计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2023,32(8):189−197 [doi: 10.15888/j.cnki.csa.009160]
©中国科学院软件研究所版权所有.
E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

改进蚁群与动态 Q 学习融合的机器人路径规划①

薛颂东, 余　欢

(太原科技大学 计算机科学与技术学院, 太原 030024)
通信作者: 薛颂东, E-mail: xuesongdong@tyust.edu.cn

摘　要: 基本 Q 学习算法应用于路径规划时, 动作选择的随机性导致算法前期搜索效率较低, 规划耗时长, 甚至不能找到完整的可行路径, 故提出一种改进蚁群与动态 Q 学习融合的机器人路径规划算法. 利用精英蚂蚁模型和排序蚂蚁模型的信息素增量机制, 设计了一种新的信息素增量更新方法, 以提高机器人的探索效率; 利用改进蚁群算法的信息素矩阵为 Q 表赋值, 以减少机器人初期的无效探索; 设计了一种动态选择策略, 同时提高收敛速度和算法稳定性. 在不同障碍物等级的二维静态栅格地图下进行的仿真结果表明, 所提方法能够有效减少寻优过程中的迭代次数与寻优耗时.

关键词: Q 学习; 路径规划; 信息素; 动态搜索; 栅格地图

引用格式: 薛颂东,余欢.改进蚁群与动态 Q 学习融合的机器人路径规划.计算机系统应用,2023,32(8):189–197. http://www.c-s-a.org.cn/1003-3254/9160.html

Robotic Path Planning Integrating Improved Ant Colony Optimization and Dynamic Q-learning

XUE Song-Dong, YU Huan

(College of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China)

Abstract: When the basic Q-learning algorithm is applied to path planning, the randomness of action selection makes the early search efficiency of the algorithm low and the planning time-consuming, and even a complete and feasible path cannot be found. Therefore, a path planning algorithm of robots based on improved ant colony optimization (ACO) and dynamic Q-learning fusion is proposed. The pheromone increment mechanism of the elite ant model and sorting ant model is used, and a new pheromone increment updating method is designed to improve the exploration efficiency of robots. The pheromone matrix of the improved ant colony optimization algorithm is used to assign values to the Q table, so as to reduce the ineffective exploration of the robot at the initial stage. In addition, a dynamic selection strategy is designed to improve the convergence speed and the stability of the algorithm. Finally, different simulation experiments are carried out on two-dimensional static grid maps with different obstacle levels. The results show that the proposed method can effectively reduce the number of iterations and optimization time consumption in the optimization process.

Key words: Q-learning; path planning; pheromone; dynamic search; raster map

路径规划是移动机器人领域的一个研究热点, 是移动机器人实现自主导航的关键技术之一. 目的是在已知或者未知的障碍物环境中搜索出一条从起始位置到目标位置的最优安全无碰撞路径, 评估标准一般为距离最短、耗时最短等. 路径规划算法大致可分为 4 类: 传统路径规划算法、图形学方法、智能仿生学算法和其他算法, 其中强化学习算法当前较为热门[1–4].

关于强化学习在智能机器人路径规划和避障[5,6]

① 基金项目: 教育部产学合作协同育人项目 (202102076011); 山西省高等学校科技创新项目 (2021L322); 山西省基础研究计划自由探索类项目 (20210302124165); 山西省高等学校教学改革创新项目 (J2021441)
收稿时间: 2023-01-08; 修改时间: 2023-02-09; 采用时间: 2023-02-13; csa 在线出版时间: 2023-05-22
CNKI 网络首发时间: 2023-05-24

© 中国科学院软件研究所 http://www.c-s-a.org.cn

中的应用, Q 学习算法是较为常用的强化学习方法[7]. 但是基本的强化学习方法也有一些缺点. 首先, 当机器人处于较为复杂的环境中时, 容易出现“维数诅咒”. 此外, 容易反复探索次优路径, 造成算法收敛速度缓慢, 陷入局部最优. 训练机器人时一般需要花费很长时间, 而启发式知识可为机器人的动作选择提供指导[8], 帮助其更快地收敛.

Low 等人[9] 引入了部分引导 Q 学习的概念, 利用花授粉算法来改进 Q 学习的初始化, 可以加速其收敛速度, 但该方法在复杂度较高的环境难以保证收敛到最优路径. 毛国君等人[10] 引入了动态探索因子的策略, 根据运行过程中环境的反馈, 调整贪婪因子的大小, 减少了寻优耗时, 但在运行初期仍需要花费大量时间随机探索未知环境. 徐晓苏等人[11] 利用人工势场法初始化 Q 表, 为机器人提供关于环境的先验知识, 大幅减少了前期的随机探索, 但该方法在迭代后期作用较小, 且适用范围较为局限. Pei 等人[12] 基于机器人与目标点之间的距离提出了一种启发式搜索策略和距离度量标准, 在算法迭代过程中能够指引移动机器人向目标靠近, 但是该策略与实际环境的相关度较高, 不同的环境要设计不同的启发式函数, 且需要不断修正 Q 函数. Hao 等人[13] 提出了一种潜在的动态 Q 学习方法, 将 Q 学习与人工势场法和动态奖励函数相结合来生成可行路径, 在路径长度和转弯角度两个指标上表现优异, 但与经典算法相比, 在计算时间和收敛速度方面没有明显提高. Chang 等人[14] 将动态窗口法与 Q 学习相结合, 增加了新的评价功能, 提高了全局导航性能, 但其全局搜索能力较小, 且依赖于参考路径. 可见, 在考虑 Q 学习的改进时需要考虑动作选择的指导作用以及探索与利用的关系对收敛性和寻优耗时上的影响.

