

控制与决策

Control and Decision

基于进化蚁群算法的移动机器人路径优化

李涛, 赵宏生

引用本文:

李涛,赵宏生. 基于进化蚁群算法的移动机器人路径优化[J]. *控制与决策*, 2023, 38(3): 612–620.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2021.1324>

您可能感兴趣的其他文章

Articles you may be interested in

基于改进双层蚁群算法的移动机器人路径规划

Mobile robot path planning using improved double-layer ant colony algorithm

控制与决策. 2022, 37(2): 303–313 <https://doi.org/10.13195/j.kzyjc.2020.0610>

基于改进蚁群和DWA算法的机器人动态路径规划

Robot dynamic path planning based on improved ant colony and DWA algorithm

控制与决策. 2022, 37(9): 2211–2216 <https://doi.org/10.13195/j.kzyjc.2021.1804>

基于16方向24邻域改进蚁群算法的移动机器人路径规划

Mobile robots path planning based on 16-directions 24-neighborhoods improved ant colony algorithm

控制与决策. 2021, 36(5): 1137–1146 <https://doi.org/10.13195/j.kzyjc.2019.0600>

基于改进蚁群算法的水面无人艇路径规划

Path planning for unmanned surface vehicle based on improved ant colony algorithm

控制与决策. 2021, 36(4): 847–856 <https://doi.org/10.13195/j.kzyjc.2019.0839>

基于 $\text{pm}3\sigma$ 正态概率区间分族遗传蚁群算法的移动机器人路径规划

Path planning of mobile robot based on $\text{pm}3\sigma$ normal probability interval population division using genetic ant-colony algorithm

控制与决策. 2021, 36(12): 2861–2870 <https://doi.org/10.13195/j.kzyjc.2020.0745>

基于进化蚁群算法的移动机器人路径优化

李涛^{1,2†}, 赵宏生¹

(1. 南京信息工程大学 自动化学院, 南京 210044; 2. 大气环境与装备技术协同创新中心, 南京 210044)

摘要: 针对蚁群算法进行路径规划中出现的运行时间长、搜索效率低和容易出现死锁的问题, 提出一种基于达尔文进化论思想的蚁群算法. 首先, 针对空白栅格搜索效率低的问题, 提出一种蚁群算法简易模式; 然后在启发函数中引入目标影响因子和障碍物影响因子以提高算法的全局搜索能力, 避免陷入死锁; 最后利用达尔文的进化论改进蚁群算法的信息素更新规则用于加快算法的迭代速度, 缩小运行时间. 在不同规模的栅格地图环境下的实验表明, 所提出的进化蚁群算法能够加快迭代速度, 提高搜索效率, 实现最优路径并避免算法死锁问题.

关键词: 移动机器人; 蚁群算法; 路径规划; 死锁; 信息素更新; 达尔文进化论

中图分类号: TP242

文献标志码: A

DOI: 10.13195/j.kzyjc.2021.1324

引用格式: 李涛, 赵宏生. 基于进化蚁群算法的移动机器人路径优化[J]. 控制与决策, 2023, 38(3): 612-620.

Path optimization for mobile robot based on evolutionary ant colony algorithm

LI Tao^{1,2†}, ZHAO Hong-sheng¹

(1. School of Automation, Nanjing University of Information Science and Technology, Nanjing 210044, China;
2. Collaborative Innovation Center of Atmospheric Environment and Equipment Technology, Nanjing 210044, China)

Abstract: In order to solve the problems of long running time, low searching efficiency and frequent deadlock in the path planning of ant colony algorithms, this paper proposes an ant colony algorithm based on the Darwin's theory of evolution. Firstly, a simple mode of the ant colony algorithm is proposed to solve the problem of blind search in blank grids. Then, in order to improve the global search ability and avoid falling into deadlock, the target influence factor and obstacle influence factor are introduced into the heuristic function. Finally, the pheromone updating rules of ant colony algorithm are improved using the Darwin's theory of evolution to accelerate the iteration speed and shorten the running time of the algorithm. Experiments on raster maps of different scales show that the evolutionary ant colony algorithm proposed in this paper can speed up the iteration speed, improve the search efficiency, achieve the optimal path and avoid the deadlock.

Keywords: mobile robot; ant colony algorithm; path planning; deadlock; pheromone update; Darwin's theory of evolution

0 引言

移动机器人的路径规划一直是机器人领域的热门话题, 也是机器人自主导航的关键所在^[1]. 传感器技术的发展使得移动机器人可以感知周围的环境, 这是自主导航的前提, 移动机器人可以在传感器技术的基础上利用路径规划算法进行自主高效的路径规划. 路径规划算法主要包括蚁群算法、粒子群算法^[2]、A*算法^[3]等. 这些算法中, 蚁群算法因为具有操作简单、鲁棒性强的优点, 被许多学者用于路径规划并取得了很好的成效. Miao 等^[4]利用蚁群算法进行多目

标优化, 将路径长度、安全性和油耗综合考虑, 利用自适应改进蚁群算法得出多目标的最优路径. 王晓燕等^[5]将蚁群算法与人工势场法相结合对不同的栅格位置赋予不同的初始信息素, 降低蚁群搜索的盲目性, 提高算法的搜索效率. 敖邦乾等^[6]使用 Dubins 曲线算法和贝塞尔三阶曲线理论对水面无人艇路径的连接点与路径进行平滑处理, 使得路径在满足最小旋转半径的同时也满足动力学特性. 徐菱等^[7]另辟蹊径, 将蚂蚁的搜索方式拓展到 16 方向 24 邻域, 扩大了搜索范围, 提高了蚁群算法的路径寻优效果和搜索效

收稿日期: 2021-07-29; 录用日期: 2022-01-12.

基金项目: 国家自然科学基金项目(61973168); 江苏省“333工程”项目(BRA2020067).

责任编辑: 誉斌.

†通讯作者. E-mail: litaojia@163.com.

率. 张恒等^[8]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避免在迭代过程中陷入局部最优. 尽管上述文献已经用蚁群算法改进了移动机器人的路径规划性能, 但对于空白地图以及障碍物分布较分散的随机地图依然欠缺考虑, 对于算法的运行时间和迭代速度的优化研究还不够深入.

