Path Planning for Lunar Surface Robots Based on Improved Ant Colony Algorithm

SONG Ting[1], *SUN Yuqi*[2], *YUAN Jianping*[1], *YANG Haiyue*[2], *WU Xiande*[2*]

1. School of Astronautics, Northwestern Polytechnical University, Xi'an 710072, P. R. China;
2. College of Aerospace and Civil Engineering, Harbin Engineering University, Harbin 150001, P. R. China

(Received 17 June 2022; revised 26 October 2022; accepted 26 November 2022)

Abstract: In the real-world situation, the lunar missions' scale and terrain are different according to various operational regions or worksheets, which requests a more flexible and efficient algorithm to generate task paths. A multi-scale ant colony planning method for the lunar robot is designed to meet the requirements of large scale and complex terrain in lunar space. In the algorithm, the actual lunar surface image is meshed into a gird map, the path planning algorithm is modeled on it, and then the actual path is projected to the original lunar surface and mission. The classical ant colony planning algorithm is rewritten utilizing a multi-scale method to address the diverse task problem. Moreover, the path smoothness is also considered to reduce the magnitude of the steering angle. Finally, several typical conditions to verify the efficiency and feasibility of the proposed algorithm are presented.

Key words: ant colony algorithm; grid map; multi scale; path smoothing

CLC number: TN925 **Document code**: A **Article ID**: 1005-1120(2022)06-0672-12

0 Introduction

Lunar base construction and resource development is an important aspect of expanding the human playground in the future. The lunar surface robot will be a critical assistance and vital tool for astronauts to develop and utilize lunar resources. Lunar robot task allocation and path planning will be large-scale planning problems, with the increasing number of robots. The uncertainty of path planning application scenarios and terrain conditions has further increased based on the improved robot performance and expanded planning scale. Diverse planning situation requires searching algorithms having the flexibility to adapt to various scales and high computing efficiency to fit complex terrain. Therefore, it is particularly critical to plan a short, safe and smooth path for robots to reach the target position on the lunar surface.

In-depth research on path planning has been carried out for decades, which can be divided into two types, one is based on the theoretical model, and the other is based on the random search algorithm. In terms of random search-based planning methods, Li[1] proposed an improved method for the A* algorithm. This method combines the AGV state with the fuzzy logic rules as the heuristic information and dynamically adjusts the heuristic function weight. Alireza et al.[2] enhanced the mutated cuckoo optimization algorithm (MCOA) which has the ability to deal with several unmapped objects and takes less time than the GA and A* algorithm. A new genetic algorithm with higher efficiency than A* algorithm on larger scale, based on the path network, was proposed by Li et al[3]. To enhance the calculating efficiency and reduce the number of nodes, a combination algorithm of probabilistic roadmap algorithm (PRM) and probability-based bi-directional rapidly-exploring random trees (P-Bi-RRT) was discussed by Xu et al[4].

The ant colony algorithm is widely used in searching for the optimal path due to its stability and

*Corresponding author, E-mail address: xiande_wu@163.com.

How to cite this article: SONG Ting, SUN Yuqi, YUAN Jianping, et al. Path planning for lunar surface robots based on improved ant colony algorithm[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2022, 39(6): 672-683.
http://dx.doi.org/10.16356/j.1005-1120.2022.06.004

reliability. By imitating ants' foraging behavior, the ant colony algorithm promotes optimization ability on intelligence, efficiency, accuracy, and application scope. Compared with the A* algorithm, GA algorithm, and other random search algorithms, the ant colony algorithm has positive feedback, and the algorithm can adopt distributed computing framework and heuristic probabilistic search method, making it easier and more efficient to find the global optimal solution. So, the ant colony algorithm is widely used to solve path planning problems. A fusion method of the segment and global path planning was proposed by Shi et al[5], which searched globally utilizing the potential ant colony algorithm and locally with the improved artificial potential field (APF) method. The article by Chen et al.[6] suggested that the ant colony optimization algorithm can solve the unmanned aerial vehicle (UAV) path planning problem with outstanding robustness and scalability. Based on the ant colony algorithm, Zhao et al.[7] improved the calculating efficiency through increasing the adaptive adjustment into heuristic function. Shao et al.[8] suggested that setting a dynamical limit of pheromone's attenuation coefficient can enhance the searching efficiency and reduce the probability of local optimal solution. Whereas, for the large-scale path planning problem of complex terrain, the current ant colony algorithm remains problems of searching blindly in the initial stage and having limits in searching scale.

A multi-scale improved ant colony algorithm is proposed in this paper, combined with the smoothness method for lunar robot intelligent path planning to adapt to the large-scale and complex lunar surface, hence the path searching efficiency, algorithm adaptation flexibility, and path safety are enhanced. The proposed algorithm can adapt various search-scale, reduce the probability of local optimum and enhance the convergence efficiency. Compared with the present method, there are three main contributions in this paper:

(1) Map the global space matrix into a new matrix with corresponding terrain complexity to enhance the searching scale of the ant colony algorithm;

(2) Propose a multi-scale method to escape from the local optimization quickly;

(3) Suggest that the ant colony algorithm can meet the requirements of large-scale and complex terrain on the lunar surface.

The paper is organized as follows. The basic idea of the multi-scale ant colony model is introduced in Section 1. Section 2 describes the algorithm design of the multi-scale ant colony model in detail. In Section 3, the proposed algorithm is used for lunar surface path planning, and the results are compared with the results obtained by the A* algorithm to illustrate the effectiveness of the multi-scale ant colony algorithm. Finally, some conclusions are drawn in Section 4.

1 Problem Description

(1) Symbol design

To explain the path planning problem, a simple mathematical model is designed, as shown in Fig.1.

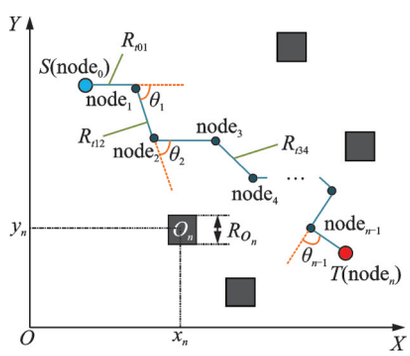


Fig.1 Mathematical model

In coordinate XOY, path nodes are denoted as a set $\text{Node}=\{\text{node}_i\}$, node_i is the ith node and $\text{node}_i=\{\text{node}_{x_i},\text{node}_{y_i}\}$, where node_{x_i} and node_{y_i} are node_i's position. The first node node_0 is also denoted as S, and the last one node_n is also denoted as T, where n denotes the last number of nodes.