蚁群算法是一种分布式的智能仿生算法, 模拟了自然界中蚂蚁觅食的特征[15], 具有正反馈、分布式计算及鲁棒性强等优势[16], 其候选解构建过程与路径规划过程相似. 因此, 选用蚁群算法作为优化 Q 学习算法的对象. 田晓航等人[17] 引入蚁群算法的信息素机制, 提出了一种寻优范围优化方法, 减少智能体的无效探索次数, 但没有充分考虑到智能体探索的目的性. 石立培的方法[18] 中蚂蚁移动一次, 就对路径上的信息素进行一次局部的更新, 选择动作时同时考虑信息素浓度与 Q 值, 虽然可以实现指导后续搜索的效果, 但频繁的局部更新增加了耗时, 且与柯逸[19] 类似, 未考虑改进搜索策略对算法收敛性的影响, 复杂环境下性能提升较小.

因此, 针对基本 Q 学习在算法运行前期出现的搜索效率低, 寻优时间较长的问题, 利用改进后的自适应蚁群算法先探索未知环境, 通过产生的信息素矩阵实现 Q 表的不均匀初始化, 对机器人的寻优范围进行优化, 指导机器人在初期的探索和寻优过程. 另外, 设计了一个动态搜索策略, 通过环境的变化动态地改变搜索因子的大小, 提高机器人的探索效率.

1 本文基础算法思想及策略

1.1 蚁群算法

蚂蚁在寻找最优路径的过程中, 会根据可选路径上信息素浓度的大小选择动作, 同时会在其走过的路径上留下信息素. 在时刻t, 蚂蚁k由第i个节点选择下一节点j是根据信息素和启发式信息的大小来判定的. 蚁群算法的转移概率如式 (1) 和式 (2) 所示.

$$P_{ij}^{k}=\begin{cases}\dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum\limits_{j\in allowed_k}[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}, & j\in allowed_k\\ 0, & \text{otherwise}\end{cases} \tag{1}$$

$$\eta_{ij}=\frac{1}{d_{ij}} \tag{2}$$

其中, α为信息素启发因子, 表示路径选择时信息素浓度的重要性; β为期望启发因子, 表示启发式信息在路径选择时的重要性; $allowed_k$表示蚂蚁k在当前节点i可到达的下一节点j的集合, $\tau_{ij}(t)$为路径ij上的信息素浓度, $\eta_{ij}(t)$表示路径ij上的启发信息, d_{ij}表示路径ij之间的欧式距离.

信息素更新策略为在一次循环中所有蚂蚁都完成迭代之后, 进行一次全局信息素更新, 经过多次循环后, 信息素呈现出不均匀的分布, 最优路径附近的信息素浓度高于其他区域. 全局信息素更新如式 (3)–式 (5) 所示:

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

$$\Delta\tau_{ij}(t)=\sum_{k=1}^{m}\Delta\tau_{ij}^{k}(t) \tag{4}$$

$$\Delta\tau_{ij}^{k}(t)=\begin{cases}\dfrac{Q}{L_k}, & k\text{经过路径}ij\\ 0, & \text{otherwise}\end{cases} \tag{5}$$

其中, ρ为全局信息素挥发系数, $\rho\in[0,1]$, $\Delta\tau_{ij}(t)$为一次迭代后所有蚂蚁留下的信息素, m为蚂蚁总数. Q为

© 中国科学院软件研究所 http://www.c-s-a.org.cn

信息素强度, L_k表示第k只蚂蚁在本次迭代中走过的路径总长度.

1.2 Q 学习算法

Q 学习是强化学习中的一种无模型算法[20], 是一种近似于动态规划的技术. Q 值表示机器人在执行一个动作后, 由当前状态转向下一个有效状态的奖励值. 强化学习的框架如图 1 所示, 代理根据 Q 表选择一个动作a并执行它, 然后环境返回给机器人的一个状态s和执行动作a的奖励r, 以更新 Q 表, 强化学习框架见图 1.

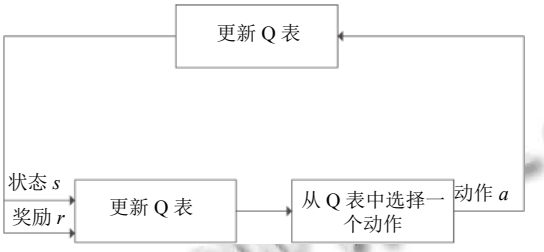

图 1 强化学习框架图

在 Q 学习中, Q 表是一个最优策略动作值函数, 按式 (6) 所示规律更新.

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right] \tag{6}$$

其中, α为学习率, γ为折扣因子, r为即时奖励, s为机器人当前的状态, a为机器人选择的动作, s'为执行动作a后的下一个状态, a'为对应的下一个动作, $\max Q(s',a')$为状态s'对应的最大累积奖励值.