本文针对上述问题进行深入的研究与实验, 主要贡献归纳如下: 1) 针对移动机器人路径规划中出现的盲目搜索问题, 提出一种蚁群算法简易模式, 使得算法不仅可以得到最优路径而且提高了搜索效率; 2) 在启发函数中引入目标影响因子和障碍物影响因子使得全局搜索能力得到增强, 以解决路径规划中的死锁问题; 3) 根据达尔文的进化思想改进信息素的更新规则加快了算法的迭代速度, 减少寻找最优路径的时间.

1 问题描述

1.1 移动机器人的路径规划准则

路径规划目的是在给定的工作环境中根据一定的准则为机器人寻找从起点移动到终点的无碰撞路径, 这里的准则根据主体的不同也各有不同. 无人机的路径规划准则^[9]主要包括: 安全性原则, 即无人机在飞行过程中不能触碰到障碍物; 最优性原则, 即无人机的飞行路径长度要短; 机动性原则, 即飞行的俯仰、偏航和横滚角度不能超出限制. 机械臂的路径规划准则^[10]包括: 安全性原则, 即机械臂不能与障碍物发生碰撞; 稳定性原则, 机械臂运动过程中不能出现停滞或者反复振荡. 本文研究的是移动机器人的路径规划, 其准则主要包括:

- 1) 安全性原则, 即移动机器人不能与障碍物碰撞;
- 2) 最优性原则, 即移动机器人所走的路径要达到最短;
- 3) 时间原则, 即寻找最优路径的时间要尽可能短.

本文工作主要围绕这3个方面开展.

1.2 空间建模

栅格图法是一种简单有效的建模方法, 它不仅对于障碍物有着很强的适应性, 还能有效减小环境的复杂度. 笛卡尔坐标系法则是一种直观的建模方法, 它可以清楚地表示出环境中的每一个点. 本文结合两个建模方法的长处, 既有利于路径规划也可以辅助算法的改进, 两者融合在一起, 将移动机器人的活动空间转化为图1所示的栅格地图模型.

图1中白色方格表示可行区域, 黑色方格表示障

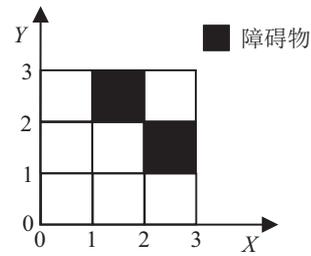


图1 建模图

碍物区域. 假设第*i*个栅格的坐标为(*x_i*, *y_i*), 则两者之间的关系为

$$\begin{cases} x_i = a(\text{mod}(i, N_x) - 0.5), \\ y_i = a(N_y + 0.5 - \text{ceil}(i/N_y)). \end{cases} \quad (1)$$

其中: *a*为每个栅格的长度, *N_x*和*N_y*分别为栅格图的行数与列数, mod为取余函数, ceil(*i/N_y*)为大于或等于*i/N_y*的最小整数函数.

2 改进蚁群算法

2.1 经典蚁群算法

经典蚁群算法(以下简称“蚁群算法”)是从蚁群寻找食物的过程中受到启发进而被提出, 研究表明: 生物世界中的蚂蚁有能力在没有任何可见提示的情况下找到一条从窝巢到食物之间的最短路径, 并且可以随着环境的改变而改变, 适应性地找到新的最优路径. 蚁群算法可以应用到TSP问题、车间调度问题、路径规划问题等^[1], 主要流程如下:

- 1) 蚁群初始化, 确定蚁群的数量num以及信息素*q*的分布.
- 2) 根据启发信息随机选择方向向下一节点运动, 启发信息定义为

$$\eta_{i,j} = 1/d_{i,j}, \quad j \in \text{allow}_k. \quad (2)$$

其中: $\eta_{i,j}$ 为启发信息, $d_{i,j}$ 为第*i*个节点到第*j*个节点的欧氏距离, allow_k 为当前蚂蚁可以选择的下一个目标点的集合, 且

$$d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \quad (3)$$

x_i, x_j, y_i, y_j 为*i*节点和*j*节点的横纵坐标.

- 3) 所有蚂蚁都完成一次迭代后, 更新每条路径的信息素, 有

$$\tau_{i,j}^{\text{new}} = (1 - \rho)\tau_{i,j}^{\text{old}} + \sum_{k=1}^m \Delta\tau_{i,j}^k. \quad (4)$$

其中: $\tau_{i,j}^{\text{old}}$ 为原来路段(*i, j*)的信息素浓度; ρ 为信息素衰减率; *m*为蚂蚁个数; $\Delta\tau_{i,j}^k$ 为第*k*只蚂蚁在路段(*i, j*)上留下的单位长度轨迹信息素数量, 有

$$\Delta\tau_{i,j}^k = \begin{cases} Q/L_k, & (i, j) \text{在路径上;} \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

这里: Q 为体现蚂蚁所留轨迹数量的一个常数, L_k 为本次迭代中第 k 只蚂蚁所走的路径长度. 设第 k 只蚂蚁当前所在地点为 i , 则其下一个节点 j 根据伪随机策略进行选择, 有

$$j = \begin{cases} \operatorname{argmax}\{[\tau_{i,j}]^\alpha \times [\eta_{i,j}]^\beta\}, & q \leq q_0; \\ P_{i,j}^k, & \text{otherwise.} \end{cases} \quad (6)$$

其中: α 为信息素重要程度因子, β 为启发函数重要程度因子, q 为 $(0,1)$ 区间的一个随机数, q_0 为转移概率阈值. 当系统产生的随机数小于等于转移概率阈值时, 选取可选节点中信息素浓度与启发信息乘积最大的节点作为下一节点, 否则根据概率 $P_{i,j}^k$ 进行选择, 有

$$P_{i,j}^k = \begin{cases} \frac{[\tau_{i,j}]^\alpha \times [\eta_{i,j}]^\beta}{\sum_{u \in \text{allow}_k} [\tau_{i,u}]^\alpha \times [\eta_{i,u}]^\beta}, & j \in \text{allow}_k; \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

其中: u 为可选节点中的任意一个节点, α 和 β 与式(6)相同.

4) 判断是否满足结束条件, 若不满足则继续迭代, 若满足则结束迭代, 输出最优路径.

2.2 简易模式

针对在空白网格中利用蚁群算法中进行路径规划时出现的盲目搜索现象, 设计一种蚁群算法的简易模式, 实现搜索效率的提升. 空白网格分为两种情况: 一是全局空白网格, 即没有障碍物的空白网格; 二是局部空白网格, 即栅格图的一部分没有障碍物.

