

# 基于改进跳点搜索和蚁群算法的 机器人多目标点巡检规划\*

芮宏斌<sup>1†</sup> 李未<sup>1</sup> 解晓琳<sup>2</sup> 彭家璇<sup>1</sup> 郭旋<sup>1</sup>

(1. 西安理工大学 机械与精密仪器工程学院, 西安 710048)

(2. 河南科技大学 农业装备工程学院, 洛阳 471000)

**摘要** 针对移动机器人的多目标点巡检规划问题,本文提出了一种融合改进跳点搜索算法(JPS)与蚁群算法(ACO)的路径规划算法。首先,在JPS算法的评估函数中引入角度引导因子,使路径具有更强的导向性;然后,综合考虑路径距离、平滑度、安全性对评估函数的影响,以获得综合性能更优的路径;其次,提出了双向的逆向跳点剔除规则,筛选了多余节点,从而进一步降低路径长度并提高路径平滑度;最后,将多目标优化得到的路径综合性能替代传统旅行商问题(TSP)中的距离因子,并使用自适应蚁群算法来实现多巡检点的路径规划问题。仿真结果表明,改进JPS算法与传统JPS算法相比,具有更好的综合性能;同时应用于多巡检点规划时,具有更强的有效性和实用性。

**关键词** 巡检机器人, 路径规划, 跳点搜索算法, 多目标优化, 蚁群系统算法

中图分类号:TP242

文献标志码:A

## Robot Multi-Target Inspection Planning Based on Improved Jump Point Search and Ant Colony Algorithm\*

Rui Hongbin<sup>1†</sup> Li Lei<sup>1</sup> Xie Xiaolin<sup>2</sup> Peng Jiaxuan<sup>1</sup> Guo Xuan<sup>1</sup>

(1. School of Mechanical and Precision Instrument Engineering, Xi'an University of Technology, Xi'an 710048, China)

(2. College of Agricultural Equipment Engineering, Henan University of Science and Technology, Luoyang 471000, China)

**Abstract** For the problem of multi-target point inspection planning for mobile robots, this paper proposes a path planning algorithm that integrates the Improved Jump Point Search algorithm (JPS) with the Ant Colony Optimization algorithm (ACO). Firstly, an angle-guided factor is introduced into the evaluation function of the JPS algorithm to provide stronger directional guidance for the path. Then, considering the influences of path distance, smoothness, and safety on the evaluation function, a path with better comprehensive performance is obtained. Next, a bidirectional reverse jump point pruning rule is proposed to eliminate redundant nodes, further reducing path length and improving path smoothness. Finally, the path obtained from multi-objective optimization is used to replace the distance factor in the traditional Traveling Salesman Problem (TSP), and an adaptive ant colony algorithm is used to solve the multi-target point path planning problem. Simulation results show that the improved JPS algorithm has better comprehensive performance compared to the traditional JPS algorithm. When applied to multi-target point planning, it demonstrates stronger effectiveness and practicality.

2023-10-19 收到第 1 稿,2023-11-06 收到修改稿。

\* 国家自然科学基金青年科学基金项目(51905154);2022 年陕西省教育厅重点科研计划项目(22JY051);2023 年陕西省科技厅重点研发计划项目(2023-YBGY-357)和陕西省技术创新引导专项(2018ZKC-160), National Natural Science Youth Science Foundation Project (51905154); 2022 Shaanxi Provincial Department of Education Key Scientific Research Program Project (22JY051); 2023 Shaanxi Provincial Department of Science and Technology Key Research and Development Program Project (2023-YBGY-357) and Shaanxi Provincial Technological Innovation Guiding Special Project (2018ZKC-160).

† 通信作者 E-mail:Hongbin\_rui@126.com

**Key words** inspection robot, path planning, jump point search algorithm, multi-objective optimization, ant colony system algorithm

## 引言

随着人工智能的飞速发展,路径规划技术被大量应用于移动巡检机器人领域<sup>[1,2]</sup>.目前常用的路径规划法有 Dijkstra 算法<sup>[3]</sup>、人工势场法<sup>[4]</sup>、A\* 算法<sup>[5]</sup>、遗传算法<sup>[6]</sup>、粒子群算法<sup>[7]</sup>、蚁群算法<sup>[8]</sup>.

在两点之间的路径规划中,相较其他方法而言,A\* 算法是一种效率高,搜索时间短,且能搜索到最优路径的算法<sup>[9]</sup>,但使用 A\* 算法时,浪费了大量多余时间用于搜索节点,针对 A\* 算法的改进,许多国内外学者使用跳点搜索的方法以减少其搜索节点.文献[10]结合了 JPS 和人工势场法,给跳点搜索提供了方向,能有效提高路径搜索效率;文献[11]在 JPS 算法中融合了向量叉积策略以减少相同代价的节点;文献[12]提出了一种基于路径碰撞的改进 JPS 算法,将障碍物进行膨化处理,提高了路径的安全性;文献[13]使用了双向同步跳点搜索算法,提高了路径规划的实时性与高效性.上述算法都对 A\* 算法进行了有效改进,但是没有考虑路径的平滑性、安全性等综合性能.

在多目标点巡检规划中,常用蚁群算法来求解这类 TSP 问题<sup>[14]</sup>.文献[15]采用贪心策略构造次优路径来增加该路径上的初始信息素浓度,提高了搜索初期跳出局部最优的能力.文献[16]采用基于收敛系数的动态信息素更新策略,有效加快收敛速度;上述算法虽然有效改进了蚁群算法,但在处理路径规划问题时忽略了地图中的障碍物,这可能导致生成的路径缺乏可行性.文献[17]在路径规划中引入了障碍物因子,但其使用遗传算法来进行多目标点巡检规划倾向于在搜索空间中进行局部搜索,这可能导致算法陷入局部最优解.

因此,本文提出了一种基于 JPS 的多目标评价函数,同时考虑了路径的长度、平滑度和安全性,以得到综合性能较好的路径;并且引入了双向的逆向跳点剔除策略,以进一步优化路径的长度和平滑度,得到了改进的 ERJPS 算法;最后将 ERJPS 算法与采用了自适应信息素更新策略的蚁群算法结合解决了移动机器人多任务点的巡检规划问题.

## 1 相关算法

### 1.1 JPS 算法

JPS 算法是一种启发式搜索算法,具体评估函数如式(1):

$$f(n) = h(n) + g(n) \quad (1)$$

式中, $f(n)$ 是指经由节点  $n$  到达目标点的代价; $g(n)$ 是指从当前节点到节点  $n$  的实际代价; $h(n)$ 是从节点  $n$  到目标节点的估计代价.