Path segments between adjacent nodes are denoted as a set $R_t=\{R_{tij}\}$, where R_{tij} is the path segment between the ith node and the jth one. Here $j=i+1$, $0\leqslant i<n$.

Path segment angle is denoted as a set $\text{Ang}=\{\theta_i\}$, $\theta\in[0,\pi)$, where θ_i is the steeling angle be-

tween R_{tij} and prolonged line of $R_{t(i-1)i}$.

Obstacle is denoted as a set $\mathrm{Obs}=\{O_i\}$, where O_i is the ith obstacle. O_i is denoted as a third-tuple set $O_i=\{O_{x_i}, O_{y_i}, R_{O_i}\}$, where O_{x_i} and O_{y_i} are O_i's position, and R_{O_i} is the length of O_i's side.

The searching space is R_{search}.

The R_{search}'s mean altitude is A_R and lunar robot's motion has an altitude limit of A_{lim}.

(2) Path planning model

The lunar robot path planning problem can be defined as searching for an optimal or best path from S to T, as shown in Fig.2. Path planning model can be described by $S, T, R_t, \mathrm{Ang}, \mathrm{Obs}$, and so on.

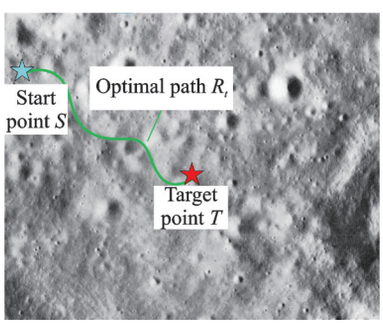


Fig.2 Lunar surface path planning

Obstacle avoidance model is

$$\mathrm{Node}\cap\mathrm{Obs}=\varnothing$$

$$\sqrt{(\mathrm{node}_{x_i}-O_{x_n})^2+(\mathrm{node}_{y_i}-O_{y_n})^2}\geqslant R_{o_n} \quad (1)$$

where O_{x_n} and O_{y_n} is the nearest obstacle to node_i.

The issue of path planning is to search for a shorter smoothing path with less calculating time t_{calcu} and smaller average steeling angle $\overline{\theta}=\dfrac{\sum_{i=1}^{n-1}\theta_i}{n}$.

The lunar space environment model is assumed as follows:

(i) If the altitude of position $A<A_R-A_{\mathrm{lim}}$ or $A>A_R+A_{\mathrm{lim}}$, the position is defined as an obstacle.

(ii) The lunar image is meshed into a grid map with different scales. The grid map is denoted as a matrix $\boldsymbol{L}$, and it's element L_{ij} is 0 or 1, where 0 means passable and 1 impassable, according to the results of assumption (i). In this way, R_{search} is turned into a matrix $\boldsymbol{L}$.

The mapping process is shown in Fig.3.

The classical ant colony algorithm has the disadvantage of low searching efficiency in the initial

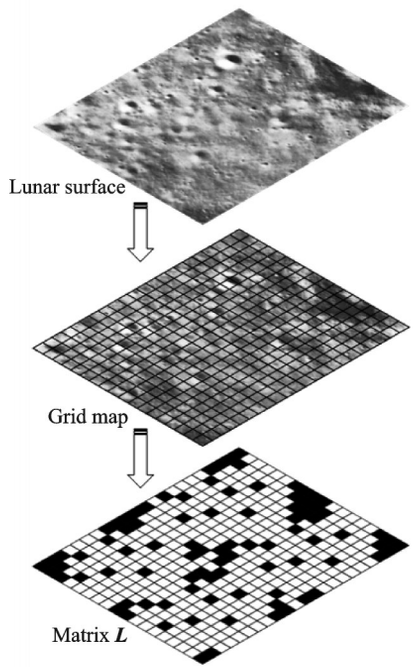


Fig.3 Lunar surface information processing

stage, especially in large spaces. Since the region scale of the lunar task could be large, $\boldsymbol{L}$ might contain a great number of elements. The ant pheromone parameter τ is difficult to spread into the whole space in several periods with huge calculating amounts. Hence, the calculation efficiency is significantly reduced in large scale problems. Changing the number of elements in the initial matrix and narrowing down the searching range are utilized in the algorithm to reduce the influence of the scale.

Before the path planning, map every bunch of elements $h\times h$ into one element. In other words, split the initial matrix $\boldsymbol{L}$ into a set of matrices L_{small} and integrate each matrix as an element in new matrix $\boldsymbol{L}_{\mathrm{New}}$. $\boldsymbol{L}_{\mathrm{small}\_m}$ is the mth matrix in L_{small}, which corresponds with $\boldsymbol{L}_{\mathrm{New}\_m}$, the mth element in $\boldsymbol{L}_{\mathrm{New}}$. The matrix $\boldsymbol{L}_{\mathrm{New}}$ consists of new elements with terrain complexity TC, which is indicated with different grey levels, as shown in Fig.4. The start point S and target point T are also mapped in $\boldsymbol{L}_{\mathrm{New}}$.

Plan the path with a new matrix $\boldsymbol{L}_{\mathrm{New}}$ and let passing elements map into the initial matrix to find the passing matrices L_{pass} in L_{small}. Plan path R_{t_small} with the passing matrices L_{pass}, combine them with each other in turn, and grow into the initial path R_t, as shown in Fig.5.