简易模式工作原理: 在蚂蚁前行的过程中会生成一个禁忌表和可选节点集合^[11], 蚂蚁当前节点的邻域有8个, 其中障碍物和上一节点放置在禁忌表中, 空白节点均作为可选节点. 如图2(a)所示, 当可选节点的个数为7时可以将当前所在区域视作一个空白邻域, 此时根据目标的方位直接确定下一个节点的位置, 而如图2(b)则需利用伪随机策略以及轮盘赌确定下一节点, 这也是简易模式可以提高搜索效率的根本原因.

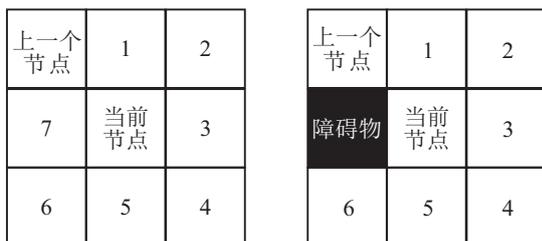


图2 邻域节点分布图

简易模式下一个节点的选择公式如下:

$$j = \begin{cases} i + N_x + 1, & \text{amount} = 7; \\ j', & \text{otherwise.} \end{cases} \quad (8)$$

其中: amount 为可选节点的数量, j' 为利用式(6)选择的下一个节点.

注1 简易模式可以在有空白栅格的地图中提高算法的搜索效率并缩小运行时间, 其根本原因是将轮盘赌的随机选择转化为定向选择, 这会降低解的多样性. 当出现障碍物的分布较为集中、空白空间较少的复杂地形时, 简易模式会出现容易陷入死锁的问题, 因此下文的改进点主要针对死锁问题.

2.3 调整距离启发函数

蚁群算法的启发函数只考虑当前网格与可选节点的距离, 这必然会导致算法的全局搜索能力变差^[10-15], 且容易陷入死锁. 为了提高算法的全局搜索能力, 在启发函数中引入可选节点与终点的距离, 使得启发函数兼顾局部搜索与全局搜索. 与此同时, 将当前节点的周围障碍物信息加入启发函数, 使得启发函数随着节点环境的不同产生变化, 更改的启发函数为

$$\eta_{i,j} = \frac{1}{k_1 \times d_{i,G} + k_2 \times \sum_{\text{obs} \in P} d_{i,\text{obs}}}, \quad j \in \text{allow}_k. \quad (9)$$

其中: $d_{i,G}$ 为目标影响因子, 表示第 i 个节点到终点 G 的距离; $\sum_{\text{obs} \in P} d_{i,\text{obs}}$ 为障碍物影响因子, 表示障碍物影响范围 P 内所有到第 i 个节点的障碍物的距离之和. 障碍物影响范围如图3所示.

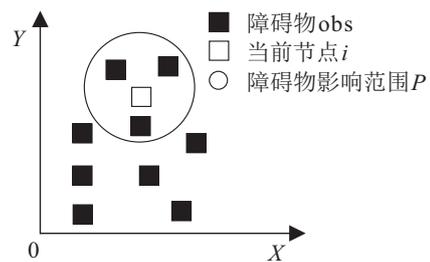


图3 障碍物影响范围

图3中, 当前节点 i 的障碍物影响范围 P 内有3个障碍物, 障碍物影响范围 P 外有6个障碍物, 那么 $\sum_{\text{obs} \in P} d_{i,\text{obs}}$ 则表示当前节点 i 到障碍物影响范围 P 内3个障碍物的距离之和.

式(9)中 k_1 和 k_2 分别表示启发函数中目标影响因子和障碍物影响因子的权重系数, 目标影响因子表示当前节点到目标点的距离, 这是从全局确定蚂蚁移动的方向; 障碍物影响因子表示在障碍物影响范围内障碍物到当前节点的距离之和, 这是从局部确定蚂

蚁移动的方向. 这两个因子分别代表的即为全局搜索和局部搜索. 在蚂蚁移动的前期, 目标影响因子较大, 所以权重系数要相对较小才能平衡全局搜索和局部搜索的能力, 而在蚂蚁移动的后期, 目标影响因子较小, 所以权重系数要相对较大才能平衡全局搜索和局部搜索的能力. 基于此, 参数设置为

$$\begin{cases} k_1 = 0.1, k_2 = 0.9, i < N_x \times N_y / 2; \\ k_1 = 0.9, k_2 = 0.1, i \geq N_x \times N_y / 2. \end{cases} \quad (10)$$

其中: i 为当前节点, N_y 为栅格地图的行数, N_x 为栅格地图的列数.

注2 1) 根据蚂蚁行走的不同阶段设置目标影响因子 $d_{i,obs}$ 和障碍物影响因子 $\sum_{obs \in P} d_{i,obs}$ 的权重系数 k_1 和 k_2 , 不仅可以加快算法的搜索速度还能兼顾局部搜索能力与全局搜索能力.

2) 式(9)与(2)的区别在于, 式(2)中 $d_{i,j}$ 只考虑当前节点 i 附近的可行节点, 而根据图2可知, 每一个节点到周围可行节点的距离都不会变化, 无论蚂蚁走到了地图的任何位置 $d_{i,j}$ 都不会变化, 这便容易出现死锁现象. 但是式(9)中目标影响因子 $d_{i,G}$ 会随着蚂蚁不断行走而产生变化, 这样动态的变化是死锁现象消失的主要原因. 也可以参考人工势场法的原理进行解释, 启发函数是下一节点对当前节点的吸引力, 如果距离目标越近则吸引力越大, 启发函数的值便越大; 障碍物越多则意味着斥力越大, 启发函数值越小. 在启发函数里综合考虑这两个因素便是看下一个节点到当前节点的合力有多大, 合力越大则蚂蚁选择该点的概率越大. 产生死锁的位置其合力必定小于其他节点, 所以陷入死锁的可能性大大减低甚至消失.

3) 障碍物影响因子 $\sum_{obs \in P} d_{i,obs}$ 表示当前节点与周围障碍物的距离, 这从反面表现出蚁群算法中当前节点的可行节点信息, 且搜索的范围比蚁群算法大, 所以障碍物影响因子 $\sum_{obs \in P} d_{i,obs}$ 是蚁群算法中可行节点信息的加强版.

