

控制与决策

Control and Decision

基于自适应机制改进蚁群算法的移动机器人全局路径规划

毛文平, 李帅永, 谢现乐, 杨雪梅, 聂嘉炜

引用本文:

毛文平, 李帅永, 谢现乐, 杨雪梅, 聂嘉炜. 基于自适应机制改进蚁群算法的移动机器人全局路径规划[J]. 控制与决策, 2023, 38(9): 2520–2528.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2022.0026>

您可能感兴趣的其他文章

Articles you may be interested in

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 $\text{pm}3\sigma$ 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>

基于改进RRT*FN算法的机器人路径规划

Robot path planning based on improved RRT*FN algorithm
控制与决策. 2021, 36(8): 1834–1840 <https://doi.org/10.13195/j.kzyjc.2019.1713>

一种基于免疫机理的确定性移动机器人路径规划算法

A path planning algorithm of deterministic mobile robot based on immune mechanism
控制与决策. 2021, 36(10): 2418–2426 <https://doi.org/10.13195/j.kzyjc.2020.0059>

基于自适应机制改进蚁群算法的移动机器人全局路径规划

毛文平, 李帅永[†], 谢现乐, 杨雪梅, 聂嘉炜

(重庆邮电大学 工业物联网与网络化控制教育部重点实验室, 重庆 400065)

摘要: 针对基本蚁群算法在二维静态栅格地图下进行移动机器人路径规划时出现的搜索效率低下、收敛速度缓慢、局部最优解等问题, 提出一种自适应机制改进蚁群算法, 用于移动机器人在二维栅格地图下的路径规划. 首先采用伪随机状态转移规则进行路径选择, 定义一种动态选择因子以自适应更新选择比例, 引入距离参数计算转移概率, 提高算法的全局搜索能力以及搜索效率; 然后基于最大最小蚂蚁模型和精英蚂蚁模型, 提出一种奖励惩罚机制更新信息素增量, 提高算法收敛速度; 最后定义一种信息素自适应挥发因子, 限制信息素浓度的上下限, 提高算法全局性的同时提高算法的收敛速度. 在不同规格的二维静态栅格地图下进行移动机器人全局路径规划对比实验, 实验结果表明自适应机制改进蚁群算法具有较快的收敛速度, 搜索效率明显提高且具有较好的全局搜索能力, 验证了所提算法的实用性和优越性.

关键词: 路径规划; 蚁群算法; 自适应机制; 移动机器人; 信息素浓度; 栅格地图

中图分类号: TP242.6

文献标志码: A

DOI: 10.13195/j.kzyjc.2022.0026

引用格式: 毛文平, 李帅永, 谢现乐, 等. 基于自适应机制改进蚁群算法的移动机器人全局路径规划[J]. 控制与决策, 2023, 38(9): 2520-2528.

Global path planning of mobile robot based on adaptive mechanism improved ant colony algorithm

MAO Wen-ping, LI Shuai-yong[†], XIE Xian-le, YANG Xue-mei, NIE Jia-wei

(Key Laboratory of Industrial Internet of Things & Networked Control of Ministry of Education, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

Abstract: Aiming at the problems of low search efficiency, slow convergence rate, and local optimal solution when a basic ant colony algorithm is applied to mobile robot path planning under a 2D static grid map, an adaptive mechanism improved ant colony algorithm for the path planning of mobile robots under the two-dimensional grid map environment is proposed. Firstly, pseudo-random state transition rules are used. For path selection, a dynamic selection factor is defined to adaptively update the selection ratio and a distance parameter is introduced to calculate the transition probability. Hence, the global search ability and search efficiency of the algorithm is improved. Then, based on the maximum and minimum ant model and the elite ant model, a reward and punishment mechanism is proposed to update the pheromone increment. Finally, a pheromone dynamic volatilization factor is defined to limit the upper and lower range of the pheromone concentration to improve the convergence speed of the algorithm and the global search ability. The global path planning comparison experiments of the mobile robot are carried out under two-dimensional static grid maps of different specifications, and the experimental results show that the adaptive mechanism improved ant colony algorithm has a faster convergence speed and significantly improved search efficiency, and it has a better global search ability, which verifies the effectiveness and superiority of the algorithm.

Keywords: path planning; ant colony algorithm; adaptive mechanism; mobile robot; pheromone concentration; raster map

0 引言

随着科技不断进步以及产业智能升级的不断推进, 移动机器人的应用领域越来越广, 如工厂自动

化、建筑、采矿、排险、军事、服务、农业等方面. 移动机器人的应用离不开路径规划, 路径规划是移动机器人研究的关键技术^[1]. 比如利用移动机器人在餐厅

收稿日期: 2022-01-05; 录用日期: 2022-04-15.

基金项目: 国家自然科学基金项目(61703066); 重庆市基础研究与前沿探索项目(cstc2018jcyjAX0536); 重庆市技术创新与应用发展专项(cstc2018jszx-cyztzxX0028, cstc2019jscx-fxydX0042, cstc2019jscx-zdztzxX0053).

责任编辑: 易建强.

[†]通讯作者. E-mail: lishuaiyong@cqupt.edu.cn.

