doi:10.6041/j.issn.1000-1298.2022.08.032

基于改进避障策略和双优化蚁群算法的机器人路径规划

郝　琨　张慧杰　李志圣　刘永磊
(天津城建大学计算机与信息工程学院, 天津 300384)

摘要: 针对传统蚁群算法在移动机器人路径规划中存在的收敛速度慢、收敛路径质量低、死锁以及动态避障能力差的问题,本文提出基于改进避障策略和双优化蚁群算法(Double optimization ant colony algorithm,DOACO)的路径规划方法。首先,设计新的概率转移函数并对函数中的各分量权重进行自适应调整,以优化算法的收敛速度;然后,利用碰撞检测策略对路径进行再优化,进一步提高算法的性能;最后,针对常规避障策略避障能力差、实时性不足等问题,提出避障行为与局部路径重规划相结合的避障策略。实验结果表明,DOACO 算法相对于传统的蚁群算法,不仅能规划出更优的路径,收敛速度也更快,而且新的避障策略也可以有效地应对多种碰撞情况。

关键词: 路径规划; 蚁群算法; 避障策略; 路径优化; 死锁问题
中图分类号: TP18; TP242　　**文献标识码**: A　　**文章编号**: 1000-1298(2022)08-0303-10　　**OSID**:

Path Planning of Mobile Robot Based on Improved Obstacle Avoidance Strategy and Double Optimization Ant Colony Algorithm

HAO Kun　ZHANG Huijie　LI Zhisheng　LIU Yonglei
(*School of Computer and Information Engineering*, *Tianjin Chengjian University*, *Tianjin* 300384, *China*)

Abstract: In order to solve the problems of traditional ant colony algorithm in mobile robot path planning, such as low convergence speed, low quality of convergence path, deadlock, and poor dynamic obstacle avoidance capability, a path planning method based on improved obstacle avoidance strategy and double optimization ant colony algorithm (DOACO) was proposed. Firstly, a probability transfer method was designed. The pseudo-random probability adjustment factor was introduced to adjust the selection degree of high-quality path points in the probability transfer function. It avoided the problem that the probability of selecting high-quality path points in traditional ant colony algorithm was too low. Secondly, the weight of each component of the probability transfer function was adaptively adjusted to optimize the convergence speed of the algorithm. Then the elite saving strategy was introduced to prevent the data falling back of the algorithm. The elite saving strategy can also improve the quality of the path. In order to further improve the quality of path, a path optimization strategy was proposed based on key path points. This strategy tried to generate better path segments by looking for key path points. Finally, a obstacle avoidance strategy based on obstacle avoidance behavior and local path replanning was proposed to solve the problems of poor obstacle avoidance ability and lack of real-time performance. The experimental results showed that compared with the traditional ant colony algorithm, DOACO algorithm can not only plan a better path, but also can have a faster convergence speed, and the obstacle avoidance strategy can effectively deal with a variety of collision situations.

Key words: path planning; ant colony algorithm; obstacle avoidance strategy; path optimization; deadlock problem

0　引言

随着科技的发展,移动机器人已经被广泛应用于地面自主导航[1]、水下环境勘探[2]、应急信息采集[3]等领域中。路径规划技术是实现移动机器人智能化的关键技术之一,它是指移动机器人根据一

收稿日期: 2021-08-10　修回日期: 2021-10-03
基金项目: 国家自然科学基金项目(61902273)
作者简介: 郝琨(1979—),女,副教授,博士,主要从事水下传感器网络和网络性能优化研究,E-mail: kunhao@tcu.edu.cn

种或多种性能指标,在复杂的空间环境中,寻找一条从起点到终点且没有碰撞的最优或次优路径[4-5]。目前移动机器人路径规划的主流算法分为两大类:传统算法与仿生学智能算法。其中,传统算法主要有 A* 算法、人工势场法等。智能算法主要有遗传算法、蚁群算法、粒子群算法、免疫算法等[6-8]。传统算法在简单的地图环境中性能较好,但不适用于复杂的地图环境中。而智能仿生学算法则存在过早收敛、易陷入局部极值等问题[9]。

蚁群算法具有良好的并行性和鲁棒性,因此被广泛地应用于路径规划领域[10-12]。但蚁群算法在路径规划领域也存在前期搜索盲目、收敛速度慢、容易陷入局部最优和死锁等问题。文献[13]利用人工势场法修改了蚁群算法的启发值参数,并提出了吸引素使蚁群算法更快收敛。文献[14]提出了一种基于合作博弈机制的多蚁群协同优化算法(CCACO)来加快算法的收敛速度,但该算法在大规模地图中运行效率低。文献[15]将人工势场法与蚁群算法相结合来提高算法后期的全局搜索能力。并采用基于运动路径的几何方法实现动态障碍物避障,但该算法不适合在未知环境下使用。文献[16]提出了一种改进蚁群算法(Improved ant colony algorithm,IACO)。该算法通过使初始信息素不均匀分布避免算法的早期盲目搜索,加快算法收敛速度,并利用动态惩罚方法减少蚂蚁陷入死锁的数量,但是该方法不能有效解决深度死锁问题。文献[17]提出了一种改进双层蚁群算法,将蚁群划分为引导层蚁群和普通层蚁群以应对不同的情况。然而,该算法将引导层蚁群与普通蚁群并行进行使引导层蚁群并未起到相应作用。

针对蚁群算法在路径规划中存在收敛速度慢、收敛路径质量低、死锁以及动态避障能力差的问题,本文提出基于改进避障策略和双优化蚁群算法(Double optimization ant colony algorithm,DOACO)的路径规划方法,该算法通过改进概率转移函数解决蚁群算法收敛速度慢的问题;通过回溯机制结合路径长度清零机制解决死锁问题;并通过路径优化与精英保存策略进一步提高收敛路径的质量;同时针对动态环境中的避障问题,提出一种避障行为与局部路径重规划相结合的避障策略。

1 空间环境模型

采用栅格法[18]建立空间环境模型,整个二维平面被划分为 20×20 大小一致的栅格,如图 1 所示。其中,白色栅格表示机器人可活动的自由区域,黑色栅格表示静态障碍物,红色栅格表示动态障碍物。