2.4 进化论思想的信息素更新

达尔文的进化论“物竞天择, 适者生存”讲述了自然界生物进化的一般规律, 说明不适应环境便会被环境所淘汰, 只有适应环境才能生存. 蚁群算法是一种智能仿真算法, 而蚂蚁种群的进化遵循进化论, 因此达尔文的进化论对于蚁群算法的改进具有指导意义. 本文正是利用这一规律在蚁群算法的基础上对信息素的更新规则进行调整, 使得算法能够更快地寻找到最优路径. 将所有蚂蚁分成两类, 一类是基层蚂蚁, 一类是进化蚂蚁, 在一个种群中基层蚂蚁的个数总是多于进化蚂蚁, 但是进化蚂蚁往往决定种群进化

的方向, 因此本文对不同的蚂蚁采取不同的信息素处理方式: 基层蚂蚁信息素随着时间慢慢减少, 而进化蚂蚁会根据进化程度的不同增加对应的信息素. 但是, 这并不意味着基层蚂蚁只能慢慢消失, 只要基层蚂蚁在信息素的更新后找到了更优的路径, 便能变成进化蚂蚁. 具体更新规则如下:

$$\tau_{i,j}^{new} = \begin{cases} (1 - \rho)\tau_{i,j}^{old}, (i, j) \in \text{基层路径}; \\ (1 - \rho)\tau_{i,j}^{old} + \rho\Delta\tau_{i,j}, (i, j) \in \text{进化路径}. \end{cases} \quad (11)$$

其中 $\Delta\tau_{i,j}$ 为信息素增量, 有

$$\Delta\tau_{i,j} = \sigma / (L_k - L_{ave}), \quad (12)$$

σ 为进化蚂蚁影响因子, L_k 为第 k 只进化蚂蚁的路径长度, L_{ave} 为所有蚂蚁走过的路径的平均长度.

注3 式(11)与(4)的区别在于, 本文利用达尔文的进化论思想将蚂蚁分成两种基因类型, 不同基因类型的蚂蚁所走的道路会有不同的信息素更新规则. 而引入达尔文的进化论使得算法能够更快找到最优解的本质原因是: 达尔文的进化论使得原来的种群不断进化, 即基层蚂蚁变成进化蚂蚁, 而进化蚂蚁变成了更好的进化蚂蚁, 这样不断地进化使得原有种群中不适应环境的蚂蚁逐渐被淘汰, 从而使得整个种群的基因越来越好, 而找到的路径越来越短. 在相同的基数下, 最优解的数量越多找到最优解的概率越高, 速度越快.

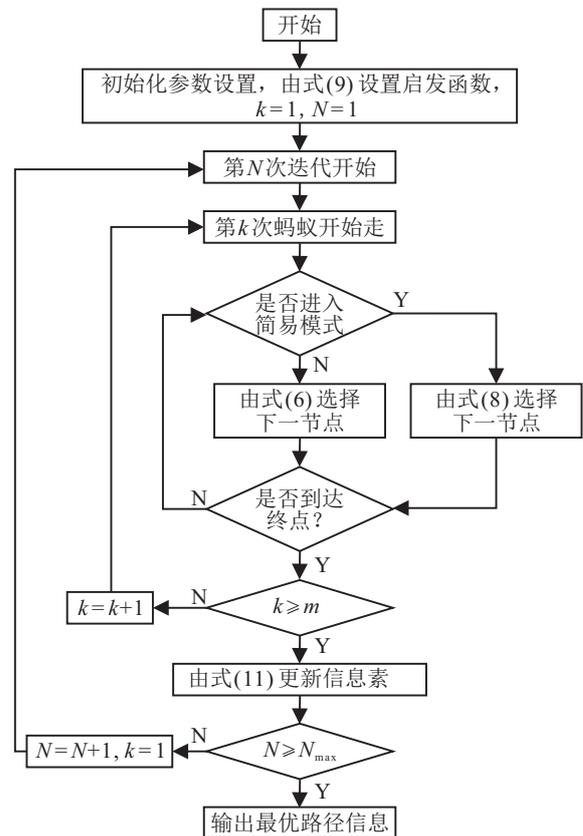


图4 改进蚁群算法流程

2.5 算法流程

算法流程如图4所示,具体步骤如下.

- step 1: 设置初始化参数,并由式(9)设置启发函数,第 N 次迭代开始, $N = 1, k = 1$;
- step 2: 第 k 只蚂蚁开始寻路,判断是否进入简易模式,并由式(6)或(8)选择下一节点;
- step 3: 判断蚂蚁是否到达终点,若是则执行 step 4,否则转至 step 2;
- step 4: 判断是否所有蚂蚁均走完,若是则执行 step 5,否则转至 step 2,令 $k + 1$;
- step 5: 由式(11)更新信息素,判断迭代是否完成,若完成则执行 step 6,否则转至 step 2,令 $N = N + 1, k = 1$;
- step 6: 输出最优路径信息.

3 仿真实验分析

首先对蚁群算法的改进点逐个进行实验验证,然后在两种不同规模的地图下利用蚁群算法、排序精英蚁群算法^[16]、文献[4]算法和改进蚁群算法分别进行路径规划的仿真实验. 算法运行环境 Windows 10 64 bit; Matlab R2018a; 处理器 AMD Ryzen 7 3700X 8-Core Processor; 内存 16 GB. 以文献[8]的参数设置为基础,各算法参数设置如表1所示.

表1 算法参数

参数名称	数值大小
蚂蚁个数 m	30
最大迭代次数 N_{max}	50
信息素重要程度因子 α	1.5
启发函数重要程度因子 β	6.0
信息素衰减率 ρ	0.4
转移概率阈值 q_0	0.8

3.1 简易模式实验论证

将简易模式加入蚁群算法中,在保持各参数一致的情况下将两者的运行时间进行对比,从而验证简易模式是否能够在空白栅格地图中提高算法的搜索效率和运行速度. 在地图中绿色栅格为起点,红色栅格为终点.

3.1.1 空白地图

在 20×20 的空白栅格图中,分别运行蚁群算法和带有简易模式的蚁群算法各 10 次,将得到的运行时间进行对比,结果如图5所示. 由图(5)可见,在空白栅格图中简易模式的运行时间比蚁群算法要短. 计算 10 次运行时间的平均值得出蚁群算法为 0.239 5 s,简

易模式为 0.154 7 s,简易模式的运行速度比蚁群模式提高了 35.4%. 可见,简易模式在空白栅格图中可以提高蚁群算法的运行速度.