给客人送餐时,采用较好的路径规划技术可以提高移动机器人的送餐效率,减少移动机器人的磨损,同时节约大量的人力资源。目前国内外学者对于移动机器人路径规划做了大量的研究,并提出了相应的解决方法,大致可以分为传统路径规划算法与智能仿生路径规划算法两大类。传统的路径规划算法有Dijkstra算法^[2]、A*算法^[3]、人工势场法^[4]。Dijkstra算法的搜索时间长,计算复杂度高;采用A*算法进行移动机器人的路径规划时,规划出的路径转折点较多,路径不平滑且搜索时间长;采用人工势场法进行路径规划时,容易出现目标点不可达以及易陷入局部最优等问题。相对于传统路径规划算法而言,智能仿生算法更适合解决大规模复杂环境下的路径规划问题。智能仿生路径规划算法有蚁群算法^[5]、遗传算法^[6]、粒子群算法^[7]、人工神经网络^[8]等。采用遗传算法进行移动机器人路径规划时,得到的最优路径长度较长且转折点多,收敛速度慢且局部搜索能力差;采用粒子群算法进行路径规划时,前期收敛速度快,后期易陷入局部最优;采用人工神经网络进行路径规划时,学习时间长且容易陷入局部最优。因此,研究搜索效率更高、寻优能力更强、环境适应能力更强的优化算法是目前的重点。

蚁群算法是一种模拟蚂蚁觅食行为的分布式智能仿生算法,可以实现从起始节点到目标节点的路径规划。蚁群算法具有正反馈高、鲁棒性强、分布式计算等优点,但是蚁群算法在算法初期搜索过于随机,使得算法搜索效率低,以至于算法整体收敛速度缓慢;信息素浓度在算法中后期过于集中,导致算法搜索停滞和易陷入局部最优解。针对这些问题,许多学者作出了改进优化,文献[9]在势场蚁群算法的基础上,融入跳点搜索算法,提高了路径的平滑度与搜索效率;文献[10]增加避障策略到转移概率函数中,并且进行二次路径规划,优化了最优路径长度,减少了路径的节点个数,提高了机器人运行的平稳度;文献[11]改进了初始信息素浓度分配,使起始节点与目标节点之间所有节点的信息素浓度高于其他节点,提高了算法的搜索效率,但是起始节点与目标节点之间的节点信息素浓度依旧相同,且不合适大型复杂地图,改进效果一般。

为了解决基本蚁群算法应用于移动机器人路径规划时出现局部最优、收敛性差、搜索效率低等问题,本文提出一种自适应机制改进蚁群算法(adaptive mechanism improved ant colony algorithm, AMACA)。该算法首先采用伪随机状态转移规则进

行路径选择,定义一种动态选择因子以自适应更新选择比例,提高算法全局搜索能力以及搜索效率;其次基于最大最小蚂蚁模型和精英蚂蚁模型,提出一种奖励惩罚机制更新信息素增量,提高算法收敛速度;最后采用自适应信息素挥发因子,并且为了防止信息素浓度过大或过小,限制信息素浓度的上下限,提高算法的全局性以及收敛速度,最终使移动机器人寻得全局最优路径。

1 蚁群算法描述

1.1 基本蚁群算法

1.1.1 基本蚁群算法描述

蚁群算法是通过模拟大自然中蚂蚁的觅食行为而衍生出来的一种智能仿生算法。蚁群算法有两个关键步骤,分别是计算状态转移概率与信息素更新^[12-13]。

1.1.2 状态转移概率

在移动机器人路径规划过程中,假设在起始点有 M 只蚂蚁,蚂蚁选择下一节点主要是由路径之间的信息素浓度以及启发信息来判断。基本蚁群算法的转移概率如下所示:

$$P_{ij}^k = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{j \in \text{allowed}_k} [\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}, & j \in \text{allowed}_k; \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

其中: α 表示信息素启发因子,反应了在进行路径选择时信息素浓度的受重视程度; β 表示期望启发因子,反映了启发式信息在路径选择时的重要性; allowed_k 表示蚂蚁在当前节点 i 可选择的下一节点 j 的集合; $\tau_{ij}(t)$ 表示路径 $\langle i, j \rangle$ 的信息素浓度; $\eta_{ij}(t)$ 表示路径 $\langle i, j \rangle$ 的启发信息,其表达式为

$$\eta_{ij} = 1/d_{ij}, \quad (2)$$

d_{ij} 表示路径 $\langle i, j \rangle$ 的欧氏距离。

1.1.3 信息素更新

当第 t 代蚂蚁完成迭代搜索之后,采用信息素更新规则对当前蚂蚁走过的路径进行信息素浓度更新,更新之后会使下一代蚂蚁选择信息素浓度高的路径,信息素浓度通过下式进行更新:

$$\tau_{ij}(t) = (1 - \rho)\tau_{ij}(t - 1) + \rho\Delta\tau_{ij}(t). \quad (3)$$

其中: ρ 为信息素挥发系数, $\Delta\tau_{ij}(t)$ 为蚂蚁本次迭代过程中在路径 $\langle i, j \rangle$ 上的信息素增量,计算公式为

$$\Delta\tau_{ij}(t) = \begin{cases} \frac{Q}{L^k}, & ij \in L^k; \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

这里 Q 为信息素强度, L^k 为第 k 只蚂蚁在此次迭代中所走路径的总长度.

1.2 改进蚁群算法

1.2.1 改进状态转移规则

在算法前期, 路径之间的信息素浓度差距较小, 影响蚂蚁路径的选择, 会导致算法的搜索效率降低. 为了提高算法前期的搜索效率, 首先对状态转移公式进行改进, 为了提高算法的搜索效率, 采用伪随机状态转移规则. 蚂蚁选择下一节点时, 将根据下式进行选择:

$$j = \begin{cases} \arg \min_{j \in \text{allowed}_k} \{[\tau_{ij}(t)]^\alpha \times [\eta_{Sj}(t)]^\beta \times [\varphi_{jE}(t)]^\chi\}, \\ q \leq q_0; \\ p_{ij}^k(t), q > q_0. \end{cases} \quad (5)$$

