

控制与决策

Control and Decision

基于滚动优化和分散捕食者猎物模型的全覆盖路径规划算法

阮贵航, 陈教料, 胥芳

引用本文:

阮贵航, 陈教料, 胥芳. 基于滚动优化和分散捕食者猎物模型的全覆盖路径规划算法[J]. *控制与决策*, 2023, 38(9): 2545–2553.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2021.2149>

您可能感兴趣的其他文章

Articles you may be interested in

[基于生物启发神经网络和DMPC的多机器人协同搜索算法](#)

Multi-robot cooperative search algorithm based on bio-inspired neural network and DMPC

控制与决策. 2021, 36(11): 2699–2706 <https://doi.org/10.13195/j.kzyjc.2020.0959>

[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 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>

[凸优化与A*算法结合的路径避障算法](#)

Convex optimization and A-star algorithm combined path planning and obstacle avoidance algorithm

控制与决策. 2020, 35(12): 2907–2914 <https://doi.org/10.13195/j.kzyjc.2019.0351>

基于滚动优化和分散捕食者猎物模型的全覆盖路径规划算法

阮贵航, 陈教料[†], 胥芳

(浙江工业大学 机械工程学院, 杭州 310013)

摘要: 针对多机器人执行全覆盖任务效果差的问题, 提出一种基于滚动优化和分散捕食者猎物模型的多机器人全覆盖路径规划算法。首先, 利用栅格地图表示作业的环境空间, 并基于栅格地图修正捕食者猎物算法中的避开捕食者奖励, 添加移动代价奖励和死区回溯机制构建分散捕食者猎物模型; 然后, 引入滚动优化方法, 避免机器人陷入局部最优, 预测周期内机器人覆盖栅格的累计奖励值作为适应度函数, 并使用鲸鱼优化算法(WOA)求解最优移动序列; 最后, 在不同环境下进行仿真实验, 得到的平均路径长度与生物激励神经网络算法(BINN)和牛耕式A*算法(BA*)相比分别减少了16.69%~17.33%、10.32%~20.03%, 验证了所提出算法在多机器人全覆盖路径规划中的可行性和有效性。

关键词: 多机器人; 全覆盖路径规划; 栅格地图; 分散捕食者猎物模型; 滚动优化; 鲸鱼优化算法

中图分类号: TP242

文献标志码: A

DOI: 10.13195/j.kzyjc.2021.2149

开放科学(资源服务)标识码(OSID):



引用格式: 阮贵航, 陈教料, 胥芳. 基于滚动优化和分散捕食者猎物模型的全覆盖路径规划算法[J]. 控制与决策, 2023, 38(9): 2545-2553.

Complete coverage path planning algorithm based on rolling optimization and decentralized predator-prey model

RUAN Gui-hang, CHEN Jiao-liao[†], XU Fang

(College of Mechanical Engineering, Zhejiang University of Technology, Hangzhou 310013, China)

Abstract: A multi-robot complete coverage path planning algorithm based on rolling optimization and decentralized predator-prey models is proposed to solve the problem of poor performance of multi-robot performing coverage tasks. The raster map is used to represent the environment space of the job. And based on the raster map, the reward function of avoiding predators in the predator-prey algorithm is modified, the moving cost reward function and dead-zone backtracking mechanism are added to build a decentralized predator-prey model. The rolling optimization method is introduced to avoid the robot falling into local optimum. The cumulative reward value of the robot covered grid during the prediction period is used as the fitness function, and the optimal movement sequence is solved using the whale optimization algorithm (WOA). Finally, simulation experiments are carried out in different environments. Compared with the biologically inspired neural network algorithm (BINN) and the boustrophedon-A* algorithm (BA*), the average path length planned by the proposed method is reduced by 16.69%~17.33% and 10.32%~20.03%, respectively, which verifies the feasibility and effectiveness of the proposed method in multi-robot full coverage path planning.

Keywords: multi-robot; complete coverage path planning; raster map; decentralized predator-prey model; rolling optimization; whale optimization algorithm

0 引言

全覆盖路径规划是机器人路径规划中一个重点问题, 其主要任务是为机器人寻找到一条可到达空间中各个位置的路径^[1]. 全覆盖路径规划在排雷、

清洁和巡检领域发挥了巨大作用, 现已成为当代移动机器人学研究的重要内容^[2-4]. 目前机器人覆盖路径规划方法的研究主要集中于覆盖效率, 在确保能够覆盖目标区域和较低重复率的前提下, 减少覆盖的路径

收稿日期: 2021-12-12; 录用日期: 2022-04-15.

基金项目: 国家重点研发计划项目(2018YFC1309404); 浙江省公益技术应用研究项目(LGG18E050023).

[†]通讯作者. E-mail: jlchen@zjut.edu.cn.

长度^[5]. 多机器人覆盖路径规划(multi-robot coverage path planning, MCPP)相较于单机器人覆盖路径规划(coverage path planning, CPP)可更快地完成覆盖任务,并具有较好的系统鲁棒性,逐渐成为覆盖路径规划的热点.

为了解决 MCPP 问题,一类方法是将工作空间划分为多个子区域,将机器人分配至对应的子区域执行覆盖任务. Hungerford 等^[6]根据机器人的起始位置,将环境分解为若干个 Voronoi 单元,机器人覆盖各自的 Voronoi 单元^[7]. Sun 等^[8]利用环境的拓扑结构将地图划分为多个区域,使用遗传算法(genetic algorithm, GA)将任务分配给机器人. 罗志远等^[9]通过模糊 C 均值聚类划分子区域,使用分步遗传算法规划子区域内的机器人路径. Nair 等^[10]提出了基于测地线曼哈顿距离的 Voronoi 分区覆盖方法(geodesic-Manhattan Voronoi-partition-based coverage, GM-VPC),有效地划分工作空间. Li 等^[11]利用矩形分解法划分工作区域,通过遗传算法将子区域覆盖任务分配给各机器人,并借助 Dijkstra 算法完成子区域间的路径规划. 上述方法是以解耦的形式将 MCPP 问题转换为 CPP 问题,简化了规划的复杂度,但面对未知环境则难以划分合理的子区域分配给机器人,易出现各机器人工作负载不均衡的现象. 另一类方法是建立机器人运动模型或构造运动模板指引机器人完成覆盖任务. 符小卫等^[12]建立了无人机运动模型和搜索模型,并结合预测控制完成覆盖任务. Luo 等^[13]提出了基于生物激励神经网络法(biologically inspired neural network, BINN)的全覆盖算法,根据神经元活性引导机器人向未覆盖区域覆盖移动. 朱大奇等^[14]将 BINN 与模板策略相结合,实现了多 AUV 在水下的全覆盖作业任务. Hassan 等^[15]提出了一种捕食者猎物全覆盖法(predator-prey coverage path, planning, PPCPP),根据建立的机器人奖励值决策模型,指引机器人向奖励值最大的区域移动. Lin 等^[16]设计了一个成本函数评估机器人覆盖运动的收益,通过考虑单个机器人的局部增益和目标选择的全局增益,优化整体覆盖效率. Sanna 等^[17]借助预先训练的人工神经网络引导无人机运动,从而得到每一步覆盖的最大化收益. 但上述方法会引起机器人覆盖范围重叠的现象,导致路径重复率增高.