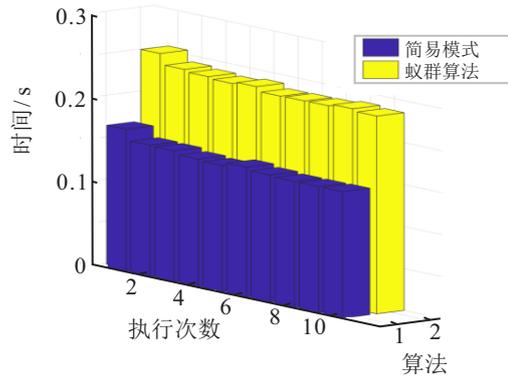


图5 不同算法运行时间对比(空白地图)

不同算法节点分布对比如图6所示,蓝色上三角表示蚁群算法的搜索节点,红色下三角表示简易模式的搜索节点. 由图6可见,蚁群算法搜索到的节点有 134 个,简易模式搜索到的节点只有 20 个,在空白地图中蚁群算法会出现盲目搜索现象,而简易模式可以大大提高蚁群算法的搜索效率.

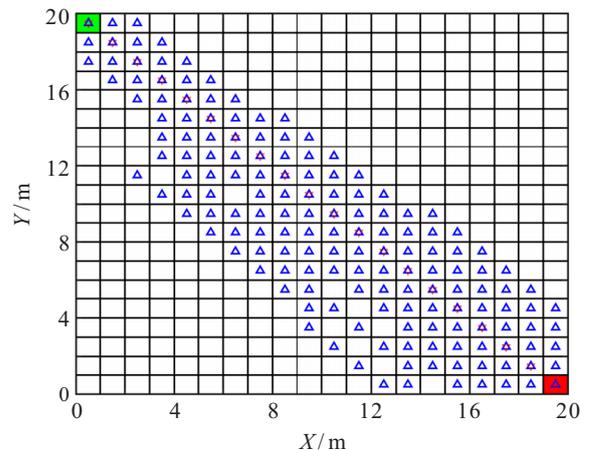


图6 不同算法节点分布对比(空白地图)

3.1.2 有部分空白的地图

不同算法运行时间对比如图7所示. 由图(7)可见,在有部分空白的栅格图中,简易模式的运行时间

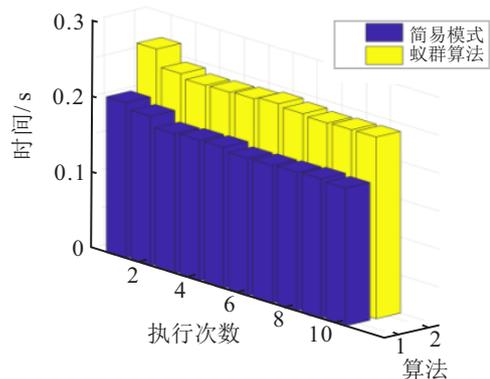


图7 不同算法运行时间对比(部分空白地图)

比蚁群算法要短. 计算10次运行时间的平均值得出蚁群算法为0.2428s, 简易模式为0.1864s, 简易模式的运行速度比蚁群算法提高了23.2%. 可见, 简易模式在有部分空白的栅格图中可以提高蚁群算法的运行速度.

不同算法节点分布对比如图8所示, 红色下三角表示简易模式节点, 蓝色上三角表示蚁群算法节点. 由图8可见, 蚁群算法搜索到的节点有139个, 简易模式搜索到的节点只有50个, 在部分空白的栅格图中简易模式可以减少不必要的节点搜索, 提高搜索效率.

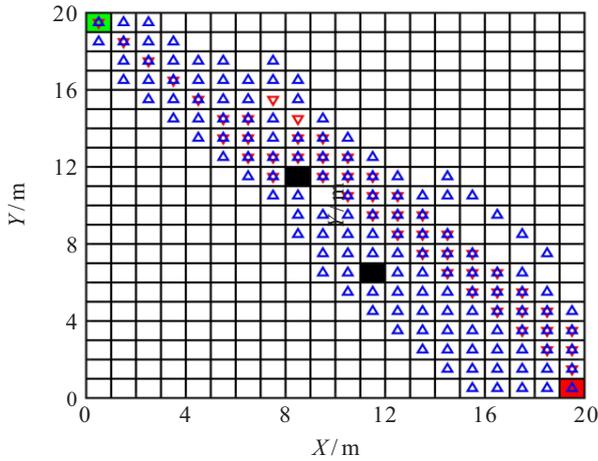


图8 不同算法节点分布对比(部分空白地图)

3.1.3 复杂地图

如注1所言, 简易模式在障碍物分布比较集中、空白空间较少的复杂地形中容易陷入死锁, 且无法避免地陷入局部最优. 本文在复杂地图中利用蚁群算法和简易模式在复杂地图中进行了多次对比实验, 分别从平均时间、平均最优路径长度和死锁蚂蚁个数3方面进行分析, 结果如表2所示.

表2 复杂地形运行结果对比

算法	平均时间/s	平均最优路径长度/m	死锁蚂蚁个数
简易模式	0.3627	33.1973	188.4
蚁群算法	0.3476	33.0659	156.7

由表2可知, 简易模式在复杂地形下的运行时间比蚁群算法长, 路径长度比蚁群算法长, 死锁蚂蚁个数比蚁群算法多. 由此可知, 简易模式在复杂地形下的适应性不好.

3.2 改进启发信息实验论证

为了验证本文对启发函数改进的有效性, 在蚁群算法的基础上只加入改进的启发式信息, 并将最后得出的路径信息与蚁群算法和文献[4]算法进行时间、死锁蚂蚁数量、最优路径3方面对比.

不同算法运行时间对比如图9所示. 由图9可见, 本文算法10次平均时间为0.2517s, 文献[4]算法为0.2923s, 蚁群算法为0.4148s. 本文算法比文献[4]算法运行时间提高了13.9%, 比蚁群算法提高了39.3%, 可见本文算法可以减少运行时间.

