

基于多级视野自适应蚁群算法的移动机器人路径规划

许建民 邓冬冬 宋雷 杨炜

(厦门理工学院机械与汽车工程学院, 厦门 361024)

摘要: 针对传统蚁群算法(Ant colony optimization, ACO)在应用于移动机器人路径规划时存在优化能力差,易于死锁,搜索效率低等问题,提出一种多级视野自适应蚁群(Multi-level field of view adaptive ant colony optimization, MLFVAACO)算法。首先在 ACO 的基础上依次扩展 2 级视野使得规划出的路径更加平滑;其次设计了自适应全局初始信息素更新策略,既避免了蚂蚁在算法初期出现盲目搜索现象又加强了蚂蚁选择可选区域的指导作用;然后对算法迭代过程中的死锁蚂蚁进行优化,以提高蚁群的利用率和增加搜索解的多样性;最后对蚂蚁的状态转移规则进行改进来避免蚂蚁陷入局部最优解。通过仿真选取 MLFVAACO 算法的最优参数,在 2 种不同复杂程度的格栅地图中分别与传统 ACO 算法、改进 ACO 算法和图搜索算法进行对比,验证 MLFVAACO 算法的可行性和有效性。仿真结果表明,在简单与复杂环境中,MLFVAACO 算法相较于传统 ACO 算法最优路径分别缩短 12.74% 和 4.38%,路径转折点分别减少 50% 和 63.16%,蚂蚁利用率分别提升 99.99% 和 99.95%,搜索效率分别提高 60.14% 和 62.17%;相较于改进 ACO 算法和图搜索算法,MLFVAACO 算法能够规划出路径平滑度更好的最短路径,同时搜索解的质量也更好。这充分验证了 MLFVAACO 算法在应用于移动机器人路径规划时具有出色的综合性能。

关键词: 移动机器人;全局路径规划;蚁群算法;多级视野;死锁

中图分类号: TP242.6 **文献标识码:** A **文章编号:** 1000-1298(2024)11-0475-11

OSID:



Mobile Robot Path Planning Based on Multi-level Field of View Adaptive Ant Colony Algorithm

XU Jianmin DENG Dongdong SONG Lei YANG Wei

(School of Mechanical and Automotive Engineering, Xiamen University of Technology, Xiamen 361024, China)

Abstract: Aiming at the problems of poor optimization ability, easy deadlock, and low search efficiency of the traditional ant colony optimization (ACO) when applied to mobile robot path planning, a multi-level field of view adaptive ant colony optimization (MLFVAACO) algorithm was proposed. Firstly, on the basis of ACO, the two levels field of view was expanded sequentially to make the planned path smooth. Secondly, an adaptive global initial pheromone update strategy was designed, which not only avoided the blind search phenomenon of ants in the early stage of the algorithm but also strengthened the guiding role of ants in selecting optional areas. Then the deadlock ants in the algorithm iteration process were optimized to improve the utilization of the ant colony and increase the diversity of search solutions. Finally, the state transition rule of ants was improved to prevent ants from falling into the local optimal solution. The optimal parameters of the MLFVAACO algorithm were selected through simulation analysis, and the feasibility and effectiveness of the MLFVAACO algorithm were verified by comparing it with the traditional ACO algorithm, the improved ACO algorithms, and the graph search algorithms, respectively, in two kinds of grid maps with different levels of complexity. The simulation results showed that in simple and complex environments, compared with the traditional ACO algorithm, the optimal path of the MLFVAACO algorithm was shortened by 12.74% and 4.38%, respectively, the turning points of the path were reduced by 50% and 63.16%, respectively, the ant utilization rate was increased by 99.99% and 99.95%, respectively, and the search efficiency was increased by 60.14% and 62.17%, respectively. Compared with the improved ACO algorithms and the graph search algorithms, MLFVAACO

收稿日期: 2023-12-29 修回日期: 2024-02-03

基金项目: 国家重点研发计划项目(2023YFB3406500)和国家自然科学基金项目(51975499)

作者简介: 许建民(1981—),男,副教授,博士,主要从事移动机器人路径规划和精密减速器等研究,E-mail: xujianmin1020@163.com

algorithm can plan the shortest path with better path smoothness, while the quality of the search solutions was also better. This fully validated the excellent performance of MLFVAACO algorithm when applied to mobile robot path planning.

Key words: mobile robot; global path planning; ant colony algorithm; multi-level field of view; deadlock

0 引言

随着数字经济与人工智能的发展,移动机器人在仓储物流^[1]、智能制造^[2]、智慧城市^[3]、医疗服务^[4]、智慧农业^[5]等领域都有广泛的应用前景。路径规划是实现移动机器人自主导航的关键技术之一,根据环境信息类型,路径规划问题可以分为全局路径规划和局部路径规划^[6]。全局路径规划作为路径规划过程中的基础并且对移动机器人的自主性和智能化发挥着关键作用,其目的是在环境信息已知的情况下从起点到终点规划出一条满足约束条件(安全性高、最短路径、平滑性好等)的无碰撞最优路径。目前,主流的全局路径规划算法分为经典算法和启发式智能优化算法。经典算法主要有:Dijkstra 算法^[7]、A* 算法^[8]、D* 算法^[9]、Floyd 算法^[10]等。启发式智能优化算法主要有:遗传算法^[11]、粒子群算法^[12]、蚁群算法(ACO)^[13-14]等。

经典算法在简单环境下的路径规划性能较好,但随着环境变得复杂,其计算成本急剧上升^[15]。与经典算法相比,启发式智能优化算法能够提高路径规划的效率和准确性,能适应不同的环境变化和约束条件,因此被广泛应用于移动机器人路径规划领域中^[16-18]。在启发式智能优化算法中,ACO 具有较强的全局搜索能力、通用性和鲁棒性等优点,但其在解决移动机器人路径规划问题时存在易陷入局部最优、搜索效率低、路径转折点过多等缺点。因此,为了克服传统 ACO 的缺点,许多学者对其进行改进来提高其在不同应用场景下的路径规划性能^[19-25]。

虽然现有研究在一定程度上提高了传统 ACO 的路径规划性能,但是仍存在蚂蚁死亡率高,搜索解的质量差以及全局最优解搜索能力弱等问题没有得到有效解决。因此,本文在传统 ACO 中引入蚂蚁的多级视野范围、自适应全局初始信息素更新策略、随机过往路径节点策略、自适应伪随机状态转移策略,提出一种多级视野自适应蚁群算法(MLFVAACO)。

1 格栅地图建模和传统蚁群算法

1.1 格栅地图建模

图 1 为 10 × 10 格栅地图,黑色格栅为障碍物,用 1 表示;白色格栅为自由移动区域,用 0 表示;绿

色格栅为机器人的起点,用 S 表示;橙色格栅为机器人的终点,用 E 表示。整个格栅地图采用直角坐标系与格栅序号相结合的方式进行识别,格栅序号按从上到下、从左到右依次标记为 1, 2, ..., 100。格栅序号 i 与它对应的直角坐标 (x_i, y_i) 之间的坐标转换关系为

$$\begin{cases} x_i = \text{ceil}\left(\frac{i}{N_x}\right) + 0.5 \\ y_i = \begin{cases} \text{mod}(i, N_y) + 0.5 & (\text{mod}(i, N_y) \neq 0) \\ N_y + \text{mod}(i, N_y) + 0.5 & (\text{其它}) \end{cases} \end{cases}$$

(1)

$$N_x = N_y = \text{size}(\mathbf{G}, 1)$$

(2)

式中 $\text{ceil}()$ ——向上取整运算函数
 $\text{mod}()$ ——取模运算函数
 $\text{size}()$ ——格栅行列大小函数
 N_x ——格栅地图总列数
 N_y ——格栅地图总行数
 $\mathbf{G}$ ——格栅地图对应矩阵