多机器人在执行覆盖任务时,会出现机器人陷入死区的现象,此时机器人需要回溯到新的未覆盖区域继续执行覆盖任务. Hazon 等^[18]提出了一种回溯的多机器人生成树覆盖算法,基于螺旋生成树覆

盖算法(spanning tree coverage algorithm, STC)构造覆盖路径,允许机器人顺时针和逆时针运动以实现全覆盖,当机器人陷入死区时,借助 A* 算法脱离死区. Choi 等^[19-20]基于 boustrophedon 运动和寻路算法 A*、Theta* 分别提出了 BA* (boustrophedon-A*) 和 B-Theta* (boustrophedon-theta*) 算法,缩短了机器人回溯路径长度,提高了覆盖效率. Viet 等^[21]提出了一种基于市场的多机器人覆盖路径规划算法,其中基于贪婪 A* 算法的回溯机制,可在机器人陷入死区时指导其移动至最近的未覆盖区域. Senthilkumar 等^[22]提出了多个扩展生成树的覆盖算法,有效地完成了全覆盖任务. 但上述方法的回溯机制效率低,没有解决多个机器人回溯同一区域以及回溯区域被其他机器人抢先覆盖的问题.

针对当前 MCPP 研究中,面对未知环境难以较好地实现机器人任务分配,现有的基于运动模型或模板法的覆盖方法覆盖范围重叠高和回溯过程中回溯机制效率低等问题,本文提出一种基于滚动优化和分散捕食者猎物模型的多机器人全覆盖路径规划算法. 通过分散捕食者猎物模型中的避开捕食者机制,使得机器人在覆盖过程中相互远离,间接地实现对机器人的任务分配,同时也降低了机器人覆盖范围重叠的概率;根据死区回溯机制,阻止其他机器人覆盖回溯区域和多个机器人回溯同一区域;以分散捕食者猎物模型为预测模型进行滚动优化决策,使得在机器人覆盖过程中根据环境信息和机器人状态不断调整覆盖路径,从而有效地实现多机器人的全覆盖任务.

1 环境建模和多机器人全覆盖问题描述

环境空间建模常用的方法有栅格地图法^[23]、拓扑地图法^[24]和几何地图法^[7]. 栅格地图具有二值性,相较于其他建模方法可更为明显地表示出工作空间障碍物的占用情况,广泛应用于机器人的路径规划. 因此本文采用栅格地图法将环境空间分解为固定大小的栅格,并利用状态属性表示每个栅格的覆盖情况. 状态属性共有 3 种状态值: 1、0 和 -1 分别对应未覆盖且不为障碍物的自由栅格、已覆盖栅格和障碍物栅格,并随着机器人对地图空间的覆盖而不断更新. 工作空间的栅格状态可表示为

$$I(O_i) = \begin{cases} 1, & O_i \in FC; \\ 0, & O_i \in CC; \\ -1, & O_i \in OC. \end{cases} \quad (1)$$

其中: $I(O_i)$ 为栅格状态, O_i 为第 i 个栅格, FC 为自由栅格区域, CC 为已覆盖栅格区域, OC 为障碍物栅格区域.

在建立的栅格地图上,将栅格大小设置为机器人静态覆盖区域大小,并假设机器人一旦经过栅格中心,该栅格便被认为已经覆盖. 栅格地图由行和列组成,可用 (x, y) 表示对应的栅格位置. 假设机器人数目为 n ,则机器人的覆盖路径可表示为

$$\text{Path}_k = [(x_0, y_0), (x_1, y_1), \dots, (x_l, y_l)]. \quad (2)$$

其中: k 为第 k 个机器人, $k \in [1, n]$; Path_k 为第 k 个机器人的覆盖路径; $(x_l, y_l) \in \text{FC} \cup \text{CC}$, l 为机器人 k 的路径长度; $\text{FC} \cup \text{CC}$ 为无障碍区域. 因此,多机器人全覆盖路径规划任务可描述为找到一组机器人的覆盖路径,使得 $\text{Path}_1 \cup \text{Path}_2 \cup \dots \cup \text{Path}_n = \text{FC} \cup \text{CC}$,即工作空间中任意可达区域至少被机器人覆盖1次,也意味着全覆盖的实现.

2 捕食者猎物模型

2.1 方法原理

在捕食者猎物模型中,与机器人所在栅格相连的栅格被视为机器人的临近点,临近点最大数量为8个. 机器人选择临近点中的自由栅格作为机器人下一步移动的候选栅格,并计算当前位置到候选栅格得到的奖励值,将候选栅格中奖励值最大的栅格作为机器人下一步移动的目标位置. 奖励值函数由避开捕食者奖励函数、平滑度奖励函数和边界奖励函数组成,如下式所示:

$$R = R_d + \omega_s R_s + \omega_b R_b, \quad (3)$$

其中 ω_s 、 ω_b 分别为平滑度奖励函数和边界奖励函数的权重因子.

2.1.1 避开捕食者奖励函数

避开捕食者奖励函数为

$$R_d = \frac{D(O_j) - D_{\min}(O_{k(t)})}{D_{\max}(O_{k(t)}) - D_{\min}(O_{k(t)})}. \quad (4)$$

其中: $O_{k(t)}$ 为机器人 k 在 t 时刻所处位置, $D_{\min}(O_{k(t)})$ 、 $D_{\max}(O_{k(t)})$ 分别为机器人 k 临近点与捕食者间的最小距离和最大距离, O_j 为临近点 j 所在位置, $D(O_j)$ 为临近点 j 与捕食者间的距离.

