization (Fig. 11) indicate that the shortest path nodes obtained by this algorithm align with the previous PSO path, namely “1 →

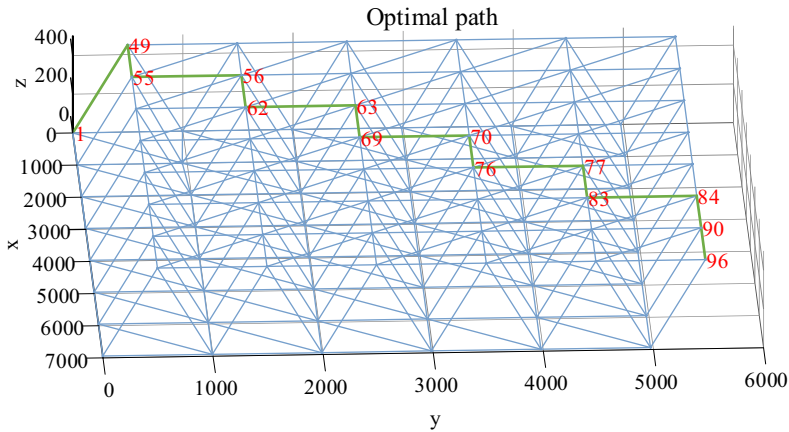


Figure 9. ACO path planning finder results.

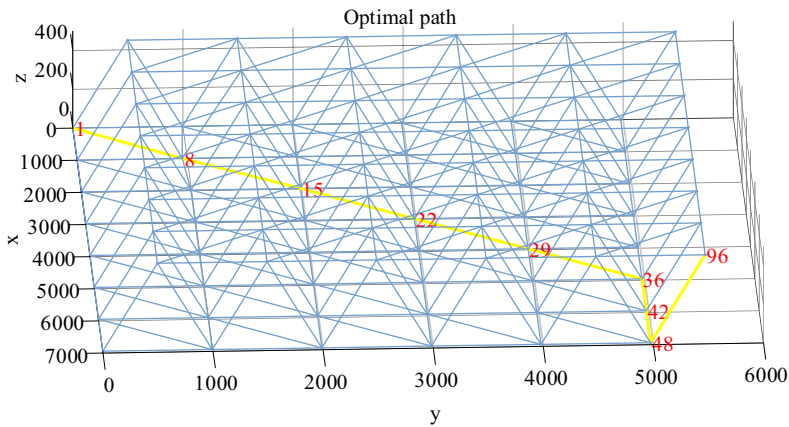


Figure 10. PSO path planning finder results.

8 → 15 → 22 → 29 → 36 → 42 → 48 → 96,” and the optimal path length remains at 9719.42 mm. Finally, Fig. 12 showcases the optimization outcomes of CSSA path planning, revealing that the shortest path nodes derived from this algorithm coincide with the path obtained by PSO and SSA, specifically “1 → 8 → 15 → 22 → 29 → 36 → 42 → 48 → 96.” The optimal path length remains unchanged at 9719.42 mm.

Through comparative analysis, it is evident that PSO, SSA, and CSSA all demonstrate the capability to identify an optimal path. However, ACO tends to find an optimal path that is characterized by a prolonged distance. This discrepancy arises due to the ACO algorithm’s susceptibility to getting trapped in local optimal solutions during the early stages, impeding its ability to escape. The convergence behavior of each algorithm’s robot path planning is depicted in Fig. 13. Notably, from Fig. 13, it is apparent that CSSA exhibits a faster convergence rate compared to PSO and SSA. Moreover, the curve exhibits smooth variations with minimal amplitude.

To ascertain the effectiveness and superiority of CSSA, the iteration times and population number are selected as independent variables, and the standard SSA is set as the experimental control group. The optimal path and reaction time are comprehensively evaluated, and the running results are shown in Table VIII.