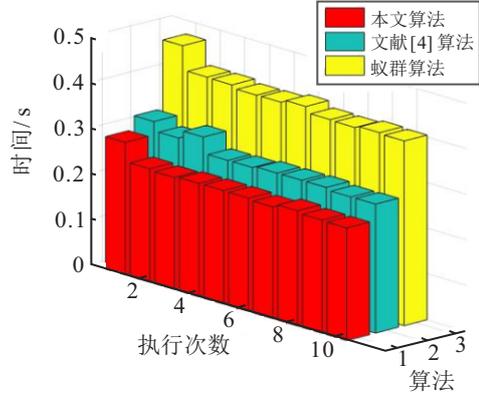


图9 不同算法运行时间对比(改进启发函数)

死锁蚂蚁个数对比如图10所示. 由图10可见, 本文算法执行10次死锁蚂蚁平均个数为0, 文献[4]算法为5.2, 蚁群算法为166.9. 可见, 本文算法可以解决蚁群算法陷入死锁的问题.

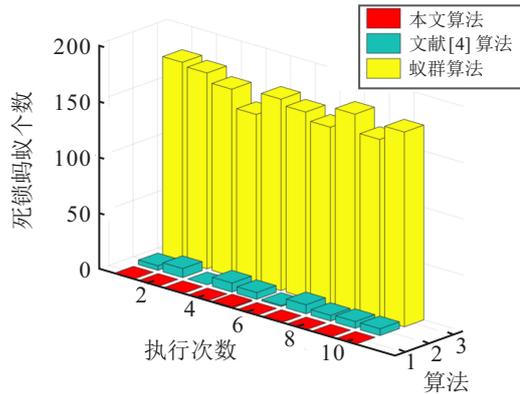


图10 死锁蚂蚁个数对比(改进启发函数)

不同算法路径规划如图11所示. 红色菱形标线为本文算法的路径规划, 蓝色上三角标线为文献[4]算法的路径规划.

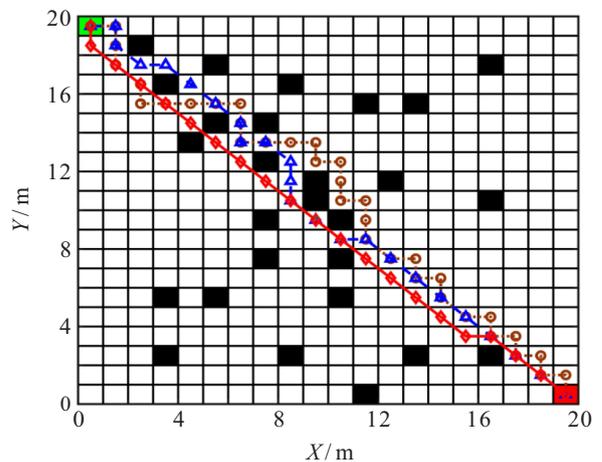


图11 不同算法路径规划(改进启发函数)

算法路径规划,棕色圆形标线为蚁群算法路径规划标线.由图11可见,本文算法路径规划长度最短.综上所述可知,改进的启发函数可以在找到最优路径的情况下减少运行时间并避免算法陷入死锁.

3.3 改进信息素更新规则实验论证

利用式(11)对蚁群算法进行改造,并将其与排序精英蚁群算法^[16]和蚁群算法从时间和迭代关系两方面进行对比.

不同算法运行时间对比如图12所示.由图12可见,本文算法执行10次的平均时间为0.2542s,排序精英蚁群算法为0.3103s,蚁群算法为0.4029s.本文算法比排序精英蚁群算法的运行时间提高了18.1%,比蚁群算法提高了34.9%.可见,本文算法可以减少更多的运行时间.

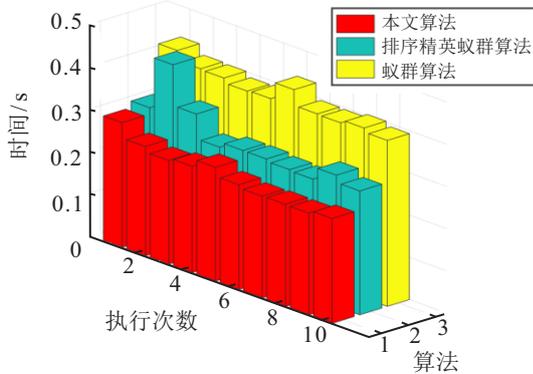


图12 不同算法运行时间对比(改进信息素更新规则)

不同算法迭代曲线如图13所示.由图13可见,蚁群算法在迭代进行到15次时收敛,找到的最优路径长度为29.15m;排序精英蚁群算法在迭代进行到6次时收敛,找到的最优路径长度为28m;本文算法在迭代进行到2次时收敛,找到的最优路径长度为27.4m.本文算法不仅可以更快地收敛,还能找到最优值.

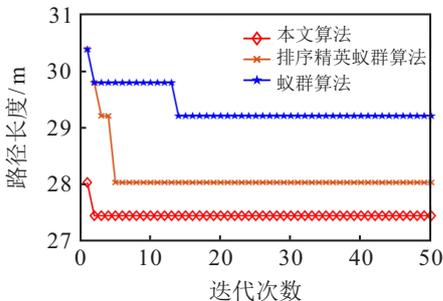


图13 不同算法迭代曲线(改进信息素更新规则)

3.4 总体实验论证

图14和图15为本文算法、蚁群算法、文献[4]算法和排序精英蚁群算法在20×20和40×40随机地图上的路径规划,两图中各算法生成路径的颜色、标

记和线形均相同,红色菱形实线为本文算法生成的最优路径;青色点下三角虚线为蚁群算法;蓝色上三角断虚线为文献[4]算法;粉色圆形长虚线为排序精英蚁群算法,具体分析见下文.