2.1.2 平滑度奖励函数

平滑度奖励函数为

$$R_s = \frac{\angle O_{k(t-1)} O_{k(t)} O_j}{180^\circ}, \quad (5)$$

其中 $\angle O_{k(t-1)} O_{k(t)} O_j$ 为机器人 k 当前移动方向与下一步移动方向之间的夹角.

2.1.3 边界奖励函数

边界奖励函数为

$$R_b = \frac{n^{N_{\max}} - n^N(O_j)}{n^{N_{\max}}}. \quad (6)$$

其中: $n^N(O_j)$ 为在位置 O_j 处属于自由栅格的临近点数量, $n^{N_{\max}}$ 为临近点的最大个数.

3 分散捕食者猎物模型

针对捕食者猎物模型应用于多机器人覆盖路径规划时,无法分散机器人、覆盖范围的重叠率高等问题,本文提出了一种分散捕食者猎物模型,对避开捕食者奖励函数进行修正,并添加了移动代价函数和死区回溯机制,以解决捕食者猎物模型中存在的缺陷和多机器人覆盖中回溯质量低的问题.

3.1 修正避开捕食者奖励函数

该函数决定了机器人在空间运动的趋势,使得机器人优先覆盖远离捕食者的区域,逐步向捕食者靠拢,如图1所示. 但在多机器人全覆盖任务中,这种运动特性会造成机器人在运动过程中向远离捕食者的一端靠拢,引起机器人覆盖区域重叠,降低了规划路径的质量.

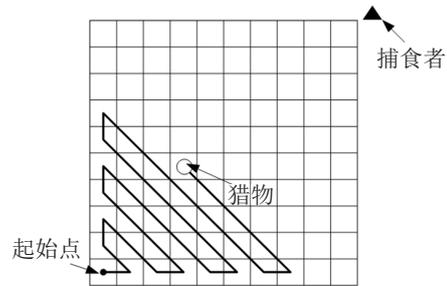


图1 机器人避开捕食者的覆盖路径

修改后的避开捕食者奖励函数,将规划的机器人自身作为猎物,除自身以外的机器人作为捕食者,借助机器人避开捕食者这一特性,使得机器人在覆盖过程中相互远离,实现对机器人工作区域的动态划分. 由于机器人的位置在不断变化,且机器人每次移动均需要计算避开捕食者奖励,为减少系统的计算量,只取当前机器人规划范围内的捕食者进行计算,并根据捕食者与机器人间的距离赋予不同的权重. 修正后的避开捕食者奖励函数为

$$R'_d(O_{k,j}) = \begin{cases} \sum_{m=1}^H \frac{1}{D(P_{k,m}, O_{k,j})} R_d, & H > 0; \\ 1, & H = 0. \end{cases} \quad (7)$$

其中: $D(\cdot)$ 为两栅格间的欧氏距离; $O_{k,j}$ 为机器人 k 临近点 j 所处位置; H 为在机器人 k 规划范围内捕食者的个数; $P_{k,i}$ 、 $P_{k,m}$ 分别为机器人 k 规划范围内的第 i 个和第 m 个机器人, $i, m \in [1, H]$.

3.2 新增移动代价奖励函数

机器人在移动过程中,过多的转弯和对角运动的消耗均会增加系统的能耗.平滑度奖励函数鼓励机器人保持直线移动,在遇到边界或进入死区时才改变方向,从而减少机器人的转弯次数.但这也造成了机器人在对角运动时,平滑度奖励也保持一个较大的值,这样的对角路径易将未覆盖区域分割为多块子区域,降低覆盖路径质量,增加回溯次数.本文新增移动代价函数,可在机器人对角运动时,使得机器人旋转 45° ,将对角运动转换为水平或竖直运动.移动代价奖励函数为

$$E_{\text{step}}(O_{k,j}) = \frac{1}{D(O_{k,j}, O_k)}, \quad (8)$$

其中 $D(O_{k,j}, O_k)$ 为机器人 k 移动至临近点 j 所需经过路径的长度.

由式(5)~(8),得到如下总奖励函数:

$$E_{\text{total}} = \alpha_1 R'_d + \alpha_2 E_{\text{step}} + \alpha_3 R_s + R_b, \quad (9)$$

其中 α_1 、 α_2 和 α_3 为权重系数,取值范围为 $[0,1]$.

3.3 新增机器人死区回溯机制

面对复杂的覆盖环境,机器人易于覆盖过程中陷入死区,需借助回溯法脱离死区.本文建立了一套机器人死区回溯机制,以提高机器人的回溯性能.

回溯点的选择.常用的死区脱离方法将贪心算法与欧氏距离相结合,将死区点与回溯点间的距离作为贪心算法的选择策略,每次回溯机器人选择与死区位置 S 最近的回溯点 S_b 作为目标点 S_g ,如下式所示:

$$S_g = \operatorname{argmin}(D(S, S_b)). \quad (10)$$

这种贪心策略在单机器人覆盖任务中是有效的,但在多机器人覆盖任务中,与陷入死区的机器人距离最近的回溯点可能在其他机器人覆盖范围内.其他机器人可能在当前机器人回溯过程中已将回溯点覆盖,使得当前机器人反复陷入死区.多个陷入死区的机器人可能会选取同一区域的回溯点,使得机器人覆盖范围重合造成资源浪费.

所提出的机器人死区回溯机制,将机器人回溯点半径 r 内的栅格区域作为机器人的回溯区域,同时拒绝其他机器人选择回溯区域自由栅格作为回溯点,如下式所示:

$$L_{\text{back},r} = \{X \mid \|X - O_{\text{back}}\| < r, X \in \text{FC}\}, \quad (11)$$

$$L_{\text{search}} = L_{\text{return}}/L_{\text{back},r}. \quad (12)$$

其中: $L_{\text{back},r}$ 为回溯点半径 r 内点的集合, O_{back} 为回溯点位置, L_{return} 为回溯点的集合, L_{search} 为回溯搜索列表.

当其他机器人选择回溯区域作为下一步移动的路径时,给予在回溯点半径 r 内临近点的奖励值惩罚,如下式所示:

$$E = c \cdot E_{\text{total}}. \quad (13)$$

其中: c 为激励函数惩罚系数, E_{total} 为累计奖励值.