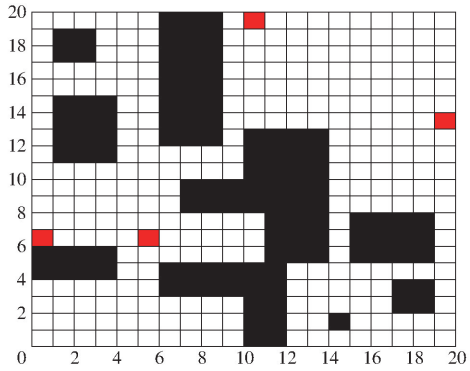


图 1 20×20 栅格地图

Fig. 1 20×20 grid map

黑色栅格和红色栅格所在区域均为机器人不可活动区域(即障碍物区域)。地图从左到右、从上到下由 1 开始为栅格添加编号直到添加至右下角的第 400 号栅格。环境地图中各栅格的坐标由其中心点坐标 (x,y) 表示,栅格编号与栅格坐标之间的转换式为

$$x=\begin{cases}\mathrm{mod}(m,n)-0.5 & (\mathrm{mod}(m,n)\neq 0)\\ n-0.5 & (\mathrm{mod}(m,n)=0)\end{cases} \tag{1}$$

$$y=n+0.5-\mathrm{ceil}(m/n) \tag{2}$$

其中

$$m=\mathrm{floor}(l-y)n+\mathrm{ceil}(x) \tag{3}$$

式中 m——栅格编号

n——栅格总列数

l——栅格总行数

(x,y)——栅格的横、纵坐标

式(1)、(2)把栅格编号转换成栅格坐标。式(3)把栅格坐标转换成栅格编号。mod( )为取余函数,floor( )为向下取整函数,ceil( )为向上取整函数。

2 DOACO 算法设计

2.1 DOACO 算法

DOACO 算法参数包括:起点栅格编号 S,目标点栅格编号 G,最大迭代次数 $N_{c\max}$,当前迭代次数 N_c,蚁群中蚂蚁的总数量 K,当前蚂蚁数量 k,信息素常量 Q,信息素因子 α,启发式因子 β,信息素挥发因子 ρ,伪随机概率调整因子 q_0。图 2 为 DOACO 算法流程图,算法流程如下:①对蚁群算法的各参数进行初始化。②令 $N_c=1$。③令 $k=1$。④将第 k 只蚂蚁放置在起点位置。⑤根据改进的概率转移公式选择下一个路径点。⑥判定蚂蚁是否陷入死锁状态,若陷入死锁,则采用死锁处理策略使得蚂蚁跳离死锁,然后转入步骤⑤;若未陷入死锁,则更新第 k 只蚂蚁的禁忌表,然后转入步骤⑦。⑦判断蚂蚁是否到达目标点,若未到达,转至步骤⑤;若已到达目标点,则转到步骤⑧。⑧判断该蚂蚁是否为当前迭代次数中的最后一只蚂蚁,若不是最后一只蚂蚁,则令 $k=$

$k+1$，然后返回步骤④；若是最后一只蚂蚁，则更新信息素并判断 $N_c > N_{cmax}$ 是否成立，若不成立则转入步骤⑨，否则转入步骤⑩。⑨$N_c = N_c + 1$，同时返回步骤③。⑩输出收敛路径，采用路径优化策略，得到最优路径，流程结束。

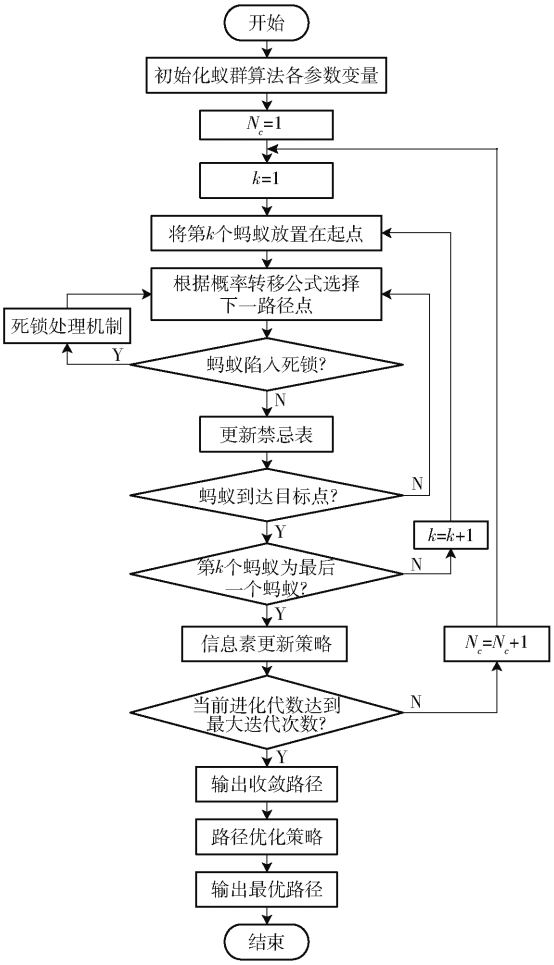


图 2　DOACO 算法流程图

Fig. 2　Flow chart of DOACO algorithm

2.2　伪随机概率转移

本文设计一种新的概率转移方法，通过引入伪随机概率调整因子 q_0 来调整概率转移函数中对优质路径点的选择程度（式(4)），避免传统蚁群算法中选择优质路径点的概率过低这一问题（式(5)）。

$$w = \begin{cases} \underset{s \in a_k}{\operatorname{argmax}}(\tau_{is}^{\alpha}\eta_{is}^{\beta}) & (q \leqslant q_0) \\ \text{Roulette}(p_{ij}^{k}) & (q > q_0) \end{cases} \tag{4}$$

$$p_{ij}^{k} = \begin{cases} \dfrac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum\limits_{s \in a_k}\tau_{is}^{\alpha}\eta_{is}^{\beta}} & (j \in a_k) \\ 0 & (\text{其他}) \end{cases} \tag{5}$$

式中　p_{ij}^{k}——第 k 只蚂蚁在路径点 i 和路径点 j 之间的转移概率

τ_{ij}——路径点 i 和路径点 j 之间的信息素浓度

η_{ij}——当前路径点 i 到路径点 j 的启发式信息

a_k——当前路径点 i 的下一个可行路径点的集合

w——通过伪随机概率转移函数所获得的路径点

$\underset{s \in a_k}{\operatorname{argmax}}(\tau_{is}^{\alpha}\eta_{is}^{\beta})$——选取最优邻居路径点的函数

