

doi: 10.12052/gdutxb.180044

未知环境下基于虚拟子目标的 对立Q学习机器人路径规划

汪盛民, 林 伟, 曾 碧

(广东工业大学 计算机学院, 广东 广州 510006)

摘要: 针对Q学习算法在复杂的未知环境下Q值更新速度慢, 容易产生维数灾难等问题, 提出了一种未知环境下基于虚拟子目标的对立Q学习机器人路径规划算法. 该算法根据移动机器人探索过的状态轨迹, 建立了2个状态链分别记录状态-动作对和状态-反向动作对, 并将每个单链当前状态的Q值, 依次反馈影响前一状态的Q值, 直到状态链的头端. 同时, 在局部探测域内通过寻找最优虚拟子目标的方法解决了大规模环境下Q学习容易产生维数灾难的问题. 实验结果表明, 在复杂的未知环境中, 该算法可以有效地加快算法学习的收敛速度, 提高学习效率, 以较优的路径完成机器人导航任务.

关键词: 移动机器人; 虚拟子目标; 对立Q学习; 未知环境
中图分类号: TP242.6 **文献标志码:** A **文章编号:** 1007-7162(2019)01-0051-06

Path Planning of Opposite Q Learning Robot Based on Virtual Sub-Target in Unknown Environment

Wang Sheng-min, Lin Wei, Zeng Bi

(School of Computers, Guangdong University of Technology, Guangzhou 510006, China)

Abstract: Aiming at the problem that in Q learning algorithm Q value is slow in updating speed in complex unknown environment and the dimensionality disaster is easy to occur, a path planning algorithm based on virtual subtarget for Q learning robot in unknown environment is proposed. According to the state trajectory explored by the mobile robot, two state chains are established to record the state-action pair and the state-reverse action pair respectively. The Q value of each single chain current state is fed back to the Q value of the previous state in turn till it affects the head of a single chain. Meanwhile, the problem that Q learning is prone to dimensionality disaster in large-scale environment is solved by finding the optimal virtual subtarget in the local detection domain. The experimental results show that the algorithm can effectively accelerate the convergence of the algorithm learning, improve the learning efficiency and complete the robot navigation task with a better path in the complex unknown environment.

Key words: mobile robot; virtual subtarget; opposite Q learning; unknown environment

随着服务、仓储物流产业的快速发展及其相关产业出现的升级难题, 移动机器人成为安全公司、物流公司的研究热点^[1-2]. 路径规划是移动机器人实现自主导航的关键技术之一, 路径规划是指在具有障碍物的环境中, 按照一定的评价标准, 寻找一条从起始位姿到目标位姿无碰撞的最优路径^[3]. 目前, 大多

数移动机器人是在高度结构化的已知地图中执行预先规定的动作序列. 例如, 在已知静态环境中的路径规划算法已经比较成熟, 主要有人工势场法^[4]、可视图法^[5]、RRT^[6]、遗传算法^[7]、粒子群算法^[8]等. 相对来说在未知、非结构化或动态的环境中时, 移动机器人对实际环境没有主动学习和自适应能力, 导致移动

收稿日期: 2018-03-16
基金项目: 广东省产学研合作专项项目(2014B090904080); 广东省应用型科技研发专项项目(2015B090922012); 东莞市产学研合作项目(2015509109107)
作者简介: 汪盛民(1990-), 男, 硕士研究生, 主要研究方向为人工智能、智能机器及物联网.
通信作者: 林伟(1965-), 男, 副教授, 主要研究方向为智能工程、嵌入式系统、计算控制. E-mail: wlin@gdut.edu.cn

机器人缺乏对环境的认识,无法确定自身的位置信息和环境中全部障碍物的信息.因此,采用强化学习使机器人在无法预先感知障碍物具体信息的前提下,在不断地与环境交互的过程中,可以自适应规划出一条从起始点到目标点且满足一定优化标准的无碰撞路径尤为重要^[9].Q学习是一种在线的、无监督的机器学习算法,这使得Q学习成为了强化学习在未知环境下移动机器人路径规划的一个研究热点^[10].