2 本文算法

2.1 改进蚁群算法信息素增量的更新机制

基于精英蚂蚁模型和排序蚂蚁模型, 对基于自适应机制的蚁群算法中信息素增量的更新规则进行改进, 提出一种梯度更新机制来更新信息素增量, 以提高算法的收敛速度. 在所有蚂蚁完成一次迭代之后, 记录所有成功到达目标节点的蚂蚁信息, 统计其节点信息和路径长度, 然后计算当代最短路径长度和所有到达目标节点的平均路径长度.

一代寻优结束后, 先对当前最短路径和小于等于平均路径长度的路径按照梯度更新公式进行一次局部信息素更新, 大于平均路径长度的路径则不再进行信息素的加强. 通过给予不同等级的信息素浓度奖励, 可以加强较优的解对后续寻优过程的指导作用, 同时减少较差解的误导. 随后, 再进行一次全局信息素更新. 最短路径和小于等于平均寻优路径长度的路径上的信息素增量更新公式按照式进行计算, 局部信息素按式 (7)–式 (9) 更新.

$$\tau_{ij}(t+1) = (1-\rho_1)\tau_{ij}(t) + \rho_1 \Delta\tau_{ij}(t) \tag{7}$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) \tag{8}$$

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \dfrac{Q_\phi}{L_k} - \dfrac{Q_\phi}{L_a}, & k\text{经过路径}ij \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

其中, L_a表示平均寻优路径长度, ρ_1表示局部信息素挥发系数, $\rho_1 \in [0,1]$.

局部信息素加强时对信息素量Q_ϕ进行改进, 改进公式如下.

$$Q_\phi(t+1) = Q_\phi(t) + \eta(L_b - L_k) \tag{10}$$

其中, L_b表示当前最短路径长度, η表示距离因子.

2.2 动态 Q 学习中 Q 表的初始化

在基本 Q 学习中, Q 表通常被初始化为某一固定的数值, 通常情况下设置为 0. 基本的 Q 学习算法没有充分利用已知的条件, 算法运行初期随机性较高, 导致收敛速度较慢, 且效率低下. 但 Q 表的值存在一定的规律性, 一般情况下越靠近最优路径的位置, Q 值应越大.

蚁群算法是一种在路径规划领域很有效的启发式算法, 其信息素表的结构与 Q 学习中 Q 表的结构类似, 二者更新方式也类似, 这是 2 种算法结合的关键. 本文规定机器人的移动方向为上、下、左、右的四叉树搜索策略, 信息素表和 Q 值表的部分结构见图 2.

$\tau(3,1)$	$\tau(3,2)$	$\tau(3,3)$	$\tau(3,4)$
$\tau(2,1)$	$\tau(2,2)$	$\tau(2,3)$	$\tau(2,4)$
$\tau(1,1)$	$\tau(1,2)$	$\tau(1,3)$	$\tau(1,4)$

(a) 信息素表的结构

$Q(3,1)$	$Q(3,2)$	$Q(3,3)$	$Q(3,4)$
$Q(2,1)$	$Q(2,2)$	$Q(2,3)$	$Q(2,4)$
$Q(1,1)$	$Q(1,2)$	$Q(1,3)$	$Q(1,4)$

(b) Q 表的结构

图 2 信息素表和 Q 表部分结构图

算法融合的规则为: 先用改进信息素更新策略的自适应蚁群算法进行路径预规划, 得到该环境下的次优路径, 不同的位置根据走过的蚂蚁数量、是否处于

© 中国科学院软件研究所 http://www.c-s-a.org.cn

较优的路径以及周围是否有障碍物有着不同的信息素浓度. 将栅格地图上不同位置的信息素与其可选择的行动的个数相乘, 得到一个不均匀分布的信息素矩阵; 再用信息素矩阵对 Q 表赋初值, 完成 Q 表的初始化, 以指导后续搜索.

假设某一位置处于边界, 则只有 2 个方向可选择, 另 2 个方向信息素浓度低于阈值, 被标记为无效状态; 假设该位置既不处于边界, 周围也没有障碍物, 则 4 个方向的动作都可选择, 可选择动作部分结构见图 3.

$\tau(5,1)$	$\tau(5,2)$	$\tau(5,3)$	$\tau(5,4)$

(a) 4 个方向都可选择

$\tau(5,1)$	$\tau(5,2)$	$\tau(5,3)$	$\tau(5,4)$

(b) 2 个方向都可选择 (动作2、3被禁止)

图 3 可选择动作图

由于不同算法之间存在差异, 蚁群信息素τ与 Q 值之间存在数量级上的差异, 因此按式 (11) 赋值.

$$Q(s,a)=\lambda\mu\tau_{ij}(t) \tag{11}$$

其中, $Q(s,a)$表示未赋初值的 Q 值, $\tau_{ij}(t)$为对应位置的信息素浓度, λ是一个大于 0 的常数, μ为所处位置的可选移动方向的个数. 融合算法中 Q 表初始化的伪码描述见算法 1.