Roulette( )——利用轮盘赌策略选取路径点的函数

q——随机数

q_0 通常为固定常数，当 $q \leqslant q_0$ 时，保留最优邻居路径点。当 $q > q_0$ 时，利用传统概率转移方法和轮盘赌策略获得路径点。

在传统概率转移函数中，假设某优质路径点被选择的概率为 p，此时引入伪随机转移概率因子 q_0，根据式(5)，设伪随机概率转移函数对该优质路径点的选择概率为 p_1，则 $p_1 = q_0 + (1 - q_0)p$，而 $p_1 - p = q_0 - pq_0 = q_0(1 - p) \geqslant 0$，故 $p_1 \geqslant p$，即引入伪随机概率调整因子后，新函数中优质路径点被选择的程度大于传统概率转移函数。在已知 p 的情况下，新函数对优质路径点的选择程度取决于 q_0，因此，通过引入伪随机概率转移因子 q_0 来调整优质路径点被选择的概率可行。

传统启发式信息函数只考虑了当前节点到下一节点的距离，且数值固定为 1 或 $1/\sqrt{2}$，引导作用较弱，计算式为

$$\eta_{ij} = \frac{1}{d_{ij}} \tag{6}$$

式中　d_{ij}——路径点 i 和路径点 j 之间的距离

针对传统启发式信息的问题，本文设计了新的启发式信息函数，充分考虑下一节点到目标点的关系，强化启发式信息的引导作用，计算式为

$$\eta_{ij} = \frac{1}{d_{jg}} \tag{7}$$

式中　d_{jg}——路径点 j 和目标点 g 之间的距离

在迭代前期，信息素对蚁群的引导作用较弱，启发式信息应发挥主导作用。此时信息素因子应较小，启发式因子应较大，强化启发式信息的主导作用。随着迭代次数的增加，各路径段上的信息素差异逐渐增大，信息素逐渐发挥主导作用。此时信息素因子逐渐增大，启发式因子逐渐减小，这样可以强化信息素的主导作用。基于上述分析，本文设计了自适应信息素因子 α_1、自适应启发式因子 β_1，以加快算法的收敛速度，计算式为

$$\alpha_1 = [(N_{cmax} + N_c)/N_{cmax}]\alpha \tag{8}$$

$$\beta_1 = \begin{cases} [(N_{cmax} - N_c)/N_{cmax}]\beta & \left(N_c < \dfrac{N_{cmax}}{2}\right) \\ 0.5\beta & \left(N_c \geqslant \dfrac{N_{cmax}}{2}\right) \end{cases} \tag{9}$$

2.3 信息素初始化及更新

信息素初始化主要有两种方式:均匀分布和不均匀分布。初始信息素的不均匀分布[19-20]使算法有倾向性的快速收敛于某一条路径,当应对复杂地图环境时极易导致算法陷入局部最优解。而信息素均匀分布可以扩大算法对解空间的搜索范围,增强算法的种群多样性,降低算法陷入局部最优解的概率。因此本文采用初始信息素均匀分布的方法。

在信息素更新方面,DOACO 仅考虑对每条路径上的信息素进行单次更新,不考虑二次更新[21-22]。信息素更新过程为

$$\tau_{ij}(t+1)=(1-\rho)\tau_{ij}(t)+\Delta\tau_{ij}\quad(\rho\in(0,1)) \tag{10}$$

其中

$$\Delta\tau_{ij}=\sum_{k=1}^{K}\Delta\tau_{ij}^{k} \tag{11}$$

$$\Delta\tau_{ij}^{k}=\begin{cases}\dfrac{Q}{L_k} & (\text{蚂蚁 } k \text{ 经过路径段}(i,j))\\ 0 & (\text{蚂蚁 } k \text{ 未经过路径段}(i,j))\end{cases} \tag{12}$$

式中 $\Delta\tau_{ij}$——路径段(i,j)上的信息素增量

$\Delta\tau_{ij}^{k}$——第 k 只蚂蚁在路径段(i,j)上的信息素增量

L_k——第 k 只蚂蚁所经过的路径长度

2.4 死锁问题处理策略

采取回溯机制和路径长度清零机制相结合的方法来解决蚁群算法的死锁问题[23]。假设当前蚂蚁行进到了第 n 个路径点且蚂蚁在第 n 个路径点处陷入死锁状态。则执行该方法流程如下:①令 $i=n$。②将第 i 个路径点和第 $i-1$ 个路径点之间的路径标记为不可行状态(即路径长度清零机制)。③蚂蚁返回到第 $i-1$ 个路径点(即回溯机制),即 $i=i-1$。④判断蚂蚁当前所在路径点是否陷入死锁现象。如果蚂蚁当前所在路径点陷入了死锁现象,则转入步骤②。如果蚂蚁当前所在路径点没有陷入死锁现象,则流程结束。蚂蚁根据概率转移函数选择下一个路径点。

该方法通过回溯并将路径长度清零重新调整蚂蚁寻路方向保证蚂蚁百分之百的存活率,提高蚂蚁对解空间的探索能力。

2.5 路径优化策略

DOACO 算法在收敛路径的基础上对收敛路径进行了二次优化,从而进一步提高收敛路径的质量,路径优化策略如下:

(1)定义关键路径点的概念并找到一条收敛路径中关键路径点的集合。关键路径点是指可以支撑起一条路径中某一个路径段的核心路径点。包括:起点、目标点、特殊的路径点(多为路径转折处及障碍物附近的路径点)。

(2)利用启发式策略设计连接操作符。启发式策略是寻找当前路径点附近的距离目标点最近的相邻路径点。连接操作符是指利用启发式策略生成两个路径点之间的路径段的操作。

(3)利用连接操作符生成两个相邻关键路径点之间的新的路径段。

(4)若两个关键路径点之间的新的路径段比之前的路径段好,则用新的路径段取代旧的路径段。若两个关键路径点之间新的路径段比之前的路径段差,则不作改变。

假设某条路径中的第 i 个路径点是关键路径点,则该条路径中的下一个关键路径点的寻找方法如下:①令 $j=i$,其中 j 为指针型变量,用作寻找关键路径点。②令 $j=j+1$。③如果第 j 个路径点是终点,则第 j 个路径点就是下一个关键路径点,流程结束。否则,转入步骤④。④如果第 i 个路径点到第 j 个路径点之间的连线不经过障碍物栅格且第 i 个路径点到第 $j+1$ 个路径点之间的连线经过障碍物栅格,则第 j 个路径点就是下一个关键路径点,流程结束。否则,转入步骤②。

本文使用二维栅格地图中的碰撞检测模型[24]来检测两个路径点之间的连线是否与障碍物栅格发生碰撞。

3 动态避障

针对传统避障策略存在的避障能力较差且实时性不足的问题[25-26],本文利用避障行为中的等待策略,并结合局部路径重规划的方法来进行动态避障。局部路径重规划方法是指当预测到移动机器人和动态障碍物即将碰撞时,利用启发式策略重新生成碰撞处的局部路径段的方法,该方法仅对全局路径进行小幅度调整。因此局部路径重规划方法具有避障代价小、避障实时性强、路径质量高的优点。