近年来,对于改进的Q学习移动机器人路径规划问题可以分为5类:(1)重新定义环境的状态空间;(2)动作的随机选择策略;(3)Q表的初始化策略;(4)Q函数的更新策略;(5)奖惩函数的设计.对此,国内外的一些学者对Q学习算法提出了一些改进方法.Konar A等^[11]提出了一种改进的确定性Q学习路径规划方法,假设知道从当前状态到下一个状态和目标之间的距离,该算法就可以通过利用Q学习的4个派生性质有效地用于更新Q表中的条目,而不是像传统Q学习算法重复更新,因此节省了存储空间,但算法时间复杂度被提高了,导致收敛速度下降的问题.Hwang H J^[12]提出了一种Q(λ)学习算法,有效加快了收敛速度,但在生成单链的过程中会去除原始状态路径中的所有环,导致算法能找到一个可行解,但可能不是最优路径.Pietersm^[13]提出了一种基于经验回放的Q学习算法,通过经验函数 $e(s,a)$ 来弥补前期对环境模型认知不足的缺点,在一定程度上加快了算法的收敛速度,但是在算法中除了要更新Q值函数的同时还要更新 $e(s,a)$,增加了算法的时间复杂性,并且忽略了在环境规模比较大时Q值存在“维数灾难”的问题.

针对上述研究现状及不足,本文提出了基于虚拟子目标的对立Q学习机器人路径规划.该算法通过建立双向的状态链,使当前状态的动作决策快速影响到前面的状态动作对,来改善传统Q学习数据传递的滞后性,提高收敛速度,同时通过在局部探测域内寻找最优虚拟子目标的方法解决了大规模环境下Q学习容易产生的维数灾难问题.

1 相关算法

1.1 激光雷达及激光数据坐标转换算法

本文采用sick的激光雷达如图1所示,截取后的扫描范围为180°,通过扫描可以得到栅格地图^[14].利用激光雷达测距速度快、精度高、抗干扰能力强和范围广等特点,机器人可以快速有效地获取周围的环境信息.机器人的感知信息来自于激光雷达旋转

180°所得的距离数据,通过旋转角度和激光末端点的测量距离,计算出机器人离周围障碍物的距离.在二维坐标系中,激光末端点坐标假设以 $s(s_x, s_y)$ 形式表示,激光雷达在全局坐标系中的位姿为 $\phi = (\phi_x, \phi_y, \phi_\theta)$,而通过转换矩阵 T_ϕ ,将 s 转换到全局坐标系中,如式(1)所示.

$$T_\phi s = \begin{pmatrix} \cos \phi_\theta & -\sin \phi_\theta \\ \sin \phi_\theta & \cos \phi_\theta \end{pmatrix} s + \begin{pmatrix} \phi_x \\ \phi_y \end{pmatrix}. \quad (1)$$



图1 激光雷达与移动平台

Fig.1 Lidar and mobile platform

1.2 经典的Q学习算法

Q学习算法^[15]采用每个状态-动作对所对应的 $Q(s,a)$ 作为估计函数通过不断地循环更新Q值,使相邻状态间Q值估计的差异达到一定的收敛条件,即

$$Q^*(s,a) = r + \gamma \sum T(s,a,s') \max_{a'} Q^*(s',a'). \quad (2)$$

其中, r 表示执行动作 a 得到的奖惩值, γ 表示折扣因子, $T(s,a,s')$ 表示在状态 s 采取动作 a 之后转换到状态 s' 的概率.通过查找给定状态对应的所有可能动作,根据贪婪策略选择一个具有最大Q值的动作执行,选择策略为

$$\pi^*(s) = \arg \max_a Q^*(s,a). \quad (3)$$

Q学习算法中Q值得更新公式为

$$Q(s_t, a_t) = Q(s_t, a_t) + \beta(r + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)). \quad (4)$$

其中, β 表示学习因子, $Q(s_t, a_t)$ 为 t 时刻的Q值.