#### 1.1.1 节点的邻域内无障碍物

在父节点  $p$  移动至节点  $n$  时,始终存在一条更短的路径,可以绕过节点  $x$  并到达灰色区域内的所有节点.在节点的邻域中,如果没有障碍物,节点具备直线和对角线两种移动方式,如图 1 所示.

对直线方向的筛选条件如式(2):

$$\text{length}(p, \dots, n | x) \leq \text{length}(p, x, n) \quad (2)$$

式中, $\text{length}$ 表示路径长度; $(p, \dots, n | x)$ 指父节点  $p$  不经过节点  $x$  到达目标点  $n$  的任意路径; $(p, x, n)$ 是指父节点  $p$  经过节点  $x$  到达目标点  $n$  的路径.

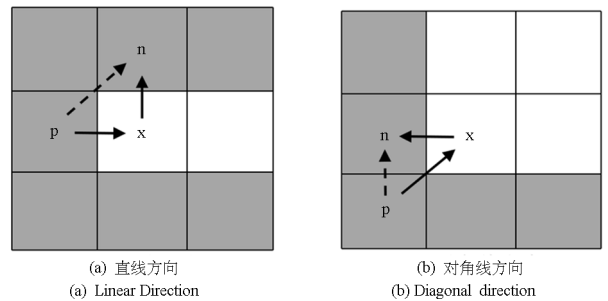


图 1 节点的邻域内无障碍物  
Fig. 1 No obstacles in the neighborhood of nodes

同样,对对角线方向的筛选条件如式(3):

$$\text{length}(p, \dots, n | x) < \text{length}(p, x, n) \quad (3)$$

#### 1.1.2 节点的邻域内有障碍物

当前节点  $x$  的非自然邻居中存在有障碍物,同时节点  $x$  的父节点  $p$  经过节点  $x$  到目标点  $n$  的路径长度比任何一条不经过  $x$  到  $n$  的路径长度更小,则  $n$  是  $x$  的被迫邻居,如图 2 所示.因此,当节点周围有障碍物时,对被迫邻居的筛选条件如式(4):

$$\text{length}(p, \dots, n | x) > \text{length}(p, x, n) \quad (4)$$

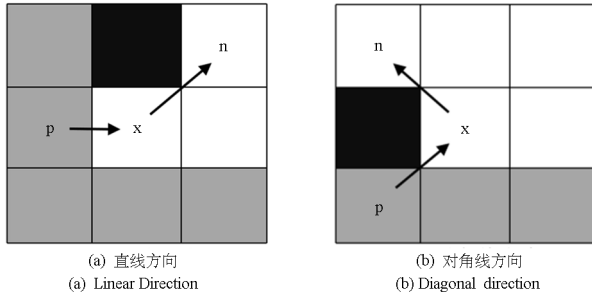


图2 节点的邻域有无障碍物

Fig. 2 Have obstacles in the neighborhood of nodes

### 1.1.3 跳点的筛选

综上所述,跳点需至少满足以下三个条件之一:

- (1)节点  $x$  可以是起始点或终止点之一;
- (2)节点  $x$  具备至少一个被迫邻居;
- (3)如果从父节点  $p$  到当前节点  $x$  的移动为对角方向,同时当前节点  $x$  在水平、垂直方向上移动时,在这两个方向上有满足条件(1)、(2)的节点.

## 1.2 蚁群算法和 TSP 问题

蚁群算法是一种智能仿生算法,为复杂的系统性难题提供了一种新的解决方案.在针对本文中所涉及的多目标点巡检问题时,运用蚁群算法的思路来解决这种 TSP 问题,有助于提高算法的效率,同时具有自适应性和全局搜索能力.

### 1.2.1 TSP 模型

TSP 问题是一种经典优化问题,其首先给定了一个巡检点的集合  $C$ ,其中包括  $n$  个巡检点的坐标,以及每两个巡检点之间的距离  $d_{ij}$ .多点巡检规划问题的目标在于找到一种最优的方案,使得在遍历所有巡检点后回到起点,从而在最短的总路径长度下完成遍历任务.该问题用数学模型描述如式(5):

$$\min T = \sum_{i=1}^{n-1} d(C_i, C_{i+1}) + d(C_n, C_1) \quad (5)$$

式中,  $C_1$  为选定的出发巡检点,  $C_i (i=1, 2, \dots, n)$  为待遍历的巡检点;  $d(C_i, C_{i+1})$  为巡检点  $i$  与巡检点  $i+1$  之间的路径长度.

## 2 改进 JPS 算法

JPS 算法是目前应用较为广泛的一种路径规划算法,通过预处理并在搜寻过程中跳过一些不必要的节点以提高搜寻效率.尽管跳点搜索法在许多情况下表现出色,但也存在路径复杂性、安全性等问题.同时,由于它搜索过程中更关注搜索效率,在选择跳跃点时可能会错过某些潜在的更优路径,可

能会陷入局部最优解.针对上述问题,本文提出了一种基于多目标改进的 JPS 算法(ERJPS).

### 2.1 基于启发函数的改进策略(EJPS)

#### 2.1.1 估计代价 $h(n)$

在传统的 JPS 算法中,通常使用对角线距离或曼哈顿距离作为启发函数来计算待选择点与目标点之间的代价.然而,这种单一的距离代价忽略了路径的方向性,这可能会导致生成的路径导向型差,偏离最佳路线.因此,本文在待选节点  $n$  到目标节点  $G$  的代价函数  $h(n)$  中引入了关于角度的引导因子如式(6):

$$h(n) = \kappa_l L_{d_{jx}}(n, G) + \kappa_\gamma \zeta \delta(n, G) \quad (6)$$

$$L_{d_{jx}}(i, j) = \sqrt{2} \min(|X_j - X_i|, |Y_j - Y_i|) + ||X_j - X_i| - |Y_j - Y_i|| \quad (7)$$

$$\zeta = L_{d_{jx}}(n, G) / L_{d_{jx}}(S, G) \quad (8)$$

式中,  $S, x, n, G$  分别为起始点、当前点、待选择点和目标点;  $\kappa_l$  为距离评估的权重系数,  $\kappa_\gamma$  为平滑度评估的权重系数;  $L_{d_{jx}}$  为对角线距离函数,  $(X_i, Y_i), (X_j, Y_j)$  分别为节点  $i, j$  的横坐标和纵坐标;  $\zeta$  为角度引导系数的自适应调节因子,在寻路开始时,  $\zeta$  无限接近于 1,可以加强角度启发信息的作用,以指引准确的路径方向;在寻路后期,距离目标点较近,  $\zeta$  无限接近于 0,此时,通过削弱角度启发信息的作用以更快地到达目标点.  $\delta$  为角度引导因子,计算公式如式(9):

$$\delta = \left| \arccos \left( \frac{l_{SG}^2 + l_{nG}^2 - l_{Sn}^2}{2l_{SG}l_{Sn}} \right) \right| \quad (9)$$

式中,  $l_{SG}, l_{nG}, l_{Sn}$  分别为起始点到目标点、待选择点到目标点、起始点到待选择点的直线距离;如图 3 所示,  $\delta$  为  $l_{SG}$  与  $l_{nG}$  之间的夹角,  $0 \leq \delta \leq \pi$ ,通过引入角度引导因子,可以指导搜索算法更注重路径的方向性.这有助于算法选择更适合实际应用的路径;  $\delta$  越小,路径就越接近于理想轨迹  $l_{SG}$ ,具有更强的导向性.