由式(9)和(13),得到如下分散捕食者猎物模型的总奖励值函数:

$$E(O_{k,j}) = \begin{cases} E_{\text{total}}, & O_{k,j} \notin L_{\text{back},r}; \\ c \cdot E_{\text{total}}, & O_{k,j} \in L_{\text{back},r}. \end{cases} \quad (14)$$

由式(14)可求得机器人 k 在 j 时刻临近点总奖励值,进而得到最大奖励值临近点的位置,确定机器人下一步移动位置,即

$$O_{k,j}^* = \operatorname{argmax}(E(O_{k,j} \in N(O_k))), \quad (15)$$

其中 $N(O_k)$ 为机器人 k 临近点中为自由栅格且不与其他机器人目标点的坐标集合.机器人 k 在计算其下一个最佳临近点时,首先会与其他机器人通讯,更新其他机器人的位置和目标点.当 $N(O_k)$ 为空时,将从 L_{search} 中选出回溯点,利用点对点规划下一步移动的位置为 $O_{k,j}^*$.

4 基于滚动优化的决策方法

当机器人面对地图环境未知时,需要根据探测到的环境信息更新地图,通过滚动优化决策可在有限域内进行局部优化以适应不确定的环境,避免机器人陷入局部最优.

4.1 滚动优化

滚动优化方法可利用机器人实时探测到的局部环境信息,以滚动的方式进行在线规划.每滚动一步,机器人更新采集的环境信息,不断地规划子路径,从子路径中选择效益最高的子路径,反馈给机器人并指导机器人移动.

在已知机器人 k 的总激励函数中各权重 α_1 、 α_2 、 α_3 的大小和机器人 k 当前时刻 t 的运动状态 $O_{k(t)}$ 的情况下,通过式(15)可预测机器人下一步的移动位置,由于后续涉及到未来 T 时刻的预测,对于式(15)中部分符号进行如下修改:

$$O_{k(t+1)}^* = \operatorname{argmax}(E(O_{k(t),j} \in N(O_{k(t)}))). \quad (16)$$

其中: $O_{k(t),j}$ 为机器人 k 在 t 时刻的临近点 j 所处位置, $O_{k(t+1)}^*$ 为预测机器人在 $t+1$ 时刻所处位置.由式(16)可预测机器人在未来 T 时刻的移动序列为

$$y_{k(t)} = [O_{k(t+1)}^*, O_{k(t+2)}^*, \dots, O_{k(t+T)}^*], \quad (17)$$

其中 $O_{k(t+T)}^*$ 为预测得到的机器人 k 在未来 T 时刻的所处位置.

根据预测的 T 步的移动序列, 将每一步的机器人奖励 E 累加, 从而得到 T 步累计覆盖效益函数为

$$L(T) = \sum_{k=0}^n E_k = \sum_{k=0}^n \sum_{i=t}^{i+T-1} E(O_{k(t+i)}^*). \quad (18)$$

其中: E_k 为机器人 k 经过 T 时刻的累计奖励函数值, n 为机器人总数.

通过优化求解 T 步累计覆盖效益函数 $L(T)$, 可确定预测的机器人未来 T 步的最佳移动序列 $y_{k(t)}$. 鲸鱼优化算法 (whale optimization algorithm, WOA) 是一种启发式全局优化算法, 主要由包围猎物、气泡攻击以及寻找猎物 3 个阶段组成, 在包围猎物阶段, 鲸鱼个体会向最佳个体方向移动; 在气泡攻击阶段, 根据 $[0, 1]$ 间的随机概率选择气泡网捕食或收缩包围, 各鲸鱼逐渐包围当前最优解; 在寻找猎物阶段, WOA 根据鲸鱼彼此间的距离更新位置, 达到随机搜索的目的^[25]. WOA 具有易于实现、参数设置少和鲁棒性强等优点, 因此本文选择 WOA 优化求解式 (18), 寻求未来 T 步最大的累计覆盖效益值. 并将最佳移动序列中最靠前的序列作用于机器人 k 引导其移动, 移动后更新机器人 k 的状态和地图信息. 在 $t + 1$ 时刻, 根据更新后机器人 k 的状态求解新的移动序列. 不断地重复这个过程, 使得机器人滚动前进, 最终实现覆盖效益函数的最大化.

4.2 整体覆盖流程

图 2 为单个机器人角度基于滚动优化和分散捕食者猎物模型的滚动优化方法 (rolling-decentralized predator-prey coverage path planning, R-DPPCPP) 的覆盖流程, 具体步骤如下.

step 1: 将工作空间栅格化, 设置机器人的初始位置、规划范围和回溯惩罚系数等相关参数.

step 2: 设置滚动优化的预测步数和 WOA 基本参数.

step 3: 搜索机器人的临近点, 若 $N(O_k)$ 不为空集, 则根据式 (15) 寻找机器人下一步的移动位置; 若 $N(O_k)$ 为空集, 则从式 (12) 中选择回溯点, 采用点对点规划决定下一步移动位置.

step 4: 记录预测出的机器人下一步的移动序列, 重复 step 3 直至 T 步路径预测完毕或在预测中地图已完全覆盖.

step 5: 通过式 (18) 计算机器人未来 T 步累计效益函数 $L(T)$, 并将其作为 WOA 的适应度函数; 使用 WOA 优化求解式 (18), 得到未来 T 步机器人的最优移动序列.

step 6: 将最优移动序列最靠前的序列作为机器人下一步移动的位置, 同时通讯更新机器人的位置状态信息和栅格地图信息.

step 7: 判断是否完成全覆盖, 若完成全覆盖任务, 跳转至 step 8; 否则跳转至 step 2.

step 8: 输出最优路径.