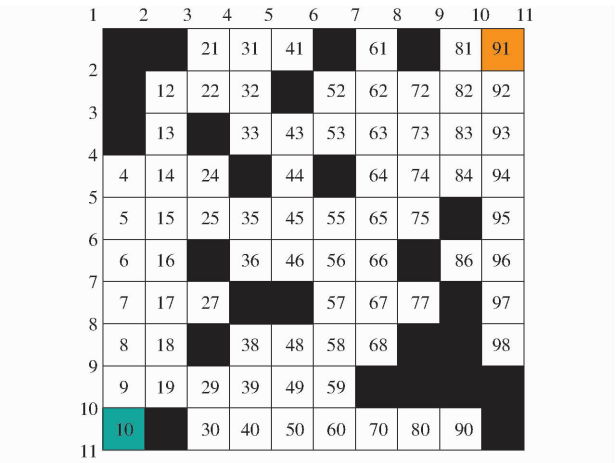


图 1 格栅地图
Fig. 1 Grid map

1.2 传统蚁群算法

1.2.1 状态转移概率

蚁群算法是一种模拟自然界中蚁群觅食过程的启发式智能优化算法。在蚁群算法迭代前,需要对参数进行初始化,设蚂蚁最大数量为 M ,开始时各个节点之间的信息素浓度相同。蚂蚁根据路径上的信息素浓度和启发式函数选择下一个待访问节点。 $P_{(i,j)}^k$ 表示蚂蚁 k 从节点 i 移动到节点 j 的状态转移概率,其计算公式为

$$P_{(i,j)}^k(t) = \begin{cases} \frac{(\tau_{(i,j)}(t))^\alpha (\eta_{(i,j)}(t))^\beta}{\sum_{q \in A_k} (\tau_{(i,q)}(t))^\alpha (\eta_{(i,q)}(t))^\beta} & (q \in A_k) \\ 0 & (q \notin A_k) \end{cases} \quad (3)$$

其中 $\eta_{(i,j)}(t) = \frac{1}{D_{(i,j)}}$ (4)

$$D_{(i,j)} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (5)$$

式中 A_k ——蚂蚁 k 待访问节点的集合
 $\tau_{(i,j)}(t)$ —— t 时刻节点 i 与节点 j 连接路径上的信息素浓度
 α ——信息素启发式因子
 β ——期望启发因子
 $\eta_{(i,j)}(t)$ ——启发式函数
 $D_{(i,j)}$ ——节点 i 与节点 j 间的欧氏距离
 (x_i, y_i) ——节点 i 的直角坐标
 (x_j, y_j) ——节点 j 的直角坐标

1.2.2 信息素更新规则

在蚂蚁释放信息素的同时,各个节点间连接路径上的信息素也在逐渐蒸发。当蚂蚁完成当前迭代后,各个节点间的连接路径上的信息素浓度需要进行全局更新,计算公式为

$$\begin{cases} \tau_{(i,j)}(t+1) = (1 - \rho) \tau_{(i,j)}(t) + \Delta \tau_{(i,j)} \\ \Delta \tau_{(i,j)} = \sum_{k=1}^M \tau_{(i,j)}^k \end{cases} \quad (6)$$

其中

$$\tau_{(i,j)}^k = \begin{cases} \frac{Q}{L_k} & (\text{蚂蚁 } k \text{ 经过 } i, j \text{ 之间路径}) \\ 0 & (\text{其它}) \end{cases} \quad (7)$$

式中 ρ ——信息素挥发系数,取 0 ~ 1
 $\Delta \tau_{(i,j)}$ ——当前迭代过程中节点 i 与节点 j 连接路径上信息素浓度增量
 $\tau_{(i,j)}^k$ ——蚂蚁 k 在节点 i 与节点 j 连接路径上释放信息素浓度
 Q ——信息素强度
 L_k ——当前迭代蚂蚁 k 规划路径总长度

2 多级视野自适应蚁群算法(MLFVAACO)

2.1 扩大蚂蚁的视野范围

对于蚂蚁而言,开阔的视野范围可以帮助它们提前规划路线,有利于快速到达食物源。但是,在格栅地图中,传统 ACO 的蚂蚁只能搜索到当前视野范围内的 8 个邻域,如图 2a 中橙色格栅所示。可以看出,此时的蚂蚁只有 1 级视野范围,这将导致它们搜索到食物源的效率大大降低,因此,为了更快速地到达目标位置,在传统 ACO 的基础上依次扩展 2 级视

野。第 1 级扩展视野具有 16 个移动方向和 16 个邻域,如图 2b 中灰色格栅所示,再与传统 ACO 的 1 级视野结合,此时的蚂蚁具有 2 级视野范围;第 2 级扩展视野具有 24 个移动方向和 24 个邻域,如图 2c 中绿色格栅所示,然后与 2 级视野范围结合,最终蚂蚁具有 3 级视野范围。与传统 ACO 相比,具有多级视野的蚂蚁在爬行过程中有更多的方向选择和更广的移动范围,有利于蚂蚁快速定位目标。图 2a ~ 2c 中的蓝色格栅为蚂蚁当前位置,其余格栅为其邻域,图 2d 为蚂蚁不同视野范围对应的搜索方向。

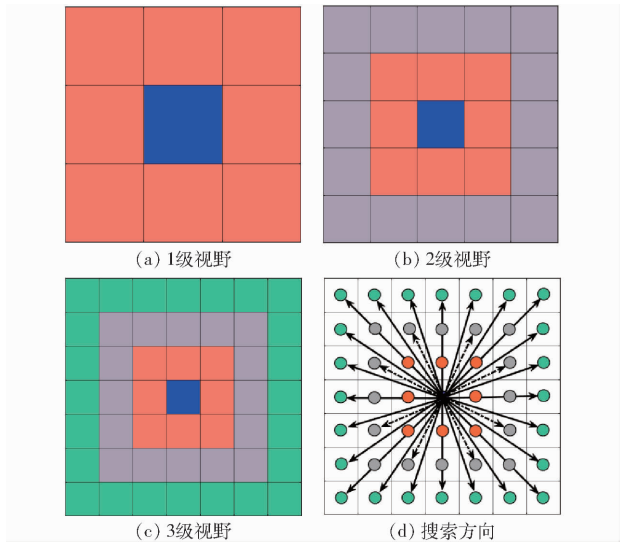


图 2 蚂蚁的视野范围和搜索方向
Fig. 2 Ant's field of view and search direction

2.1.1 具有多级视野的蚂蚁移动规则

蚂蚁在不同视野范围内的可见邻域不同,并且可见邻域的移动条件有所差异。图 3 为蚂蚁多级视野范围内可移动邻域,其中蓝色格栅 S 为当前蚂蚁所在位置,绿色格栅为多级视野范围内可移动邻域,灰色格栅为障碍物格栅。如果蚂蚁想要移动到绿色格栅,那么蚂蚁当前位置 S 与绿色格栅的连线之间不能存在障碍物格栅。例如,图 3a 中,如果蚂蚁想要移动到 G_1 ,那么 G_1 不能是障碍物格栅,否则不能移动;图 3b 中,如果蚂蚁想要移动到 G_1 ,那么 P_1 、 P_2 都不能是障碍物格栅,否则不能移动;图 3c 中,如果蚂蚁想要移动到 G_1 ,那么 P_1 、 P_2 、 P_3 、 P_4 都不能是障

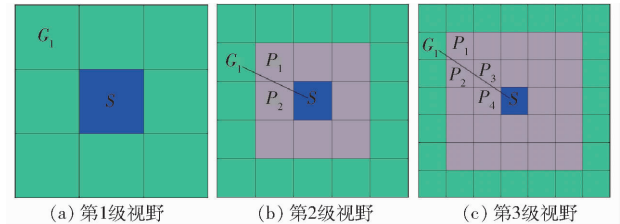


图 3 多级视野范围内可移动邻域
Fig. 3 Movable neighborhoods within multi-level field of view

碍物格栅,否则不能移动;其余同理。

2.1.2 具有多级视野的蚂蚁优势

具有多级视野的蚂蚁有更多的方向和邻域,能够在短时间内快速定位目标。下面举例说明具有多级视野的蚂蚁相对于传统蚁群算法中蚂蚁的优势。图4是蚂蚁采用不同视野范围规划出的路径,其中绿色格栅为起点,橙色格栅为终点。假设格栅的边长为1,可以计算出蚂蚁采用不同视野范围规划出的最优路径长度和对应路径上的移动次数,如表1所示。由表1可知,与传统ACO中的蚂蚁相比,具有多级视野范围的蚂蚁可以用最少的移动次数规划出更短的路径,这说明具有多级视野范围的蚂蚁能够减少路径转折点,使得规划出的路径平滑度更好,有利于蚂蚁快速到达目标位置。