Based on the findings in Table VIII, it is evident that the optimal path index exhibits significant improvement as the number of iterations and population increases. Specifically, when the number of

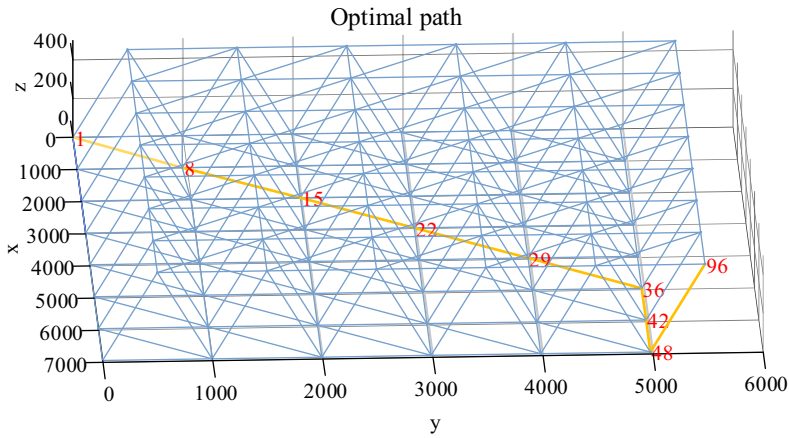


Figure 11. SSA path planning finder results.

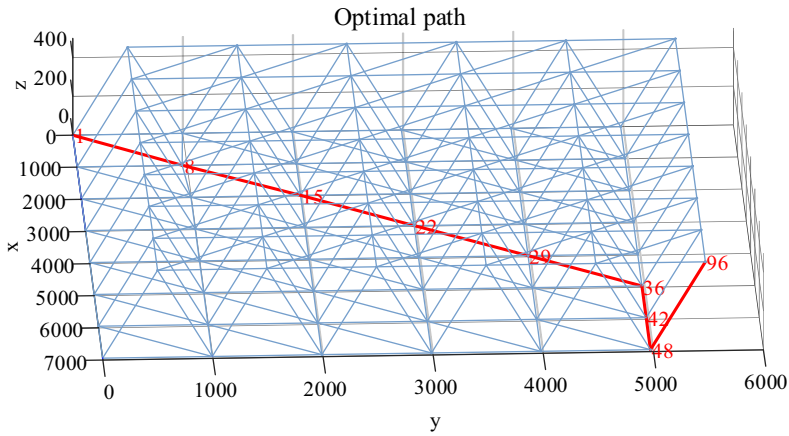


Figure 12. CSSA path planning finder results.

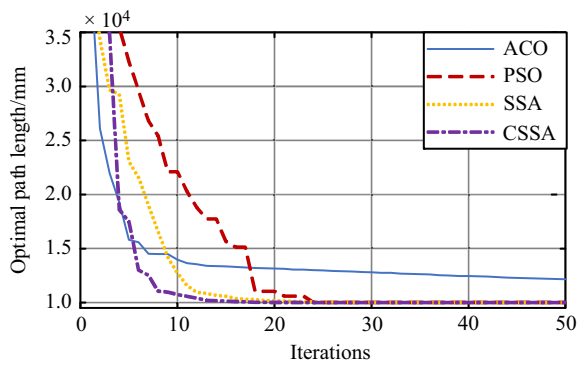


Figure 13. Iterative curves of algorithms.

Table VIII. Algorithm comparison.

Iterations	Population quantity	Algorithm	Optimal path index			Reaction time index		
			Optimal value	Average value	Standard deviation	Optimal value	Average value	Standard deviation
5	5	SSA	9910.23	9970.35	34.56	0.45	0.62	0.05
		CSSA	9880.35	9882.01	0.67	0.51	0.67	0.05
10	10	SSA	9842.29	9865.21	10.31	2.31	2.61	0.16
		CSSA	9753.66	9754.29	0.34	0.38	2.60	0.17
15	15	SSA	9739.21	9749.61	2.53	5.61	6.74	0.27
		CSSA	9726.53	9727.55	0.53	6.01	6.69	0.28
20	20	SSA	9721.93	9724.31	0.48	8.23	13.62	0.41
		CSSA	9719.42	9719.42	0.00	7.31	13.73	0.41
25	25	SSA	9720.12	9721.03	0.37	20.63	22.41	1.14
		CSSA	9719.42	9719.42	0.00	20.32	22.41	1.12
30	30	SSA	9719.42	9719.42	0.00	25.53	28.37	2.15
		CSSA	9719.42	9719.42	0.00	25.43	28.77	2.14

Table IX. Comparison of SSA and CSSA indicators.