其中: $p_{ij}^k(t)$ 为第 k 只蚂蚁从节点 i 运动到节点 j 的概率转移公式, 其表达式为

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{Sj}(t)]^\beta [\varphi_{jE}(t)]^\chi}{\sum_{j \in \text{allowed}_k} [\tau_{ij}(t)]^\alpha [\eta_{Sj}(t)]^\beta [\varphi_{jE}(t)]^\chi}, j \in \text{allowed}_k; \\ 0, \text{ otherwise.} \end{cases} \quad (6)$$

q_0 为一个随机数, $q_0 \in (0, 1)$. η_{Sj} 为起始节点 S 到下一节点 j 的欧氏距离, φ_{jE} 为下一节点 j 到目标节点 E 的欧氏距离的倒数. α 为信息素启发因子, β 和 χ 分别为起始节点到下一节点的欧氏距离以及下一节点到目标节点的欧氏距离的倒数的相对重要性. q_0 是一种动态选择因子以自适应更新选择比例, q_0 的计算方式为

$$q_0 = \frac{1}{1 - 1.5 \times \log_2 \frac{1}{N}}, \quad (7)$$

其中 N 是当前迭代次数. q_0 随着迭代次数的增加而逐渐减少, 算法前期 q_0 取值较大, 可以提高算法在初始阶段的搜索效率, 算法中后期 q_0 取值较小, 可以保证算法的全局搜索能力, 较小的 q_0 值可以确保大多数蚂蚁依靠 $p_{ij}^k(t)$ 进行节点选择. 当 $q \leq q_0$ 时, 下一节点 $j = \arg \max_{n \in \text{allowed}_k} \{[\tau_{in}]^\alpha \times [\eta_{Sn}]^\beta \times [\varphi_{nE}]^\chi\}$; 当 $q > q_0$ 时, 则根据转移概率公式 $p_{ij}^k(t)$ 计算下一节点的选择概率, 再通过轮盘赌的方式确定下一节点.

1.2.2 改进信息素增量更新规则

基于最大最小蚂蚁模型和精英蚂蚁模型, 对信息素增量更新规则进行改进, 提出一种奖励惩罚机制更

新信息素增量, 提高算法收敛速度. 每一次迭代完成之后, 抛弃陷入死锁的蚂蚁, 计算出成功到达目标节点的蚂蚁所经过路径长度的平均值, 最优蚂蚁规划出的路径上的信息素增量由下式更新:

$$\Delta\tau_{ij}(t) = \lambda_1 \times \frac{N+1}{N_{\max}} \times \frac{Q}{L^k}. \quad (8)$$

对于最优路径, 要给予信息素浓度奖励, 以增强最优解对后续迭代的引导作用. 若蚂蚁所走的路径长度大于最优蚂蚁路径长度, 但却小于路径长度平均值, 则按下式更新该蚂蚁所在路径上的信息素增量:

$$\Delta\tau_{ij}(t) = \lambda_2 \times \frac{N+1}{N_{\max}} \times \frac{Q}{L^k}; \quad (9)$$

若蚂蚁所走的路径长度大于这一代蚂蚁所走路径长度的平均值, 则按下式更新该蚂蚁所在路径上的信息素增量, 通过对较差解的惩罚来减少较差路径对后续迭代的误导作用:

$$\Delta\tau_{ij}(t) = -\lambda_3 \times \frac{N+1}{N_{\max}} \times \frac{Q}{L^k}. \quad (10)$$

每一次迭代完之后, 先通过改进之后的信息素增量更新公式更新信息素增量, 再通过信息素全局更新公式更新信息素浓度. 其中: Q 表示信息素强度; L^k 表示第 k 只蚂蚁在此次迭代中所走路径的总长度; N_{\max} 表示总的迭代次数; N 表示当前迭代次数; λ_1 、 λ_2 、 λ_3 是大于 0 的常数, 通过奖励惩罚机制更新信息素增量, 提高算法收敛速度.

1.2.3 改进信息素更新规则

信息素浓度的更新主要用于模拟天然蚂蚁信息素随时间的积累和自主挥发. 在基本蚁群算法中, 信息素挥发系数 ρ 是一个固定的常数, 这不利于保证算法前期的全局搜索能力以及算法后期的快速收敛能力. 因此, 采用信息素自适应挥发因子更新信息素浓度. 信息素浓度更新公式如下所示:

$$\tau_{ij}(t+1) = (1 - \rho(t+1)) \times \tau_{ij}(t) + \Delta\tau_{ij}(t+1). \quad (11)$$

对信息素挥发因子 ρ 采用自适应更新策略可以保证算法前期的全局搜索能力, 可以提高算法中后期的收敛速度, 能够在提高收敛速度的同时保证路径的多样性. 信息素挥发系数 ρ 随迭代次数增加而单调递减, 当递减到最小值之后, 就不再递减, 即如下所示:

$$\rho(t+1) = \begin{cases} \frac{N_{\max} - N}{N_{\max}} \times \frac{1}{e^{1.1 - \rho(t)}}, \rho \geq \rho_{\min}; \\ \rho_{\min}, \text{ otherwise.} \end{cases} \quad (12)$$

其中: ρ_{\min} 为 ρ 的最小值, N_{\max} 为最大迭代次数, N 为当前迭代次数. 通过限制信息素浓度上下限, 可以避

免算法的早熟与停滞. 在 $[\tau_{\min}, \tau_{\max}]$ 区间内定义信息素, 如下所示:

$$\tau_{ij}(t) = \begin{cases} \tau_{\max}, & \tau_{ij}(t) \geq \tau_{\max}; \\ \tau_{ij}(t), & \tau_{\min} < \tau_{ij}(t) < \tau_{\max}; \\ \tau_{\min}, & \tau_{ij}(t) \leq \tau_{\min}. \end{cases} \quad (13)$$

1.2.4 改进蚁群算法的实现

自适应机制改进蚁群算法的具体实施步骤如下.

step 1: 首先建立二维静态栅格环境地图, 初始化主要参数. 将最大蚂蚁数量 M , 起始节点 S , 目标节点 E , 最大迭代次数 N_{\max} , 信息素浓度挥发系数 ρ , 信息素启发式因子 α , 起始点期望启发因子 β 以及目标点期望启发因子 χ 等参数初始化.

step 2: 蚂蚁路径选择. 将 M 只蚂蚁投放至起始节点, 初始化禁忌表, 并将起始节点 S 加入到禁忌表中, 根据式(5)寻找下一节点 j , 然后将节点 j 加入到禁忌表中, 并且更新路径长度, 重复这个循环, 直到蚂蚁到达目的地或发生死锁状态.

step 3: 更新信息素增量. 当一代蚂蚁完成搜索后, 抛弃未走到目标点的蚂蚁, 对最优蚂蚁按式(8)加强所经过路径上的信息素增量; 对于较优的蚂蚁(路径长度低于这一代蚂蚁路径长度平均值, 但不是最优蚂蚁), 按式(9)更新所经过路径上的信息素增量; 对于较差的蚂蚁(路径长度大于这一代蚂蚁路径长度平均值), 按式(10)减弱所经过路径上的信息素.

step 4: 更新信息素浓度. 当信息素增量更新完毕之后, 限定信息素浓度上下限, 由式(11)对走到目标点的蚂蚁进行信息素更新.

step 5: 输出最优路径. 判断当前迭代次数是否等于最大迭代次数, 如果等于最大迭代次数, 则输出最优路径长度, 算法结束; 如果小于最大迭代次数, 则继续下一代迭代搜索, 寻找最优路径, 清空禁忌表中所有节点, 转到 step 2 依次循环执行, 直到当前迭代次数等于最大迭代次数.