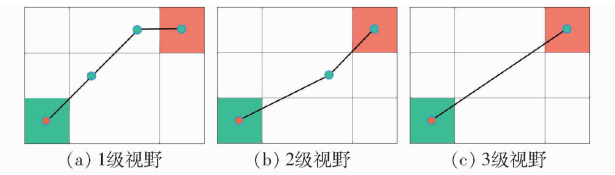


图4 不同视野范围规划的路径

Fig.4 Path planning for different fields of view

表1 不同视野范围的路径规划结果

视野范围	路径长度	移动次数
1级视野	3.828 4	3
2级视野	3.650 3	2
3级视野	3.605 6	1

2.2 自适应全局初始信息素更新策略

在ACO迭代开始前,需要对信息素进行初始化,但是ACO中的信息素初始值是固定不变的,这将导致在ACO迭代初期会出现蚂蚁盲目搜索现象,从而降低了算法的搜索效率。如图5a所示,其中左下角格栅为起点,右上角格栅为终点,红色矩形中的3只蚂蚁出现了盲目搜索现象。为了克服这一缺点,本文提出了一种自适应全局初始信息素更新策略,来加强蚂蚁在ACO早期选择可选区域的指导作用,该策略的计算公式为

$$\tau_{(i,j)}(0) = \begin{cases} -1 & (i \text{ 或 } j \text{ 是障碍物}) \\ \frac{1}{D_{(i,j)} + D_{(j,E)}} + \frac{D_{(S,E)}}{D_{(S,i)} + D_{(i,E)}} & (\text{其它}) \end{cases}$$

(8)

式中 $\tau_{(i,j)}(0)$ ——初始信息素浓度
 $D_{(i,J)}$ ——当前节点*i*到格栅地图中任意节点*J*欧氏距离
 $D_{(J,E)}$ ——任意节点*J*到终点*E*欧氏距离

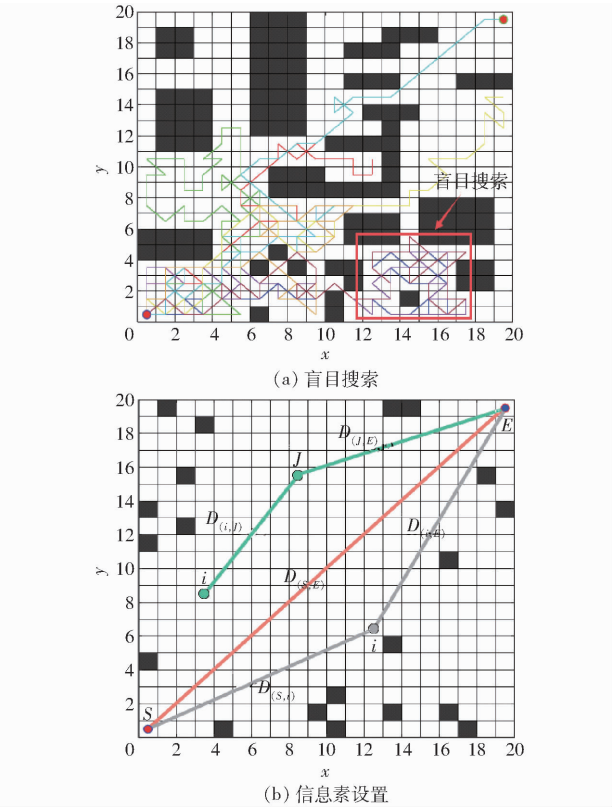


图5 自适应全局初始信息素更新策略

Fig.5 Adaptive global initial pheromone update strategy

$D_{(S,i)}$ ——起点*S*到当前节点*i*欧氏距离
 $D_{(i,E)}$ ——当前节点*i*到终点*E*欧氏距离
 $D_{(S,E)}$ ——起点*S*和终点*E*欧氏距离
 $D_{(i,J)}$ 、 $D_{(J,E)}$ 、 $D_{(S,i)}$ 、 $D_{(i,E)}$ 、 $D_{(S,E)}$ 如图5b所示。

i、*J*、*E*三者的连通路线越长,信息素会越低,反之越高,以此达到离散化信息素的效果来改善蚂蚁的盲目搜索现象;*i*、*S*、*E*三者的连通路线越短,信息素会越高,以此来加强可选区域的引导作用,使得蚂蚁能够快速到达终点。综上所述,该策略能够根据不同的格栅地图自适应地全局更新初始信息素浓度,有效地改善蚂蚁盲目搜索现象。

2.3 随机过往路径节点策略

在ACO迭代过程中,蚂蚁走过的路径节点会被存放在禁忌表中,禁止蚂蚁再次通过。然而,随着格栅地图中障碍物密度的增加,再加上禁忌表的限制和格栅地图边界的限制,这将导致蚂蚁无路可走,即死锁,如图6a所示。死锁问题会导致蚂蚁大量死亡,从而降低算法搜索解的多样性进而影响搜索解的质量。因此,本文提出了一种随机过往路径节点策略,允许死锁蚂蚁随机选择死锁点之前走过的路径中的任意节点进行再次规划路径。如图6b所示,蚂蚁1和蚂蚁2分别从起点*S*到终点*E*,蓝色路线是蚂蚁1规划的路径,此时的蚂蚁1陷入障碍物死锁中,死锁点为*p*₆,橙色路线是蚂蚁2规划的路径,

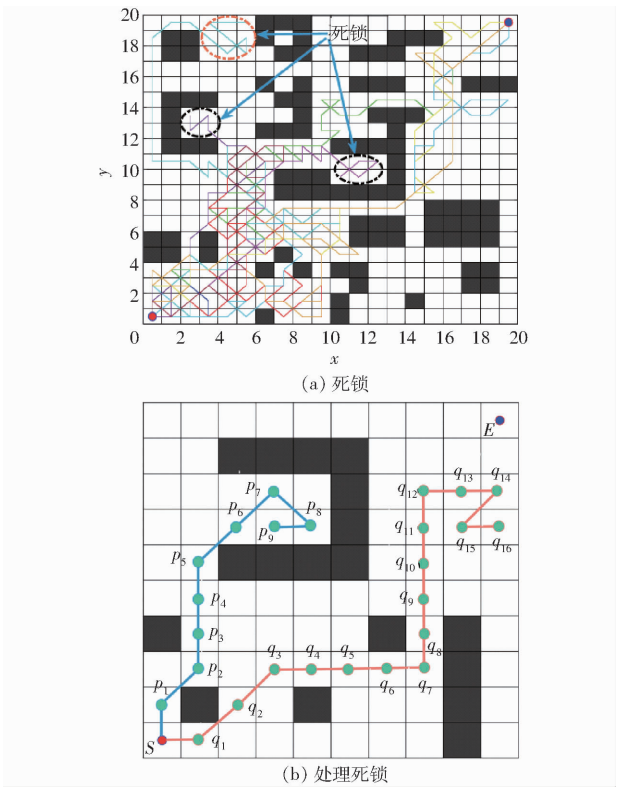


图 6 随机过往路径节点策略

Fig. 6 Random past-path-node strategy

此时的蚂蚁 2 陷入自死锁中,死锁点为 q_{14} ,随后对 2 只蚂蚁采用随机过往路径节点策略,该方法计算式为

$$\text{Start_node} = \text{random}(m_i) \quad (i = 1, 2, \cdots, n) \quad (9)$$

式中 Start_node——死锁蚂蚁新的起始路径节点
random()——随机函数

m_i ——死锁点前蚂蚁走过的路径节点集合

由式(9)可知,蚂蚁 1 可随机选择路径节点为 $p_i(i = 1, 2, 3, 4, 5)$,蚂蚁 2 可随机选择路径节点为 $q_i(i = 1, 2, \cdots, 13)$ 。综上所述,采用随机过往路径节点策略能够使死锁蚂蚁具备二次路径规划的能力,该策略能够降低蚂蚁死亡率,提高算法搜索解多样性。