Parameter index	SSA	CSSA
Shortest route/mm	9719.42	9719.42
Reaction time/s	28.37	13.73
The times of iteration to obtain the optimal path for the first time	30	20
Iteration times of population convergence	30	20
Effective iteration times/%	67.65	77.03

iterations and population reaches approximately 20, CSSA successfully generates the shortest path for the in-orbit cellular robot traversing the truss structure. At this point, the shortest path length measures 9719.42 mm, with an average path length of the same value. The corresponding reaction time is 7.31 s, surpassing SSA by 0.05% in this regard while achieving an 11.18% reduction in the optimal reflection time. Conversely, SSA only converges to an optimal path when the number of iterations and population is approximately 30.

Analyzing the standard deviation index of the optimal path reveals a notable issue with the standard SSA algorithm when confronted with path optimization challenges. Specifically, the algorithm is prone to local optimal solutions, resulting in substantial deviation of the optimal path data for the in-orbit cellular robot on the truss when the iteration number and population are set at lower values, which leads to a large deviation of the optimal path data of the in-orbit cellular robot on the truss when the iteration number and population number are low, and CSSA effectively avoids this problem. According to the reaction time index, it can be seen that the introduction of CSSA-improved strategies increases part of the running time, but it does not introduce additional time error to the overall performance of path optimization and still maintains high optimization stability.

In order to show the performance gap between CSSA and SSA in more detail, the optimization results are shown in Table IX.

From Table IX, it can be observed that the optimal path obtained by CSSA for in-orbit cellular robots remains consistent with the standard SSA. The overall reaction time has increased by 51.60%, and the effective iteration count has increased by approximately 13.87% compared to the standard SSA.

In conclusion, the enhanced SSA has achieved improvements in convergence speed and optimization stability to a certain extent, significantly reducing the number of iterations required to obtain the optimal path for in-orbit cellular robots compared to the standard SSA. The introduction of enhanced algorithmic

strategies has had no additional impact on time complexity and space complexity, maintaining their consistency. Consequently, it can be inferred that the enhanced strategies effectively enhance the search performance and optimization efficiency of the algorithm, making the ISSA well suited for addressing the path search problem in on-orbit truss cellular robots.

5. Conclusions

This study presents a novel structure and configuration for a cell robot specifically designed for climbing on trusses, based on the current path planning problem of cell robots in truss climbing. Furthermore, an ISSA is proposed. The three-dimensional truss is mathematically modeled using MATLAB, and the improved SSA is employed to solve the optimal path problem for the robot during truss climbing. The improved sparrow algorithm proposed in this paper has many advantages. The introduction of chaotic operators can increase the diversity of the search process, which helps improve the global search ability of the algorithm. Usually, the random performance of chaotic operators can make the algorithm expand the search scope, help the algorithm solve the problem of falling into local optimal solution, thus improving the performance on complex problems and enabling the improved sparrow algorithm to solve various optimization problems. However, the chaos operator also introduces an additional calculation process, including generating a chaotic sequence and integrating with the search process, which may increase the calculation cost of the algorithm, and some problems are not sensitive to chaos, so the improved sparrow algorithm proposed in this paper is not suitable for such problems. Generally speaking, using chaotic operators to improve the sparrow algorithm can increase the global search performance and diversity, but it also needs to deal with additional parameters and calculation costs, and it needs to carefully balance chaos and search efficiency to adapt to different types of optimization problems.