2 基于虚拟子目标的对立Q学习算法

针对已有算法存在的问题,本文算法的基本思想是根据移动机器人传感器的检测范围,以机器人作为中心取得一个局部的环境作为一个窗口,以窗口滚动的方式来遍历整个环境,移动机器人在局部环境中根据得到的虚拟子目标来进行对立Q学习,并得到一条局部的最优路径.移动机器人按照该路径前进一段距离 L 后,再产生一个新的窗口,再次进行对立Q学习并得到最优路径.随着窗口的移动会产生一

系列的局部最优路径并到达虚拟子目标点,将这些路径有序地组织起来就构成了一条全局优化的路径。

假设工作环境为 G ,移动机器人的起始点为 R_{start} ,终点为 R_{goal} . G 中存在的障碍物分别为 $\text{obs}_1, \text{obs}_2, \dots, \text{obs}_n$ 要求移动机器人从起始位置无碰撞地移动到目标位置。

定义1 任意两个栅格之间的距离计算公式为

$$d(g_i, g_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \quad (5)$$

定义2 S 为所有状态空间 $\{s_1, s_2, \dots, s_n\}$ 的集合, A 表示动作集合, $A = \{\text{左}, \text{左上}, \text{上}, \text{右上}, \text{右}, \text{右下}, \text{下}, \text{左下}\}$.

定义3 obs 为所有障碍物 $\{\text{obs}_1, \text{obs}_2, \dots, \text{obs}_n\}$ 的集合, g_i 表示第 i 个栅格, 如果 $g_i \in G, \wedge g_i \notin \text{obs}$, 则称 g_i 是可行点, 所有的可行点集合称之为 AV_g .

定义4 记 P 表为线性表用来记录机器人已经过的栅格集合。

定义5 记 P_{record} 和 P_{length} 分别表示当前迭代的可行路径和路径长度, $P_{\text{record_his}}$ 和 $P_{\text{length_his}}$ 分别表示历史最优的可行路径和路径长度。

定义6 机器人的视野区域 $V \in (d(g_i, g_j) < D)$, 其中 D 为激光雷达探测的最远距离。

2.1 对立Q学习算法

对立意味着在每个时间 t , 如果移动机器人在给定状态 s_2 下接受采取动作 a_2 的奖励, 那么在这一时刻移动机器人也可以在相同状态下接受针对相反动作的惩罚, 而不需采取相反的动作来训练. 它将会同时更新2个 Q 值, 如图2所示, 分别为 v_1, v_2 (v_2 为 v_1 状态相反动作值). 因此, 移动机器人可以同时探索正向行动和反向行动, 每个时间步长更新当前正向动作 Q 值的同时, 更新了相反动作的 Q 值, “双重更新”加快了学习过程。

	a_1	a_2	a_3	a_4	a'_1	a'_2	a'_3	a'_4
s_1	—	—	—	—	—	—	—	—
s_2	—	v_1	—	—	—	v_2	—	—
s_3	—	—	—	—	—	—	—	—
s_4	—	—	—	—	—	—	—	—

图2 Q 值矩阵

Fig.2 Q Matrix

其中, 定义 $M(t) \leftarrow [s_t, a_t, r_t]$ 来记录移动机器人所经历过的状态-动作对, r_t 表示为 t 时刻回报值. 定义 $M'(t) \leftarrow [s_t, a'_t, r'_t]$ 来记录移动机器人所经历过的状态-

反向动作对, 则可以通过记忆矩阵中的状态-动作对的回溯来更新 Q 值, 其更新过程如下:

$$\text{For } k = t-1, t-2 \dots 3, 2, 1 \\ Q(s_k, a_k) \leftarrow (1-\beta)Q(s_k, a_k) + \beta[r_k + \gamma \max_{a_{k+1} \in A} Q(s_{k+1}, a_{k+1})]. \quad (6)$$

$$Q(s_k, a'_k) \leftarrow (1-\beta)Q(s_k, a'_k) + \beta[r'_k + \gamma \max_{a_{k+1} \in A} Q(s_{k+1}, a_{k+1})]. \\ \text{Until } k = 1 \quad \text{end} \quad (7)$$