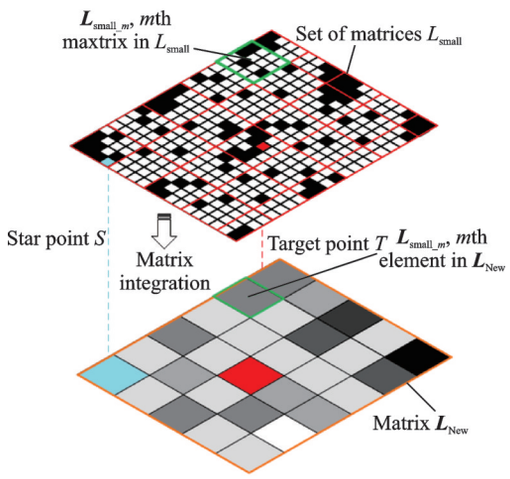


Fig.4 Matrix integration

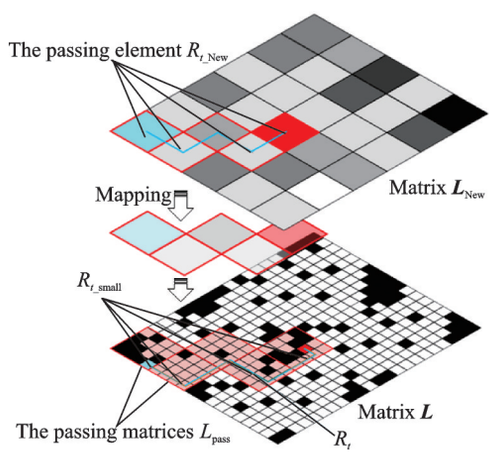


Fig.5 Illustration of mapping new matrix into initial matrix

The initial path R_t is remained to be optimized using ant colony algorithm. Firstly, collect path segment R_{t_seg} by means of stochastic method and get the corresponding matrix $\boldsymbol{M}_{New}$. Next, optimize the path R_{t_seg} in $\boldsymbol{M}_{New}$ without changing the start point and end point, then get the result of optimization R_{t_New}. Finally, replace R_{t_seg} with R_{t_New} and update the path R_t with local optimization mentioned above, as shown in Fig.6.

The actual lunar robot cannot drive on a polyline path; hence the path generated in the grid map is remained to be smoothed. The algorithm completes the smoothness control utilizing curve smoothing method, and the feasibility and efficiency of the final path are improved with the smoothing process, as shown in Fig.7.

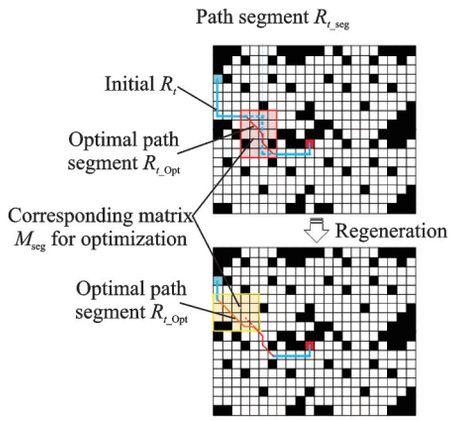


Fig.6 Path optimization

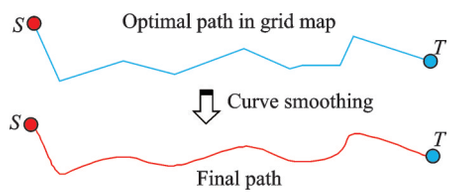


Fig.7 Path smoothing

2 Algorithm Design

2.1 Image process

First, convert the colored moon image into a grayscale picture, then obtain the picture's pixels to gain a gray value matrix for each pixel MG. Calculate the mean pixels' value mean_{rgb} which reflects as mean altitude. Regard mean_{rgb} as a pixel reference value and set value proportion range p_{rgb}. Finally, generate the global matrix $\boldsymbol{L}=\{l_{rc}\}$, where l_{rc} is an element in the rth row and the cth column. According to the assumption of the lunar surface environment, the global matrix can be defined as

$$\text{high}_{rgb}=\text{floor}((\max_{rgb}-\text{mean}_{rgb})\times p_{rgb}+\text{mean}_{rgb}) \tag{2}$$

$$\text{low}_{rgb}=\text{floor}(\text{mean}_{rgb}-(\text{mean}_{rgb}-\min_{rgb})\times p_{rgb}) \tag{3}$$

$$l_{rc}=\begin{cases}0 & \text{low}_{rgb}\leqslant \text{MG}_{rc}\leqslant \text{high}_{rgb}\\ 1 & \text{else}\end{cases} \tag{4}$$

where 0 means accessible and 1 inaccessible; $\max_{rgb}$ and $\min_{rgb}$ indicate the maximum and minimum value in MG, respectively; high_{rgb} and low_{rgb} respectively mean the upper and lower boundary in MG, corresponding to p_{rgb}. floor(A) rounds the elements of A to the nearest integer less than or equal to A.

2.2 Improved ant colony algorithm

Express the heuristic function $\eta_{ij}(t)$ is in inverse proportion to the distance from next node j to the target[9].

$$\eta_{ij}(t)=\frac{1}{d(j,T)} \tag{5}$$

Set the lower limit of τ_{ij} to simulate the error probability in ant colony and increase the searching diversity.

$$\tau_{ij}=\begin{cases}\tau_{ij} & \tau_{ij}>\tau_{\min}\\ \tau_{\min} & \tau_{ij}\leqslant\tau_{\min}\end{cases} \tag{6}$$

Mortify the recording of pheromone[10-11]. Only record pheromone of ants which have reached T, and the collection of these ants in the tth generation is $\text{arrive}_{t,T}$. k denotes the kth ant in $\text{arrive}_{t,T}$. Here L_k denotes the length of the path taken by the kth ant in $\text{arrive}_{t,T}$.

$$\Delta^k\tau_{ij}(t)=\begin{cases}Q/L_k & \{i,j\}\in \overset{k}{\text{arrive}}_{t,T}\\ 0 & \text{else}\end{cases} \tag{7}$$

$$\Delta\tau_{ij}(t)=\sum_{k=1}^{a}\Delta^k\tau_{ij}(t) \qquad a=\text{size}(\text{arrive}) \tag{8}$$

In traditional ant colony algorithm, the value of pheromone is assumed to degrade in each iteration. However, the main idea of the improved ant colony algorithm is to select path segment randomly for optimization, so as to obtain the global optimization result. Due to this randomness, this assumption is not applicable to the improved ant colony algorithm.

2.3 Path mapping method

The path mapping method is adopted to generate the initial solution, which enhances the calculation efficiency. The general idea is splitting global grid into several small bunch of matrices L_{small}, and mapping each matrix of L_{small} into each element in matrix $\boldsymbol{L}_{\text{New}}$ with terrain complexity TC. After generating the optimal path R_{t_New} in $\boldsymbol{L}_{\text{New}}$, map the corresponding matrices L_{pass} and combine the optimal path R_{t_small} with each other in turn, then generate the initial path R_t.

The regional splitting process is as follows:

(1) Confirm the number of elements $M_1\times M_2$ in $\boldsymbol{L}$.

(2) Set the number of elements $h\times h$ in $\boldsymbol{L}_{\text{small}\_m}$.

(3) Split $\boldsymbol{L}$ into $\text{ceil}\left(\frac{M_1}{h}\right)\times\text{ceil}\left(\frac{M_2}{h}\right)$ matrices in L_{small}. $\text{ceil}(A)$ rounds the elements of A to the nearest integer greater than or equal to A.