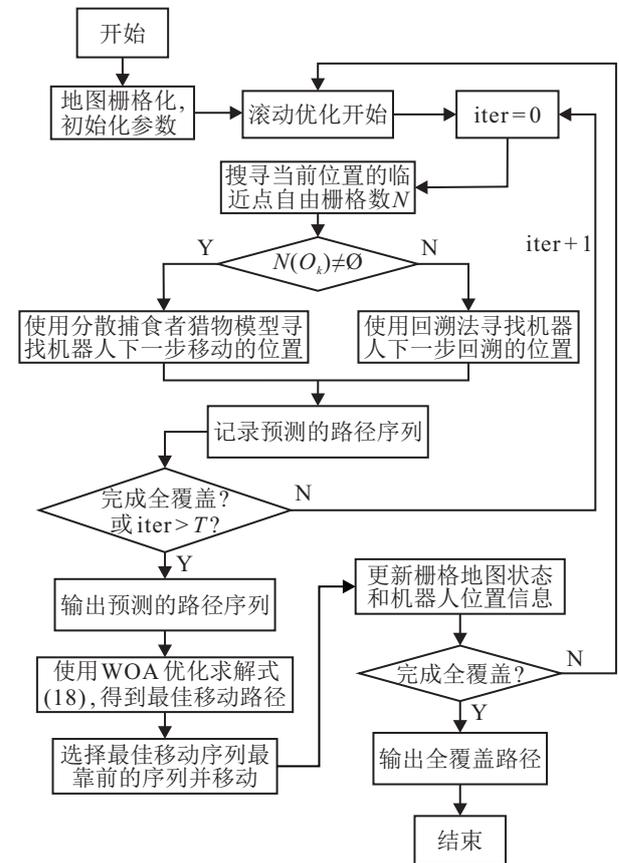


图 2 覆盖流程

4.3 计算复杂度

假设关于环境信息已覆盖栅格、未覆盖栅格和障碍物栅格的存储为二进制向量, 通过索引可以更新, 可认为是计算复杂度的常数, 时间复杂度为 $O(n)$, 使用 $k-d$ 树数据结构存储可覆盖栅格, 利用二进制向量更新其状态, 则找到最大奖励值邻居的复杂度为 $O(m)$. 对于滚动优化是找到 T 步最大奖励值临近点的复杂度 $O(m \times T)$, 鲸鱼优化算法的时间复杂度为 $O(P \times I)$, 则滚动优化求解最大奖励值临近点为 $O(m \times T \times P \times I)$. 回溯的复杂度取决于死区脱离方法, 所有算法均需要扫描和更新的过程, 本文分析中不再计算. 因此在忽略回溯方法以及地图的扫描和更新的情况下, 算法的复杂度为 $O(m \times T \times P \times I) + O(n)$, 其中 n 、 m 、 T 、 P 和 I 分别为机器人数目、工作空间的栅格数、预测步数、鲸鱼优化算法中的种群数量和迭代次数.

5 仿真实验与分析

为了验证所提出算法,对多机器人全覆盖路径规划的可行性和有效性进行仿真实验,在 Intel (R) Core(TM) i5-4 790 CPU, 3.60 GHz, 8 GB 内存的服务器中使用 Matlab 2018 b 进行仿真.

5.1 R-DPPCPP的改进性能验证

为了验证分散捕食者猎物模型和滚动优化方法的有效性,分别将两种方法单独作用于多机器人的全覆盖任务. 仿真中障碍物随机生成,环境为 20×20 的栅格,通讯条件为理想状态, R-DPPCPP 中激励函数惩罚系数为 $c = 0.5$, 预测步数为 $T = 3$, 规划范围为 $H = 5$, 回溯半径为 $r = 3$, 机器人最大临近点个数为 $n^{N_{\max}} = 8$, 机器人数量为 $n = 3$; WOA 中最大迭代次数设置为 50, 种群数量为 20. 分别对 PPCPP、通过滚动优化后的 PPCPP (RPPCPP)、基于分散捕食者猎物模型的 PPCPP (DPPCPP) 和 R-DPPCPP 进行 20 次仿真实验, 由于仿真结果相类似, 选取其中一次覆盖路径如图 3 所示.

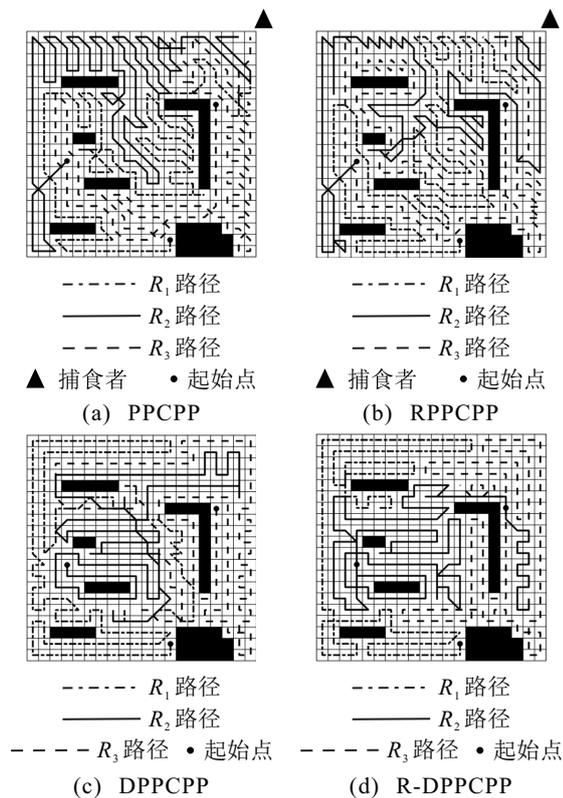


图 3 改进方法分别作用于 PPCPP 的路径结果

对角运动使得机器人单步路径增加且易产生子区域, 路径交汇使得机器人重复路径增加, 这些均造成了机器人路径质量下降. 图 3(a) 为 PPCPP 规划的多机器人路径, 路径中存在较多的对角运动, 且在捕食者附近存在大量的路径交汇现象. 图 3(b) 为 RPPCPP 规划的路径, 靠近捕食者区域的路径交汇

大幅减少, 但是仍然存在较多的对角运动. 图 3(c) 为 DPPCPP 规划的路径, 对角运动得到有效遏制, 但当多个机器人在同一区域覆盖时, 路径交汇率大幅提升. 当两种方法均优化时, 此时路径性能是最好的, 路径交汇和机器人对角运动均是最少的, 整体路径平滑, 如图 3(d) 所示.

表 1 为在图 3 环境中, PPCPP、RPPCPP、DPPCPP 和 R-DPPCPP 的全覆盖路径规划的实验结果. 每次规划机器人的起始位置随机生成, 所有结果取 20 次实验数据平均值. 由表 1 可见, RPPCPP 与 DPPCPP 的回溯次数均优于 PPCPP. 当机器人进入死区后, 需通过回溯以脱离死区. 但在机器人回溯移动时不产生奖励值, 所以滚动优化决策将会选择不回溯或回溯路径短的路径, 从而有效地减少回溯次数. DPPCPP 利用机器人回溯时的信息, 对靠近回溯区域的机器人给予奖励值惩罚, 同时拒绝多个机器人回溯同一区域, 减少了由于回溯点提前被覆盖而导致的重复回溯次数.