3.1 碰撞类型

按照不同的碰撞方式将碰撞类型分为正面碰撞、侧面碰撞以及障碍物停留在全局路径上的静态碰撞。其中,正面碰撞又分为接触式正面碰撞和非接触式正面碰撞。正面碰撞是指移动机器人与动态障碍物因运动方向共线且相反而产生的面对面碰撞的情况。假设移动机器人当前位置为 $r_i(r_{i_x},r_{i_y})$,移动机器人下一个位置为 $r_{i+1}(r_{i+1_x},r_{i+1_y})$。动态障碍物当前位置为 $o_i(o_{i_x},o_{i_y})$,动态障碍物下一个位置为 $o_{i+1}(o_{i+1_x},o_{i+1_y})$。

若移动机器人与动态障碍物发生正面碰撞时有公共碰撞点,则这种碰撞情况被称为非接触式正面

碰撞(图 3a)。即

$$\begin{cases} r_{i+1_x} = = o_{i+1_x} \\ r_{i+1_y} = = o_{i+1_y} \\ \dfrac{r_{i_x} + o_{i_x}}{2} = = r_{i+1_x} \\ \dfrac{r_{i_y} + o_{i_y}}{2} = = r_{i+1_y} \end{cases} \tag{13}$$

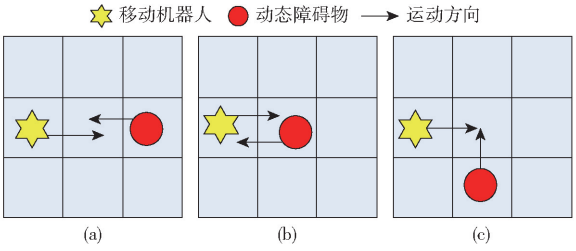


图 3　碰撞类型示意图

Fig. 3　Schematic of collision types

若移动机器人与动态障碍物发生正面碰撞时没有公共碰撞点,即移动机器人与动态障碍物的下一步互为行进栅格。在这种情况下虽无碰撞点但两者也会发生碰撞,这时判断这种碰撞情况为接触式正面碰撞(图 3b)。即

$$\begin{cases} r_{i_x} = = o_{i+1_x} \\ r_{i_y} = = o_{i+1_y} \\ r_{i+1_x} = = o_{i_x} \\ r_{i+1_y} = = o_{i_y} \end{cases} \tag{14}$$

侧面碰撞是指移动机器人与动态障碍物运动方向不共线,但在某一时刻两者恰好产生公共碰撞点的情况(图 3c)。即

$$\begin{cases} r_{i+1_x} = = o_{i+1_x} \\ r_{i+1_y} = = o_{i+1_y} \end{cases} \tag{15}$$

障碍物停留在全局路径上的静态碰撞是指动态障碍物由于某种原因恰好停留在了全局路径上,导致移动机器人与障碍物必定会发生碰撞的情况。即

$$\begin{cases} r_{i+1_x} = = o_{i+1_x} \\ r_{i+1_y} = = o_{i+1_y} \\ o_{i_x} = = o_{i+1_x} \\ o_{i_y} = = o_{i+1_y} \end{cases} \tag{16}$$

3.2　避障策略

本文提出的动态避障策略包含两种避障策略:原地等待策略和局部路径重规划策略。针对侧面碰撞,采取等待策略来避免碰撞的发生。即在侧面碰撞发生前,移动机器人原地暂停,等待动态障碍物通过后移动机器人再继续前进。

针对接触式正面碰撞、非接触式正面碰撞及障碍物停留在全局路径上的静态碰撞,采取局部路径重规划策略来避免碰撞的发生。假设移动机器人位于全局路径的第 i 个路径点,则把预测的公共碰撞点视为静态障碍物并利用启发式策略重新规划第 i 个路径点到第 $i+2$ 个路径点之间的局部路径。当移动机器人沿着局部路径避过障碍物后,移动机器人将会重新返回全局路径。

4　仿真实验

4.1　仿真设计

为了验证 DOACO 算法的性能,本文将 DOACO 算法、ACO 算法、IACO 算法[16]分别运行在人工仿真的 20×20 栅格地图与自然环境仿真图书馆的 50×50 栅格地图下进行比较。20×20 栅格地图环境为人为设置,而 50×50 栅格地图以真实图书馆环境为原型设置。评价指标包括:路径生成、路径长度、算法收敛所需的迭代次数、算法收敛所需的程序运行时间、蚂蚁存活率等。另外,通过在上述自然仿真的 50×50 地图中增加不同时间、不同方向出发的动态障碍物模拟图书馆中的学生来验证动态避障策略的有效性。

4.2　人工仿真实验

人工仿真实验使用 20×20 的栅格地图(图 1),单位栅格长度设为 1 m,移动机器人从 1 号栅格进入,从 400 号栅格离开。蚁群算法初始参数如表 1 所示。

表 1　蚁群算法初始参数

Tab. 1　Parameters of ant colony algorithm

参数	ACO 算法	IACO 算法	DOACO 算法
初始信息素浓度	1	不均匀分布	1
蚂蚁数量	50	50	50
最大迭代次数	200	200	200
信息素因子	1	1	1
启发式信息因子	7	7	7
信息素挥发因子	0.3	0.3	0.3
信息素常量	1	1	1
伪随机概率调整因子		自适应值	0.3
死锁惩罚因子		0.2	

4.2.1　路径生成

图 4 为 ACO 算法和 IACO 算法的机器人运动轨迹图。从图 4a 可看出,ACO 算法生成路径的质量明显较差,而 IACO 算法生成的路径(图 4b)质量有了一定的提高。图 5 为未添加优化策略的 DOACO 算法和添加优化策略的 DOACO 算法的机器人运动轨迹图。从图 5a 可看出,未添加优化策略的 DOACO 算法由于改进了启发信息等以及使用精英保存策略,生成路径的质量明显提高,但是该路径仍

有进一步优化的空间。从图 5b 可看出,添加优化策略的 DOACO 算法生成路径的质量最好。实验结果表明,DOACO 算法生成的路径明显优于 ACO 算法和 IACO 算法。