2.4 自适应伪随机状态转移策略

在 ACO 迭代过程中需要计算蚂蚁视野范围内的每一个可移动邻域的概率,然后再计算这些邻域的概率分布,最后通过轮盘赌方式选择下一个可移动邻域。然而,这种方式使得 ACO 过早收敛于局部最优解。为了克服上述问题,本文提出了一种自适应伪随机状态转移策略来改变蚂蚁选择下一节点的概率方式。 $S_{(i,j)}^k(t)$ 为 t 时刻蚂蚁 k 从节点 i 转移到节点 j 的概率,其计算式为

$$S_{(i,j)}^k(t) = \begin{cases} \arg\max_{j \in A_k} [(\tau_{(i,j)}(t))^\alpha (\eta_{(i,j)}(t))^\beta] & (Q_1 \leq Q_0) \\ P_{(i,j)}^k(t) & (Q_1 > Q_0) \end{cases} \quad (10)$$

式中 Q_1 ——均匀分布于 $[0, 1]$ 的随机数
 Q_0 —— Q_1 的临界值,取值一般在 $[0, 1]$
 $\arg\max(f(x))$ ——使得 $f(x)$ 取得最大值所对应的变量 x

由式(10)可知,当 $Q_1 \leq Q_0$ 时,下一个可选节点是由信息素浓度与启发式函数乘积的最大值决定的,这是一种确定性的选择方式,有利于提高算法收敛速度;当 $Q_1 > Q_0$ 时,由传统蚁群算法中的轮盘赌方式决定下一个可选节点,具有一定随机性,有利于扩大搜索解的范围以提高蚁群全局搜索能力。因此, Q_0 决定了选择确定性方式和轮盘赌方式的比重,同时对算法收敛速度和全局搜索能力有着重要影响。为了更好地平衡上述 2 种选择方式以避免算法陷入局部最优解,本文设计了一种能自适应计算 Q_0 的方法,计算公式为

$$Q_0 = \begin{cases} \frac{S_{\text{best}}}{S_{\text{worst}}} & (K_c \leq K_0) \\ \frac{D_{(S,E)}}{L_{gb}} & (K_c > K_0) \end{cases} \quad (11)$$

其中 $K_0 = 0.6K$

式中 S_{best} ——上次迭代中最短路径长度
 S_{worst} ——上次迭代中最差路径长度
 L_{gb} ——全局最短路径
 K_c ——当前迭代次数
 K_0 ——迭代次数临界值
 K ——最大迭代次数

在图 6a 的格栅地图上以 50 只蚂蚁 100 次迭代为例, Q_0 变化曲线如图 7 所示。在前 60 次迭代中,根据 S_{best} 和 S_{worst} 比值可以看出, Q_0 偏小且在动态随机的变化中,此时选择轮盘赌方式的概率更大,有利于扩大算法前期种群的多样性以提高算法的全局搜索能力;在 60 次迭代后,随着迭代次数的增加,全局最短路径质量也越来越好, Q_0 变大,此时选择确定性方式的概率更大,有利于加快算法后期收敛速度。

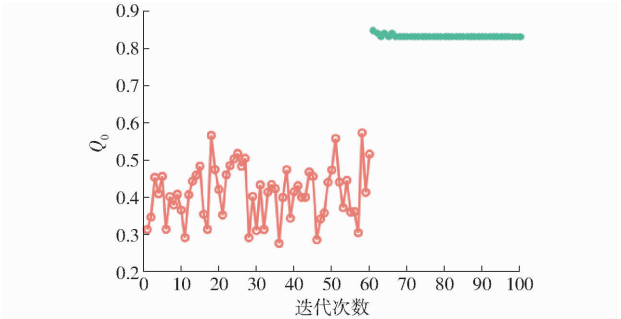


图 7 自适应计算 Q_0

Fig. 7 Adaptive calculation of Q_0

此外,将路径上的信息素控制在 $\tau_{\min} \sim \tau_{\max}$ 内来避免算法出现停滞和早熟,信息素更新规则为

$$\tau_{(i,j)}(t) = \begin{cases} \tau_{\max} & (\tau_{(i,j)}(t) > \tau_{\max}) \\ \tau_{(i,j)}(t) & (\tau_{\min} \leq \tau_{(i,j)}(t) \leq \tau_{\max}) \\ \tau_{\min} & (\tau_{(i,j)}(t) < \tau_{\min}) \end{cases}$$

(12)

其中

$$\begin{cases} \tau_{\max} = \frac{M}{\rho L_{gb}} \\ \tau_{\min} = -\frac{\tau_{\max}}{0.1M} \end{cases}$$

(13)

式中 $\tau_{\max}$ ——最大信息素浓度
 $\tau_{\min}$ ——最小信息素浓度

综上所述,通过引入自适应伪随机状态转移策略能够有效避免算法过早收敛于局部最优解。

2.5 MLFVAACO 算法的路径规划流程

综合以上提出的 4 种改进机制(蚂蚁的多级视野范围、自适应全局初始信息素更新策略、随机过往路径节点策略、自适应伪随机状态转移策略)与传统 ACO 相结合,形成一种新的 ACO 变体,称为多级视野自适应蚁群(MLFVAACO)算法。MLFVAACO 算法在移动机器人路径规划中的应用流程如图 8 所示,首先建立格栅地图环境模型并初始化参数,其次在算法迭代的不同时期分别采用上述 4 种改进机制来搜索最优解,最后算法迭代结束输出最优路径。

3 仿真与分析

为验证本文所提 MLFVAACO 算法在解决移动机器人路径规划问题上的有效性和优越性,分别与传统 ACO 算法、改进 ACO 算法(文献[19]中的 IAACO 算法、文献[20]中的 IHMACO 算法、文献[21]中的 AIACSE 算法、文献[22]中的 MAACO 算法)和图搜索算法(A* 算法、Dijkstra 算法、Floyd 算法)在不同环境中进行对比。实验平台的计算机操作系统为 Windows 11,所有算法的编码平台为 Matlab R2022a。

3.1 仿真参数设置

在传统 ACO 中,各个参数的作用是密切相关的,其中信息素启发式因子 α 、期望启发因子 β 、信息素挥发系数 ρ 对算法性能起主要作用,它们的选取方法和选取原则直接影响算法的求解效率和求解质量。由于不同 ACO 变体对参数的适应性是不同的,到目前为止,还没有完善的分析方法能够直接确定统一的最优参数组合。因此,本文在文献[22]推荐的参数组合基础上,针对 MLFVAACO 算法,通过设置不同的参数值进行仿真分析,最后根据仿真结果的优缺点来选取适合 MLFVAACO 算法的参数组合。仿真规则如下:在每组测试中,仅改变其中一个参数,其它参数保持不变,为了保证实验的准确性以及