表 1 各改进方法在图 3 环境中 20 次仿真平均值对比

	PPCPP	RPPCPP	DPPCPP	R-DPPCPP
对角长度	95.49	44.12	58.80	26.49
回溯次数	19	10	13	9
回溯长度	95.49	44.12	58.80	26.49

融合了滚动优化与分散捕食者猎物法的 R-DPPCPP, 较 RPPCPP 回溯次数只相差 1 次, 但回溯长度却只有 RPPCPP 的 60.04%. 分散捕食者猎物法中的避开捕食者奖励函数有效地分离了机器人覆盖区域, 为滚动优化提供了更大的优化空间, 使得每次回溯的回溯距离减少. RPPCPP 通过滚动优化使得回溯区域被提前覆盖以及多个机器人回溯同一区域的概率减少, 从而融合后的 R-DPPCPP 回溯次数相对于 RPPCPP 提升并不明显.

5.2 不同环境下的对比实验验证

为了验证 R-DPPCPP 能够有效地解决多机器人全覆盖任务, 将 R-DPPCPP 与 BINN^[13] 和 BA*^[21] 算法进行对比实验, 机器人的数目为 4 个, R-DPPCPP 其余参数设置不变. 本文根据障碍物分布、障碍物形状和地图尺寸这 3 个特点选取 3 个环境空间执行全覆盖仿真, 环境 1 和环境 2 为 30×30 的栅格, 环境 3 为 50×50 的栅格, 障碍物分布如图 4 中黑色区域所示.

图 4 环境中分别使用 R-DPPCPP、BINN 和 BA* 算法进行路径规划, 由于机器人起始位置不同会影响全覆盖路径规划的结果, 为了防止实验出现偶然性, 每次仿真根据机器人数目选取 20 组机器人的起始位置, 同次模拟中 R-DPPCPP、BA* 与 BINN 的机器人初

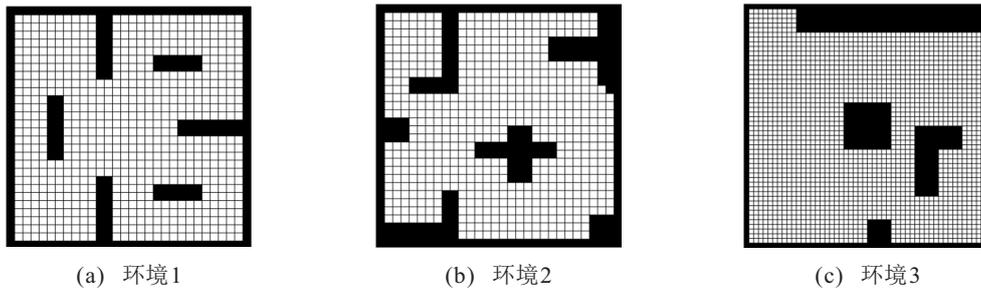


图4 模拟环境场景

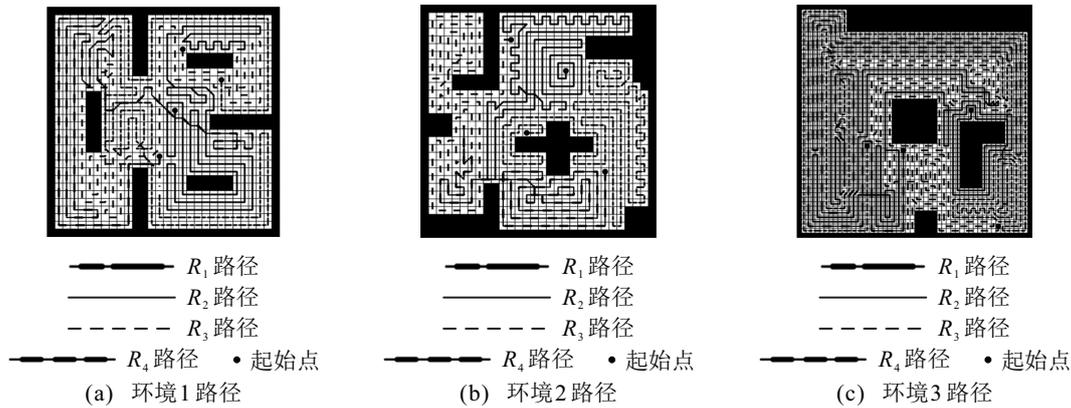


图5 R-DPPCPP在环境1、环境2和环境3中规划的路径

始位置相同. R-DPPCPP算法对每个环境均规划了全覆盖路径,从每个环境中随机抽取了一个路径规划结果,如图5所示,不同线型的线段为各个机器人的行走路径. 由图5可见,面对环境中障碍物增多、地图边界变化以及地图环境扩大的情况,R-DPPCPP依然可以完成对工作区域的覆盖规划任务.

R-DPPCPP、BA*与BINN同一环境下仿真结果相似,部分仿真数据如表2所示. 表2为机器人数目为4个时,不同算法应用于各机器人在环境1、环境2和环境3中规划得到的路径长度,加粗数据为使用同种算法的4个机器人中最长覆盖路径长度, $R_1 \sim R_4$ 为机器人. 多机器人系统的覆盖时间取决于覆盖最长

路径的机器人所需要的时间. 由表2可见,R-DPPCPP规划得到的机器人最长覆盖路径优于BINN与BA*算法,表明了R-DPPCPP的覆盖效率高于BINN与BA*算法. 但是R-DPPCPP的机器人路径长度的极差大于BINN与BA*,工作量分配均匀度低. 这是因为在覆盖规划后期,自由栅格数目减少,且存在于其他机器人覆盖范围中,此时机器人通过回溯带来时间上的回报是较小的,甚至会因为资源竞争延长覆盖时间. R-DPPCPP将其他机器人覆盖区域的点从机器人的候选回溯点中删除,使得部分机器人在规划后期不再进行回溯,有效避免了这种现象,降低了覆盖的重复率,但同时舍弃了部分机器人路径的均匀度.