#### 2.1.2 实际代价 $g(n)$

JPS 算法通常选择最优的跳跃点来迅速到达目标点,这在大多数情况下可以显著提高搜寻效率,但可能致使路径安全性差、不够光滑,出现快速转折和大角度变换等现象.因此本文在计算当前节点  $x$  到待选节点  $n$  的实际代价  $g(n)$  时,采用多目标优化策略,将距离、平滑度和安全性这三个评价

指标作为优化的约束,如式(10).

$$g(n) = \kappa_l l(x, n) + \kappa_\gamma \gamma \langle \vec{px}, \vec{xn} \rangle + \kappa_b b(n) \quad (10)$$

其中,  $\kappa_l$  为距离影响因子的权重系数,  $\kappa_\gamma$  为路径平滑度影响因子的权重系数[同式(6)中启发函数的权重系数];  $\kappa_b$  为路径安全性影响因子的权重系数.

距离影响因子计算公式如式(11);平滑度影响因子计算公式如式(12);安全性影响因子的计算公式如式(13).

$$l(x, n) = \sqrt{(Y_n - Y_x)^2 + (X_n - X_x)^2} \quad (11)$$

$$\gamma \langle \vec{px}, \vec{xn} \rangle = \left| \arccos \frac{\vec{px} \cdot \vec{xn}}{|\vec{px}| |\vec{xn}|} \right| \quad (12)$$

$$b(n) = \sum_{i=x+1}^n (1 / \sqrt{(Y_i - Y_o)^2 + (X_i - X_o)^2}) \quad (13)$$

式中,  $l(x, n)$  表示当前节点  $x$  到待选节点  $n$  的距离;  $\gamma \langle \vec{px}, \vec{xn} \rangle$  表示当前点  $x$  到待选节点  $n$  的向量与父节点  $p$  到当前点  $x$  的向量之间的夹角,如图3中所示;  $b(n)$  表示当前点  $x$  移动方向上的下一点  $x+1$  与跳点  $n$  之间的所有节点与其最近的障碍物的直线距离的倒数和,  $X_o$ 、 $Y_o$  表示与待选择点  $i$  最近的障碍物点的横、纵坐标.

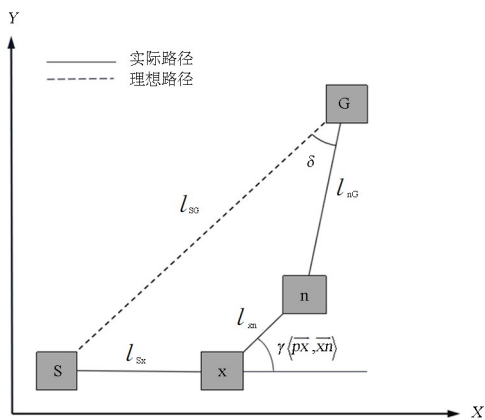


图3 角度引导原理图

Fig. 3 Schematic diagram of angle guidance

综上所述,由式(6)、式(10)可将路径规划问题转变为一个单目标优化问题,如式(14)所示:

$$\begin{aligned} \min J_F &= h(n) + g(n) \\ &= \kappa_l [l(x, n) + L_{d_{jx}}(n, G)] + \kappa_b b(n) + \\ &\quad \kappa_\gamma \gamma \langle \vec{px}, \vec{xn} \rangle + \zeta \delta(n, G) \end{aligned} \quad (14)$$

式中,  $J_F$  表示本次路径规划的综合考量路径长度、平滑度和安全性的代价;代价  $J_F$  越小,则路径综合性能更强,代价  $J_F$  越大,则路径综合性能更弱.

## 2.2 基于跳点筛选规则的改进策略(RJPS)

### 2.2.1 逆向跳点剔除规则建立

在跳点搜索过程中,虽然可以显著减少扩展节点的数量,只选择一些关键节点来进行跳跃,但仍然存在路径节点冗余问题.因此本文在使用JPS算法的过程中,使用跳点剔除规则再次剔除冗余节点,以实现路径的进一步平滑化,并同时保持路径的可行性和合理性.

跳点剔除是一种路径优化策略,其核心思想是在路径中选择某一节点与其后续节点相连,然后检查这两个节点所形成的线段是否被障碍物所穿越,以此来判断是否剔除中间的冗余节点.在文献[18]中,使用的改进跳点剔除方法,虽然处理后路径转折节点变少、变平滑,但其生成的路径仍受限于栅格的离散坐标点.在文献[19]中,进行了连续空间路径规划,减少了路径节点,但其按顺序( $p_2, p_3, \dots, p_{n-1}$ )依次筛选所有路径节点,会遗漏部分的更优解.

本文提出了一种改进的跳点剔除规则(RJPS).首先将初始节点S和目标节点G的连线作为第一次的剔除判定线段,若线段(SG)未与障碍物干涉,则剔除中间所有节点;若出现障碍物穿越,则循序遍历初始节点S与节点( $p_{n-1}, p_{n-2}, \dots, p_2$ )的连线,直至出现线段(S,  $p_i$ )不与障碍物干涉,则剔除节点( $p_2, p_3, \dots, p_{i-1}$ );接着,将剔除得到的节点  $i$  与目标节点G之间的线段,进行下一轮的障碍物相交判定,直至达到无法进一步剔除的情况为止.

如图4(a)所示为文献[19]的算法仿真过程,该剔除规则在路径优化的第一步中,连接了初始点S与后续节点  $p_3$ ,但该连线穿越了障碍物.于是继续选择节点  $p_2$  作为新的判断点,连接节点  $p_2$  与  $p_4$ ,然而同样这条连线也穿越了障碍物.经过两轮搜寻,得到了路径( $S \rightarrow p_2 \rightarrow p_3 \rightarrow p_4$ ),显然忽视了更优的路径( $S \rightarrow p_4$ ).