(4) The matrix $\boldsymbol{L}_{\text{New}}$ consist of new elements, and each element takes value in $[0,1]$. The mapping relationship between the elements in $\boldsymbol{L}_{\text{New}}$ and the matrix in L_{small} is reflected by TC, start point S and target point T.

$$\text{TC}=\frac{\text{num}_{\text{obs}}}{\text{num}_{\text{total}}} \tag{9}$$

$$L_{\text{New}\_m}=\begin{cases}1 & \text{TC}\geqslant\text{TC}_{\max}\\ \text{TC} & \text{else}\end{cases}$$

$$m\in\left[1,\text{ceil}\left(\frac{M_1}{h}\right)\times\text{ceil}\left(\frac{M_2}{h}\right)\right] \tag{10}$$

where num_{obs} is the obstacle number in L_{small}'s matrix and $\text{num}_{\text{total}}$ the total number of grids in L_{small}'s matrix.

Cancel the ant motion degree of freedom (DOF) of the upper left, lower left, upper right and lower right during the process of generating R_{t_New} in $\boldsymbol{L}_{\text{New}}$, to ensure the existence of accessible path between neighbor matrices in L_{small}. Then, map the R_{t_New} nodes into matrices L_{pass}.

The mapping method of generating R_t from S to T in L is indicated as below.

(1) Iterate every pair of neighbor matrices in L_{pass} and select accessible path between neighbor matrices. $\boldsymbol{L}_{\text{pass}\_n}$ and $\boldsymbol{L}_{\text{pass}\_m}$ mean the upper and later matrix in L_{pass}. J_n and J_m are the connected edge collections in $\boldsymbol{L}_{\text{pass}\_n}$ and $\boldsymbol{L}_{\text{pass}\_m}$.

$$J_n+J_m=J_{nm} \tag{11}$$

(2) Select J_{nm}'s elements, whose values are 0, into J_{nm0} as accessible collection. Select the posth elements J_{pos} in J_{nm0} by means of stochastic method. The target position in L_{pass_n} is T_{pass_n}, $T_{\text{pass}\_n}=J_n\cap J_{\text{pos}}$, and the start position in L_{pass_m} is S_{pass_m}, $S_{\text{pass}\_m}=J_m\cap J_{\text{pos}}$.

(3) Calculate the start position and the target in every L_{pass}'s matrix, according to the processes (1) and (2).

(4) Generate path in L_{pass}'s matrix, and splice the path one by one. Then, obtain R_t in global matrix $\boldsymbol{L}$, as shown in Fig.8.

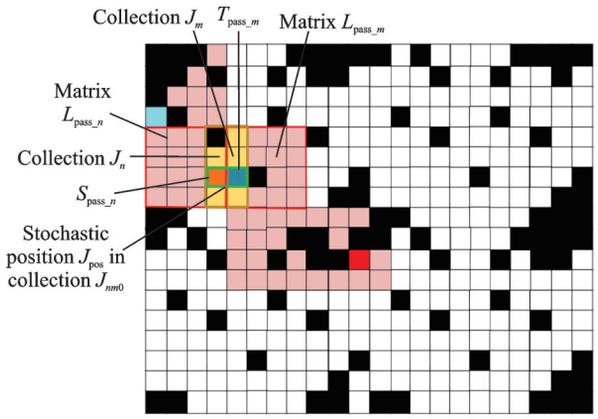


Fig.8 Searching start point and target in local matrix

2.4 Space splitting optimal method

The space splitting optimal method can optimize R_t with stochastic method, reduce the probability of local optimal solution, and enhance the calculation efficiency. The overall process of space splitting optimal method is indicated as below.

(1) Set parameters of path segment optimization: O_{num} is the number of optimization, $P_{max}(0<P_{max}<1)$ the upper boundary of segment-to-R_t ratio; $P_{min}(0<P_{min}<P_{max})$ lower boundary of segment-to-R_t ratio; N_{Opt0} the initial number of generations; M_{Opt0} the initial number of ants in the algorithm.

(2) Search the start position S_{Opt} in regional optimization, and determine the number of intercept nodes num. Then gain the certain path segment R_{t_Seg}.

$$\text{num}=\max[\text{floor}(((O_{num}-O_i)\times P_{max}\times L_R)/O_{num}),L_R\times P_{min}] \tag{12}$$

$$s=\text{rand}[1,L_R-\text{num}] \quad s\notin \text{Tabu} \tag{13}$$

$$S_{Opt}=Rt(s) \tag{14}$$

$$T_{Opt}=Rt(s+\text{num}) \tag{15}$$

$$N_{Opt}=\text{floor}(\text{num}\times N_{Opt0}/(P_{max}\times L_R)) \tag{16}$$

$$M_{Opt}=\text{floor}(\text{num}\times M_{Opt0}/(P_{max}\times L_R)) \tag{17}$$

where $\text{rand}[a,b]$ can generate random number from a to b; $\max[a,b]$ means selecting maximum one from a and b; L_R is the number of R_t's nodes; T_{Opt} the end nodes in R_{t_Seg}; N_{Opt} the current regional optimization ant colony iterate times; M_{Opt} the number of ants.

Tabu list is set in this step within the process of regional path optimization. And the used S_{Opt} position is recorded in the tabu list, $\text{Tabu}=\{S_{small}(1),S_{small}(2),\cdots,S_{small}(N_{small})\}$. If the new start position is already in Tabu, then reselect.

(3) Get the corresponding matrix $\boldsymbol{M}_{Seg}$ to R_{t_Seg}. Iterate every node in R_{t_Seg} and find the maximum and minimum value in x-axis and y-axis. Obtain $\boldsymbol{M}_{Seg}$ according to the maximum and minimum value mentioned above. Path planning is implemented in $\boldsymbol{M}_{Seg}$ to obtain the solution R_{t_Opt}.

(4) Replace R_{t_Opt} with R_{t_Seg} and update the path R_t with local optimization mentioned above.

(5) Repeat Steps (2—4) until reach O_{num}.

2.5 Path smoothing process

After path mapping and space splitting method, obtain $\text{Node}=\{\text{node}_i\}$ which needs to be smoothed. This work adopts the smoothing method proposed by Refs.[12-14] to reduce the magnitude of the steering angle and improve path safety.

The definition of weight function $w(x)$ is described as

$$w(x)=e^{-\frac{-|x-x_0|}{\delta}} \tag{18}$$

where δ is the smoothing factor.

While calculating the node_k's new position, select $2m$ points around node_k as the smoothing collection and establish a local coordinate system centered on node_k.