表2 不同算法在图4环境中各机器人路径长度

方法	机器人	环境1路径长度			环境2路径长度			环境3路径长度		
		1	2	3	1	2	3	1	2	3
R-DPPCPP	R_1	179.5	203.4	177.7	171.9	207.3	171.5	532.8	437.5	508.3
	R_2	186.4	169.8	185.5	174.1	173.6	178.7	517.7	568.1	554.6
	R_3	196.9	172.3	179.1	193.1	142.7	141.5	555.2	580.0	551.5
	R_4	191.7	197.1	209.6	156.2	148.9	187.0	531.7	577.4	536.5
	极差	17.4	33.6	31.9	36.9	64.6	45.5	37.5	142.5	46.3
BINN	R_1	208.5	229.6	217.7	199.5	216.1	211.1	623.1	672.3	589.1
	R_2	206.3	214.3	215.6	191.7	206.8	222.4	616.8	674.2	603.3
	R_3	213.7	225.7	218.7	194.1	213.0	212.7	626.9	658.9	597.9
	R_4	208.4	217.3	208.4	201.0	197.9	217.9	639.5	681.2	595.0
	极差	7.4	15.3	10.3	9.3	18.2	11.3	22.7	22.3	14.2
BA*	R_1	206.3	227.1	214.9	206.0	220.3	221.5	565.0	673.9	583.1
	R_2	203.4	228.7	211.5	209.6	219.3	224.8	560.1	677.2	576.1
	R_3	204.7	226.7	210.5	208.7	218.1	219.8	561.7	674.9	573.2
	R_4	205.9	225.3	210.7	204.1	220.6	217.4	562.1	676.0	576.1
	极差	2.9	3.4	4.4	5.5	2.5	7.4	4.9	3.3	9.9

环境2相较于环境1障碍物更多,机器人需要覆盖的栅格数减少,但应用BA*算法的机器人总覆盖路径长度没有明显地减少。这是因为BA*只能重复进行往复运动,进入死区后寻找自由栅格然后脱离。障碍物的增多使得BA*算法规划的路径中重复路径长度增加,路径质量差。R-DPPCPP利用环境信息和机器人覆盖情况规划路径,在不同环境和起始位置规划动态调整分散捕食者模型中的参数以获取最大的累计奖励值,实现根据环境调整机器人路径的目的,总覆盖路径长度随着覆盖栅格数的减少而减少。

环境3相较于环境1与环境2扩大了需要覆盖的空间,由表2可见,R-DPPCPP的总覆盖长度和机器人最长覆盖长度仍然小于其他两种算法,表明R-DPPCPP相对其他两种算法在大范围工作空间中执行覆盖任务更具有稳定性。

表3数据为在图4环境中,4个机器人应用R-DPPCPP、BA*和BINN进行20次多机器人全覆盖实验得到的平均值。由表3可见:在3个环境中,R-DPPCPP得到的平均路径长度与BINN、BA*算法相比路径分别减少了16.69%~17.33%和10.32%~20.03%,体现了R-DPPCPP的优越性;在环境1、环境2和环境3中,R-DPPCPP、BINN和BA*均完成了全覆盖任务;R-DPPCPP的平均路径重复率低于BINN与BA*算法,表明了R-DPPCPP对于环境的适应性好,规划的路径质量高,可有效地应用于多机器人的全覆盖路径规划。

表3 不同算法路径性能对比

指标		BINN	BA*	R-DPPCPP
平均路径长度	环境1	888.13	854.79	748.82
	环境2	821.49	849.15	679.10
	环境3	2601.5	2397.9	2150.5
平均重复率	环境1	19.93%	21.91%	6.18%
	环境2	21.05%	31.32%	4.87%
	环境3	17.04%	14.78%	3.21%
覆盖率	环境1	1	1	1
	环境2	1	1	1
	环境3	1	1	1

为了验证R-DPPCPP在机器人数量增多的情况的可行性,选用了6个机器人在如图4(b)所示的环境2中进行20次仿真实验,机器人初始位置随机生成并与BINN和BA*算法比较,仿真得到结果的平均值如表4所示。由表3和表4可见,随着机器人数量的增多,BINN与BA*算法的平均路径重复率和平均路径长度明显增加,R-DPPCPP平均重复率和平均路径长度分别只增加了0.36%和0.77%,但其平均极差却增加了14.29%。因为在覆盖中后期,剩余覆盖区域已被

机器人占据,R-DPPCPP不引导机器人回溯进入其他机器人的覆盖区域,降低了重复率和路径长度,但也造成了机器人路径不均匀,同时也表明了合理的机器人数量能够提高规划路径的质量。

表4 6个机器人在环境2中的仿真数据

方法	平均路径长度	平均重复率	平均极差
BINN	898.77	29.94%	5
BA*	880.50	35.60%	6
R-DPPCPP	684.34	5.23%	56

6 结论和展望

本文针对多机器人未知地图全覆盖效果差的难题,提出了一种基于滚动优化和分散捕食者猎物模型的多机器人全覆盖路径规划算法。首先,通过栅格法对环境空间进行建模,对捕食者猎物模型中的避开捕食者奖励函数进行改进,并添加移动代价奖励函数和回溯区域惩罚机制,构建分散捕食者猎物模型;然后,将分散捕食者猎物模型与滚动优化策略相结合求解机器人的覆盖路径;最后,分别对改进策略的有效性和R-DPPCPP作用于多机器人覆盖规划的可行性进行仿真验证。仿真结果表明,分散捕食者猎物模型和滚动优化均提升了PPCPP算法规划的路径质量,R-DPPCPP的平均路径和平均重复率与BINN、PPCPP算法相比具有明显优势,路径长度分别减少了16.69%~17.33%和10.32%~20.03%,表明了所提出算法能够规划高质量的路径,可有效地应用于多机器人的全覆盖路径规划。

在后续工作中,将进一步探究机器人工作量的均匀程度与路径长度和重复率之间的关系。同时本文通过二维仿真得到了实验结果,将其与轨迹规划相结合应用于实际机器人也是后续研究的重点。

参考文献(References)