图4(b)中所展示的RJPS剔除策略通过逆向的遍历方式进行节点剔除.这种逆向的遍历节点的方法通常能够找到更优的路径[如图4(b)中的路径: $S \rightarrow p_4$ ],因为它能够考虑到路径的整体情况,而不只是局部的节点连线情况.

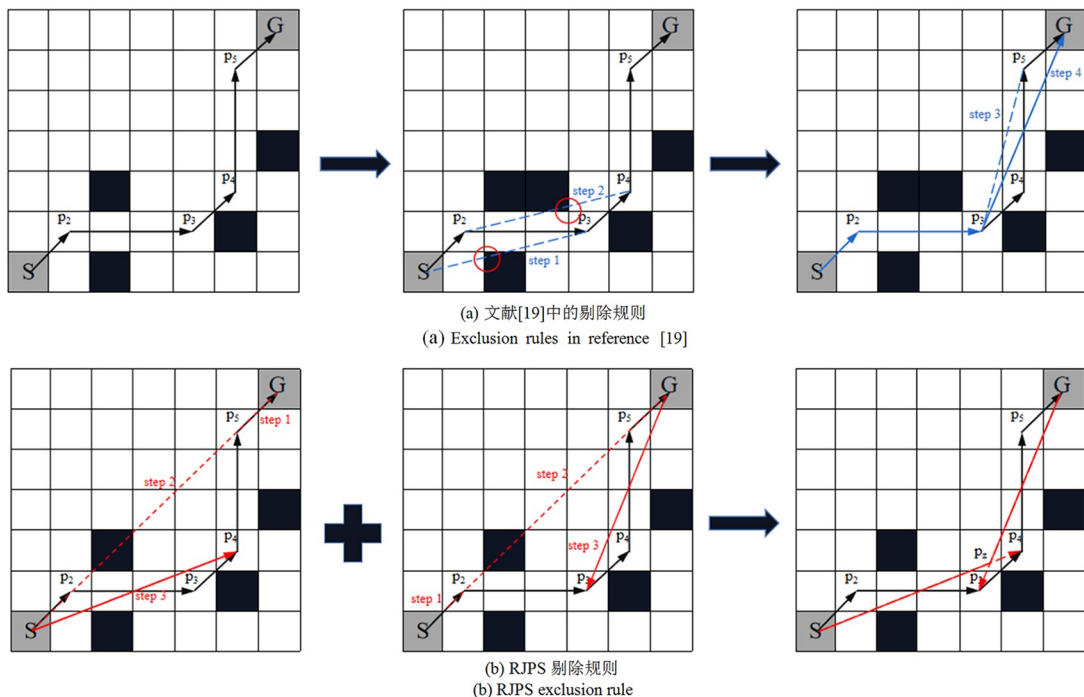


图4 基于JPS算法的冗余点删除示意图

Fig. 4 Schematic diagram of redundant point deletion based on JPS algorithm

### 2.2.2 双向跳点剔除规则建立

然而,采用逆向的跳点筛选规则可能会导致需要检查的节点数量增加,从而影响路径规划的效率.为了克服这一问题,本文在此基础上使用了双向的跳点剔除策略,即从初始节点和目标节点同时开始跳点剔除,实现了双向的RJPS策略.从初始点 $S$ 和目标点 $G$ 同时开始逆向在节点剔除,如图4(b)中前两张图,当双向规划的路径出现相交时,得到新的路径( $S \rightarrow p_z \rightarrow G$ ).通过使用这种策略,不仅在保证路径质量的前提下,还显著提高了节点剔除速度.

## 3 基于改进ACO算法求解TSP问题

在巡检机器人的多目标点路径规划中,通常会采用TSP模型来确定巡检顺序.然而,传统的TSP模型仅考虑了两个巡检点之间的直线距离,忽视了实际道路中可能存在的障碍物,因此在实际应用中可能欠缺准确性和实用性.

为了更好地解决这个问题,除了考虑路径长度外,还应综合考虑路径的平滑度和安全性等因素,因此,本文提出了一种改进的ACO算法,在使用TSP模型计算两点之间代价时,使用ERJPS算法计算得到的多目标综合性能作为评估因子.这个评估因子涵盖了长度、平滑度和安全性,通过这种优

化方式,我们可以更准确地计划巡检顺序,以更高效地完成巡检任务.

蚂蚁 $k$ 根据各个巡检点路径上的信息素水平来选择下一个巡检点.设 $P$ 表示 $t$ 时刻蚂蚁 $k$ 从巡检点 $i$ 转移到巡检点 $j$ 的概率,如式(15):

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha (J_E)^{-\beta}}{\sum_{j \in allowed_k} [\tau_{ij}(t)]^\alpha (J_E)^{-\beta}}, & j \in allowed_k \\ 0 & , j \notin allowed_k \end{cases} \quad (15)$$

式中, $allowed_k$ 为蚂蚁 $k$ 待移动点的集合; $\tau_{ij}(t)$ 为 $t$ 时刻,第 $i$ 个巡检点与第 $j$ 个巡检点之间的信息素浓度; $J_E$ 表示本次路径规划的多目标综合代价; $\alpha$ 为信息素浓度的权重系数,其值越大,则表示蚂蚁在移动方向的选择中受信息素浓度的影响越大; $\beta$ 为路径的综合性能的权重系数,其值越大,蚂蚁更可能移动到路径综合性能高的巡检点,即路径更短、更平滑、更安全.

### 3.1 路径综合代价

在多目标规划模型 $\min J_F = h(n) + g(n)$ 中, $h(n)$ 代表估计代价,它在跳点搜索的仿真过程中起启发作用,目的是引导算法搜索的方向,并不直接反映路径所具有的实际代价.然而, $g(n)$ 表示路

径的实际长度、平滑度、安全性等实际代价。

因此,在TSP问题中,我们将每条路径的真实代价 $g(n)$ [式(10)]作为多目标点路径规划的优化模型,如式(16):

$$\min J_E = g(n) = \kappa_l l(x, n) + \kappa_\gamma \gamma \langle \vec{p}_x, \vec{x}_n \rangle + \kappa_b b(n) \quad (16)$$

通过引入这种综合性的评价指标,蚂蚁在选择路径的过程中将不再仅仅考虑两个巡检点之间的直线距离,而是更加全面地考虑其综合性能。

### 3.2 自适应信息素更新

巡检点 $i$ 到 $j$ 的信息素浓度更新规则,如式(17)、式(18):