2 实验研究与仿真分析

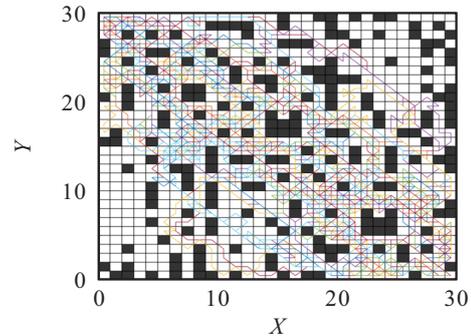
为了验证本文算法 AMACA 的可行性与有效性, 本文在 PC 机上进行大量的仿真实验, 使用基本蚁群算法(original ant colony algorithm, OACA)、自适应多态蚁群算法^[14](adaptive polymorphic ant colony algorithm, APACA)和自适应机制改进蚁群算法(AMACA) 3种算法在不同复杂程度(15×15 , 20×20 , 30×30 , 50×50 , 60×60)的二维静态栅格地图下进行移动机器人路径规划对比实验. 二维栅格地图的单位长度为 1 m, 算法运行环境为 Windows10,

处理器为 Intel(R) Core(TM) i5-7500, 主频为 3.40 GHz, 内存 12 G, 仿真软件为 Matlab r2018b.

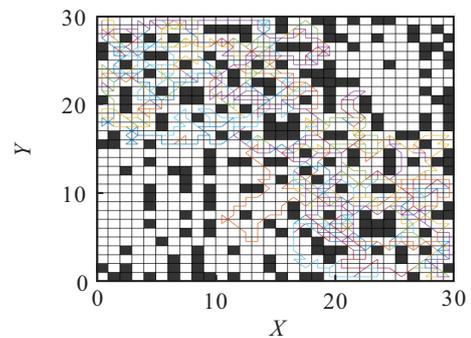
2.1 算法实用性与有效性分析

2.1.1 改进状态转移规则实验验证

为了验证状态转移规则对改进算法起到的作用, 在基本蚁群算法的基础上只加入本文改进的状态转移规则, AMACATest 为在基本蚁群算法的基础上只加入本文改进的状态转移规则的算法. 在 30×30 栅格地图下进行路径规划实验, 绘制出本文算法 AMACA 与基本蚁群算法中所有成功到达目标点的蚂蚁所走过的路径. 本文算法的参数取值设为 $M = 100$, $\alpha = 1$, $\beta = 3$, $\chi = 1.2$, $N_{\max} = 100$, $\rho = 0.8$, $\lambda_1 = 11.5$, $\lambda_2 = 5$, $\lambda_3 = 2$, $Q = 100$. 在进行算法实用性与有效性分析时, 参数配置都如上所示. 图 1 为应用 AMACATest 以及 OACA 进行移动机器人路径规划实验时成功到达目标点的蚂蚁走过的路径.



(a) AMACATest 中蚂蚁所走过的路径



(b) OACA 中蚂蚁所走过的路径

图 1 两种算法的蚂蚁爬行图

由图 1 可以看出, AMACATest 中蚂蚁走过的路径比基本蚁群算法中蚂蚁走过的路径具有更高的全局性, 成功到达目标点的蚂蚁数目更多, 这样更易于找到最优路径. 由此可以看出, 改进状态转移规则能够提高蚂蚁搜索的全局性, 避免算法陷入局部最优.

2.1.2 改进信息素增量更新规则实验验证

为了验证改进信息素增量更新规则对本文算法所起到的作用, 在基本蚁群算法的基础上只加入本文改进的信息素增量更新规则, 并且与基本蚁群算法

以及AMACA进行对比实验.主要对比3个方面,一是最优路径长度,二是最优路径长度收敛次数,三是平均路径长度收敛次数.在 30×30 栅格地图下分别采用3种算法进行路径规划仿真实验,仿真次数为10

次.实验结果如图2所示,其中AMACATest为在基本蚁群算法的基础上只加入本文改进的信息素增量更新规则的算法.