算法 1. Q 表初始化

输入: α、β、ρ、ρ_l、λ、Q、迭代次数 *iter_number*、蚂蚁数 *ant_number*、起始节点 *start*、目标节点 *goal*
输出: Q值表

```
for NC=1:iter_number do
  for ant=1:ant_number do
    初始化禁忌表, start 加入禁忌表
    找到可行节点, 以供循环
    通过式 (1) 选择下一个节点
    while(蚂蚁k没有到达终点) do
      重新计算可行节点
      allowed_k ←可行节点
    end
    记录蚂蚁k走过的节点信息、路径长度
    通过式 (7) 局部信息素加强
    通过式 (3) 全局信息素更新
  end
end
生成信息素矩阵
通过式 (11) 的融合规则给 Q 表赋值
```

本文算法的奖励函数设置如式 (12) 所示.

$$r(s,a)=\begin{cases}-5, & \text{遇到障碍物}\\ 1, & \text{正常通行}\\ 5, & \text{目标位置}\end{cases} \tag{12}$$

2.3 动态搜索策略

基本 Q 学习的动作选择策略通常采用ε-贪心策略, ε值越大, 机器人在选择动作时, 进行随机探索的概率就越大; 反之, 寻优过程的目的性较强. 基于贪婪搜索策略改进的动态策略能够有效地解决容易陷入局部最优的缺点.

为使算法在运行的前、中、后期都有合适的探索概率, 结合迭代过程中的标准差, 设计了动态调整探索因子的方法, 以实现ε值的动态变化过程.

首先计算迭代过程的标准差, 并进行归一化处理; 由于运行初期机器人要充分探索未知环境, 因此标准差的值较大, ε可保持在最大值ε_{Max}, 以提高探索的随机性, 机器人选择随机动作的概率较大, 避免陷入局部最优; 随着迭代次数的增加, 算法逐渐收敛, 标准差降低, ε值随着标准差的降低逐渐减小; 最后稳定在最小值ε_{Min}, 机器人探索的目的性不断增强, 使算法快速收敛. 改进后的动态搜索策略伪码描述见算法 2.

算法 2. 动态搜索策略

输入: 迭代步数 *iter*、标准差 *std_n*、ε_{Max}、ε_{Min}
输出: 执行的动作a

```
计算迭代过程的标准差 std_n
对 std_n 进行归一化处理, 使 std_n∈(0, 1)
if std_n>ε_Max then
    ε←ε_Max
else if std_n<ε_Min then
    ε←ε_Min
else
    ε←std_n
end
end
for (each action a∈A(s)) do
  if rand()<ε then
    a′←a random action from s
  else
    a′←ε-greedy get a
    执行动作a
  end
end
```

3 实验分析

为了验证本文所提算法的可行性和有效性, 基于 Matlab 设计了 3 种不同规格障碍物的环境地图进行仿

© 中国科学院软件研究所 http://www.c-s-a.org.cn

真实验, 包括 20×20 的随机障碍物地图、30×30 的特殊障碍物地图、40×40 的随机障碍物地图. 在 3 类不同的地图上综合对比了基本 Q 学习算法 (basic Q-learning, BAS-Q)、文献 [10] 提出的算法 (DSS-Q)、利用自适应蚁群算法与 Q 学习融合的算法 (ant colony optimization Q-learning, ACO-Q) 和改进蚁群与动态 Q 学习融合的算法 (DSSACO-Q), ACO-Q 与 DSSACO-Q 的区别在于是否使用了本文所提出的动态搜索策略.

设定每次运行尝试的最大迭代次数为 5 000, 每次运行最大的探索步数为 600, 每种算法分别重复执行 10 次. 统计平均路径长度、平均迭代次数和平均寻优耗时, 取其中一次迭代结果绘制路径规划图和迭代曲线对比图. 本文方法中蚁群信息素参数见表 1, Q-learning 参数见表 2.

表 1 蚁群信息素参数

参数	α	β	ρ	ρ_l	m	$N_{\max}$	Q
数值	2	3	0.1	0.08	40	50	5

表 2 Q-learning 参数

参数	α	γ	λ	ε	M
BAS-Q	0.8	0.95	—	0.8	600
DSS-Q	0.8	0.95	—	0.1–1	600
ACO-Q	0.8	0.95	100	0.8	600
DSSACO-Q	0.8	0.95	100	0.01–0.4	600

注: “—”表示此算法未用到参数λ

3.1 20×20 障碍物环境

在 Matlab 中分别生成障碍物占比为 20%、30%、40% 和 50% (不考虑障碍物重叠) 的 4 张 20×20 栅格地图, 作为仿真验证环境.

图 4 展示了 4 种算法运行中的一次路径规划结果, 图 4(a)–(d) 分别为 20%–50% 障碍物占比的路径规划结果. 如图所示, 4 种算法在不同难度等级的障碍物环境下都找到了目标点. 图 5 展示了 4 种算法运行中一次的迭代曲线, 图 5(a)–(d) 分别为 20%–50% 障碍物占比的 4 种算法迭代曲线. 可以看出本文提出的方法能在算法前期迅速找到最优解, 且在迭代后期保持稳定, 波动较小, 明显优于其他 3 种算法.

表 3 展示了 4 种算法在不同难度等级障碍物下的平均寻优耗时对比和平均迭代次数对比. 不难发现, 在使用本文方法寻找到最优路径的前提下, 较 BAS-Q 算法的平均耗时约有 49% 的提升; 较 DSS-Q 算法的平均耗时约有 15% 的提升. ACO-Q 算法的平均寻优时间高于 DSS-Q 算法的原因是, 该算法在运行初期先利用改进后的蚁群算法探索未知的障碍物环境, 然后再进行 Q 表的初始化过程, 增加了耗时. 但在加入本文提出的动态搜索策略后, 寻优耗时明显减少, 验证了动态搜索策略减少平均寻优耗时的有效性.