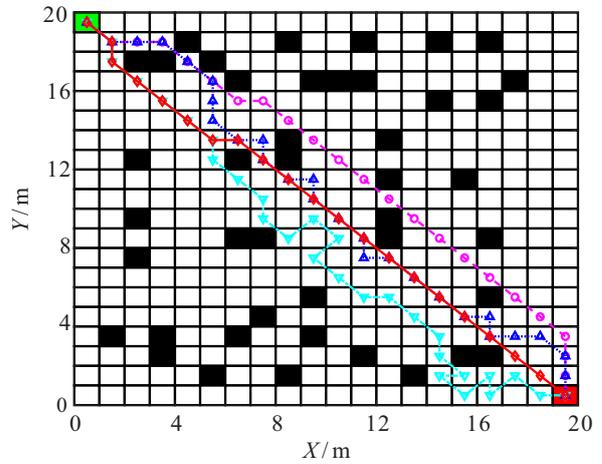


图14 20×20地图路径规划

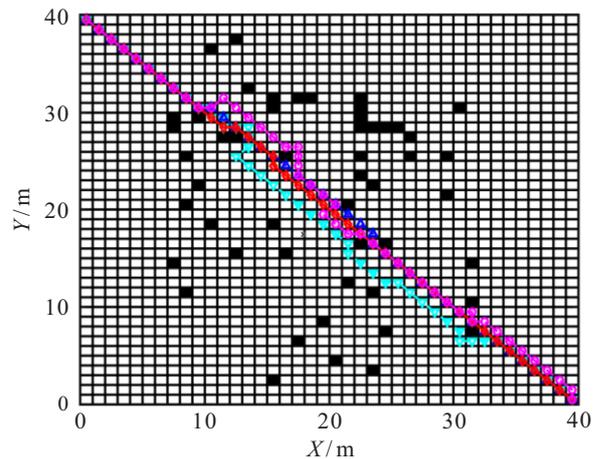


图15 40×40地图路径规划

3.4.1 20×20地图

本节将第3.1节、3.2节、3.3节中对于蚁群算法的改进点融合在一起,在20×20的地图上进行实验,并将其与蚁群算法、文献[4]算法、排序精英蚁群算法进行对比,从运行时间、迭代曲线和死锁蚂蚁个数等

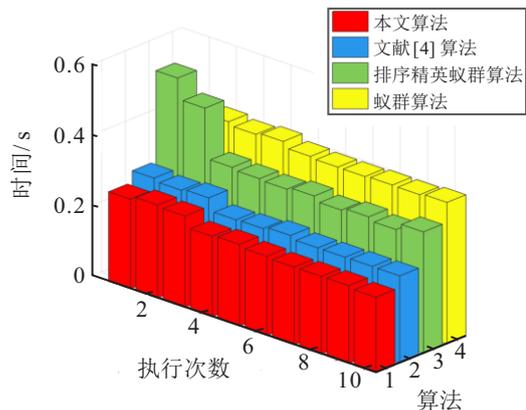


图16 不同算法运行时间对比(20×20地图)

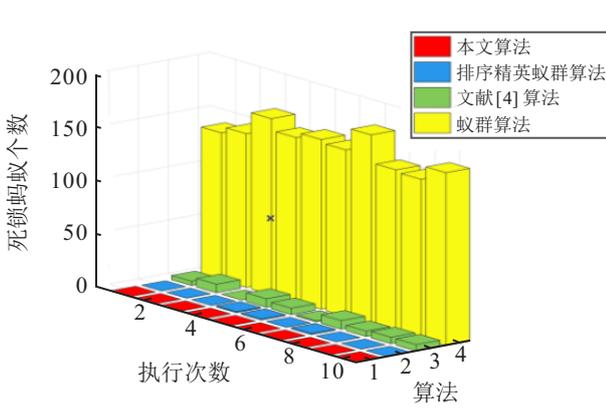


图 17 死锁蚂蚁个数对比(20 × 20 地图)

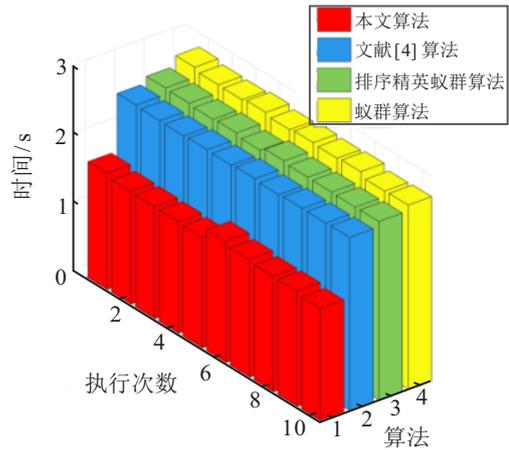


图 19 不同算法运行时间对比(40 × 40 地图)

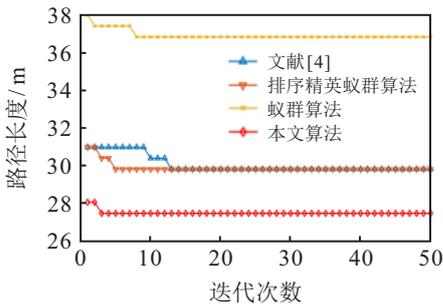


图 18 不同算法迭代曲线(20 × 20 地图)

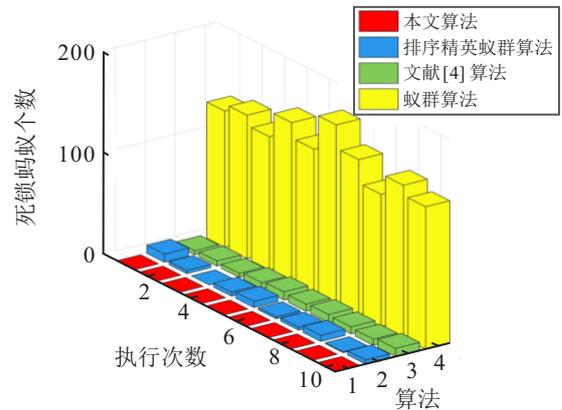


图 20 死锁蚂蚁个数对比(40 × 40 地图)

角度对算法进行评价. 由图 16~图 18 可见: 在 20 × 20 网格地图中, 本文算法较其他 3 种算法不仅运行时间少, 找到的最优路径短, 而且避免了死锁问题.

3.4.2 40 × 40 地图

将 3.4.1 节的实验环境变为 40 × 40 栅格地图, 其他设置不变, 实验结果如图 19~图 21 所示. 可以看出: 在 40 × 40 的网格地图中, 本文算法较其他 3 种算法不仅运行时间少, 找到的最优路径短, 而且避免了死锁问题.

为了方便地看出本文算法与其他算法的优劣, 将运行结果列于表 3 和表 4.