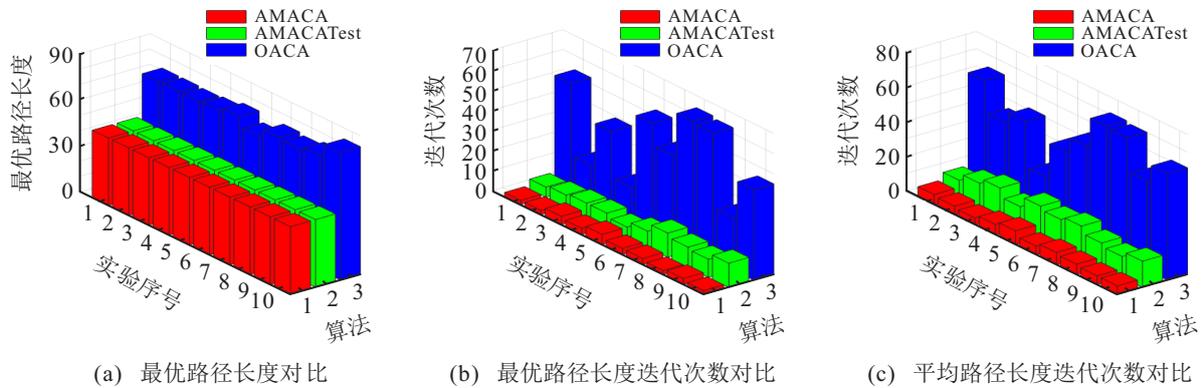


图2 改进信息素增量更新规则实验结果对比

由图2(a)可知,本文算法AMACA与AMACATest找到的平均最优路径长度分别为43.9411和44.1749,OACA找到的平均最优路径长度为72.3053;AMACATest相比于OACA寻得的最优路径长度提高了38.91%,但是没有本文算法的寻优能力强.由图2(b)可知,本文算法AMACA与AMACATest的平均最优路径长度迭代次数分别为2.3和11.7,OACA的平均最优路径迭代次数为23;AMACATest相比于OACA的最优路径迭代次数减少了49.13%.由图2(c)可知,本文算法AMACA与AMACATest的平均路径长度迭代次数分别为6和20,OACA的平均路径迭代次数为36;AMACATest相比于OACA的平

均路径迭代次数减少了44.44%.综上所述,采用奖励惩罚机制来更新信息素增量可以提高算法的收敛速度,能够使算法更快地趋于稳定状态.

2.1.3 改进信息素更新规则实验验证

为了验证改进信息素更新规则可以提高蚁群的收敛速度,并且能够寻得更优的路径,在基本蚁群算法的基础上只加入本文改进的信息素更新规则,在 30×30 的栅格地图中进行10次仿真实验,并且与本文算法AMACA以及OACA进行对比分析.实验结果如图3所示,其中AMACATest为在基本蚁群算法的基础上只加入本文改进的信息素更新规则的算法.

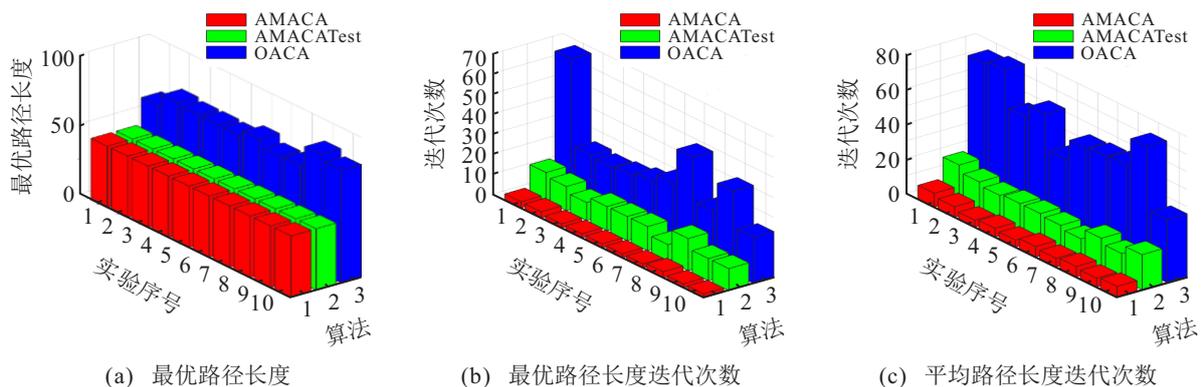


图3 改进信息素更新规则实验结果对比

由图3(a)可知,本文算法AMACA与AMACATest找到的平均最优路径长度分别为43.9411和44.0580,OACA找到的平均最优路径长度为73.1154,AMACATest相比于OACA寻得的最优路径长度提高了39.74%.由图3(b)可知,本文算法AMACA与AMACATest的平均最优路径长度迭代

次数分别为2.9和8.1,OACA的平均最优路径迭代次数为41.8,AMACATest相比于OACA的最优路径迭代次数减少了80.62%.由图3(c)可知,本文算法AMACA与AMACATest的平均路径长度迭代次数分别为5.7和14.6,OACA的平均路径迭代次数为53.6,AMACATest相比于OACA的平均路径迭代次数减少

了72.76%。采用自适应信息素挥发系数更新信息素浓度可以提高基本蚁群算法的收敛速度,提高算法的全局寻优能力。

2.2 20 × 20 仿真环境下的路径规划实验

仿真环境1为20 × 20的二维静态栅格地图,移动机器人起始节点坐标为(1, 20),目标节点为(20, 1)。本文算法的参数取值设为 $M = 100, \alpha = 1, \beta = 5, \chi = 1.2, N_{max} = 50, \rho = 0.8, \lambda_1 = 1.5, \lambda_2 = 0.8, \lambda_3 = 0.5, Q = 100$ 。在同一栅格地图下,分别采用3种算法进行路径规划仿真,仿真实验次数为10次。

如图4(a)所示,红色曲线为AMACA规划的路径,绿色曲线为文献[14]提出的APACA所规划出的路径,OACA走出的最优路径如图中的蓝色曲线所

示。本文算法AMACA所规划的路径长度比其他两种算法短,路径转折点最少,路径更加平滑,表明本文算法提高了算法的全局搜索能力,能够找到更优的路径。由图4(b)所示的3种算法最优路径收敛曲线可以看出,AMACA能够快速有效找到最优路径,收敛曲线相对平稳,基本没有波动,表明通过采用奖励惩罚机制更新信息素增量,采用改进信息素浓度更新规则可以提高算法的收敛速度。由图4(c)所示的平均路径收敛曲线可以看出,OACA的波动性大,这表明OACA不能快速有效地找到最优路径,导致平均路径波动性大,AMACA和APACA基本能在10次迭代左右达到稳定;本文提出的AMACA解决了OACA搜索效率低下、收敛速度缓慢且容易陷入局部最优的问题。