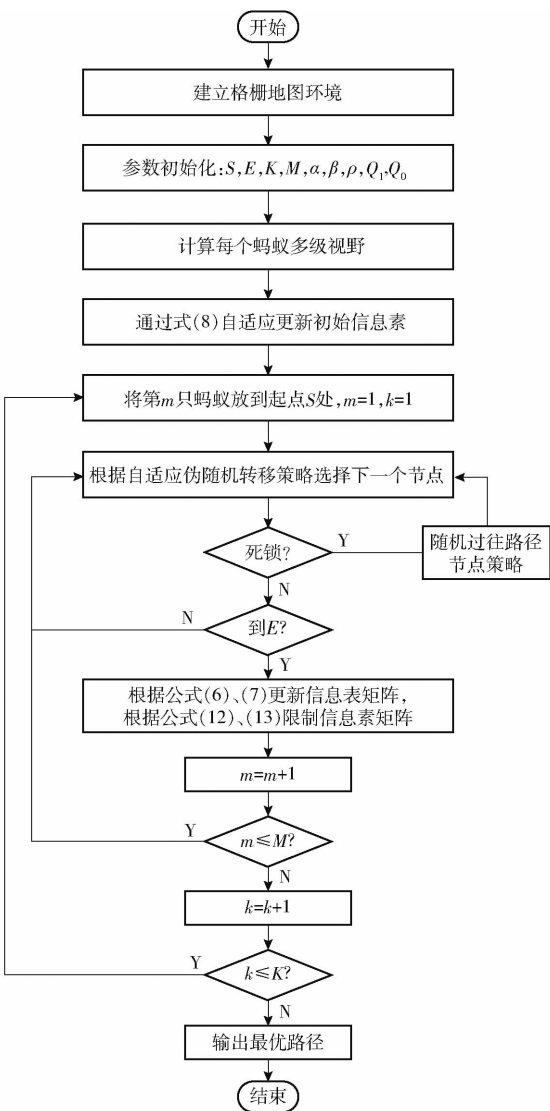


图 8 MLFVAACO 算法的路径规划流程图
Fig. 8 Path planning flowchart of MLFVAACO algorithm

减少偶然实验造成的误差影响,每个参数值进行 10 次实验,然后取 10 次实验结果的平均最优路径长度、平均迭代次数和平均路径转折点进行分析。

以图 6a 的格栅地图作为本次仿真的测试环境,设置默认推荐参数: $K=100, M=50, \alpha=1, \beta=7, \rho=0.2, Q=1, K_0=60$ 。首先测试信息素启发式因子 α , 当 $\alpha \in \{0.5, 0.6, \dots, 1.4, 1.5\}$ 时,仿真结果如图 9 所示。当 α 较小时,不论是 ACO 算法还是 MLFVAACO 算法都无法搜索到全局最优解。当 $\alpha \in [0.9, 1.0, 1.1]$ 时,2 种算法均具有较好的寻优能力和较高的搜索效率。随着 α 进一步增大,当 $\alpha \in [1.2, 1.5]$ 时,虽然 MLFVAACO 算法的搜索效率随之提高,但是寻优能力逐渐减弱,这使得其更容易陷入局部最优。此外, α 的变化对 MLFVAACO 算法路径上的转折点影响较小。综上所述,本文将 α 设置为 0.9,对应的平均最优路径长度为 27.903 1、平均迭代次数为 14.5、平均路径转折点为 5.6,此时

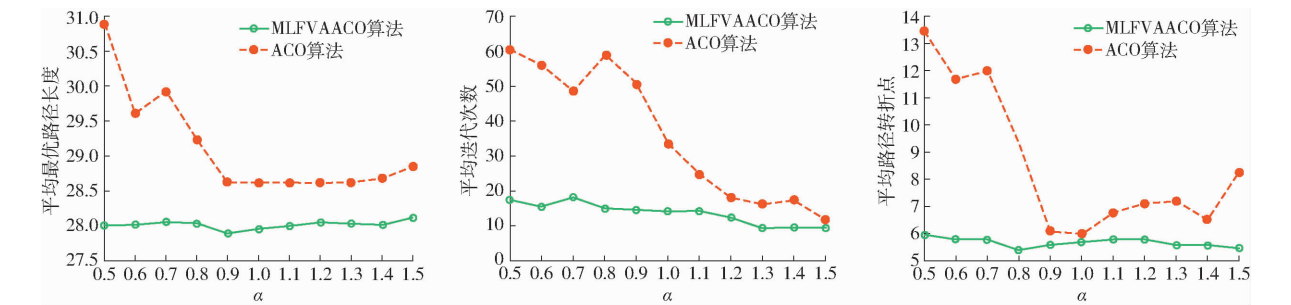


图 9 信息素启发式因子 α 对 MLFVAACO 算法和 ACO 算法性能的影响

Fig. 9 Influences of pheromone heuristic factor α on MLFVAACO and ACO performance

MLFVAACO 算法不仅具有较高的搜索效率而且具有较好的寻优能力。

再测试期望启发因子 β , 在同样的测试环境下, 参数设置如下: $K=100, M=50, \alpha=0.9, \beta=7, \rho=0.2, Q=1, K_0=60$ 。当 $\beta \in \{1, 2, \dots, 9, 10\}$ 时, 仿真结果如图 10 所示。随着 β 不断增大, MLFVAACO

算法的搜索效率不断提高。当 $\beta \in [8, 9, 10]$ 时, MLFVAACO 算法的综合性能(平均最优路径长度、平均迭代次数、平均路径转折点)较为出色。综上所述, 本文将 β 设置为 9, 对应的平均最优路径长度为 27.869 1、平均迭代次数为 9.9、平均路径转折点为 5.5, 此时 MLFVAACO 算法不仅具有较好的搜索

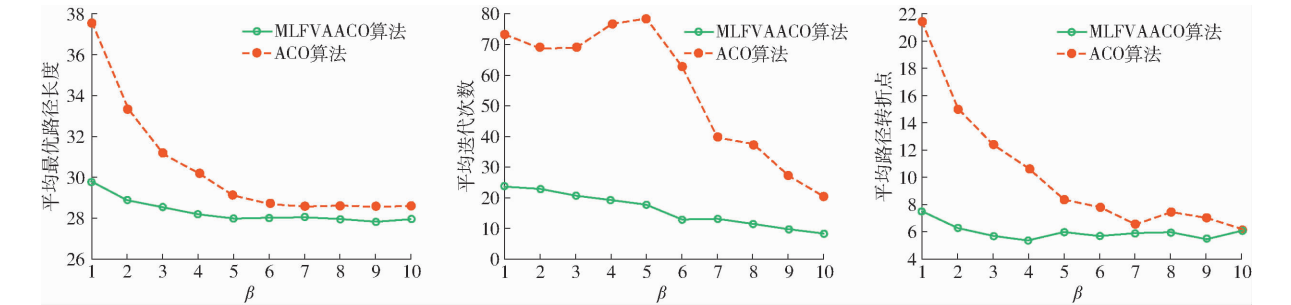


图 10 期望启发因子 β 对 MLFVAACO 算法和 ACO 算法性能的影响

Fig. 10 Influences of expectation heuristic factor β on MLFVAACO and ACO performance

效率而且具有路径平滑度较好的最优路径。

最后测试信息素挥发系数 ρ , 在同样的测试环境下, 参数设置如下: $K=100, M=50, \alpha=0.9, \beta=9, \rho=0.2, Q=1, K_0=60$ 。当 $\rho \in [0.1, 0.2, \dots, 0.8, 0.9]$ 时, 仿真结果如图 11 所示。当 $\rho \in [0.3, 0.4, \dots, 0.9]$ 时, 随着 ρ 不断增大, MLFVAACO 算法的搜索

效率不断提高, 但是寻优能力逐渐减弱, 这使得算法易于陷入局部最优解。当 ρ 为 0.2 时, 对应的平均最优路径长度为 27.861 1、平均迭代次数为 13.7、平均路径转折点为 5.9, 此时虽然 MLFVAACO 算法的搜索效率一般, 但是 MLFVAACO 算法具有较好的寻优能力和路径平滑度, 因此本文将 ρ 设置为 0.2。