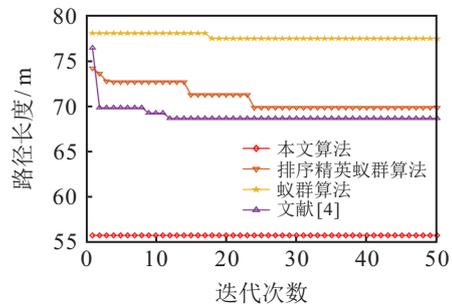


图 21 不同算法迭代曲线(40 × 40 地图)

表 3 20 × 20 地图各算法运行结果对比

算法	平均运行时间 /s	死锁蚂蚁个数	最优路径长度 /m	迭代次数
本文算法	0.2341	0	27.4558	2
文献[4]算法	0.2595	5.2	30.9706	12
排序精英蚁群算法	0.3717	0.3	29.2132	4
蚁群算法	0.4055	155.5	38	10

表 4 40 × 40 地图各算法运行结果对比

算法	平均运行时间 /s	死锁蚂蚁个数	最优路径长度 /m	迭代次数
本文算法	1.6765	0	55.7401	1
文献[4]算法	2.5293	6.3	70.9706	17
排序精英蚁群算法	2.6171	3.7	69.7990	13
蚁群算法	2.7401	145.2	76.8284	36

4 结论

本文提出了一种进化蚁群算法以解决蚁群算法进行路径规划中出现的运行时间长、搜索效率低、死锁等问题. 首先, 针对在空白栅格中出现的盲目搜索问题, 提出了一种蚁群算法简易模式; 其次在启发函数中引入目标影响因子和障碍物影响因子, 提高算法全局搜索能力的同时避免了死锁问题; 最后通过改进蚁群算法的信息素更新规则使得整个种群的基因不断进化从而加快算法迭代速度, 缩短运行时间. 本文算法虽缩短了系统运行时间, 但没有考虑移动机器人的限制问题, 这将是下一阶段的研究重点.

参考文献(References)

- [1] Wahab M N A, Nefti-Meziani S, Atyabi A. A comparative review on mobile robot path planning: Classical or meta-heuristic methods?[J]. *Annual Reviews in Control*, 2020, 50: 233-252.
- [2] Cai C X, Sun C S, Han Y, et al. Clinical flexible needle puncture path planning based on particle swarm optimization[J]. *Computer Methods and Programs in Biomedicine*, 2020, 193: 105511.
- [3] Zhong X Y, Tian J, Hu H S, et al. Hybrid path planning based on safe A* algorithm and adaptive window approach for mobile robot in large-scale dynamic environment[J]. *Journal of Intelligent & Robotic Systems*, 2020, 99(1): 65-77.
- [4] Miao C W, Chen G Z, Yan C L, et al. Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm[J]. *Computers & Industrial Engineering*, 2021, 156: 107230.
- [5] 王晓燕, 杨乐, 张宇, 等. 基于改进势场蚁群算法的机器人路径规划[J]. *控制与决策*, 2018, 33(10): 1775-1781.
(Wang X Y, Yang L, Zhang Y, et al. Robot path planning based on improved ant colony algorithm with potential field heuristic[J]. *Control and Decision*, 2018, 33(10): 1775-1781.)
- [6] 敖邦乾, 杨莎, 叶振环. 改进蚁群算法水面无人艇平滑路径规划[J]. *控制理论与应用*, 2021, 38(7): 1006-1014.
(Ao B Q, Yang S, Ye Z H. Improved ant colony algorithm for unmanned surface vehicle smooth path planning[J]. *Control Theory & Applications*, 2021, 38(7): 1006-1014.)
- [7] 徐菱, 付文浩, 江文辉, 等. 基于16方向24邻域改进蚁群算法的机器人路径规划[J]. *控制与决策*, 2021, 36(5): 1137-1146.
(Xu L, Fu W H, Jiang W H, et al. Mobile robots path planning based on 16-directions 24-neighborhoods improved ant colony algorithm[J]. *Control and Decision*, 2021, 36(5): 1137-1146.)
- [8] 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 H, He L, Yuan L, et al. Mobile robot path planning using improved double-layer ant colony algorithm[J]. *Control and Decision*, 2022, 37(2): 303-313.)
- [9] Huang Z R, Chen C C, Pan M X. Multiobjective UAV path planning for emergency information collection and transmission[J]. *IEEE Internet of Things Journal*, 2020, 7(8): 6993-7009.
- [10] Khan A T, Li S, Kadry S, et al. Control framework for trajectory planning of soft manipulator using optimized RRT algorithm[J]. *IEEE Access*, 2020, 8: 171730-171743.
- [11] Li S D, You X M, Liu S. Co-evolutionary multi-colony ant colony optimization based on adaptive guidance mechanism and its application[J]. *Arabian Journal for Science and Engineering*, 2021, 46(9): 9045-9063.
- [12] Li C Q, Xiao J, Liu Y, et al. An adaptive immune ant colony optimization for reducing energy consumption of automatic inspection path planning in industrial wireless sensor networks[J]. *Journal of Sensors*, 2021, 2021: 9960043.
- [13] 孙功武, 苏义鑫, 顾轶超, 等. 基于改进蚁群算法的水面无人艇路径规划[J]. *控制与决策*, 2021, 36(4): 847-856.
(Sun G W, Su Y X, Gu Y C, et al. Path planning for unmanned surface vehicle based on improved ant colony algorithm[J]. *Control and Decision*, 2021, 36(4): 847-856.)
- [14] Huang R X, Ning J Y, Mei Z H, et al. Study of delivery path optimization solution based on improved ant colony model[J]. *Multimedia Tools and Applications*, 2021, 80(19): 28975-28987.
- [15] 熊亿民. 基于改进蚁群算法的全向移动机器人全遍历路径规划[J]. *计算机系统应用*, 2021, 30(6): 209-214.
(Xiong Y M. Full traversal path planning of omnidirectional mobile robot based on improved ant colony algorithm[J]. *Computer Systems & Applications*, 2021, 30(6): 209-214.)
- [16] Song Q, Zhao Q L, Wang S X, et al. Dynamic path planning for unmanned vehicles based on fuzzy logic and improved ant colony optimization[J]. *IEEE Access*, 2020, 8: 62107-62115.

作者简介

李涛(1979—), 男, 教授, 博士生导师, 从事非线性系统控制理论及应用、故障诊断与自愈控制等研究, E-mail: litaojia@163.com;

赵宏生(1998—), 男, 硕士生, 从事智能算法、路径规划的研究, E-mail: zx1215907843@163.com.

(责任编辑: 郑晓蕾)