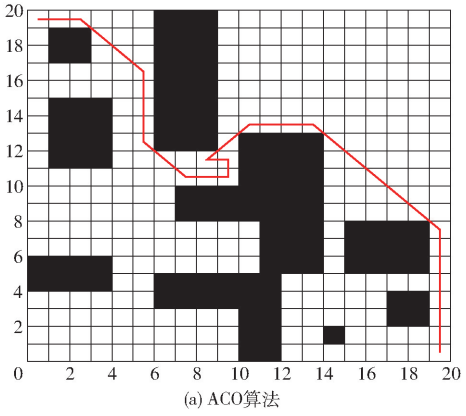


图 4 机器人运动轨迹图

Fig. 4 Robot motion trajectory diagrams

4.2.2 收敛曲线分析

图 6 为 3 种算法的收敛曲线。从图 6 可看出 ACO 算法在 108 代收敛,收敛后的最优路径长度为 37.8 m。IACO 算法在 41 代收敛,收敛后的最优路径长度为 37.56 m,该算法采用改进信息素的初始分布,因此收敛次数减少。而 DOACO 算法在 15 代就已经收敛,收敛后的最优路径长度为 34.38 m。从收敛路径长度和收敛迭代次数来看,DOACO 算法优于 ACO 算法和 IACO 算法,这主要是由于 DOACO 采用了路径优化策略以及改进的概率转移函数。此外,从图 6 中还可以发现,ACO 算法和 IACO 算法均存在数据回落现象,而 DOACO 算法由于采用了精英保存策略,有效避免了数据回落现象。

4.2.3 蚂蚁存活率分析

3 种算法的蚂蚁存活率变化曲线如图 7 所示。从图 7 可以看出,在 100 代之前,随着迭代次数的增加,ACO 算法的蚂蚁存活率呈逐渐上升趋势。在 100 代之后,ACO 算法的蚂蚁存活率会有波动,但是总体上稳定。IACO 算法则是在 93 代之前,随着迭代次数的增加,蚂蚁的存活率逐渐上升。到了 93 代

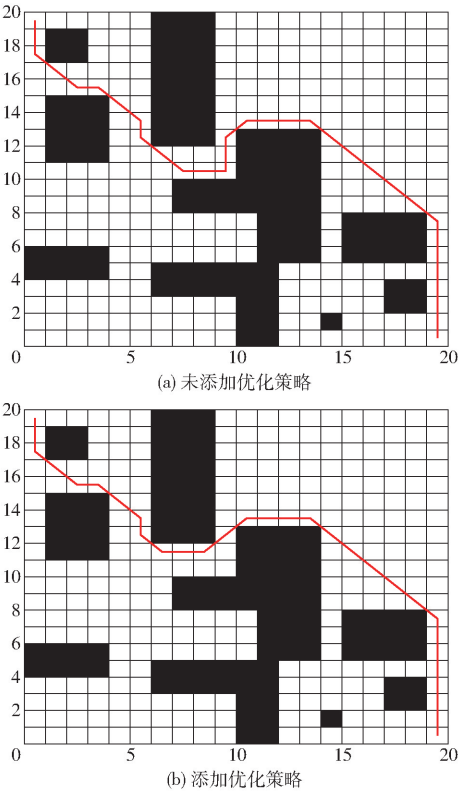


图 5 DOACO 算法的机器人运动轨迹图

Fig. 5 Robot motion trajectory diagrams of DOACO algorithm

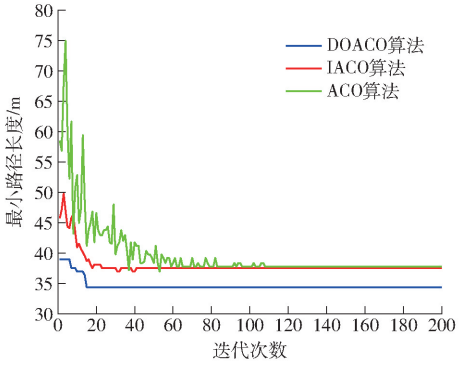


图 6 3 种算法收敛曲线

Fig. 6 Comparison of convergence curves of three algorithms

之后,蚂蚁存活率达到 1。这说明 IACO 算法的死锁惩罚因子起到了一定的效果。但是在算法迭代初期的时候,蚂蚁存活率依旧较低。而 DOACO 算法通过回溯机制保证了蚂蚁可以存活,路径长度清零机制引导蚂蚁调整前进方向跳离死锁,使蚂蚁存活率始终保持为 1,从而有效解决死锁问题。

4.2.4 综合对比

在 20×20 栅格地图的仿真环境下,本文将每种算法各仿真 20 次取平均值,结果如表 2 所示。从表 2 可以看出,DOACO 算法在平均路径长度、平均收敛迭代次数、平均算法收敛时间这 3 个性能指标上均优于 ACO 算法和 IACO 算法。算法收敛时间

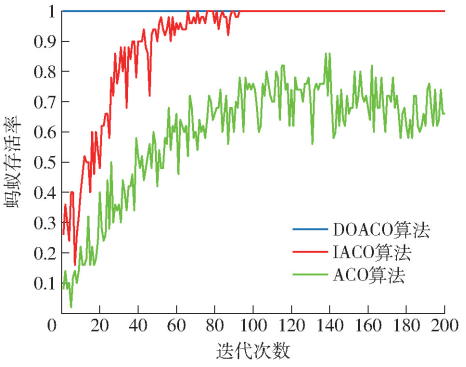


图 7　3 种算法蚂蚁存活率变化曲线

Fig. 7　Comparison of ant survival rate changes of three algorithms

表 2　3 种算法仿真结果

Tab. 2　Simulation results of three algorithms

算法	平均路径长度/m	平均收敛迭代次数	平均算法收敛时间/s
DOACO	34. 76	19. 35	1. 107 1
IACO	37. 77	77. 2	4. 358 7
ACO	38. 03	117. 65	6. 665 2

计算式为

$$t = \frac{N_{c1}}{N_{cmax}} T \tag{17}$$

式中　T——整个程序的运行时间

N_{c1}——算法收敛时的迭代次数

t——算法收敛时的程序运行时间(即算法收敛时间)