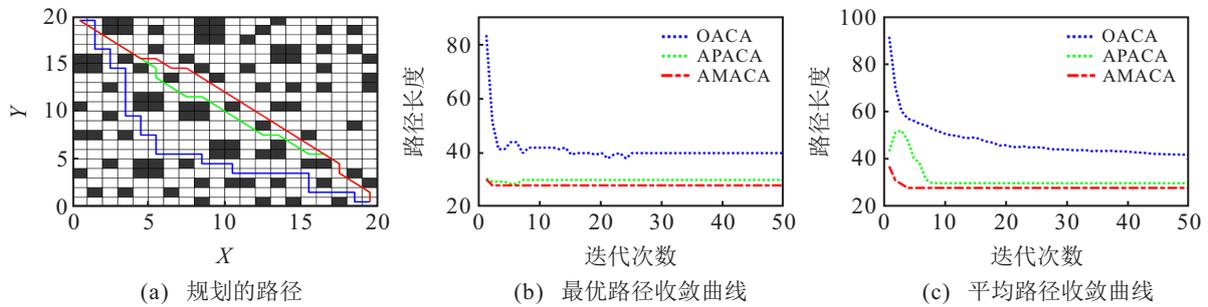


图4 20 × 20 栅格地图下3种算法仿真结果

2.3 30 × 30 仿真环境下的路径规划实验

为了对比在复杂环境下AMACA与OACA和APACA的差异,仿真环境2为30 × 30的二维静态栅格地图,机器人起始节点坐标为(1, 30),目标节点为

(30, 1)。本文算法的参数取值设为 $M = 100, \alpha = 1, \beta = 3, \chi = 1.2, N_{max} = 50, \rho = 0.8, \lambda_1 = 11.5, \lambda_2 = 5, \lambda_3 = 2, Q = 100$ 。使用文献[15]所搭建的栅格地图,分别采用3种算法进行10次路径规划仿真实验,图5为10次仿真实验结果。

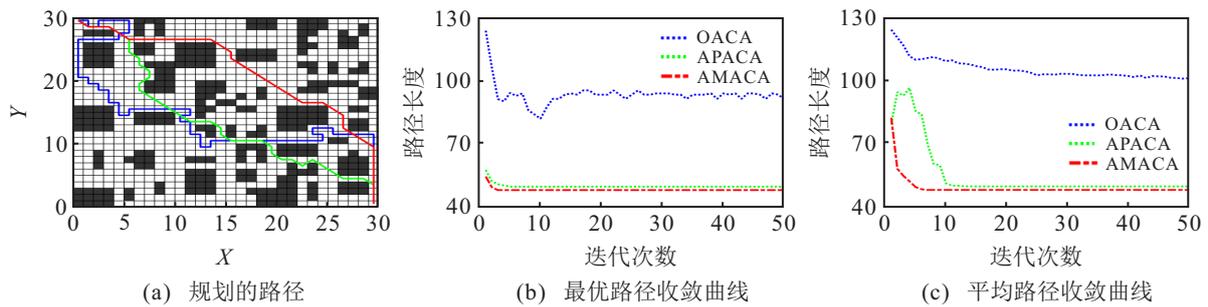


图5 30 × 30 栅格地图下3种算法仿真结果

由图5(a)可以看出,在30 × 30静态栅格地图下,就最优路径而言,本文算法AMACA搜索到的最优路径长度最短,路径转折节点最少;由图5(b)所示的3种算法最优路径收敛曲线可以看出,AMACA的收敛速度比OACA和APACA的收敛速度更快,收敛曲线更加平稳;由图5(c)所示的3种算法平均路径收敛曲线可以看出,AMACA最先达到稳定,且波动性较小,说

明本文算法最稳定,验证了本文算法在相对复杂环境下的实用性与优越性。

2.4 其他仿真环境下的路径规划实验

为了进一步验证本文算法的有效性与可行性,在不同复杂程度的障碍物(15 × 15, 50 × 50, 60 × 60栅格地图)中进行仿真实验,仿真次数为10次。表1和表2分别给出了在15 × 15, 20 × 20, 30 × 30, 50 × 50, 60

× 60 栅格地图中 AMACA、OACA、APACA 三种路径规划算法的最优路径长度、最优路径收敛迭代次数、平均路径收敛迭代次数、算法运行时间的性能对比结果. 表 3 给出了移动机器人在 15 × 15, 20 × 20, 30 × 30, 50 × 50, 60 × 60 栅格地图中进行路径规划实

验时, 本文算法 AMACA 与 OACA 以及 APACA 的相对性能对比结果. 相对性能的计算公式如下:

$$\sigma\% = \frac{x_c - x_0}{x_0} \times 100\% \quad (14)$$

其中: x_c 表示本文算法 AMACA 的实验结果, x_0 表示 OACA 或者 APACA 的实验结果.

表 1 3种算法最优路径长度与最优路径收敛迭代次数对比

地图大小	最优路径长度/m			最优路径收敛迭代次数		
	OACA	APACA	AMACA	OACA	APACA	AMACA
15 × 15	28.9313	21.5563	21.5563	22.1	8.3	1.2
20 × 20	39.8953	28.5931	28.0416	27.2	8.2	2
30 × 30	68.6932	49.6621	48.0416	31	5	3.3
50 × 50	/	81.6369	72.8112	/	12.6	6.2
60 × 60	/	138.9949	106.6102	/	19.2	10.5

注: 在 50 × 50, 60 × 60 的栅格地图下无法进行 OACA 实验, 表格中的 “/” 表示没有实验结果.

表 2 3种算法平均路径收敛迭代次数与算法运行时间对比

地图大小	平均路径收敛迭代次数			算法运行时间/s		
	OACA	APACA	AMACA	OACA	APACA	AMACA
15 × 15	不收敛	9.1	3	1.0763	2.5516	1.0846
20 × 20	不收敛	10	5.2	5.2303	6.0909	4.7876
30 × 30	不收敛	11.9	6.4	25.6602	26.1755	25.9262
50 × 50	/	18.3	10.5	/	180.2952	176.1716
60 × 60	/	26.3	15.2	/	188.8493	178.9713

注: 在 50 × 50, 60 × 60 的栅格地图下无法进行 OACA 实验, 表格中的 “/” 表示没有实验结果.