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij} \quad (17)$$

$$\Delta\tau_{ij} = \sum_{k=1}^m \Delta\tau_{ij}^k \quad (18)$$

式中, $\Delta\tau_{ij}$ 为信息量增量; $m$ 为每轮迭代蚂蚁的数量; $\rho$ 为信息素挥发因子, $\rho < 1$ 。信息素的挥发速度与参数 $\rho$ 成正比,较大的 $\rho$ 值会加速算法的收敛速度,但可能导致算法在局部最优解附近停滞,而较小的 $\rho$ 值则可能使算法对全局搜索的能力更强,但也可能会得到局部最优解,因此本文采用了一种信息素挥发因子,如式(19):

$$\rho = e^{-\varphi \left( \frac{N_{\max}}{N_c} \right)} \quad (19)$$

式中, $N_{\max}$ 表示最大迭代次数; $N_c$ 表示本次迭代的次数;在算法初期,由于 $N_{\max}/N_c$ 较大,信息素浓度挥发因子 $\rho$ 小,蚁群信息素的指导作用减小,蚂蚁会趋向于探寻不同的路线,从而能够更容易发现潜存的优质解。在算法后期,由于迭代次数增加, $N_{\max}/N_c$ 越来越小,信息素挥发因子 $\rho$ 增大,使得ACO算法能够更快速收敛以得到优质解。 $\varphi$ 表示信息素挥发因子的影响因子; $\Delta\tau_{ij}$ 表示在本次迭代中,蚂蚁 $k$ 释放在巡检点 $i$ 和 $j$ 之间的信息素。

根据本轮搜寻所得到的综合性能最优、最差路径,使用一种奖惩机制,以引导蚂蚁在搜索空间中找到更优的路径,摒弃最差的路径,如式(20):

$$\sum_{k=1}^m \Delta\tau_{ij}^k = \begin{cases} \frac{Q}{J_E(k)}, J_E(k) \neq J_{\max}, J_{\min} \\ -\psi \frac{Q}{J_E(k)}, J_E(k) = J_{\max} \\ \psi \frac{Q}{J_E(k)}, J_E(k) = J_{\min} \\ 0, \text{蚂蚁 } k \text{ 未经过 } ij \end{cases} \quad (20)$$

式中, $Q$ 为蚂蚁在本次迭代中具有的信息素总量。 $J_E(k)$ 为蚂蚁 $k$ 在本次迭代中规划的路径的综合性能代价; $J_{\min}$ 为本次迭代过程中,综合性能代价最低的路径; $J_{\max}$ 为本次迭代过程中,综合性能代价最高的路径; $\psi$ 为奖惩机制的比例系数, $\psi > 1$ ,其值越大,意味着在信息素更新中奖励和惩罚的影响越强烈,使蚂蚁更加聚焦于寻找综合性能较好的路径。

## 4 仿真实验与分析

实验平台的计算机操作系统为Win11(64bit),处理器为i7-11800H,内存为16GB。使用MATLAB R2021a软件,分别分析了以下几点:多目标权重系数选择、不同环境下路径规划仿真与对比分析、多巡检点的整体路径规划。

### 4.1 多目标仿真实验

ERJPS受到距离、平滑度和安全性多个因素的交互影响,这些因素协同作用,难以通过简单的理论进行确定其权重系数。因此,通过实验仿真验证来找到最切合具体问题的参数仍然是一种常用的方法<sup>[20]</sup>。

在针对多目标的优化问题中[如式(14)],各个单目标函数具有不同的重要性,很难直接确定权重系数 $K_l$ 、 $K_\gamma$ 和 $K_b$ 的取值。本文权重系数取值的确定过程如下:

由于跳点搜索法在算法实现的过程中,仍主要以距离为导向来进行路径规划的,平滑度和安全性只是作为辅助优化目标,所以首先将距离权重系数 $K_l$ 设为1,然后分别将另外两个权重系数设为(0,0)、(1,0)、(0,1);随后,进行三种不同目标优化的仿真实验。仿真结果如图5所示,其中初始点坐标为(1,30),目标点坐标为(30,1)。

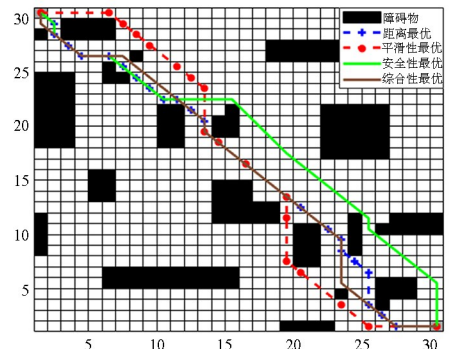


图5 多目标仿真实验结果

Fig. 5 Results of multi-objective simulation experiment

由仿真测试结果可知,当距离影响因子的权重系数为1,其他两个权重系数为0时,获得最短的路径;当平滑度影响因子的权重系数为1时,安全性影响因子的权重系数为0,得到转弯次数较少的路径;当安全性影响因子的权重系数为1时,平滑度影响因子的权重系数为0,生成具有更好安全性的路径。

表1 多目标权重优化  
Table 1 multi-objective weight optimization

优化目标	距离	平滑	安全	综合性能
$\kappa_l$	1	1	1	1
$\kappa_\gamma$	0	1	0	9.32
$\kappa_b$	0	0	1	2.51
路径长度 $l$	43.94	46.87	45.11	43.94
平滑度 $\gamma$	10.21	4.71	7.07	6.28
安全性 $b$	23.94	22.59	17.45	19.08
路径长度 ( $E_l = \kappa_l \cdot l$ )	43.94	46.87	45.11	43.94
平滑度 ( $E_\gamma = \kappa_\gamma \cdot \gamma$ )	0	4.71	0	58.52
安全性 ( $E_b = \kappa_b \cdot b$ )	0	0	17.45	47.89

如表1所示,路径长度、平滑度、安全性的值具有不同的数量级;为了平衡各个单目标参数的作用,必须选择合适的权重系数 $\kappa_l$ 、 $\kappa_\gamma$ 和 $\kappa_b$ 的值来对三个目标参数进行标准化.首先,分别计算以上三次不同侧重的仿真中路径长度、平滑度、安全性的最优值.然后,将距离侧重的仿真中,路径长度的最优值作为标准数量级,以此来确定另外两个指标的权重系数.如式(21)~式(23):

$$\kappa_l = 1 \quad (21)$$

$$\kappa_\gamma = l_{\min} / \gamma_{\min} \quad (22)$$

$$\kappa_b = l_{\min} / b_{\min} \quad (23)$$

通过该方法获得的权重系数将每个单目标优化的函数值放置在相同的数量级上.从表1的数据可以观察到,通过权重系数标准化后,三个单目标的影响程度  $E$  值的大小基本相同,因此它们对于总目标值的影响相近.这种标准化有助于实现综合规划路径,平衡各个目标在路径规划中的重要性。