其中, s 是状态, a_k 是 k 时刻的动作, a'_k 是 k 时刻反向动作. 在不断的迭代过程中, $t+1$ 时刻最新状态 s_t 对应 Q 值的同时, 会对两个单链中状态 s_t 之前的状态 $s_{t-1}, s_{t-2}, \dots, s_3, s_2, s_1$ 也进行更新操作。

第1步训练:

$$Q(s_0, a_0) \leftarrow Q(s_1, a_1), \\ Q(s_0, a'_0) \leftarrow Q(s_1, a'_1).$$

第2步训练:

$$Q(s_0, a_0) \leftarrow Q(s_1, a_1) \leftarrow Q(s_2, a_2), \\ Q(s_0, a'_0) \leftarrow Q(s_1, a'_1) \leftarrow Q(s_2, a'_2).$$

...

第 n 步训练:

$$Q(s_0, a_0) \leftarrow Q(s_1, a_1) \leftarrow \dots \leftarrow Q(s_{n-1}, a_{n-1}) \leftarrow Q(s_n, a_n), \\ Q(s_0, a'_0) \leftarrow Q(s_1, a'_1) \leftarrow \dots \leftarrow Q(s_{n-1}, a'_{n-1}) \leftarrow Q(s_n, a'_n).$$

根据对立链的数据更新过程可知, 状态 s_0 要获得状态 s_n 的反馈信息只需要一次回溯过程, 无需训练 n 次. 算法步骤如下。

步骤1 对当前的环境进行栅格的划分, 设置初始位置 R_{start} 和目标位置 R_{goal} 并将机器人置于起始位置, 初始化迭代次数 $\text{count} = 0$, 设置最大的迭代次数 N , 初始化先验信息即对每个栅格的 Q 值进行式(8)的计算。

$$Q(s_i, a) = 1/d(g_i, g_{\text{goal}}). \quad (4)$$

步骤2 机器人对当前位置 g_i 的所有邻域进行判断, 根据当前栅格 g_i 的状态根据式(3)选择一个动作 a . 执行动作 a 后, 机器人移动到 g_j , 如 $g_j \in \text{obs}$, 跳转到步骤3; 如果 $g_j = R_{\text{goal}}$, 则跳转到步骤4; 如果 $g_j \in AV_g$, 跳转到步骤5。

步骤3 对当前状态-动作给予一定的惩罚, 对立状态-动作对给予一定的奖励。

步骤4 到达目标点, 记录下当前的 P_{record} 和 P_{length} , 如 $P_{\text{length}} < P_{\text{length_his}}$ 将 $P_{\text{record_his}}$ 和 $P_{\text{length_his}}$ 更新为当前的 P_{record} 和 P_{length} . 对当前状态-动作给予最高的奖励。

步骤5 将 g_j 加入到 P 线性表中, 对当前状态-

动作给予一定的奖励,同时根据式(6)、(7)更新 Q 值函数,并跳转到步骤2.

步骤6 $\text{count} = \text{count} + 1$, 如果 $\text{count} < N$, 跳转到步骤2; 如果 $\text{count} > N$, 跳转到步骤7.

步骤7 迭代结束, 根据 $P_{\text{record_his}}$ 得到一条从起始位置到终点位置的最优路径.

按照上述步骤, 其流程图如图3所示.