Control the smoothness of the curve through parameters, set the node_i's parameter as t_i. Then the parameterization of $\text{node}_i=\{\text{node}_{x_i},\text{node}_{y_i}\}$ generates two components in the X and Y directions: (t_i,node_{x_i}) and (t_i,node_{y_i}).

$$t_i=\begin{cases}0 & i=0\\ t_{i-1}+|\text{node}_i\text{node}_{i-1}| & 1\leqslant i\leqslant n\end{cases} \tag{19}$$

Set the parameter of farthest node as t_j and the weight as α. Then obtain the initial smoothness factor δ_0.

$$\delta_0=-\frac{|t_j-t_i|}{\ln\alpha} \tag{20}$$

Calculate node_i's weight matrix $\boldsymbol{W}$ according to the function (18) and δ_0.

$$\boldsymbol{W}=\begin{bmatrix}w_1 & 0 & \cdots & 0\\ 0 & w_2 & 0 & \vdots\\ \vdots & 0 & \ddots & 0\\ 0 & \cdots & 0 & w_m\end{bmatrix} \tag{21}$$

The node_i's new component in the X direction

is $x_i'=a_x$ and the new component in the Y direction is $y_i'=a_y$. So the new position $\text{node}_i=(x_i', y_i')$.

$$B=\begin{bmatrix} 1 & (t_1-t_i) & (t_1-t_i)^2 \\ 1 & (t_2-t_i) & (t_2-t_i)^2 \\ \vdots & \vdots & \vdots \\ 1 & (t_m-t_i) & (t_m-t_i)^2 \end{bmatrix} \tag{22}$$

$$\boldsymbol{X}=[\text{node}_{x_1},\text{node}_{x_2},\cdots,\text{node}_{x_m}]^{\mathrm{T}} \tag{23}$$

$$\boldsymbol{Y}=[\text{node}_{y_1},\text{node}_{y_2},\cdots,\text{node}_{y_m}]^{\mathrm{T}} \tag{24}$$

$$\boldsymbol{A}_x=(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{B})^{-1}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{X}=[a_x,b_x,c_x]^{\mathrm{T}} \tag{25}$$

$$\boldsymbol{A}_y=(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{B})^{-1}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{Y}=[a_y,b_y,c_y]^{\mathrm{T}} \tag{26}$$

Smooth residual d_i is used to denote the distance between the original position $\text{node}_i=\{\text{node}_{x_i},\text{node}_{y_i}\}$ and the new position $\text{node}_i=(x_i', y_i')$. To ensure nodes are still in the original grid, select the node node_k with the maximum smooth residual d_k and suppress d_k within 0.5 by decreasing the value of δ.

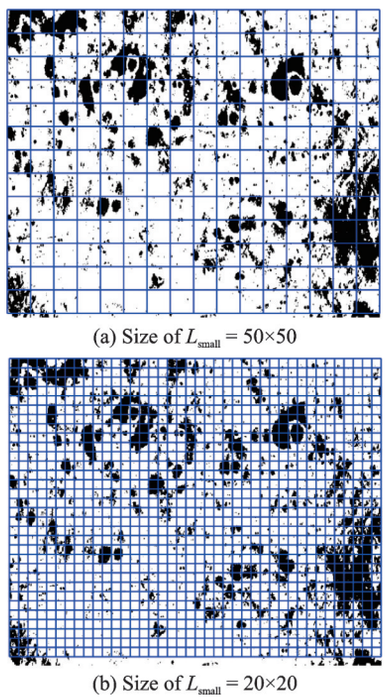
3 Simulation and Result Analysis

The lunar surface image is shown in Fig.9. Transfer Fig.9 into grid map $\boldsymbol{L}$, which contains 658×800 elements. Then, split $\boldsymbol{L}$ into L_{small}'s matrices. (a) denotes matrices' scale in L_{small} is 50×50 and (b) matrices' scale in L_{small} is 20×20, in Figs.10—15.

Fig.9 Lunar surface

Set $\text{TC}_{\max}=0.4$ and the grid level is proportional to L_{New_m}'s value. The path planning parameters in $\boldsymbol{L}_{\text{New}}$ are: $N=30$, $M=60$, $\tau_{\min}=0.1$. The start position in L is at the 150th row and the 1st column and the target at the 350th row and the 375th column, as shown in Fig.11.

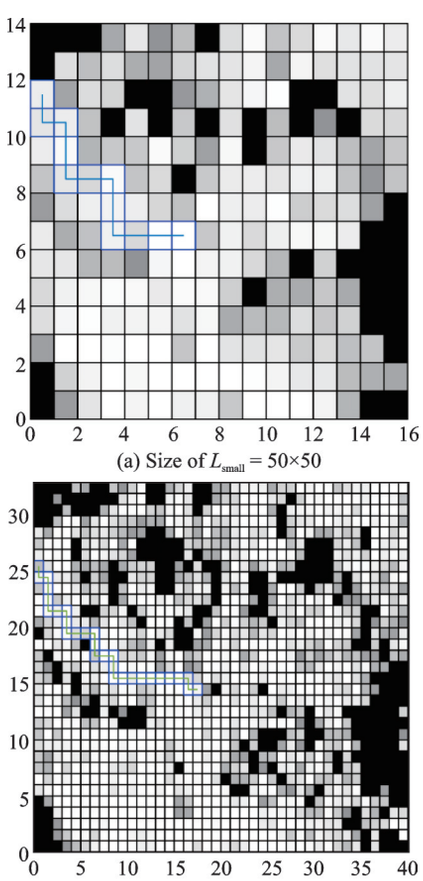
The path planning parameters in L_{pass} are: $N=10$, $M=50$, and $\tau_{\min}=0.1$. The initial path R_t is shown in Fig.12.