#### 4.2 不同环境下的仿真实验

为了使实验效果更加具有说服力,本文采用了尺寸为 $30 \times 30$ 但障碍物密度不同的3种栅格地图来进行仿真实验.在不同复杂程度的栅格地图上,

进行传统JPS算法、文献[13]中双向同步JPS算法、多目标优化后的EJPS算法(其中: $\kappa_l = 1$ 、 $\kappa_\gamma = 9.32$ 和 $\kappa_b = 2.51$ )和引入了逆向跳点剔除方法的ERJPS算法仿真对比实验,如图6~图8所示。

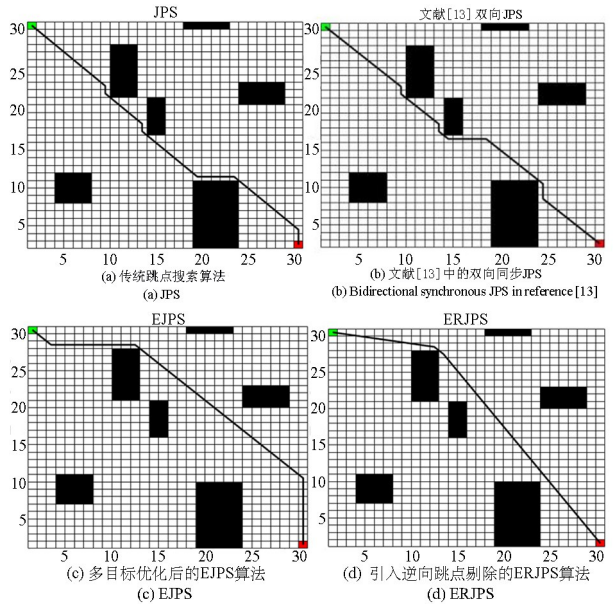


图6 地图1环境下的四种算法对比

Fig. 6 Comparison of four algorithms in map 1 environment

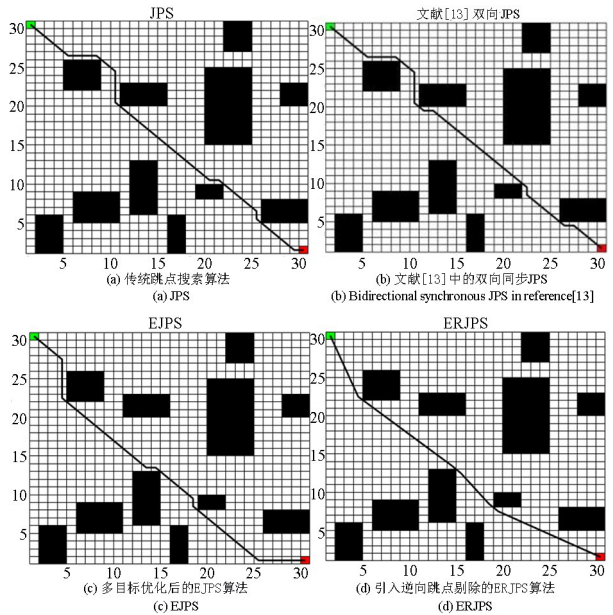


图7 地图2环境下的四种算法对比

Fig. 7 Comparison of four algorithms in map 2 environment

表2为四种不同算法在不同栅格环境下的各项仿真数据.其中,在对四种算法规划的路径进行综合性能比较时,由于ERJPS算法是在EJPS算法的基础上使用了跳点剔除规则,且生成的路径不完全沿着栅格点的中心点移动,无法直接计算所有节点与障碍物的距离倒数和,因此在此处将ERJPS算法的安全性指数近似等效为EJPS算法的安全性指数。

表 2 不同环境下实验结果对比  
Table 2 Comparison of experimental results in different environments

地图型号	路径长度				平滑度			
	JPS	文献[13]改进 JPS	EJPS	ERJPS	JPS	文献[13]改进 JPS	EJPS	ERJPS
地图 1	43.36	43.36	46.28	43.02	5.50	6.28	2.36	0.79
地图 2	43.94	43.94	44.53	42.36	7.07	7.85	5.50	0.77
地图 3	46.28	46.28	48.63	47.02	13.35	11.00	7.07	4.90

---

地图型号	路径长度				平滑度			
	JPS	文献[13]改进 JPS	EJPS	ERJPS	JPS	文献[13]改进 JPS	EJPS	ERJPS
地图 1	19.75	18.16	11.25	11.25	144.19	147.47	96.48	78.62
地图 2	22.51	23.01	18.47	18.47	166.33	174.86	142.15	95.90
地图 3	30.06	32.48	26.36	26.36	246.15	230.32	180.69	158.85

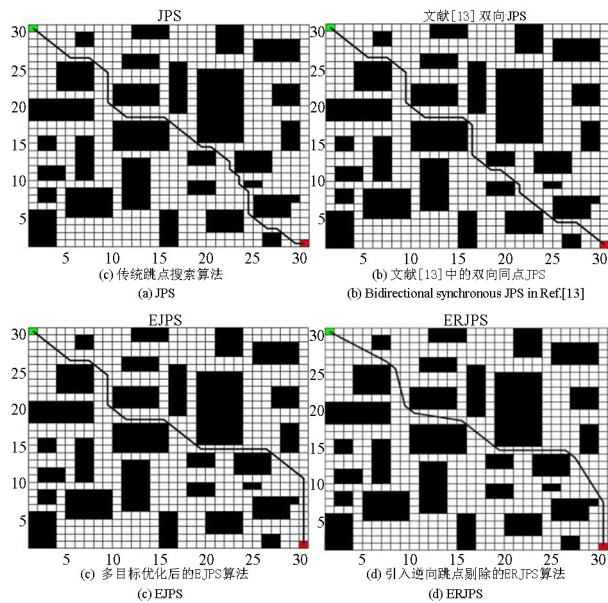


图 8 地图 3 环境下四种算法对比  
Fig. 8 Comparison of four algorithms in map 3 environment

由仿真得到的结果和实验数据可知,传统的 JPS 算法和文献[13]中的双向同步 JPS 算法规划的路径拐点多、不够光滑、缺乏安全性。而 EJPS 算法规划的路径在综合性能上明显优于前两种算法,说明改进的 EJPS 算法具有更强的全局优化能力。ERJPS 算法是针对 EJPS 算法的优化,通过引入逆向的跳点剔除方法,能大幅降低路径长度,并进一步提高路径平滑度。在三种环境中,ERJPS 算法较传统 JPS 算法相比综合性能分别增加了 45.50%、42.34%、35.47%,较文献[13]中的双向 7 同步 JPS 算法相比综合性能分别增加了 46.69%、45.16%、31.03%;这表明,在简单环境中,ERJPS 有明显的优化效果;同时在复杂环境下,ERJPS 算法仍然具有良好的综合性能优化效果。