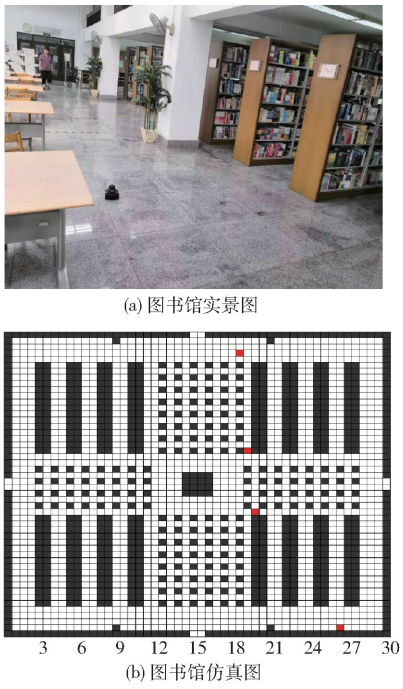
本文将算法收敛时间作为衡量蚁群算法的时间尺度。如表 2 所示,在路径长度方面,DOACO 算法得到的路径长度约为 IACO 算法的 92. 03%,ACO 算法的 91. 40%,对路径长度有一定的缩减。在收敛迭代次数方面,DOACO 算法收敛所用的迭代次数约为 IACO 算法的 25. 06%,ACO 算法的 16. 45%,收敛次数大幅度减少,同时也使得平均算法收敛时间大大缩短,充分体现了 DOACO 算法的优越性。

4. 3　自然环境仿真实验

自然仿真环境参照我校图书馆,实景图如图 8a 所示,图书馆面积为 30 m×30 m,将整个地图划分为 50×50 栅格模型,每个栅格的实际面积为 0. 6 m×0. 6 m。图 8b 为参考实际场景建立的地图模型,其中黑色障碍物代表图书馆中的墙壁、书架和桌子等不可行区域,红色障碍物代表图书馆中移动的行人。移动机器人从 25 号栅格进入,从 2 449 号栅格离开。蚁群算法的参数设计如表 3 所示。

4. 3. 1　路径生成

图 9 为 ACO 算法和 IACO 算法在图书馆自然仿真环境下的机器人运动轨迹图。从图 9a 可以看出,



(a) 图书馆实景图

(b) 图书馆仿真图

图 8　图书馆实景与仿真环境

Fig. 8　Library real scene and simulation environment

表 3　蚁群算法参数设计

Tab. 3　Parameters of ant colony algorithm

参数	ACO 算法	IACO 算法	DOACO 算法
初始信息素浓度	1	不均匀分布	1
蚂蚁数量	50	50	50
最大迭代次数	500	500	500
信息素因子	1	1	1
启发式信息因子	7	7	7
信息素挥发因子	0. 3	0. 3	0. 3
信息素常量	1	1	1
伪随机概率调整因子		自适应值	0. 3
死锁惩罚因子		0. 2	

ACO 算法生成路径的质量明显较差。IACO 算法生成路径(图 9b)的质量甚至不如 ACO 算法。这充分说明在路径生成方面 IACO 算法并不适用于图书馆这类大规模多障碍物地图。图 10 为未添加优化策略的 DOACO 算法和添加优化策略的 DOACO 算法在图书馆地图下的机器人运动轨迹图。从图 10a 可以看出,未添加优化策略的 DOACO 算法生成路径的质量明显更好些,但是该路径仍有进一步优化的空间。从图 10b 可以看出,添加优化策略的 DOACO 算法生成路径的质量最优。实验结果所示,DOACO 算法生成的路径明显优于 ACO 算法和 IACO 算法。

4. 3. 2　收敛曲线分析

图 11 为 3 种算法收敛曲线。从图 11 可以看出,ACO 算法在 461 代收敛,收敛后的最优路径长度为 39. 71 m。IACO 算法在 43 代收敛,收敛后的最优路径长度为 43. 52 m。该算法主要通过改进信息

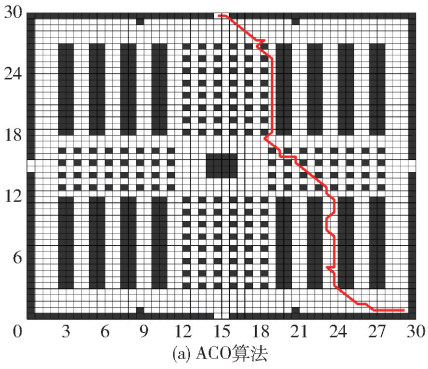


图 9 图书馆地图下的机器人运动轨迹

Fig. 9 Robot motion trajectory diagram in library map

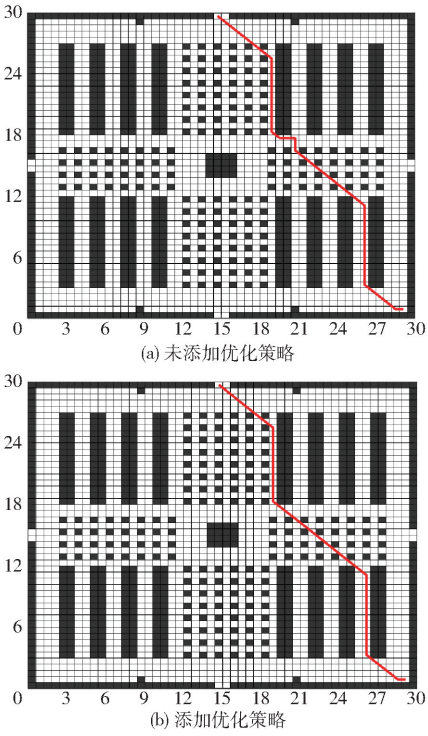


图 10 DOACO 算法在图书馆地图下的机器人运动轨迹

Fig. 10 Robot motion trajectory diagram of DOACO algorithm in library map

素的初始分布使收敛次数减少,但同时也使算法更易陷入局部最优解,导致最优路径长度甚至不如 ACO 算法。而 DOACO 算法在 25 代就已经收敛,收敛后的最优路径长度为 35.12 m。从收敛路径长度和收敛迭代次数来看,在图书馆仿真场景中,

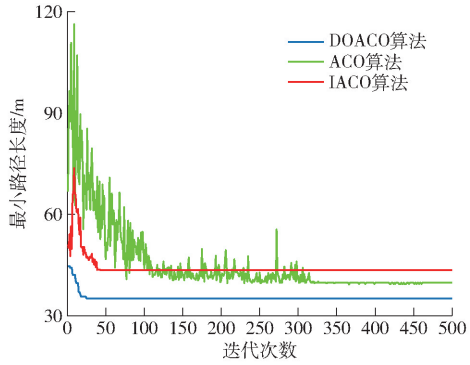


图 11 3 种算法图书馆地图收敛曲线对比

Fig. 11 Comparison of convergence curves of three algorithms in library map

DOACO 算法比 ACO 算法和 IACO 算法性能更优,充分展现了 DOACO 算法中路径优化策略以及改进的概率转移方法的科学性。此外,从图 11 中还可以发现 ACO 算法和 IACO 算法均存在数据回落现象,而 DOACO 算法由于采用了精英保存策略,有效避免了数据回落现象。