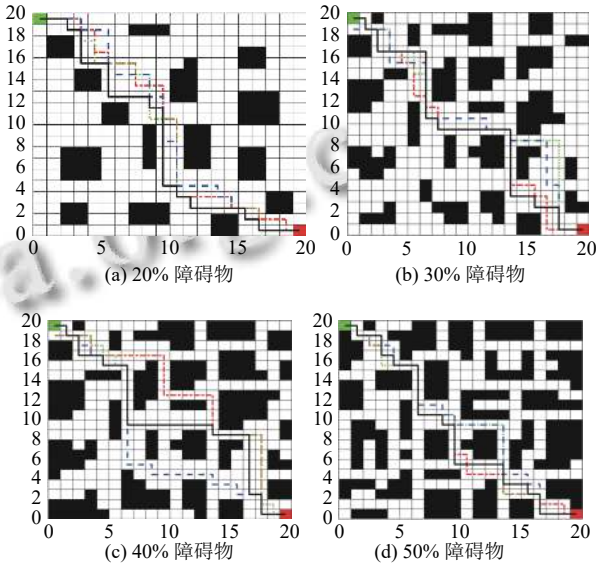

图 4 20×20 路径规划图

随着障碍物占比的不断提高, 算法的平均迭代次数呈下降趋势, 原因是在障碍物占比较少的环境中机器人有着更大的搜索空间, 需要更多的步骤去探索未知的环境, 增加了算法迭代次数. 与 BAS-Q 相比, DSSACO-Q 能够更快收敛于最优路径, 平均迭代次数减少约 28%; 与 DSS-Q 相比, DSSACO-Q 的平均迭代次数约为 DSS-Q 的 90%; 与 ACO-Q 相比, DSSACO-Q 因为加入了动态搜索策略, 使得算法在寻优过程中的目的性增强, 随着障碍物占比的增加, 迭代次数下降更快, 且更快达到平稳, 验证了本文所提出的改进方法的有效性.

3.2 30×30 障碍物环境

在 30×30 的栅格环境下, 障碍物的设置参照了文献 [17] 中的地图, 相较于随机的障碍物环境, 该地图中存在一定数量的凹形障碍物, 且该地图中障碍物体积较大, 机器人在寻优过程中更加容易陷入局部最优, 容易给算法造成额外的负担.

从表 4 中的数据可以看出 4 种算法找到的最优路径在长度上相等, 都找到了最优解. 图 6 展示了 4 种算法的一次路径规划结果, 图 7 展示了 4 种算法运行中

© 中国科学院软件研究所 http://www.c-s-a.org.cn

一次的迭代曲线. 可以看出 DSSACO-Q 最先达到收敛, 且在迭代后期保持稳定, 优于其他 3 种算法. BAS-Q 和 DSS-Q 一直到迭代后期才趋于平稳, ACO-Q 同样较快达到了稳定, 但差于 DSSACO-Q. 本文算法在平均迭代次数上相较于 BAS-Q 减少了 41.7%, 较 DSS-Q 减少了 15.2%; 平均寻优耗时上相较于 BAS-Q 减少了 43.4%, 相较于 DSS-Q 减少了 16.7%; 与未使用动态搜索策略的 ACO-Q 相比, DSSACO-Q 平均迭代次数减少了 7.3%, 平均寻优耗时减少了 40.5%, 验证了本文提出的 2 个改进点的有效性.