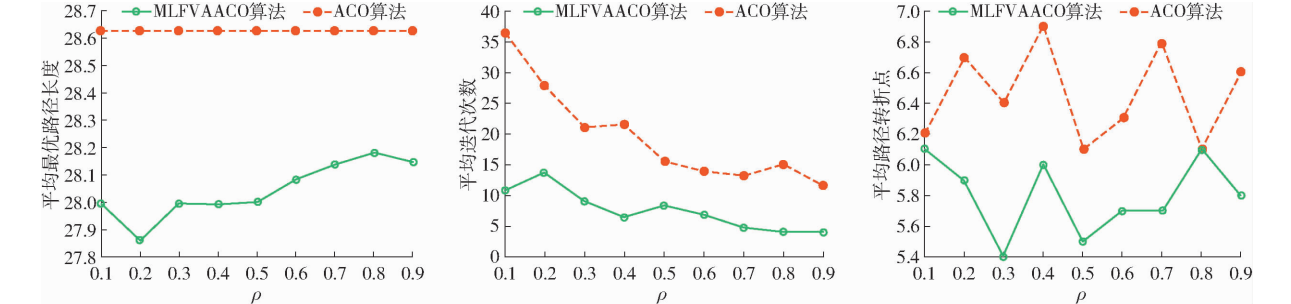


图 11 信息素挥发系数 ρ 对 MLFVAACO 算法和 ACO 算法性能的影响

Fig. 11 Influences of pheromone volatility coefficient ρ on MLFVAACO and ACO performance

综上所述, 从图 9 ~ 11 不难看出, α, β, ρ 对 MLFVAACO 算法的影响小于传统 ACO 算法并且 MLFVAACO 算法的各项性能指标(平均最优路径长度、平均迭代次数、平均路径转折点)相较于传统 ACO 算法有很大的提高。这主要得益于

MLFVAACO 算法中引入 4 种改进机制, 包括蚂蚁的多级视野范围、自适应全局初始信息素更新策略、随机过往路径节点策略和自适应的伪随机状态转移策略, 使得 MLFVAACO 算法具有更好的稳定性和鲁棒性。

3.2 算法仿真对比

3.2.1 简单环境

为验证 MLFVAACO 算法在不同环境中性能的优越性,分别与传统 ACO 算法、IAACO 算法^[19]、IHMACO 算法^[20]、图搜索算法(A*算法、Dijkstra 算法、Floyd 算法)在障碍物密度为 28% 的简单格栅地图中进行对比。首先,统一采用文献[19]中的格栅地图作为本次测试的环境模型,尺寸为 20×20,起点 S 的行列式坐标为(1,1),终点 E 的行列式坐标为(20,20),如图 12a 所示。其次,MLFVAACO 算法参数设置为 K=100,M=50,Q=1,α=0.9,β=9,ρ=0.2,K₀=60。最后,7 种算法分别独立运行 20 次来求解各项性能指标。仿真实验结果如图 12 和表 2 所示。采用的性能指标有:最优路径长度、平均路径长度、路径长度标准差(Std)、平均死亡蚂蚁数、蚂蚁死亡率 ζ、平均迭代次数、最优路径上的转折点。此外,障碍物密度计算式为

ω = (μ_{Black_grids} / N_xN_y) × 100% (14)

式中 μ_{Black_grids}——障碍物格栅总数

ω 是衡量格栅地图复杂程度的重要参考指标之一,ω 越大,则格栅地图越复杂,越能够检验算法的性能。

蚂蚁死亡率计算式为

ζ = (η_{Ave_ant} / KM) × 100% (15)

式中 η_{Ave_ant}——平均死亡蚂蚁数

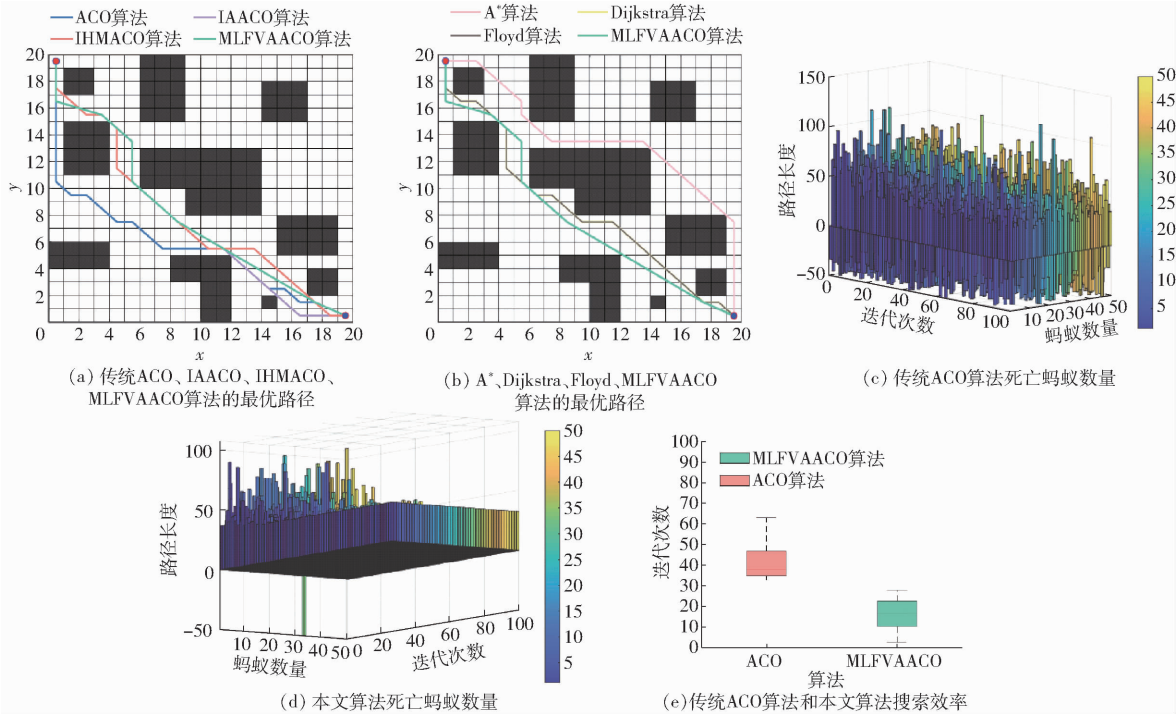


图 12 简单环境下仿真实验结果对比

Fig. 12 Comparisons of simulation experiment results in simple environment

ζ 直接反映蚂蚁利用率,进而影响搜索解质量。ζ 越大,说明死锁蚂蚁数量较多,导致蚂蚁死亡,那么蚂蚁利用率会偏低并且搜索解的多样性也会随之减少。

性能指标优化率 φ_{Im_rate} 计算式为

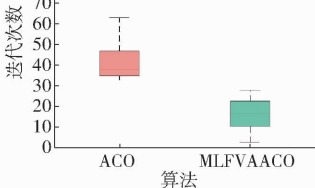
φ_{Im_rate} = | (φ_{Proposed} - φ_{Original}) / φ_{Original} | × 100% (16)

式中 φ_{Proposed}——MLFVAACO 算法性能指标

φ_{Original}——其它算法性能指标

由表 2 可知,在简单环境中,MLFVAACO 算法规划最优路径长度和平均路径长度分别为 29.286 1 和 29.379 2,这 2 种性能指标均优于其它 6 种算法,说明本文算法不仅具有较好的全局最优解搜索能力而且搜索解的质量更佳。就标准差而言,虽 MLFVAACO 算法稳定性不如图搜索算法 IHMACO 算法^[20],但比传统 ACO 算法和 IAACO 算法^[19]稳定。MLFVAACO 算法最优路径上的转折点为 6,比传统 ACO 算法减少 50%,比 IAACO 算法^[19]和 IHMAACO 算法^[20]减少 25%,比 Dijkstra 算法和 Floyd 算法减少 45.45%,同时结合图 12a 和图 12b,可以得出本文所提算法具有更好的路径平滑度。图 12c 和图 12d 分别为传统 ACO 算法死亡蚂蚁数量和 MLFVAACO 算法死亡蚂蚁数量,为了方便观察,将死亡蚂蚁路径长度设置为 -50~0 之间随机整数。可以看出,传统 ACO 算法中有大量蚂蚁死亡,平均死亡蚂蚁数为 1 543.20 只,蚂蚁死亡率为 30.86%,这将会减少算法搜索解多样性进而影响搜