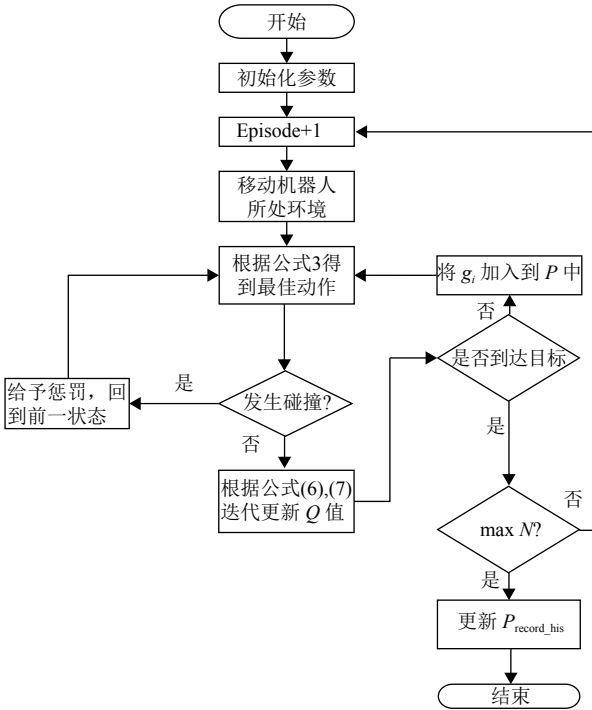


图 3 对立Q学习路径规划
Fig.3 Opposite Q learning path planing

2.2 虚拟子目标的选取策略

根据移动机器人传感器的检测范围, 以机器人作为中心取得一个局部的环境作为一个窗口, 将窗口内的环境栅格化, 根据移动机器人的当前位置计算每个栅格的坐标位置, 并将目标点的位置作为指引, 如果 $R_{\text{goal}} \notin V$, 将通过 R_{goal} 和障碍物信息选择一个虚拟目标点作为局部窗口的目标位置. 如图4所示.

虚线圆表示的是激光雷达的最大扫描范围, 在扫描区域与障碍物膨胀边界交汇的点分别为 P_1 、 P_2 . 这两个点将作为虚拟子目标的候选点. 由于 P_1 、 P_2 到目标点 R_{goal} 的距离 $d_{R_1 \rightarrow P_1} + d_{P_1 \rightarrow R_{\text{goal}}} > d_{R_1 \rightarrow P_2} + d_{P_2 \rightarrow R_{\text{goal}}}$. 因此将 P_2 视为最佳的子目标点. 同样, 在位置 R_2 得到机器人最优的虚拟子目标点为 P_3 .

由上面的分析, 可得式(9)为当前探测域内最优虚拟子目标的策略:

$$\min(d_{R \rightarrow P_i} + d_{P_i \rightarrow R_{\text{goal}}}), i \in n. \quad (5)$$

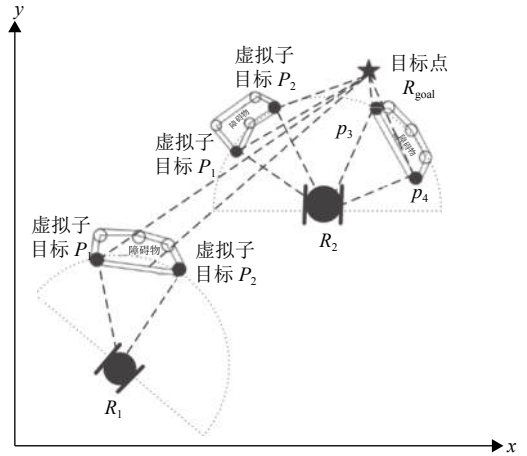


图 4 虚拟子目标点的选取
Fig.4 Choose virtual sub targets

其中 P_i 表示在当前区域内的虚拟子目标候选节点, R 表示机器人的位置, R_{goal} 表示目标点的位置.

虚拟子目标生成算法的基本步骤如下.

步骤1 根据激光雷达的扫描范围和障碍物的信息, 获得 n 个虚拟子目标候选点.

步骤2 将间隔小于某个固定值 d 的相邻可视点组合成一个障碍物群组.

步骤3 根据式(9)计算出当前探测区域中最优虚拟子目标.