- [1] 简毅,张月. 移动机器人全局覆盖路径规划算法研究进展与展望[J]. 计算机应用, 2014, 34(10): 2844-2849. (Jian Y, Zhang Y. Complete coverage path planning algorithm for mobile robot: Progress and prospect[J]. Journal of Computer Applications, 2014, 34(10): 2844-2849.)
- [2] Dakulovic M, Petrovic I. Complete coverage path planning of mobile robots for humanitarian demining[J]. Industrial Robot—The International Journal of Robotics Research and Application, 2012, 39(5): 484-493.
- [3] Miao X, Lee J, Kang B Y. Scalable coverage path planning for cleaning robots using rectangular map decomposition on large environments[J]. IEEE Access, 2018, 6: 38200-38215.
- [4] Chen K, Liu Y. Optimal complete coverage planning of wall-climbing robot using improved biologically inspired

- neural network[C]. IEEE International Conference on Real-Time Computing and Robotics. Okinawa, 2017: 587-592.
- [5] Lee T K, Baek S H, Choi Y H, et al. Smooth coverage path planning and control of mobile robots based on high-resolution grid map representation[J]. *Robotics and Autonomous Systems*, 2011, 59(10): 801-812.
- [6] Hungerford K, Dasgupta P, Guruprasad K R. Distributed, complete, multi-robot coverage of initially unknown environments using repartitioning[C]. *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems*. Paris: Assoc Comp Machinery, 2014: 1453-1454.
- [7] Zhao Y, Liu S J, Zhang Y H. Spatial density voronoi diagram and construction[J]. *Journal of Computers*, 2012, 7(8): 2007-2014.
- [8] Sun R C, Tang C H, Zheng J Y, et al. Multi-robot path planning for complete coverage with genetic algorithms[C]. *International Conference on Intelligent Robotics and Applications*. Shenyang, 2019, 11744: 349-361.
- [9] 罗志远, 丰硕, 刘小峰, 等. 一种基于分步遗传算法的多无人清洁车区域覆盖路径规划方法[J]. *电子测量与仪器学报*, 2020, 34(8): 43-50.
(Luo Z Y, Feng S, Liu X F, et al. Method of area coverage path planning of multi-unmanned cleaning vehicles based on step by step genetic algorithm[J]. *Journal of Electronic Measurement and Instrumentation*, 2020, 34(8): 43-50.)
- [10] Nair V G, Guruprasad K R. GM-VPC: An algorithm for multi-robot coverage of known spaces using generalized voronoi partition[J]. *Robotica*, 2020, 38(5): 845-860.
- [11] Li W H, Zhao T, Dian S Y. A multi-robot coverage path planning method based on genetic algorithm[C]. *International Conference on Security, Pattern Analysis, and Cybernetics*. Chengdu, 2021: 13-18.
- [12] 符小卫, 魏广伟, 高晓光. 不确定环境下多无人机协同区域搜索算法[J]. *系统工程与电子技术*, 2016, 38(4): 821-827.
(Fu X W, Wei G W, Gao X G. Cooperative area search algorithm for multi-UAVs in uncertainty environment[J]. *Systems Engineering and Electronics*, 2016, 38(4): 821-827.)
- [13] Luo C M, Yang S X, Li X D, et al. Neural-dynamics-driven complete area coverage navigation through cooperation of multiple mobile robots[J]. *IEEE Transactions on Industrial Electronics*, 2017, 64(1): 750-760.
- [14] 朱大奇, 朱婷婷, 颜明重. 基于改进神经网络的多AUV全覆盖路径规划[J]. *系统仿真学报*, 2020, 32(8): 1505-1514.
(Zhu D Q, Zhu T T, Yan M Z. Multi-AUV complete coverage path planning based on improved neural network[J]. *Journal of System Simulation*, 2020, 32(8): 1505-1514.)
- [15] Hassan M, Liu D K. PPCPP: A predator-prey-based approach to adaptive coverage path planning[J]. *IEEE Transactions on Robotics*, 2020, 36(1): 284-301.
- [16] Lin H Y, Huang Y C. Collaborative complete coverage and path planning for multi-robot exploration[J]. *Sensors*, 2021, 21(11): 3709.
- [17] Sanna G, Godio S, Guglieri G. Neural network based algorithm for multi-UAV coverage path planning[C]. *International Conference on Unmanned Aircraft Systems*. Athens, 2021: 1210-1217.
- [18] Hazon N, Kaminka G A. On redundancy, efficiency, and robustness in coverage for multiple robots[J]. *Robotics and Autonomous Systems*, 2008, 56(12): 1102-1114.
- [19] Choi S, Lee S, Viet H H, et al. B-theta*: An efficient online coverage algorithm for autonomous cleaning robots[J]. *Journal of Intelligent & Robotic Systems*, 2017, 87(2): 265-290.
- [20] Viet H H, Dang V H, Laskar M N U, et al. BA*: An online complete coverage algorithm for cleaning robots[J]. *Applied Intelligence*, 2013, 39(2): 217-235.
- [21] Viet H H, Dang V H, Choi S, et al. BoB: An online coverage approach for multi-robot systems[J]. *Applied Intelligence*, 2015, 42(2): 157-173.
- [22] Senthilkumar K S, Bharadwaj K K. Multi-robot exploration and terrain coverage in an unknown environment[J]. *Robotics and Autonomous Systems*, 2012, 60(1): 123-132.
- [23] 徐梦颖, 王娇娇, 刘宝, 等. 基于改进遗传算法的机器人路径规划[J]. *石河子大学学报: 自然科学版*, 2021, 39(3): 391-396.
(Xu M Y, Wang J J, Liu B, et al. Path planning for robot based on improved genetic algorithm[J]. *Journal of Shihezi University: Natural Science*, 2021, 39(3): 391-396.)
- [24] Shi C X, Wang Y Q, Yang J Y. Online topological map building and qualitative localization in large-scale environment[J]. *Robotics and Autonomous Systems*, 2010, 58(5): 488-496.
- [25] 蔡雨岑, 杜鹏桢. 基于平衡鲸鱼优化算法的无人车路径规划[J]. *控制与决策*, 2021, 36(11): 2647-2655.
(Cai Y C, Du P Z. Path planning of unmanned ground vehicle based on balanced whale optimization algorithm[J]. *Control and Decision*, 2021, 36(11): 2647-2655.)

作者简介

阮贵航(1997—), 男, 硕士生, 从事机器人路径规划的研究, E-mail: 925590386@qq.com;

陈教科(1979—), 男, 副教授, 博士, 从事机器人智能装备等研究, E-mail: jlchen@zjut.edu.cn;

胥芳(1964—), 女, 教授, 博士生导师, 从事机器人智能装备等研究, E-mail: fangx@zjut.edu.cn.