MLFVAACO 算法
ACO 算法



(e) 传统 ACO 算法和本文算法搜索效率

表 2 简单环境下仿真实验结果

Tab.2 Simulation experiment results in simple environment

算法	路径长度			平均死亡蚂蚁 数量/只	$\xi/\%$	平均迭代次数	路径转折点
	最优值	平均值	标准差				
MLFVAACO	29.286 1	29.379 2	0.195 5	0.15	0.003	16.60	6
ACO	33.560 0	36.610 0	3.418 9	1 543.20	30.86	41.65	12
IAACO	29.790 0	30.120 0	0.395 3				8
IHMACO	29.790 0	29.790 0	0				8
A*	31.553 6	31.553 6	0				6
Dijkstra	29.799 0	29.799 0	0				11
Floyd	29.799 0	29.799 0	0				11

索解质量;反观 MLFVAACO 算法只有极少数蚂蚁死亡,平均死亡蚂蚁数为 0.15 只,蚂蚁死亡率仅为 0.003%,相较于传统 ACO 算法,蚂蚁利用率提升 99.99%。图 12e 为传统 ACO 算法和 MLFVAACO 算法在 20 次实验中每次搜索到最优路径时对应的迭代次数,可以看出,本文所提算法搜索速度均快于传统 ACO 算法,平均搜索到最优解迭代次数为 16.60,相较于传统 ACO 算法的 41.65,搜索效率提升 60.14%。

3.2.2 复杂环境

为进一步验证 MLFVAACO 算法在不同环境中

性能的优越性,分别与传统 ACO 算法、AIACSE 算法^[21]、MAACO 算法^[22]、图搜索算法(A* 算法、Dijkstra 算法、Floyd 算法)在障碍物密度为 37% 的复杂格栅地图中进行对比。首先,采用文献[21]中的格栅地图作为本次测试的环境模型,尺寸为 20 × 20,起点 S 的行列式坐标为(1,1),终点 E 的行列式坐标为(20,20),如图 13a 所示。其次,MLFVAACO 算法参数设置为 $K=100,M=50,Q=1,\alpha=0.9,\beta=9,\rho=0.2,K_0=60$ 。最后,7 种算法分别独立运行 20 次来求解各项性能指标。仿真实验结果如表 3 和图 13 所示。

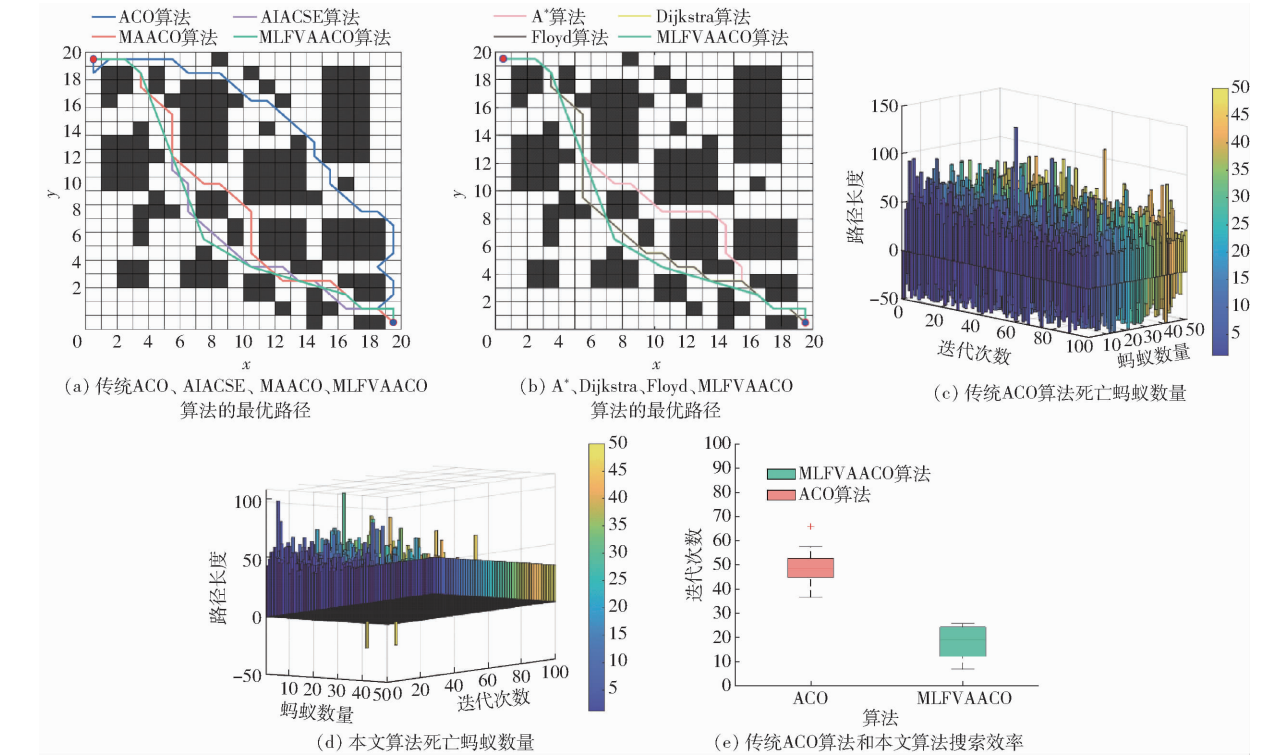


图 13 复杂环境下仿真实验结果对比

Fig.13 Comparisons of simulation experiment results in complex environment

由表 3 可知,在复杂环境中,MLFVAACO 算法最优路径为 30.407 6,平均路径为 30.407 6,这 2 种性能指标均优于其它 6 种算法,说明本文算法不仅具有较好的全局最优解搜索能力而且搜索解质量更

佳。由标准差可得,传统 ACO 算法稳定性最差,其余 6 种算法稳定性一致。MLFVAACO 算法最优路径上转折点最少,比传统 ACO 算法减少 63.16%,比 AIACSE 算法和 MAACO 算法减少 46.15%,比

表 3 复杂环境下仿真实验结果

Tab.3 Simulation experiment results in complex environment

算法	路径长度			平均死亡蚂蚁 数量/只	$\xi/\%$	平均迭代次数	路径转折点
	最优值	平均值	标准差				
MLFVAACO	30.407 6	30.407 6	0	0.85	0.017	18.50	7
ACO	31.800 0	33.764 0	1.048 4	1 758.05	35.16	48.90	19
AIACSE	30.970 6	30.970 6	0				13
MAACO	30.970 0	30.970 0	0				13
A *	30.970 6	30.970 6	0				15
Dijkstra	30.970 6	30.970 6	0				13
Floyd	30.970 6	30.970 6	0				13

A*算法减少 53.33%,比 Dijkstra 算法和 Floyd 算法减少 46.15%,同时结合图 13a、13b,可得本文所提算法路径平滑度更好。从图 13c、13d 可看出,随着环境变得复杂,传统 ACO 算法中死亡蚂蚁数量不断增大,平均死亡蚂蚁数为 1 758.05 只,蚂蚁死亡率为 35.16%,而 MLFVAACO 算法中的死亡蚂蚁数量几乎没有增加,平均死亡蚂蚁数为 0.85 只,蚂蚁死亡率仅为 0.017%,相较于传统 ACO 算法,蚂蚁利用率提升 99.95%。从图 13e 可看出,本文所提算法的整体搜索速度均快于传统 ACO 算法,平均搜索到最优解迭代次数为 18.50,相较于传统 ACO 算法的 48.90,搜索效率提升 62.17%。

4 结论

(1)提出了一种多级视野自适应蚁群算法来克服传统蚁群算法在应用于移动机器人路径规划时存在蚂蚁死亡率高,搜索解的质量差以及全局最优解搜索能力弱等缺点。首先通过扩大蚂蚁的视野范围来减少路径上的转折点,其次设计了一种自适应全