2.3 基于虚拟子目标的对立Q学习算法步骤

为了提高全局规划的速度将给出障碍物的密度函数为

$$P_{\text{obs}} = n_{\text{obs}} / n_{\text{total}}. \quad (6)$$

其中, n_{obs} 表示窗口中障碍物的个数, n_{total} 表示窗口中栅格的总个数.

在一个窗口中前进的距离 L 定义为

$$L = \begin{cases} 1 \text{ 个栅格长度,} & P_{\text{obs}} > n_2; \\ 2 \text{ 个栅格长度,} & n_1 < P_{\text{obs}} < n_2; \\ 1/3 \text{ 局部路径长度,} & P_{\text{obs}} < n_1. \end{cases} \quad (7)$$

其中, $0 < n_1 < n_2 < 1$, 当在窗口中障碍物较少时, 机器人会走1/3的局部路径长度, 然后重新规划新的路径; 当障碍物较多时每走2个栅格就规划一次; 当障碍物特别多时每走一个栅格就规划一次, 这样就相对来说减少了局部路径规划的次数. 局部路径长度的取值需要进行大量的实验来确定, 取值的原则就是局部路径叠加后得到的全局路径最优或近似最优. 同样窗口大小的选取也会影响到全局的最优结果, 如果取值太小容易陷入到局部最优, 如果取值太大, 会降低算法的收敛速度. 因此 L 的取值和窗口大小的

取值都需要大量的实验设定, 本文设置的窗口大小通过设置传感器的检测范围来确定. 其流程图如图5所示.

步骤1 初始化移动机器人的激光雷达的视野半径 D , 阈值 n_1 和 n_2 , 起始点 g_{start} 和终点 g_{goal} .

步骤2 如果 g_{goal} 是在移动机器人当前的视野 V 范围内, 则使用对立Q学习算法规划出一条从机器人当前位置到目标点位置的优化路径, 并且算法结束.

步骤3 根据式(9)计算最终选取最优虚拟子目标 g_{goal} , 对立Q学习算法规划出一条从机器人当前位置到目标点位置的优化路径, 并记录下路径长度.

步骤4 根据式(10)计算 P_{obs} , 根据式(11)计算 L , 并且移动沿局部规划路径移动 L .

步骤5 根据当前位置和视野 V , 更新当前环境, 并对当前环境进行栅格化处理, 然后回到第二步骤.

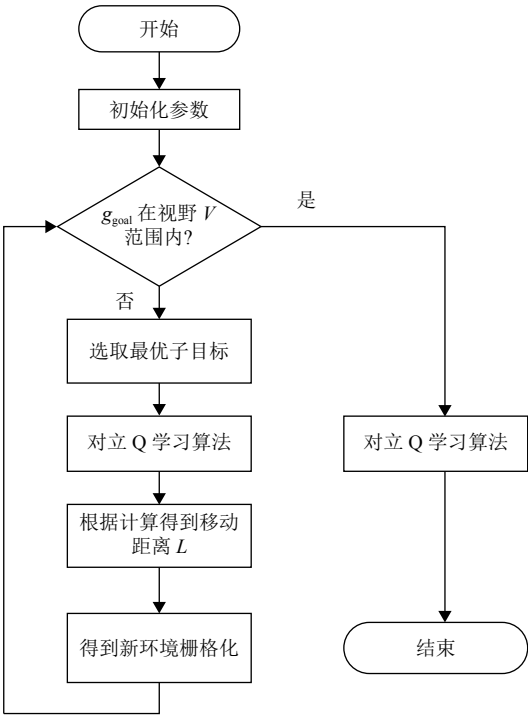


图 5 本文算法流程图
Fig.5 The flow chart of the algorithm in this paper

3 实验结果与分析

为了验证在大规模环境下基于虚拟子目标对立Q学习算法的效果, 本文在ROS机器人仿真平台上搭建未知环境, 环境中包括了四周的墙壁、箱子和挡板等障碍物. 在50×50的未知环境下进行实验, 实验中相关参数设置如下: 学习因子 $\beta=0.4$, 折扣因子 $\gamma=0.95$, 最大迭代数 $count_{max}=120$. 每一次迭代, 移动