4.3.3 蚂蚁存活率分析

3 种算法在图书馆地图下的蚂蚁存活率变化曲线如图 12 所示。从图 12 可以看出,在 388 代之前,随着迭代次数的增加,ACO 算法的蚂蚁存活率呈逐渐上升趋势。在 388 代之后蚂蚁存活率达到 1 并维持稳定。文献[16]提出的 IACO 算法则是在 245 代蚂蚁存活率达到 1。这说明 IACO 算法的死锁惩罚因子起到了一定的效果。但是在算法迭代初期时,蚂蚁存活率依旧较低。而本文提出的 DOACO 算法通过回溯机制保证了蚂蚁可以存活,路径长度清零机制引导蚂蚁调整前进方向跳离死锁,即使在实际场景较为复杂的地图下也依然可以保证蚂蚁的存活率始终为 1,从而解决死锁问题。

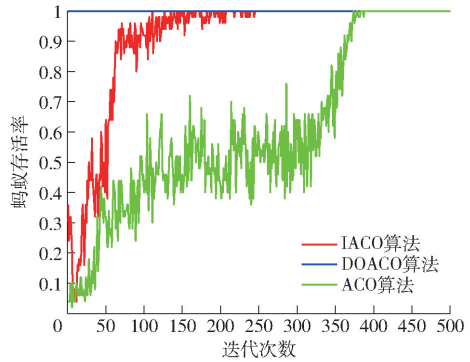


图 12 3 种算法图书馆地图蚂蚁存活率对比

Fig. 12 Comparison of ant survival rate changes of three algorithms in library map

4.3.4 综合对比

在图书馆仿真环境下,将每种算法各仿真 20 次取平均值,结果如表 4 所示。由表可知,DOACO 算

法在平均路径长度、平均收敛迭代次数、平均算法收敛时间这 3 个性能指标上均优于 ACO 算法和 IACO 算法。在路径长度方面，DOACO 算法得到的路径长度约为 IACO 算法的 81.25%，ACO 算法的 88.71%，对路径长度有了一定的缩减。在收敛迭代次数方面，DOACO 算法收敛所用的迭代次数约为 IACO 算法的 19.27%，ACO 算法的 5.23%。经实验验证，在自然环境仿真地图中 DOACO 算法的收敛次数大幅度减少，同时也使得平均算法收敛时间大大缩短，充分体现了 DOACO 算法的优越性。

表 4　3 种算法图书馆地图仿真结果

Tab. 4　Simulation results of three algorithms in library map

算法	平均路径长度/m	平均收敛迭代次数	平均算法收敛时间/s
DOACO	35.12	20.45	18.921 6
IACO	43.22	106.15	100.803 6
ACO	39.59	391.05	359.647 8

4.3.5　动态避障

在基于图书馆地图 50×50 栅格（30 m×30 m）的仿真环境中，本文在完成静态全局路径规划的基础上添加了动态障碍物。实验共设置 4 个方向不同且出发时间不同的动态障碍物表示移动的行人，以此检验本文提出的避障策略的可行性和有效性。移动机器人动态避障的路径轨迹如图 13 所示。

图 13 中动态障碍物 1、2、3 同时出发。动态障碍物行进速度均为每次一个单位长度。动态障碍物 1 由 181 号栅格向左行进，动态障碍物 2 由 982 号栅格向上行进，动态障碍物 3 由 1 483 号栅格向上行进。根据 3.1 节所述的碰撞类型检测机制，动态障碍物 1 在向左行进的过程中会与移动机器人发生侧面碰撞。碰撞点为 178 号栅格。根据 3.2 节的避障策略，移动机器人执行等待行为。即当移动机器人预测到碰撞后，移动机器人暂时停止行进，等待动态障碍物 1 通过后，移动机器人再继续前进（图 13a）。

动态障碍物 2、3 在移动过程中分别与移动机器人发生非接触式正面碰撞、静态碰撞，碰撞点分别为 482 号栅格和 1 033 号栅格。根据 3.2 节的避障策略，移动机器人均执行局部路径重规划策略。即当移动机器人预测到碰撞后，重新规划当前路径点到碰撞点之后的下一路径点的路径（图 13b、13c）。

动态障碍物 4 在移动机器人行进 35 步后开始运动，由 2 444 号栅格向上行进。当移动机器人行进至 1 944 号栅格时刚好与动态障碍物 4 正面相遇。根据 3.1 节所述的碰撞类型检测机制可以判断两者为接触式正面碰撞，故无碰撞点。根据 3.2 节的避障策略，移动机器人采用局部路径重新规划来避开障碍物。即移动机器人重新规划 1 944 号栅格到 2 044 号栅格之间的路径。移动机器人再按照重新规划的局部路径继续前进。避障完成后，移动机器人的运动轨迹如图 13d 所示。