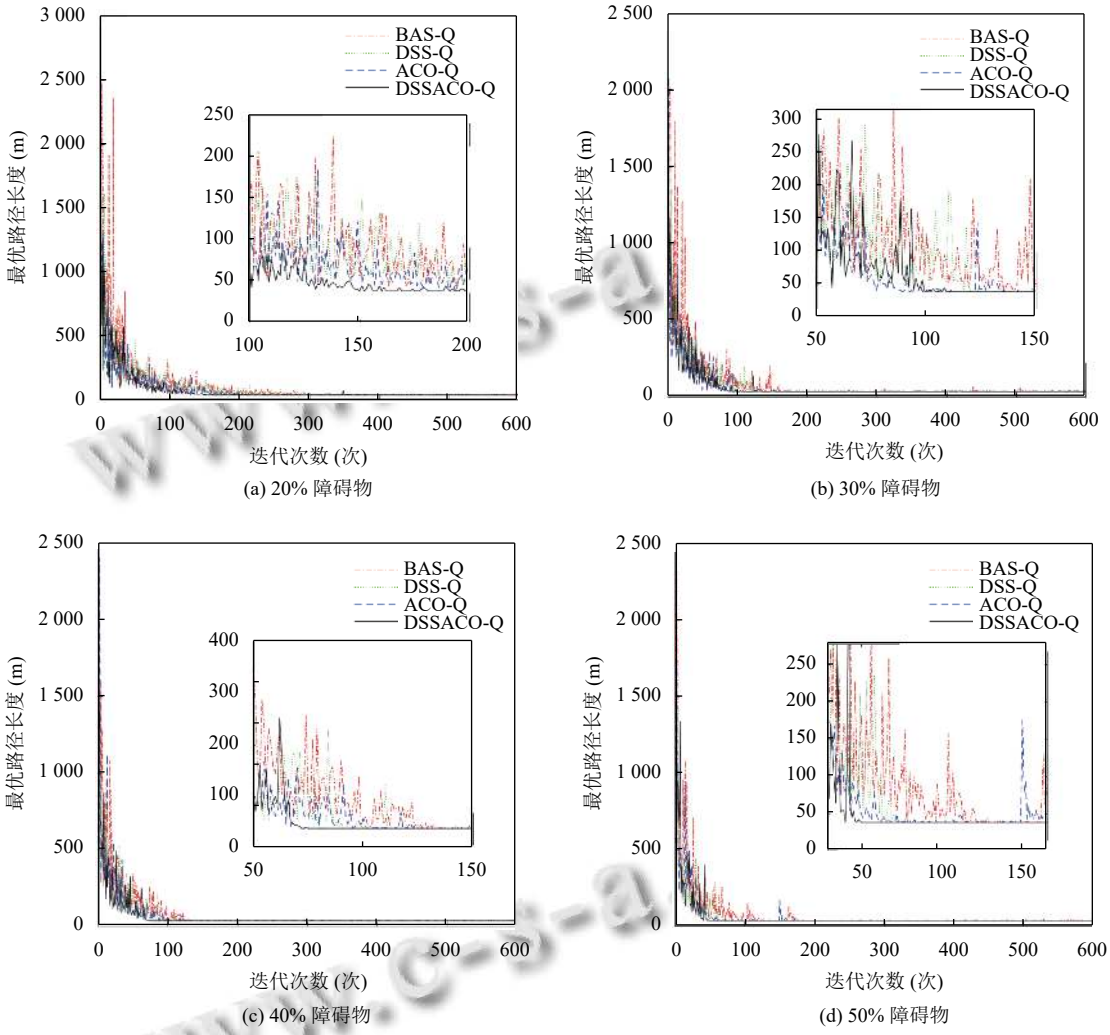

图 5 20×20 迭代曲线图

3.3 40×40 障碍物环境

为了进一步验证本文算法的有效性和可行性, 在 40×40 的栅格环境下, 设置了障碍物占比分别为 20% 和 30% (不考虑障碍物重叠) 的未知环境, 相较于其他两种障碍物环境, 该地图难度进一步加大.

在 40×40 的障碍物环境下进行了 2 组不同难度等级的实验, 图 8 展示了 4 种算法在 2 种不同障碍物占比下的路径图, 图 8(a) 和图 8(b) 分别为 20% 和 30% 障碍物占比的路径规划结果. 图 9 展示了 4 种算法的迭代曲线对比, 图 9(a) 和图 9(b) 分别为 20% 和 30% 障碍物占比的 4 种算法迭代曲线. 从整体实验结果来看, 本文算法在收敛速度、全局搜索能力和寻优耗时上相比基本 Q 学习和文献 [10] 提出的方法都有一定的提升.

如表 5 所示, 在平均寻优耗时方面, 本文方法与 BAS-Q 相比, 约减少 45%; 与 DSS-Q 相比, 约减少 28%; 与 ACO-Q 相比, 约减少 44%. ACO-Q 的寻优时间高于 DSS-Q, 同样是因为该算法在运行初期先利用

© 中国科学院软件研究所 http://www.c-s-a.org.cn

改进蚁群算法探索出次优路径, 然后进行 Q 表的初始化, 增加了耗时. 但在加入本文提出的动态搜索策略后, 寻优耗时大幅下降.

表 3 20×20 算法性能对比表

障碍物占比 (%)	算法	平均路径长度 (m)	平均迭代次数 (次)	平均寻优耗时 (s)
20	BAS-Q	38	122.1383	0.2607
	DSS-Q	38	100.1167	0.1572
	ACO-Q	38	86.8550	0.2452
	DSSACO-Q	38	**86.7983**	**0.1344**
30	BAS-Q	38	104.4817	0.2646
	DSS-Q	38	84.2550	0.1589
	ACO-Q	38	79.6100	0.2396
	DSSACO-Q	38	**76.6717**	**0.1345**
40	BAS-Q	38	95.3200	0.2579
	DSS-Q	38	74.6167	0.1602
	ACO-Q	38	70.4617	0.2405
	DSSACO-Q	38	**67.5100**	**0.1363**
50	BAS-Q	38	93.4533	0.2651
	DSS-Q	38	75.1817	0.1604
	ACO-Q	38	73.0983	0.2392
	DSSACO-Q	38	**67.3567**	**0.1322**

表 4 30×30 算法性能对比表

算法	平均路径长度 (m)	平均迭代次数 (次)	平均寻优耗时 (s)
BAS-Q	58	337.4783	0.7362
DSS-Q	58	231.6950	0.5000
ACO-Q	58	212.0683	0.7001
DSSACO-Q	58	**196.5133**	**0.4167**