机器人都是从规定的起始位置开始训练, 当迭代次数达到最大的时候结束整个的训练过程. 机器人的视野半径为 $D=4$ 根据前面大量实验的经验总结, 设置 $n_1=0.28, n_2=0.71$. 在第一个视野范围内, 机器人采用对立Q学习算法规划出一条局部路径为 L_1 , 根据公式(11)计算出 $P_{obs}=0.25 < n_1$, 所以移动机器人将 L 设置为 L_1 长度的1/3(即 $L=3$); 然后沿着 L_1 前进3步之后, 更新环境信息重新设置虚拟子目标并根据对立Q学习规划路径, 重新计算 L ; 重复这个过程, 直到探测到最终的目标. 通过实时的调整 L , 减少了局部规划的次数, 提高了算法的效率. 下面4种算法都是在相同的条件下进行的, 根据收敛的 Q 指导移动机器人完成最终的导航, 如图6所示.

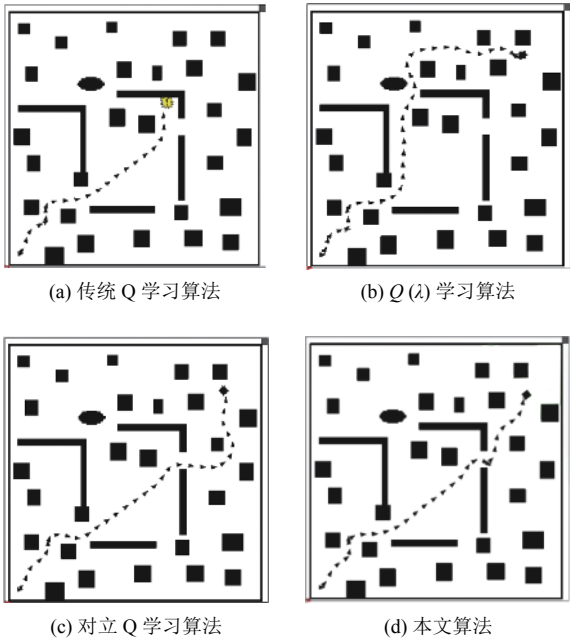


图 6 复杂环境中机器人导航路径
Fig.6 Robot navigation path in complex environment

从图6可以看到, 每个算法的导航路径有所差异, 传统Q学习算法由于在U型区域无法找到合适的路径, 与障碍物发生了碰撞, $Q(\lambda)$ 学习算法走了35步, 对立Q学习算法走了27步, 本文算法走了20步.

图7中传统的Q学习算法在训练的过程中陷入到U型区域一直在角落徘徊, 导致无法到达目的地, 其他3种算法都能到达目标点, 并且可以看出本文算法在训练到第8次左右就开始收敛, 优于其他3种算法.

为了验证本文算法的实时性, 在这32个障碍物随机变化的环境中进行了20组实验, 测得的规划时间结果如图8所示. 平均规划时间为0.152 s, 基本能满足移动机器人动态路径规划的实时性的要求.