局初始信息素更新策略来避免蚂蚁在算法迭代初期出现盲目搜索现象,然后提出了一种随机过往路径节点策略对死锁蚂蚁进行优化以提高蚁群的利用率和增加搜索解的多样性,最后引入了一种自适应的伪随机状态转移策略来避免算法过早收敛于局部最优解。

(2)仿真结果表明,与传统 ACO 相比,MLFVAACO 算法最优路径缩短 12.74%,路径转折点减少 63.16%,蚂蚁利用率提升 99.99%,搜索效率提升 62.17%;与现有文献中的改进 ACO 相比,MLFVAACO 算法最优路径缩短 1.82%,平均路径缩短 2.77%,路径转折点减少 46.15%;与图搜索算法相比,MLFVAACO 算法最优路径缩短 7.19%,平均路径缩短 7.19%,路径转折点减少 53.33%。

(3)不同环境中,MLFVAACO 算法在寻优能力、搜索解的质量、路径平滑度和蚂蚁利用率等方面均表现出出色的综合性能,并且 MLFVAACO 算法在复杂环境中的适应性更强。

参 考 文 献

[1] BERNARDO R, SOUSA J M C, GONÇALVES P J S, et al. Survey on robotic systems for internal logistics[J]. Journal of Manufacturing Systems, 2022,65:339 – 350.

[2] BAI Y, LV Y, ZHANG J,et al. Smart mobile robot fleet management based on hierarchical multi-agent deep Q network towards intelligent manufacturing[J]. Engineering Applications of Artificial Intelligence,2023,124:106534.

[3] ISLAM N, HASEEB K, ALMOGREN A, et al. A framework for topological based map building: a solution to autonomous robot navigation in smart cities[J]. Future Generation Computer Systems, 2020,111:644 – 653.

[4] FAN Y, HU Y, JIANG L, et al. Intelligent disinfection robots assist medical institutions in controlling environmental surface disinfection[J]. Intelligent Medicine, 2021,1(1):19 – 23.

[5] 郑路,张啸,王建国,等. 基于宏微结合的田间作业机器人路径规划[J]. 农业机械学报,2023,54(9):13 – 26.

ZHENG Lu, ZHANG Xiao, WANG Jianguo, et al. Path planning of field robot based on macro-micro combination[J]. Transactions of the Chinese Society for Agricultural Machinery, 2023,54(9):13 – 26. (in Chinese)

[6] LOGANATHAN A, AHMAD N S. A systematic review on recent advances in autonomous mobile robot navigation[J]. Engineering Science and Technology, 2023,40:101343.

[7] LUO M, HOU X, YANG J. Surface optimal path planning using an extended Dijkstra algorithm[J]. IEEE Access, 2020,8:147827 – 147838.

[8] 许建民,宋雷,邓冬冬,等. 基于多尺度 A* 与优化 DWA 算法融合的移动机器人路径规划[J/OL]. 系统仿真学报:1 – 12. <https://doi.org/10.16182/j.issn1004731x.joss.223-1089>. DOI:10.16182/j.issn1004731x.joss.223-1089.

XU Jianmin, SONG Lei, DENG Dongdong, et al. Path planning of mobile robot based on the integration of multi-scale A* and

- optimized dynamic-window approach algorithm[J/OL]. Journal of System Simulation; 1 – 12. <https://doi.org/10.16182/j.issn1004731x.joss.223-1089>. DOI:10.16182/j.issn1004731x.joss.223-1089. (in Chinese)
- [9] MAUROVIĆ I, SEDER M, LENAC K, et al. Path planning for active SLAM based on the D* algorithm with negative edge weights[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2017,48(8):1321 – 1331.
- [10] LYU D, CHEN Z, CAI Z, et al. Robot path planning by leveraging the graph-encoded Floyd algorithm[J]. Future Generation Computer Systems, 2021,122:204 – 208.
- [11] 徐兴,俞旭阳,赵芸,等. 基于改进遗传算法的移动机器人全局路径规划[J]. 计算机集成制造系统,2022,28(6):1659 – 1672.
- XU Xing, YU Xuyang, ZHAO Yun, et al. Global path planning of mobile robot based on improved genetic algorithm[J]. Computer Integrated Manufacturing Systems, 2022,28(6):1659 – 1672. (in Chinese)
- [12] 李薪颖,单梁,常路,等. 复杂环境下基于多目标粒子群的 DWA 路径规划算法[J]. 国防科技大学学报,2022,44(4):52 – 59.
- LI Xinying, SHAN Liang, CHANG Lu, et al. DWA path planning algorithm based on multi-objective particle swarm optimization in complex environment[J]. Journal of National University of Defense Technology, 2022,44(4):52 – 59. (in Chinese)
- [13] 冯振,肖人彬. 基于混合反馈机制的扩展蚁群算法[J]. 控制与决策,2022,37(12):3160 – 3170.
- FENG Zhen, XIAO Renbin. Extended ant colony algorithm based on mixed feedback mechanism[J]. Control and Decision, 2022,37(12):3160 – 3170. (in Chinese)
- [14] 郝琨,张慧杰,李志圣,等. 基于改进避障策略和双优化蚁群算法的机器人路径规划[J]. 农业机械学报,2022,53(8):303 – 312,422.
- HAO Kun, ZHANG Huijie, LI Zhisheng, et al. Path planning of mobile robot based on improved obstacle avoidance strategy and double optimization ant colony algorithm[J]. Transactions of the Chinese Society for Agricultural Machinery, 2022,53(8):303 – 312,422. (in Chinese)
- [15] YANG Q, LIAN Y, XIE W. Hierarchical planning for multiple AGVs in warehouse based on global vision[J]. Simulation Modelling Practice and Theory, 2020,104:102124.
- [16] CHEN Z, WU H, CHEN Y, et al. Patrol robot path planning in nuclear power plant using an interval multi-objective particle swarm optimization algorithm[J]. Applied Soft Computing, 2022,116:108192.
- [17] 马小陆,梅宏. 基于改进势场蚁群算法的移动机器人全局路径规划[J]. 机械工程学报,2021,57(1):19 – 27.
- MA Xiaolu, MEI Hong. Mobile robot global path planning based on improved ant colony system algorithm with potential field[J]. Journal of Mechanical Engineering, 2021,57(1):19 – 27. (in Chinese)
- [18] 魏彤,龙琛. 基于改进遗传算法的移动机器人路径规划[J]. 北京航空航天大学学报,2020,46(4):703 – 711.
- WEI Tong, LONG Chen. Path planning for mobile robot based on improved genetic algorithm[J]. Journal of Beijing University of Aeronautics and Astronautics, 2020,46(4):703 – 711. (in Chinese)
- [19] MIAO C, CHEN G, YAN C, et al. Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm[J]. Computers & Industrial Engineering,2021,156:107230.
- [20] LIU C, WU L, XIAO W, et al. An improved heuristic mechanism ant colony optimization algorithm for solving path planning[J]. Knowledge-Based Systems, 2023,271:110540.
- [21] ZHANG S, PU J, SI Y. An adaptive improved ant colony system based on population information entropy for path planning of mobile robot[J]. IEEE Access, 2021,9:24933 – 24945.
- [22] WU L, HUANG X, CUI J, et al. Modified adaptive ant colony optimization algorithm and its application for solving path planning of mobile robot[J]. Expert Systems with Applications, 2023,215:119410.
- [23] HOU W, XIONG Z, WANG C, et al. Enhanced ant colony algorithm with communication mechanism for mobile robot path planning[J]. Robotics and Autonomous Systems, 2022,148:103949.
- [24] LUO Q, WANG H, ZHENG Y, et al. Research on path planning of mobile robot based on improved ant colony algorithm[J]. Neural Computing and Applications, 2020,32(6):1555 – 1566.
- [25] GAO W, TANG Q, YE B, et al. An enhanced heuristic ant colony optimization for mobile robot path planning[J]. Soft Computing, 2020,24(8):6139 – 6150.