表 3 本文算法 AMACA 与 OACA 以及 APACA 的相对性能对比结果 %

地图大小	最优路径长度		最优路径收敛迭代次数		平均路径收敛迭代次数		算法运行时间	
	OACA	APACA	OACA	APACA	OACA	APACA	OACA	APACA
15 × 15	-25.49	0	-94.57	-85.54	/	-67.03	+0.77	-57.49
20 × 20	-29.71	-1.93	-92.64	-75.61	/	-48.00	-8.46	-21.40
30 × 30	-42.99	-3.26	-83.93	-34.00	/	-46.22	+1.04	-0.95
50 × 50	/	-10.81	/	-50.79	/	-42.62	/	-2.29
60 × 60	/	-23.30	/	-45.31	/	-42.21	/	-5.23
平均	-32.73	-7.86	-90.38	-58.25	/	-49.22	-2.22	-17.47

注: 在 50 × 50, 60 × 60 的栅格地图下无法进行 OACA 实验, OACA 在 4 种栅格地图下的平均路径长度无法收敛, 表格中的 “/” 表示没有实验结果.

在 15 × 15 栅格地图中, 本文算法 AMACA 与 APACA 寻得的路径相对于 OACA 所寻得的路径更优; 对于最优路径迭代次数, AMACA 相对于 APACA 与 OACA 分别减少了 94.57% 和 85.54%; 对于平均路径收敛迭代次数, AMACA 相对于 APACA 减少了 67.03%, OACA 在最大迭代次数 (50 次) 范围内无法收敛; 对于算法运行时间, 本文算法运行时间优于 APACA, 基本蚁群算法的运行时间稍微优于本文算法.

在 20 × 20 栅格地图中, 对于最优路径长度, 本文算法 AMACA 寻得的路径相对于 OACA 与 APACA 所寻得的路径更优; 对于最优路径迭代次数, AMACA 相对于 APACA 与 OACA 分别减少了 92.64% 和 75.61%; 对于平均路径收敛迭代次数, AMACA 相对于 APACA 减少了 48%, OACA 在最大迭代次数范围内无法收敛; 对于算法运行时间, 本文算法运行时间稍微优于 APACA 与 OACA.

在 30 × 30 栅格地图中, 对于最优路径长度, 本文

算法 AMACA 寻得的路径相对于 OACA 与 APACA 所寻得的路径更优,文献[15]所寻得的最优路径长度为 48.041 6,与本文算法的最优路径长度相同;对于最优路径迭代次数,AMACA 相对于 APACA 与 OACA 分别减少了 83.93% 和 34%,文献[15]的算法最优路径迭代次数为 41 次,AMACA 相对于文献[15]算法减少了 91.95%;对于平均路径收敛迭代次数,AMACA 相对于 APACA 减少了 46.22%,OACA 在最大迭代次数范围内无法收敛;对于算法运行时间,本文算法运行时间稍微优于 APACA,OACA 稍微优于 AMACA.

在 50 × 50 栅格地图中,基本蚁群算法无法寻得最优路径,只有 AMACA 和 APACA 能够寻得最优路径,对于最优路径长度,本文算法 AMACA 寻得的路径相对于 APACA 所寻得的路径更优;对于最优路径迭代次数,AMACA 相对于 APACA 减少了 50.79%;对于平均路径收敛迭代次数,AMACA 相对于 APACA 减少了 42.62%;对于算法运行时间,本文算法运行时间优于 APACA.

在 60 × 60 的栅格地图中,基本蚁群算法无法寻得最优路径,对于最优路径长度,AMACA 相对于 APACA 减少了 23.30%;对于最优路径迭代次数,AMACA 相对于 APACA 减少了 45.31%;对于平均路径收敛迭代次数,AMACA 相对于 APACA 减少了 42.21%;对于算法运行时间,本文算法运行时间稍优

于 APACA.

从 5 种栅格地图下的平均结果来看,对于最优路径长度,AMACA 相对于 OACA 与 APACA 分别减少了 32.73% 和 7.86%;对于最优路径迭代次数,AMACA 相对于 OACA 与 APACA 分别减少了 90.38% 和 58.25%;对于平均路径收敛迭代次数,AMACA 相对于 APACA 减少了 49.22%,OACA 的平均路径收敛曲线在最大迭代次数范围内无法收敛;对于算法运行时间,AMACA 相对于 OACA 与 APACA 分别减少了 2.22% 和 17.47%,本文算法相对于基本蚁群算法提升效果不大.

从整体实验结果来看,在不考虑优化时间性能的前提下,本文算法的收敛速度、搜索效率以及全局搜索能力都优于 OACA 与 AMACA,可以在运行时间上稍作舍弃.在复杂的环境下,本文算法 AMACA 依然有较好的收敛速度以及全局搜索能力.

2.5 与其他路径规划算法对比实验

为了更进一步验证本文算法的实用性与有效性,使用文献[16]所搭建的 30 × 30 栅格地图进行移动机器人路径规划实验,并且与文献[16]的蚁群算法以及 PSO (particle swarm optimization) 进行对比实验.本文算法的参数取值设为 $M = 100, \alpha = 1, \beta = 3, \chi = 1.2, N_{\max} = 50, \rho = 0.8, \lambda_1 = 11.5, \lambda_2 = 5, \lambda_3 = 2, Q = 100$. 实验结果如表 4 所示.

表 4 3 种算法实验结果对比

算法	最优路径长度 /m	最优路径长度迭代次数	平均路径长度迭代次数	算法运行时间 /s
AMACA	44.5269	4	7	18.8274
文献[16]算法	48.4250	40	59	5.5600
PSO	54.6070	51	68	7.0630

由表 4 可以看出,对于最优路径长度,本文算法相对于文献[16]算法以及 PSO 分别减少了 8.05% 和 18.46%,本文算法全局寻优能力更强,搜索出的路径更优;对于最优路径长度迭代次数,本文算法相对于文献[16]算法以及 PSO 分别减少了 90% 和 92.16%,本文算法相比于其他两种算法大幅度提升了最优路径收敛速度,搜索效率更高;对于平均路径长度迭代次数,本文算法相对于文献[16]算法以及 PSO 分别减少了 88.14% 和 89.71%,本文算法可以快速趋于稳定,提高了算法的稳定性;对于算法运行时间,本文算法的运行时间最长,时间性能是后期的研究重点.