为了测试本文提出的避障策略的实时性，仿真

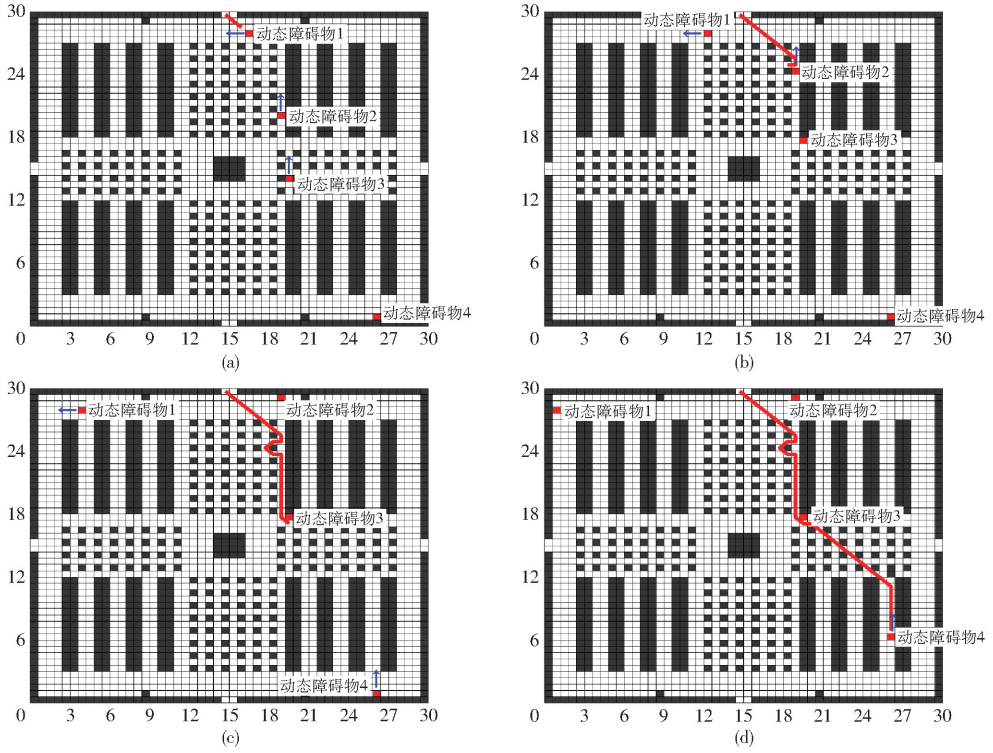


图 13　图书馆仿真环境的动态避障轨迹图

Fig. 13　Dynamic obstacle avoidance trajectory diagram in library simulation environment

程序统计了 4 次避障所用时间。4 次避障时间依次为 0.000 179、0.009 581、0.000 600、0.015 197 s。而在 DOACO 算法的基础上用传统的避障策略,4 次避障的避障时间依次为 18.597 6、14.834 3、7.409 0、0.778 40 s。由此可见,本文提出的避障策略的实时性方面具有较好的性能。

5 结束语

提出了能够生成高质量全局路径的 DOACO 算法和适用于动态环境的避障策略,且适用于实际环境。首先,提出一种新的概率转移函数优化算法的收敛速度,解决蚁群算法收敛速度慢的问题。其次,采用精英保存策略并提出基于碰撞检测的路径优化策略对收敛路径再优化,提高收敛路径的质量。此外,本文采用回溯机制和路径长度清零机制相结合这一策略来解决蚁群算法的死锁问题。最后,设计一种新的避障策略,移动机器人根据不同的碰撞类型采用不同的避障策略来躲避动态障碍物。新的避障策略可以有效地解决常规避障策略实时性不足等问题。仿真结果表明,DOACO 算法在静态和动态环境中均能生成可行有效的高质量路径,同时也能解决基本蚁群算法存在的问题。新的避障策略可以有效地应对各种潜在的碰撞情况并且避障实时性强。

参 考 文 献

[1] ORT T, GILITSCHENSKI I, RUS D. Autonomous navigation in inclement weather based on a localizing ground penetrating radar[J]. IEEE Robotics and Automation Letters, 2020, 5(2): 3267 - 3274.

[2] NAM H. Data-gathering protocol-based AUV path-planning for long-duration cooperation in underwater acoustic sensor networks [J]. IEEE Sensors Journal, 2018, 18(21): 8902 - 8912.

[3] HUANG Z, CHEN C, PAN M. Multiobjective UAV path planning for emergency information collection and transmission[J]. IEEE Internet of Things Journal, 2020, 7(8): 6993 - 7009.

[4] SUDHAKARA P, GANAPATHY V, PRIYADHARSHINI B, et al. Obstacle avoidance and navigation planning of a wheeled mobile robot using amended artificial potential field method[J]. Procedia Computer Science, 2018, 133: 998 - 1004.

[5] JIHEE H, YOONHO S. Mobile robot path planning with surrounding point set and path improvement[J]. Applied Soft Computing, 2017, 57: 35 - 47.

[6] GONZALEZ D, PEREZ J, MILANES V, et al. A review of motion planning techniques for automated vehicles[J]. IEEE Transactions on Intelligent Transportation Systems, 2016, 17(4): 1135 - 1145.

[7] 刘可,李可,宿磊,等. 基于蚁群算法与参数迁移的机器人三维路径规划方法[J]. 农业机械学报,2020,51(1):29 - 36.
LIU Ke, LI Ke, SU Lei, et al. Robot 3D path planning method based on ant colony algorithm and parameter transfer[J]. Transactions of the Chinese Society for Agricultural Machinery, 2020, 51(1): 29 - 36. (in Chinese)

[8] GUO Y, MENG Q, KONG F, et al. Research status and prospect of AUV path planning algorithms[J]. Journal of Frontiers of Computer Science and Technology, 2020, 14(12): 1981 - 1994.

[9] 刘佳,王杰. 无人水面艇避障路径规划算法综述[J]. 计算机应用与软件,2020,37(8):1 - 10,20.
LIU Jia, WANG Jie. Overview of obstacle avoidance path planning algorithm for unmanned surface vehicle[J]. Computer Applications and Software, 2020, 37(8): 1 - 10, 20. (in Chinese)

[10] YU X, CHEN W, HU X, et al. Path planning in multiple-AUV systems for difficult target traveling missions: a hybrid metaheuristic approach[J]. IEEE Transactions on Cognitive and Developmental Systems, 2020, 12(3): 561 - 574.

[11] SONG Q, ZHAO Q, WANG S, et al. Dynamic path planning for unmanned vehicles based on fuzzy logic and improved ant colony optimization[J]. IEEE Access, 2020, 8: 62107 - 62115.

[12] YU X, CHEN W, GU T, et al. ACO - A*: ant colony optimization plus A* for 3 - D traveling in environments with dense obstacles[J]. IEEE Transactions on Evolutionary Computation, 2019, 23(4): 617 - 631.

[13] 王刚,张方,严大亮,等. 基于改进蚁群算法的机器人三维路径规划[J]. 国外电子测量技术,2020,39(11):1 - 6.
WANG Gang, ZHANG Fang, YAN Daliang, et al. Robot 3D path planning based on improved ant colony algorithm[J]. Foreign Electronic Measurement Technology, 2020, 39(11): 1 - 6. (in Chinese)

[14] MENG L, YOU X, LIU S. Multi-colony collaborative ant optimization algorithm based on cooperative game mechanism[J]. IEEE Access, 2020, 8: 154153 - 154165.

[15] ZHU S, ZHU W, ZHANG X, et al. Path planning of lunar robot based on dynamic adaptive ant colony algorithm and obstacle avoidance[J]. International Journal of Advanced Robotic Systems, 2020, 17(3): 1 - 14.

[16] 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.

[17] 张恒,何丽,袁亮,等. 基于改进双层蚁群算法的移动机器人路径规划[J]. 控制与决策,2022,37(2):303 - 313.
ZHANG Heng, HE Li, YUAN Liang, et al. Mobile robot path planning using improved double-layer ant colony algoritjm[J]. Control and Decision, 2022, 37(2): 303 - 313. (in Chinese)

(下转第 422 页)