(a) Size of $L_{\text{small}}=50\times50$

(b) Size of $L_{\text{small}}=20\times20$

Fig.10 Split grid map

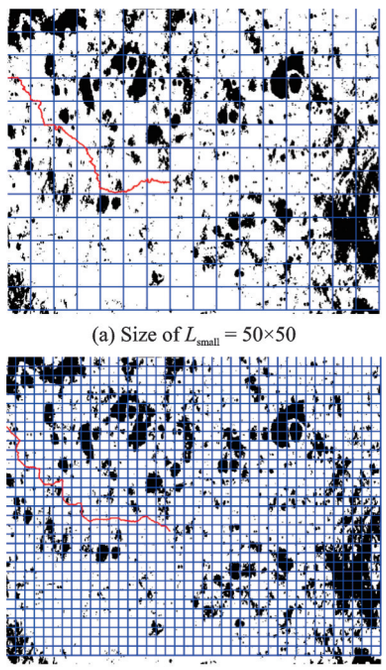


(a) Size of $L_{\text{small}}=50\times50$

(b) Size of $L_{\text{small}}=20\times20$

Fig.11 New matrix

The regional optimization parametes are set as: $O_{\text{num}}=200$, $P_{\max}=0.085$, $P_{\min}=0.05$, $N_{\text{small0}}=60$, $M_{\text{small0}}=50$, and the other constant parameters are the same as those in L_{pass}. The relation-

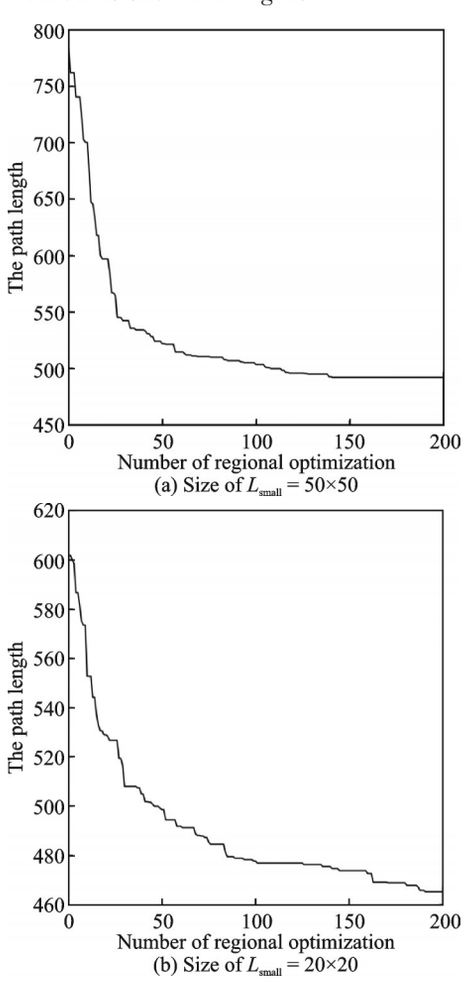


(a) Size of L_{small} = 50×50

(b) Size of L_{small} = 20×20

Fig.12 Initial path

ship between the path length and the regional optimization number is shown in Fig.13.

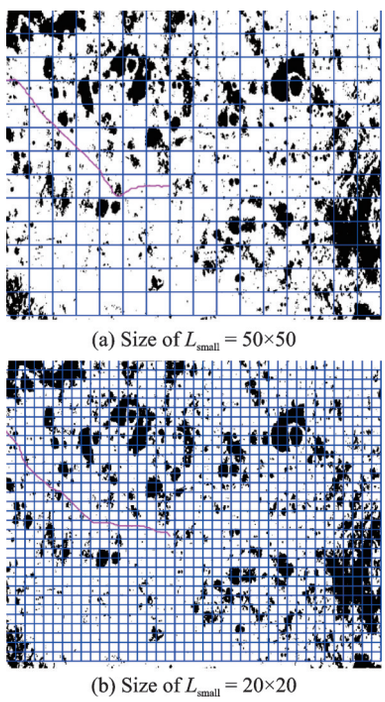


(a) Size of L_{small} = 50×50

(b) Size of L_{small} = 20×20

Fig.13 Relationship between the path length and optimization times

The optimal path before smoothing method is shown in Fig.14, where mean steering angle $\bar{\theta}$ = 14.408 867 and the number of large angles is 122. This work regards steering angle bigger than 20° as large angle, as shown in Fig.14.

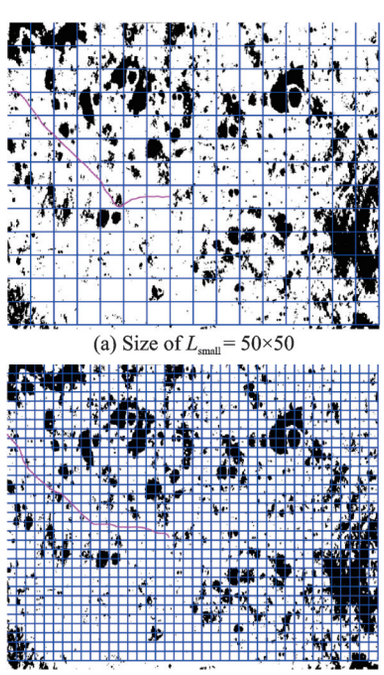


(a) Size of L_{small} = 50×50

(b) Size of L_{small} = 20×20

Fig.14 Optimal path before smoothing method

After path smoothing process, $\bar{\theta}$ = 1.981 138 and the number of large angles is 0, as shown in Fig.15.



(a) Size of L_{small} = 50×50

(b) Size of L_{small} = 20×20

Fig.15 Final optimal path

The size of L_{small} in the path mapping method is an adjustable parameter that should be determined according to the task. In this scenario, the other parameters are the same as those mentioned above.

The size of L_{small} is given from 15×15 to 50×50. The average results after removing the highest and lowest values in 8 groups of simulation are shown in Figs.16—21.

From the experimental data, the computing time of generating the path in L_{New} is in inverse proportion to the size of L_{small}, and the computing time of generating the initial path in L_{pass} is in proportion to the size of L_{small}. The total computing time has minimum value when the size of L_{small} is 19×19.

The relationship between the size of L_{small} and the computing time by space splitting optimal method is as follows.

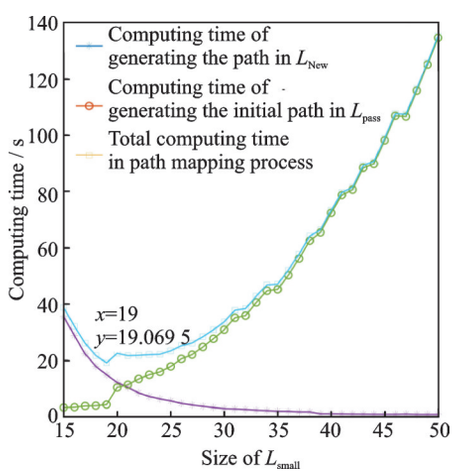


Fig.16 Relationship between the size of L_{small} and computing time by path mapping method