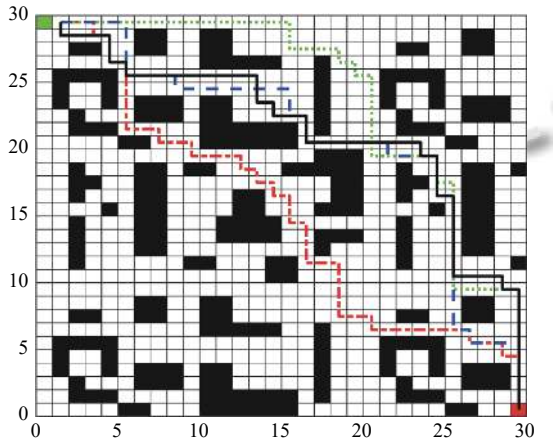

图 6 30×30 路径规划图

平均迭代次数方面, 20% 障碍物地图下的迭代次数更高, 是由于在障碍物占比较少的环境中机器人有着更大的搜索空间, 需要更多的步骤去探索未知的环境. 与 BAS-Q 相比, 本文算法的平均迭代次数在 20% 障碍物情况下减少 48.9%, 30% 情况下减少 50.6%; 与 DSS-Q 相比, 在 20% 障碍物情况下减少 16.2%, 30% 情况下减少 15.4%; 与 ACO-Q 相比, 在 20% 障碍物情况下减少 11.8%; 在 30% 情况下减少 13.1%, 验证了本文所提的算法融合方法和动态搜索策略的有效性.

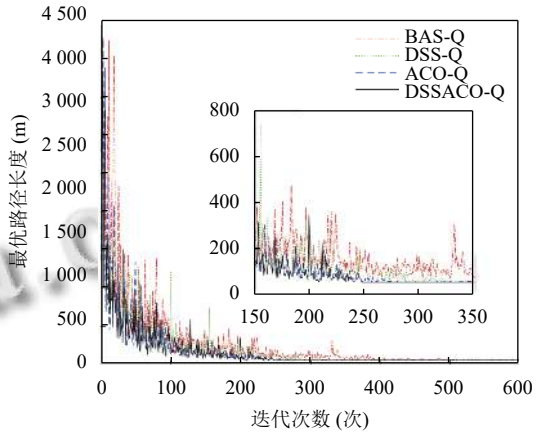

图 7 30×30 迭代曲线图

表 5 40×40 算法性能对比表

障碍物占比 (%)	算法	平均路径长度 (m)	平均迭代次数 (次)	平均寻优耗时 (s)
20	BAS-Q	78	847.9467	1.6256
	DSS-Q	78	517.1350	1.2066
	ACO-Q	78	496.8217	1.5828
	DSSACO-Q	78	**433.5917**	**0.9048**
30	BAS-Q	78	782.2833	1.5604
	DSS-Q	78	456.3767	1.2009
	ACO-Q	78	444.4217	1.5371
	DSSACO-Q	78	**386.2483**	**0.8528**

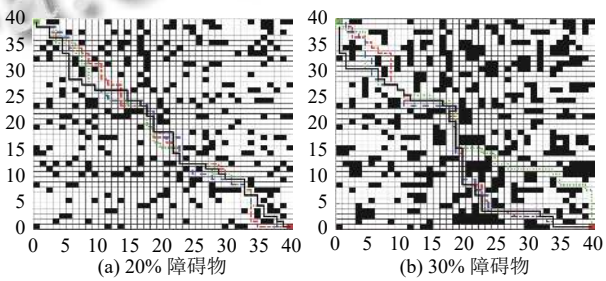

图 8 40×40 路径规划图

4 结论与展望

本文在二维静态环境下, 为了解决基本 Q 学习算法应用于移动机器人路径规划时存在的收敛速度缓慢、容易陷入局部最优等问题, 提出一种改进蚁群算法与动态 Q 学习算法融合的路径规划方法, 用于移动

© 中国科学院软件研究所 http://www.c-s-a.org.cn

机器人在不同形状、大小和布局的静态障碍物环境中进行路径规划. 通过仿真实验验证了本文所提出的路径规划算法可以避免陷入局部最优, 且能有效提高收敛速度. 通过本文的研究得出了以下结论.

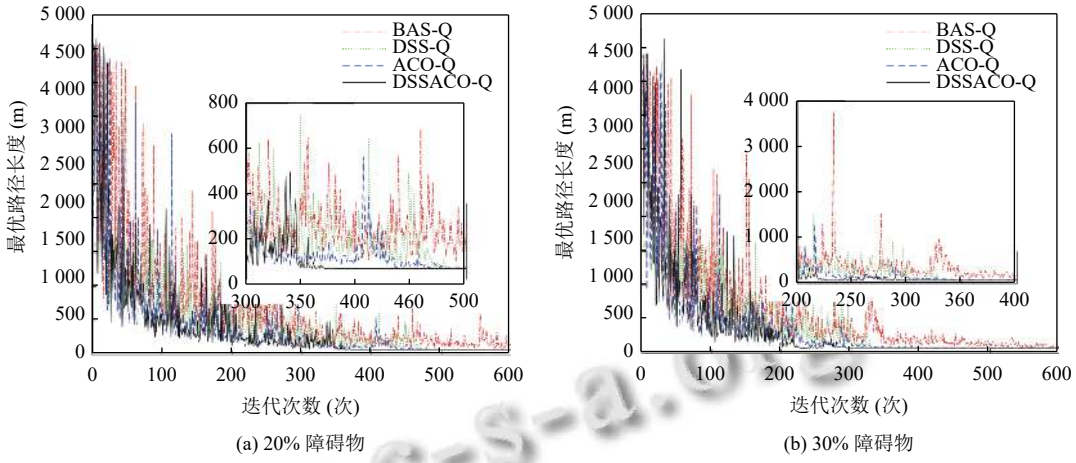

图 9 40×40 迭代曲线图

(1) 在自适应蚁群算法的基础上, 结合了精英蚂蚁模型和排序蚂蚁模型, 设计了一种新的信息素增量更新机制, 增强了较优路径对后续过程的指导作用, 以提高算法的收敛速度.