3 结 论

为了解决基本蚁群算法应用于移动机器人路径规划时存在的搜索效率低下、收敛速度缓慢以及容易陷入局部最优等问题,本文在二维静态栅格地图中

提出一种自适应机制改进蚁群算法,并通过仿真实验验证了本文提出的路径规划算法可有效提高搜索效率与收敛速度,且能避免算法陷入局部最优.通过本文的研究得出如下结论:

- 1) 针对基本蚁群算法蚂蚁前期盲目搜索,搜索效率低下等问题,提出采用伪随机状态转移规则进行路径选择,定义了一种动态选择因子以自适应更新选择比例,引入起始节点到下一节点的距离以及下一节点到目标节点的距离作为启发信息到转移概率函数中,提高了算法全局搜索能力;
- 2) 基于最大最小蚂蚁模型以及精英蚂蚁模型,提出一种奖励惩罚机制更新信息素增量,加强优质蚂蚁对后续迭代的引导作用,减少劣质蚂蚁对后续迭代的误导作用,使得算法的收敛速度得以提高;
- 3) 引入自适应调节信息素挥发系数,提高了算法

全局搜索能力以及搜索效率,通过限制信息素的上下限,以避免算法早熟和搜索停滞。

由在不同复杂程度栅格地图下的实验结果可知,本文算法AMACA的性能相对于基本蚁群算法以及其他算法有一定的提高,但是仍存在以下不足:

1) 本文算法在改进策略时引入了多个参数,在进行实验时,需要同时调节多个参数,在一定程度上增加了参数的敏感性;

2) 本文算法并没有有效地减少算法的运行时间,这是后期需要解决的问题。

参考文献(References)

- [1] 朱大奇, 颜明重. 移动机器人路径规划技术综述[J]. 控制与决策, 2010, 25(7): 961-967.
(Zhu D Q, Yan M C. Survey on technology of mobile robot path planning[J]. Control and Decision, 2010, 25(7): 961-967.)
- [2] Luo M, Hou X R, Yang J. Surface optimal path planning using an extended dijkstra algorithm[J]. IEEE Access, 2020, 8: 147827-147838.
- [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] Yao Q F, Zheng Z Y, Qi L, et al. Path planning method with improved artificial potential field—A reinforcement learning perspective[J]. IEEE Access, 2020, 8: 135513-135523.
- [5] Zeng M R, Xi L, Xiao A M. The free step length ant colony algorithm in mobile robot path planning[J]. Advanced Robotics, 2016, 30(23): 1509-1514.
- [6] Han Z L, Wang D Q, Liu F, et al. Multi-AGV path planning with double-path constraints by using an improved genetic algorithm[J]. PLoS One, 2017, 12(7): e0181747.
- [7] Tao Q Y, Sang H Y, Guo H W, et al. Improved particle swarm optimization algorithm for AGV path planning[J]. IEEE Access, 2021, 9: 33522-33531.
- [8] Zhang P C, Xiong C, Li W K, et al. Path planning for mobile robot based on modified rapidly exploring random tree method and neural network[J]. International Journal of Advanced Robotic Systems, 2018, 15(3): 1-8.
- [9] 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 X L, Mei H. 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.)
- [10] 江明, 王飞, 葛愿, 等. 基于改进蚁群算法的移动机器人路径规划研究[J]. 仪器仪表学报, 2019, 40(2): 113-121.
(Jiang M, Wang F, Ge Y, et al. Research on path planning of mobile robot based on improved ant colony algorithm[J]. Chinese Journal of Scientific Instrument, 2019, 40(2): 113-121.)
- [11] 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.)
- [12] 马小陆, 梅宏. 基于JPS策略的ACS移动机器人全局路径规划[J]. 机器人, 2020, 42(4): 494-502.
(Ma X L, Mei H. The global path planning of ant colony system mobile robot based on jump point search strategy[J]. Robot, 2020, 42(4): 494-502.)
- [13] 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.)
- [14] Jiao Z Q, Ma K, Rong Y L, et al. A path planning method using adaptive polymorphic ant colony algorithm for smart wheelchairs[J]. Journal of Computational Science, 2018, 25: 50-57.
- [15] 冯振辉, 肖人彬. 基于混合反馈机制的扩展蚁群算法[J]. 控制与决策, DOI: 10.13195/j.kzyjc.2021.0846.
(Feng Z H, Xiao R B. Extended ant colony algorithm based on mixed feedback mechanism[J]. Control and Decision, DOI: 10.13195/j.kzyjc.2021.0846.)
- [16] 张成, 凌有铸, 陈孟元. 改进蚁群算法求解移动机器人路径规划[J]. 电子测量与仪器学报, 2016, 30(11): 1758-1764.
(Zhang C, Ling Y Z, Chen M Y. Path planning of mobile robot based on an improved ant colony algorithm[J]. Journal of Electronic Measurement and Instrumentation, 2016, 30(11): 1758-1764.)

作者简介

毛文平(1995—), 男, 硕士生, 从事移动机器人SLAM、路径规划的研究, E-mail: wp168666@163.com;

李帅永(1987—), 男, 教授, 博士生导师, 从事自动驾驶与环境感知技术、SLAM与自主导航、压缩感知与超分辨重构、工业无损检测理论等研究, E-mail: lishuaiyong@cqupt.edu.cn;

谢现乐(1998—), 男, 硕士生, 从事移动机器人SLAM后端优化的研究, E-mail: xiexles@163.com;

杨雪梅(1997—), 女, 硕士生, 从事移动机器人视觉SLAM回环检测算法的研究, E-mail: m18875141812@163.com;

聂嘉炜(1999—), 男, 硕士生, 从事动态环境下SLAM视觉里程计的研究, E-mail: jv2549829187@163.com.