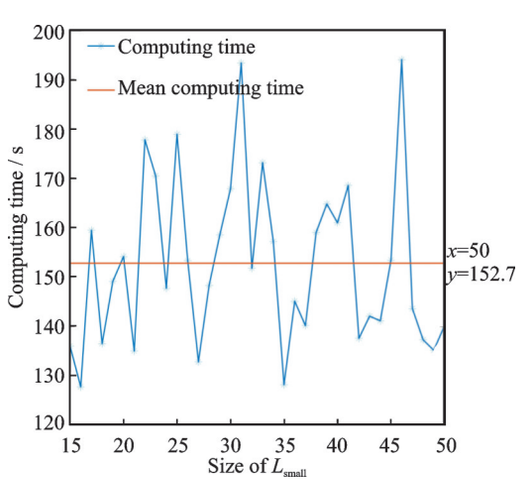


Fig.17 Relationship between the size of L_{small} and the computing time by space splitting optimal method

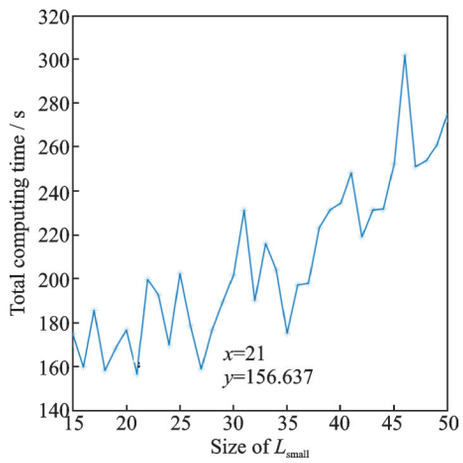


Fig.18 Relationship between the size of L_{small} and the total computing time

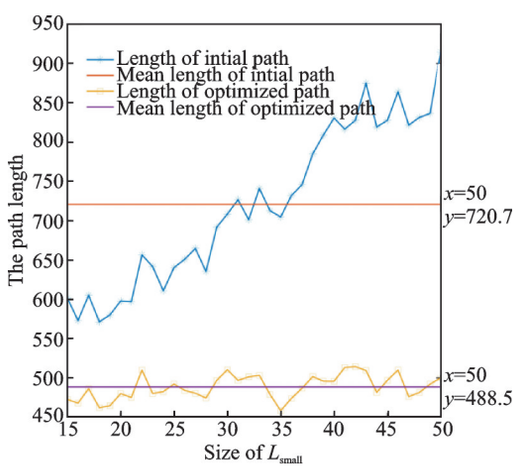


Fig.19 Relationship between the size of L_{small} and the path length

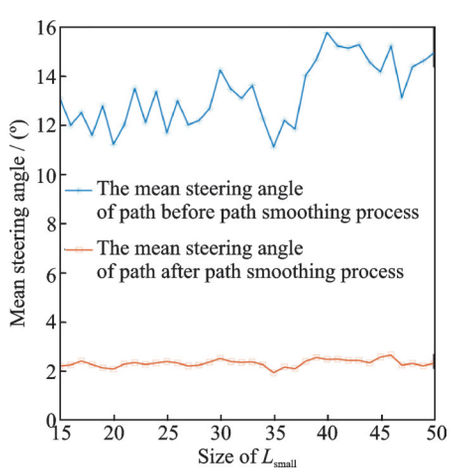


Fig.20 Relationship between the size of L_{small} and the mean steering angle

The computing time by the space splitting optimal method fluctuates constantly, but there is no pronounced changing trend.

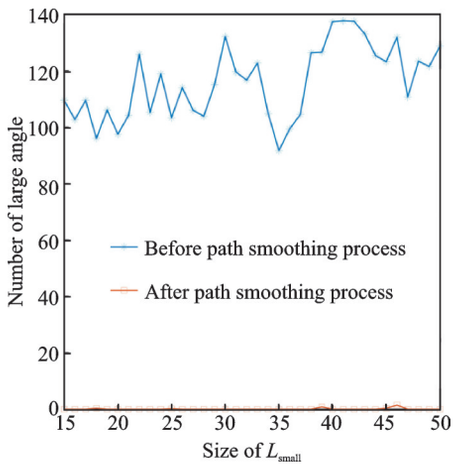


Fig.21 Relationship between the size of L_{small} and the number of large angle

By comprehensively considering the size of L_{small} and time-consuming, when the size of L_{small} is in the range of 15—27, the algorithm has less time-consuming.

The relationship between the size of L_{small} and the path length is shown in Fig.19.

The length of the initial path generated by the path mapping method has a growth trend as the size of L_{small} increases. However, the length of the optimized path always fluctuates within a small range, proving that the space splitting method can optimize the path effectively.

The relationship curve between the size of L_{small}, the mean steering angle and the number of large angles are as below.

From the simulation results, it can be concluded that the smoothness method proposed by Refs. [12-14] can be used for path planning smoothing effectively.

There remain many methods of path planning. In this scenario, the comparison curves and analysis among the multi-scale method and A* algorithm has been done to verify the efficiency of the presented method, as shown in Figs.22, 23, respectively.

The size of L is 300×300, the start position in L is at the 300th row and the 1st column, and the target is at the 2nd row and the 300th column.

In the multi-scale method, $TC_{max} = 0.4$ and the path planning parameters in L_{New} are: $N = 40$, $M = 50$, $\tau_{min} = 0.1$. The path planning parameters in L_{pass} are: $N = 30$, $M = 40$, $\tau_{min} = 0.1$, and the size of L_{small} is 15×15. Regional optimization parameter are: $O_{num} = 150$, $P_{max} = 0.085$, $P_{min} = 0.05$, $N_{small0} = 40$, and $M_{small0} = 50$.

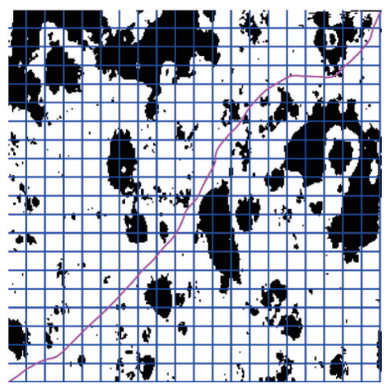


Fig.22 Final optimal path by multi-scale method

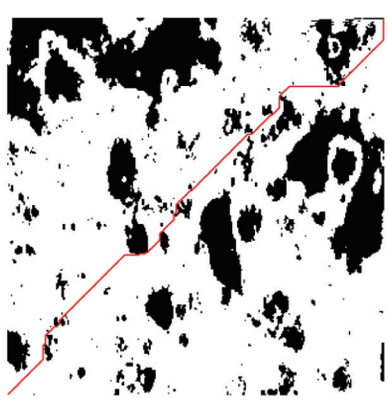


Fig.23 Final optimal path by A* algorithm

The optimal path generated by the multi-scale method is 463.92 long, which costs 79.643 s; the optimal path generated by the A* algorithm is 458.17 long, which consumes 295.987 s. Hence, the multi-scale method has significantly improved the searching speed.