(2) 将改进后的蚁群算法获得的先验知识整合到基本 Q 学习的路径规划过程中, 指导了机器人的搜索过程, 避免了不必要的探索.

(3) 设计了一种动态搜索策略, 根据环境的反馈动态地调整ε值, 更好地平衡了探索与利用的关系.

在多种不同障碍物难度等级的环境中进行的仿真实验表明, 本文方法相较于基本 Q 学习算法、其他算法在收敛性和寻优耗时上都有一定的提高, 但仍存在以下不足.

(1) 本文算法在融合启发式搜索策略时引入了多个参数, 在进行仿真实验时, 需要同时调节多个参数, 在一定程度上, 增加了敏感性.

(2) 后续将研究本文方法在动态障碍物环境下的路径规划以及求解多目标路径规划的问题.

参考文献

1 Josef S, Degani A. Deep reinforcement learning for safe local planning of a ground vehicle in unknown rough terrain. IEEE Robotics and Automation Letters, 2020, 5(4): 6748–6755. [doi: 10.1109/LRA.2020.3011912]

2 闫皎洁, 张锲石, 胡希平. 基于强化学习的路径规划技术综述. 计算机工程, 2021, 47(10): 16–25.

3 张荣霞, 武长旭, 孙同超, 等. 深度强化学习及在路径规划中的研究进展. 计算机工程与应用, 2021, 57(19): 44–56.

4 Viswanathan S, Ravichandran KS, Tapas AM, *et al*. An intelligent gain based ant colony optimisation method for path planning of unmanned ground vehicles. Defence Science Journal, 2019, 69(2): 167–172. [doi: 10.14429/dsj.69.12509]

5 Jiang L, Huang HY, Ding ZH. Path planning for intelligent robots based on deep Q-learning with experience replay and heuristic knowledge. IEEE/CAA Journal of Automatica Sinica, 2020, 7(4): 1179–1189. [doi: 10.1109/JAS.2019.1911732]

6 刘建伟, 高峰, 罗雄麟. 基于值函数和策略梯度的深度强化学习综述. 计算机学报, 2019, 42(6): 1406–1438. [doi: 10.11897/SP.J.1016.2019.01406]

7 Tan B, Peng YY, Lin JG. A local path planning method based on Q-learning. Proceedings of the 2021 International Conference on Signal Processing and Machine Learning. Stanford: ACM, 2021. 80–84.

8 马向华, 张谦. 改进蚁群算法在机器人路径规划上的研究. 计算机工程与应用, 2021, 57(5): 210–215.

9 Low ES, Ong P, Cheah KC. Solving the optimal path planning of a mobile robot using improved Q-learning. Robotics and Autonomous Systems, 2019, 115: 143–161. [doi: 10.1016/j.robot.2019.02.013]

10 毛国君, 顾世民. 改进的 Q-learning 算法及其在路径规划中的应用. 太原理工大学学报, 2021, 52(1): 91–97.

11 徐晓苏, 袁杰. 基于改进强化学习的移动机器人路径规划方法. 中国惯性技术学报, 2019, 27(3): 314–320.

© 中国科学院软件研究所 http://www.c-s-a.org.cn

12 Pei M, An H, Liu B, *et al*. An improved dyna-*Q* algorithm for mobile robot path planning in unknown dynamic environment. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2022, 52(7): 4415–4425. [doi: 10.1109/TSMC.2021.3096935]

13 Hao B, Du H, Zhao JS, *et al*. A path-planning approach based on potential and dynamic Q-learning for mobile robots in unknown environment. Computational Intelligence and Neuroscience, 2022, 2022: 2540546.

14 Chang L, Shan L, Jiang C, *et al*. Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment. Autonomous Robots, 2021, 45(1): 51–76. [doi: 10.1007/s10514-020-09947-4]

15 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. 控制与决策, 2018, 33(10): 1775–1781.

16 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. 控制与决策, 2022, 37(2): 303–313. [doi: 10.13195/j.kzyjc.2020.0610]

17 田晓航, 霍鑫, 周典乐, 等. 基于蚁群信息素辅助的 Q 学习路径规划算法. 控制与决策, 2022. [doi: 10.13195/j.kzyjc.2022.0476]

18 石立培. 基于改进 Q 学习的智能车辆动态路径规划算法的研究 [硕士学位论文]. 秦皇岛: 燕山大学, 2018.

19 柯逸. 基于蚁群 Q-learning 算法的移动机器人路径规划 [硕士学位论文]. 北京: 中国矿业大学, 2022.

20 Maoudj A, Hentout A. Optimal path planning approach based on Q-learning algorithm for mobile robots. Applied Soft Computing, 2020, 97: 106796. [doi: 10.1016/j.asoc.2020.106796]

(校对责编: 牛欣悦)

www.c-s-a.org.cn

© 中国科学院软件研究所 http://www.c-s-a.org.cn