### 4.3 基于改进 ACO 算法的多巡检点仿真实验

为了验证改进的 ACO 算法在将传统的 TSP 模型应用到巡检路径规划问题中的有效性,本文建立了一个规模为  $30 \times 30$  的栅格地图,将传统 TSP 模型、文献[16]中改进的蚁群算法、文献[17]中改进的 TSP 算法、本文中结合了 ERJPS 算法和改进 ACO 算法进行仿真实验对比。

其中迭代次数  $N_{max} = 20$ 、每轮的蚂蚁总数  $m = 20$ 、信息素的重要程度  $\alpha = 1$ 、综合性能的重要程度  $\beta = 3$ 、信息素的挥发因子的权重系数  $\Phi = 2$ 、奖惩机制的比例系数  $\Psi = 3$ 、信息素总量  $Q = 100$ 。

仿真结果如图 9 所示。由仿真结果图 9(a)和 9(b)

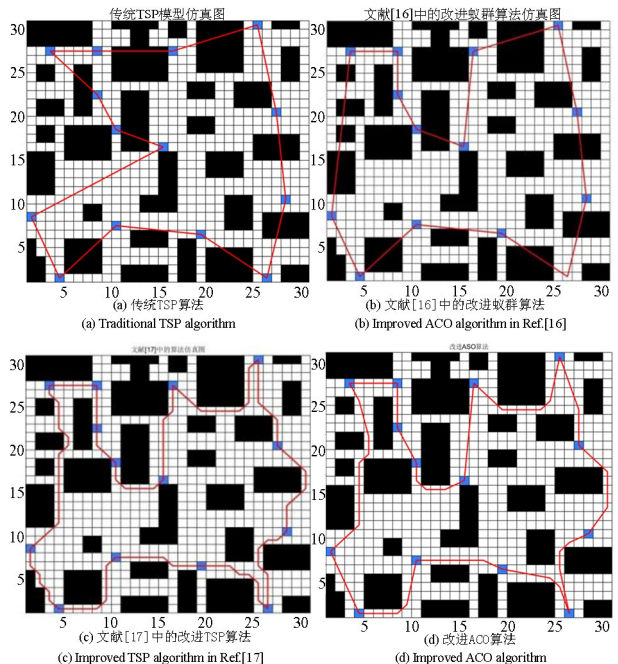


图 9 多巡检点算法仿真  
Fig. 9 Simulation of multiple inspection point algorithm



中可知,虽然文献[16]中的改进的蚁群算法相较于传统 TSP 模型能更准确找出最短路径,但这两种算法都忽视了巡检点之间的障碍物信息,不具有实用性;由图 9(c)、图 9(d)、表 3 可知,图 9(d)中改进 ACO 算法与文献[17]中的算法都能规划出考虑障碍物之后的路径,但是改进 ACO 算法相较于文献[17]中的算法,综合性能增加了 19.4%。图 10 为四种算法迭代图,横坐标为迭代次数,从迭代收敛的势可以清晰地观察到,改进 ACO 算法相较于前三种算法收敛速度分别提高了 66.7%、50%、71.4%。所以改进的 ACO 算法在路径可行性、综合性能和迭代速度均优于前三种算法。

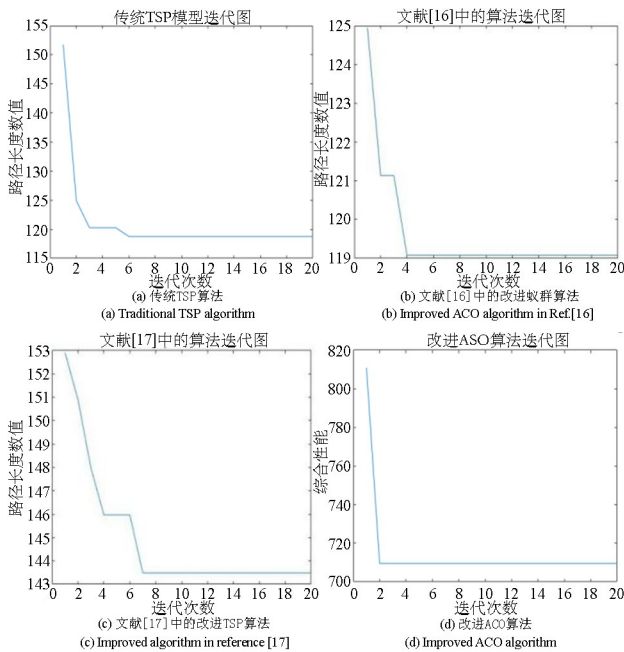


图 10 基于多巡检点的四种算法迭代图

Fig. 10 Iterative graph of four algorithms based on multiple inspection points

表 3 后两种算法的仿真数据对比

Table 3 Comparison of simulation data for the last two algorithms

评估指标	长度	平滑度	安全性	综合性能	代数
文献[17]的算法	142.50	48.67	112.82	879.27	7
改进 ACO 算法	139.29	33.38	103.08	709.12	2

## 5 总结

传统的 JPS 算法在规划路径时,存在不平滑、不安全性等缺点。首先,本文针对 JPS 算法进行改进,提出了针对评估函数进行改进的 EJPS 算法,增加了自适应的角度导向因子,以引导路径快速靠近目标点,并且综合考量了距离、平滑度和安全性

等参数,得到了综合性能更优的路径。其次,使用双向的逆向跳点剔除规则,解决了路径节点冗余现象,避免路径出现不必要的转向,提高了机器人的执行效率和平稳性。同时将改进得到的 ERJPS 算法和改进 ACO 算法结合运用到多巡检点的路径规划中。最后,在不同复杂程度的栅格环境下进行仿真测试,ERJPS 算法能够准确搜寻到路径,综合性能比传统 JPS 算法更高,而且在各种规格地图中都具有良好的效果;在进行多巡检点的仿真测试时,改进的 ACO 算法也能找寻到一条更高质量的路径,且具有更快的收敛速度。