The main findings and conclusions of this research are as follows:

1. A modular structure and robot configuration for a reconfigurable space cell robot are designed in this study. This cell robot can achieve truss climbing motion in space and accomplish truss assembly and transportation tasks. The design of the cell robot structure is sophisticated, enabling rich functionality and offering significant scientific and practical value.
2. An ISSA is proposed in this study. Specifically, the algorithm's initial population quality and convergence speed are enhanced by introducing chaotic and adaptive factors. The improved SSA is compared with other algorithms using different test functions. The results indicate that CSSA exhibits excellent optimization performance and a notable competitive advantage, regardless of whether the test functions are unimodal, multimodal, or fixed-dimension. This confirms the superiority of CSSA.
3. A path planning method suitable for three-dimensional trusses is proposed in this study. The experimental results are compared with those of the ACO algorithm, PSO algorithm, and standard SSA. It is observed that CSSA outperforms ACO, PSO, and SSA in terms of convergence speed, with smoother curve changes and smaller variations. Compared to the standard SSA, CSSA shows an overall improvement of 51.60% in response time and an approximate increase of 13.87% in effective iteration count. The experimental results validate the rationality and superiority of the ISSA in solving the optimal path problem during truss climbing.

Author contributions. The research work of this paper is the result of the joint efforts of the whole team. Conceptualization: Ye Dai and Shikun Li; Methodology: Ye Dai and Shikun Li; Software: Shikun Li; Validation: Ye Dai and Xinda Chen; Investigation: Qihao Zhang, Xinlei Nie, and Xukun Rui; Writing – original draft preparation: Shikun Li; Writing – review and editing: Ye Dai and Xinda Chen. All authors have read and agreed to the published version of the manuscript.

Financial support. This research was supported by the National Natural Science Foundation of China [grant number 52075134], the Joint Guidance Project of Natural Science Foundation of Heilongjiang Province of China [grant number LH2019E062], and

the Opening Project of the Key Laboratory of Advanced Manufacturing and Intelligent Technology Ministry of Education, Harbin University of Science and Technology [grant number KFKT202105].

Competing interests. The authors declare that they have no financial or non-financial conflict of interest.

Ethical approval. The authors declare that this work is original and does not include experiments with animals.