The proposed method can search optimal path in global space with different start position S and target position T. Fig. 24 presents S at the 150th row and the 1st column, and T at the 630th row and the 660th column.

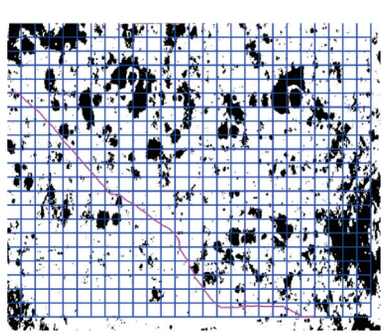


Fig.24 Different paths in global space

4 Conclusions

This paper proposes a multi-scale method based on ant colony algorithm. The certain idea are path mapping and space splitting method. Several simulations are presented to demonstrate the effectiveness of the theoretical result. The multi-scale improved ant colony algorithm can improve the search speed and find the global optimal solution effectively. Multi-scale ant colony algorithm is beneficial to plan the path of the lunar robots efficiently and stably. It can be extended to large-scale path planning problems with complex terrain requirements in various fields in the future.

References

[1] LI Yuan. Multi-AGV path planning algorithm and task allocation scheduling strategy[D]. Hangzhou: Zhejiang University, 2022.(in Chinese)

[2] ALIREZA M, VINCENT D, TONY W. Experimental study of path planning problem using EMCOA for a holonomic mobile robot[J]. Systems Engineering and Electronics, 2021, 32(6): 1450-1462.

[3] LI S, DING M, CAI C, et al. Efficient path planning method based on genetic algorithm combining path network[C]//Proceedings of 2010 Fourth International Conference on Genetic and Evolutionary Computing. [S.l.]: [s.n.], 2010.

[4] XU J, TIAN Z, HE W, et al. A fast path planning algorithm fusing PRM and P-Bi-RRT[C]//Proceedings of 2020 11th International Conference on Prognostics and System Health Management (PHM-2020 Jinan). Jinan, China: [s.n.], 2020.

[5] SHI K, WU P, LIU M. Research on path planning method of forging handling robot based on combined strategy[C]//Proceedings of 2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA). [S.l.]: IEEE, 2021.

[6] CHEN J, YE F, JIANG T. Path planning under obstacle-avoidance constraints based on ant colony optimization algorithm[C]//Proceedings of 2017 IEEE 17th International Conference on Communication Technology (ICCT). [S.l.]: IEEE, 2017: 1434-1438.

[7] ZHAO Tianliang, ZHANG Xiaojun, ZHANG Minglu, et al. Research on robot path planning method based on improved fusion ant colony algorithm[J]. Manufacturing Automation, 2022, 44(5): 185-190. (in Chinese)

[8] SHAO X Q, LV Z C, ZHAO X, et al. Research on robot path planning based on improved adaptive ant colony algorithm[C]//Proceedings of 2019 Chinese Control and Decision Conference (CCDC). [S.l.]:[s.n.], 2019.

[9] SUN Pengna. ZHANG Zhongming. Path planning and smoothing for unmanned surface vehicle based on improved ant colony optimization[J]. Electronic Science and Technology, 2022: 1-8. (in Chinese)

[10] ALI M, DEO R, XIANG Y, et al. Coupled online sequential extreme learning machine model with ant colony optimization algorithm for wheat yield prediction[J]. Scientific Reports, 2022, 12(1): 1-23.

[11] VASANT P, MARMOLEJO J A, LITVINCHEV I S, et al. Nature-inspired meta-heuristics approaches for charging plug-in hybrid electric vehicle[J]. Wireless Networks, 2020, 26(7):4753-4766.

[12] QIU Wei, GUO Bingxuan, XIAO Xiongwu. A contour smoothing method with controllable smoothness[J]. Geomatics & Spatial Information Technology, 2021, 44(7): 90-94. (in Chinese)

[13] PIEGL L, TILLER W. Non-uniform rational B spline [M]. Beijing: Tsinghua University Press, 2010.

[14] DAVID H D, THOMAS K P. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature[J]. Canadian Cartographer, 2006, 10(2): 112-122.

Acknowledgements This work was supported by the National Natural Science Foundations of China (No. 11772185), and Fundamental Research Funds for the Central Universities (No.3072022JC0202).

Authors Ms. **SONG Ting** received the M.S. degree from Harbin Institute of Technology, Harbin, China, in 2009. She is now pursuing the Ph.D. degree in Northwestern Polytechnical University. Her current research interests include aerospace control and path planning.

Prof. **WU Xiande** received the Ph.D. degree in aeronautical and astronautical science and technology from Harbin Institute of Technology, Harbin, China, in 2010. His research fields include space dynamics and planning.

Author contributions Ms. **SONG Ting** designed the study, complied the models and wrote the manuscript. Ms. **SUN Yuqi** contributed to data analysis, result interpretation and manuscript revision. Mr. **YUAN Jianping** contributed to the discussion and revision of the study. Mr. **YANG Haiyue** contributed to the experiment analysis. Prof. **WU Xiande** contributed to the design and discussion of the study. All

authors commented on the manuscript draft and approved the submission.

Competing interests The authors declare no competing interests.

(Production Editor: WANG Jing)

基于改进蚁群算法的月面机器人路径规划

宋 婷[1], 孙瑜奇[2], 袁建平[1], 杨海岳[2], 吴限德[2]

(1.西北工业大学航天学院,西安 710072,中国; 2.哈尔滨工程大学航天与建筑工程学院,哈尔滨 150001,中国)

摘要:在实际情况下,月面任务的规模和地形随着所选区域或工况而有所不同,因此需要一种更灵活和高效的算法做路径规划任务。为满足月球空间大规模、复杂地形的需求,本文针对月面机器人设计了一种混合尺度蚁群规划方法。该算法将实际的月面图像网格化处理成栅格图,并对其进行路径规划算法建模,由此模拟真实月面任务的路径规划。利用混合尺度的方法优化经典的蚁群规划算法用以适应不同的任务。此外,还采用路径平滑方法减小了路径中转向角度。最后通过几种典型场景验证了优化算法的效率和可行性。

关键词:蚁群算法;栅格图;混合尺度;路径平滑