## 参考文献

- [1] LI X D, TIAN B, HOU S D, et al. Path planning for mount robot based on improved particle swarm optimization algorithm [J]. *Electronics*, 2023, 12 (15): 3289.
- [2] ALIREZA M, VINCENT D, TONY W. Experimental study of path planning problem using EMCOA for a holonomic mobile robot [J]. *Journal of Systems Engineering and Electronics*, 2021, 32(6): 1450–1462.
- [3] 冯世盛, 徐青, 朱新铭, 等. 基于地形数据的长距离越野路径快速规划方法研究[J]. *地球信息科学学报*, 2022, 24(9): 1742–1754.  
FENG S S, XU Q, ZHU X M, et al. Fast method for long-distance off-road path planning based on terrain data [J]. *Journal of Geo-Information Science*, 2022, 24(9): 1742–1754. (in Chinese)
- [4] 余翔, 姜陈, 段思睿, 等. 改进 A\* 算法和人工势场法的路径规划[J]. *系统仿真学报*, 2024, 36(3): 782–794.  
YU X, JIANG C, DUAN S R, et al. Path planning for improvement of A\* algorithm and artificial potential field method [J]. *Journal of System Simulation*, 2024, 36(3): 782–794. (in Chinese)
- [5] LIU L S, WANG B, XU H. Research on path-planning algorithm integrating optimization A-star algorithm and artificial potential field method [J]. *Electronics*, 2022, 11(22): 3660.
- [6] CHEN Z Z, XIAO J Z, WANG G F. An effective path planning of intelligent mobile robot using improved genetic algorithm [J]. *Wireless Communications and Mobile Computing*, 2022, 2022: 9590367.

- [7] 赵传超, 郑睿, 龚莉, 等. 网联环境中新能源货车的粒子群调度方法[J]. 系统仿真学报, 2023, 35(6): 1337-1350.  
ZHAO C C, ZHENG R, GONG L, et al. Particle swarm optimization for new energy truck scheduling in network environment [J]. Journal of System Simulation, 2023, 35(6): 1337-1350. (in Chinese)
- [8] 王海琛, 吴华瑞, 朱华吉, 等. 基于改进蚁群算法的蔬菜大田无人农机路径优化[J]. 中国农机化学报, 2023, 44(4): 187-194.  
WANG H C, WU H R, ZHU H J, et al. Path optimization of unmanned agricultural machinery in vegetable field based on improved ant colony algorithm [J]. Journal of Chinese Agricultural Mechanization, 2023, 44(4): 187-194. (in Chinese)
- [9] 侯宇翔, 高焕兵, 汪子健, 等. 基于改进跳点搜索法的移动机器人路径规划[J]. 组合机床与自动化加工技术, 2023(3): 54-58.  
HOU Y X, GAO H B, WANG Z J, et al. Mobile robot path planning based on improved jump point search algorithm [J]. Modular Machine Tool & Automatic Manufacturing Technique, 2023(3): 54-58. (in Chinese)
- [10] SU Q, MA S, WANG L, et al. Artificial potential field guided JPS algorithm for fast optimal path planning in cluttered environments [J]. Journal of the Brazilian Society of Mechanical Sciences and Engineering, 2022, 44(12): 602.
- [11] 胡士强, 武美萍, 施健, 等. 融合向量叉积与跳点搜索策略的改进 A\* 算法研究[J]. 机械科学与技术, 2023, 9(5): 1-10.  
HU S Q, WU M P, SHI J, et al. Research on improved A\* algorithm with fusion of vector cross-product and jump point search strategy [J]. Journal of Mechanical Science and Technology, 2023, 9(5): 1-10. (in Chinese)
- [12] 黄智榜, 胡立坤, 张宇, 等. 基于改进跳点搜索策略的安全路径研究[J]. 计算机工程与应用, 2021, 57(1): 56-61.  
HUANG Z B, HU L K, ZHANG Y, et al. Research on security path based on improved hop search strategy [J]. Computer Engineering and Application, 2021, 57(1): 56-61.
- [13] 李艳生, 张静琦, 刘彦瑜, 等. 物流机器人调度系统及其双向同步跳点搜索算法设计[J]. 仪器仪表学报, 2023, 44(7): 121-132.  
LI Y S, ZHANG J Q, LIU Y Y, et al. Logistics robot scheduling system and the design of bidirectional synchronous jump point search algorithm [J]. Chinese Journal of Scientific Instrument, 2023, 44(7): 121-132. (in Chinese)
- [14] 王伟, 张彦斐, 宫金良, 等. 基于自适应升温模拟退火算法的农业机器人全区域覆盖策略[J]. 华南农业大学学报, 2021, 42(6): 126-132.  
WANG W, ZHANG Y F, GONG J L, et al. Whole area coverage strategy of agricultural robot based on adaptive heating simulated annealing algorithm [J]. Journal of South China Agricultural University, 2021, 42(6): 126-132. (in Chinese)
- [15] 陈颖杰, 高茂庭. 基于信息素初始分配和动态更新的蚁群算法[J]. 计算机工程与应用, 2022, 58(2): 95-101.  
CHEN Y J, GAO M T. Pheromone initialization and dynamic update based ant colony algorithm [J]. Computer Engineering and Applications, 2022, 58(2): 95-101. (in Chinese)
- [16] 马世轩, 游晓明, 刘升. 动态信息素更新和路径奖惩的蚁群算法[J]. 计算机工程与应用, 2023, 59(4): 64-76.  
MA S X, YOU X M, LIU S. Ant colony algorithm based on dynamic pheromone update and path rewards and punishments [J]. Computer Engineering and Applications, 2023, 59(4): 64-76. (in Chinese)
- [17] 杨平. 移动机器人多目标点路径规划研究[D]. 南充: 西华师范大学, 2021.  
YANG P. Multi objective point path planning for mobile robot [D]. Nanchong: China West Normal University, 2021. (in Chinese)
- [18] 马小陆, 梅宏, 龚瑞, 等. 基于改进 ACS 算法的移动机器人路径规划研究[J]. 湖南大学学报(自然科学版), 2021, 48(12): 79-88.  
MA X L, MEI H, GONG R, et al. Research on path planning of mobile robot based on improved ACS algorithm [J]. Journal of Hunan University (Natural Sciences), 2021, 48(12): 79-88. (in Chinese)
- [19] ZHANG B, ZHU D L. A new method on motion planning for mobile robots using jump point search and Bezier curves [J]. International Journal of Advanced Robotic Systems, 2021, 18(4): 1-10.
- [20] LI P, ZHU H. Parameter selection for ant colony algorithm based on bacterial foraging algorithm [J]. Mathematical Problems in Engineering, 2016, 2016: 6469721.