References

- [1] Y. H. Liu, Motion Error And Reliability Analysis of Space Cell Robots *Master's thesis* (Harbin Institute of Technology, Harbin, China, 2020).
- [2] C. Y. Liu, J. G. Liu and Y. Liu, "An open modular self-reconfigurable robot," *Modular Machine Tool & Automatic Manufacturing Technique*. **2**(1001-2265), 26–29+33 (2018).
- [3] T. Fukuda, M. Buss and Y. Kawauchi, "Communication System of Cellular Robot: CEBOT," *15th Annual Conference of IEEE Industrial Electronics Society*, Philadelphia, PA, USA (1989), vol. **3**, pp. 634–639.
- [4] H. Tanaka, "Reconfigurable cellular satellites maintained by space robots," *Robot. Mechatron.* **18**(3), 356–364 (2006).
- [5] P. J. White and M. Yim, "Reliable external actuation for full reachability in robotic modular self-reconfiguration," *Int. J. Robot. Res.* **29**(5), 598–612 (2010).
- [6] B. Piranda and J. Bourgeois, "Designing a quasi-spherical module for a huge modular robot to create programmable matter," *Auton. Robot.* **42**(8), 1619–1633 (2018).
- [7] K. Y. Liu, Design and Reconfiguration Strategy of Series Reconfiguration Modular Robot *Master's thesis* (Shanghai Jiao Tong University, Shanghai, China, 2019).
- [8] X. L. Wen, Study on Planning of Cellular Space Robot for On-orbit Truss-Climbing *Master's thesis* (Harbin Institute of Technology, Harbin, China, 2020).
- [9] X. Wu, L. Xu, R. Zhen and X. Wu, "Bi-directional adaptive A* algorithm toward optimal path planning for large-scale UAV under multi-constraints," *IEEE Access* **8**(2169-3536), 85431–85440 (2020).
- [10] D. K. Fu and P. Q. Fan, "Improved A* algorithm for 3D UAV path planning," *Intell. Comput. Appl.* **10**(12), 155–159 (2020).
- [11] I. Maurovic, M. Seder, K. Lenac and I. Petrovic, "Path planning for active SLAM based on the D* algorithm with negative edge weights," *IEEE Trans. Syst. Man Cybern. Syst.* **48**(8), 1321–1331 (2017).
- [12] S. J. Wang, L. K. Hu and Y. F. Wang, "Path planning of indoor mobile robot based on improved D* algorithm," *Comput. Eng. Des.* **41**(04), 1118–1124 (2020).
- [13] F. Zhang, W. Bai, Y. H. Qiao, B. Y. Xing and P. C. Zhou, "UAV indoor path planning based on improved D* algorithm," *CAAI Trans. Intell. Syst.* **14**(04), 662–669 (2019).
- [14] M. S. Yuan, M. Chen and Q. X. Wu, "Cooperative path planning for multiple UAVs based on NSGA-III algorithm," *J. Jilin Univ. (Inf. Sci. Ed.)* **39**(03), 295–302 (2021).
- [15] Y. V. Pehlivanoglu, "A new vibrational genetic algorithm enhanced with a voronoi diagram for path planning of autonomous UAV," *Aerosp. Sci. Technol.* **16**(1), 47–55 (2012).
- [16] S. Shao, Y. Peng, C. He and Y. Du, "Efficient path planning for UAV formation via comprehensively improved particle swarm optimization," *ISA Trans.* **97**(0019-0578), 415–430 (2020).
- [17] Q. T. Han, "Research on cooperate task allocation of multiple UAVs based on PSO algorithm," *J. Ordnance Equip. Eng.* **40**(11), 74–78 (2019).
- [18] M. P. Reynoso, "On the time-optimal trajectory planning along predetermined geometric paths and optimal control synthesis for trajectory tracking of robot manipulators," *Dissertations Theses – Gradworks* **1997**(4), 907–908 (2013).
- [19] H. K. Amir and H. Maryam, "An adaptive genetic algorithm for robot motion planning in 2D complex environments," *Comput. Electr. Eng.* **43**(0045-7906), 317–329 (2015).
- [20] V. L. Anh, P. Veerajagadheswar, S. Vinu and E. M. Rajesh, "Modified A-star algorithm for efficient coverage path planning in tetris inspired self-reconfigurable robot with integrated laser sensor," *Sensors* **18**(8), 2585 (2018).
- [21] B. Nouri Rahmat Abadi and J. A. Carretero, "Modeling and real-time motion planning of a class of kinematically redundant parallel mechanisms with reconfigurable platform," *ASME J. Mech. Robot.* **15**(2), 021004 (2023).
- [22] A. G. Dharmawan, S. Foong and G. S. Soh, "Task-constrained optimal motion planning of redundant robots via sequential expanded lagrangian homotopy," *ASME J. Mech. Robot.* **10**(3), 031010 (2018).
- [23] J. T. Zhu, "Research on robot obstacle avoidance based on neural network and sparrow algorithm," *Comput. Meas. Control* **31**(04), 258–263,288 (2023).
- [24] H. Wang, N. Tong and Q. Fu, "Sparrow search algorithm with multiple strategies fusion," *Comput. Syst. Appl.* **32**(06), 159–165 (2023).
- [25] W. J. Bai, X. C. Jia and T. Lu, "Application of improved sparrow search algorithms in three-dimensional path planning," *Control Eng. China* **29**(10), 1800–1809 (2022).
- [26] M. F. Xia, L. M. Li and R. B. Ren, "Landslide prediction model based on KPCA-SSA-GRNN," *Foreign Electron. Meas. Technol.* **41**(09), 109–115 (2022).
- [27] Y. Liu, Z. Hou, Y. Tan, H. Liu and C. Song, "Research on multi-AGVspath planning and coordination mechanism," *IEEE Access* **8**(2169-3536), 213345–213356 (2020).

- [28] C. Pozna, R.-E. Precup, E. Horvath and E. M. Petriu, “Hybrid particle filter-particle swarm optimization algorithm and application to fuzzy controlled servo systems,” *IEEE Trans. Fuzzy Syst.* **30**(10), 4286–4297 (2022).
- [29] B. Gao, W. Shen, H. Guan, L. Zheng and W. Zhang, “Research on multistrategy improved evolutionary sparrow search algorithm and its application,” *IEEE Access* **10**(2169-3536), 62520–62534 (2022).