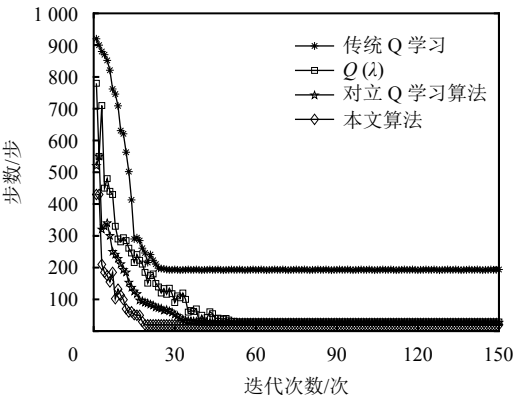


图 7 4种算法迭代收敛速度

Fig.7 Iterative convergence rate of four algorithms

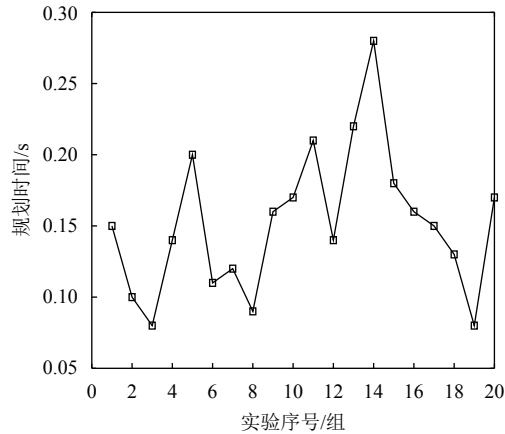


图 8 测试规划时间

Fig.8 The measurements experimental planning

根据获得的导航路径表明本文提出的强化学习算法可以在未知环境中完成导航任务. 同时在相同的环境条件下, 与传统的Q学习, $Q(\lambda)$ 算法和对立Q学习算法进行对比, 各个算法的性能如表1所示.

表 1 收敛性能对比

Tab.1 Convergence performance comparison

算法	N_1 /次		N_2 /步	
	简单环境	复杂环境	简单环境	复杂环境
传统Q学习算法	60	181	9 006	36 897
$Q(\lambda)$ 算法	39	53	4 681	10 698
对立Q学习算法	25	37	3 654	9 698
本文算法	15	20	1 947	3 628

其中, N_1 表示收敛尝试的次数, N_2 表示收敛时行走的总步数, 从表1可以看出在简单环境中本文算法收敛的速度是对立Q学习算法的1.67倍, 是 $Q(\lambda)$ 的2.6倍, 是传统Q学习算法的4倍. 在复杂环境中本文算法收敛的速度是对立Q学习算法要快1.85倍, 是 $Q(\lambda)$ 的

2.65倍, 是传统Q学习算法的9.05倍. 在复杂环境中更能体现本文算法的优越性.

为了进一步验证本文算法对环境适应能力, 在障碍物更加复杂的环境中进行仿真实验, 如图9所示, 在复杂的环境中移动机器人也能规划一条无碰撞的路径, 表明本文算法对各种环境有较强的适应能力.

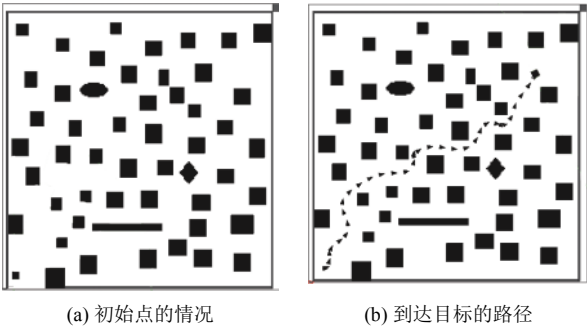


图 9 复杂环境下的规划路径

Fig.9 Path planning in complex environment

4 结语

虽然传统的Q学习算法能够在未知的环境下找到一条到达目标点的无碰撞路径, 但是由于对环境的先验知识不足, 使得算法迭代速度慢, 特别是在环境规模增大, 状态空间会急剧增大导致维数灾难. 本文针对上述问题, 在传统Q学习的基础上, 依据实际情况, 提出了基于虚拟子目标的对立Q学习, 该算法通过建立两条单链来回溯更新Q值并通过设置虚拟子目标将复杂大环境转化为局部简单环境. 采取这些措施后, 大幅度地提高了算法的实时性和对环境的适应能力. 实验结果表明该算法简单、速度快、对环境的适应能力强等特点, 特别是当环境规模较大时, 更能体现算法的优越性.

参考文献:

[1] 赵晓东, 鲍方. 清洁机器人路径规划算法研究综述[J]. 机电工程, 2013, 30(11): 1440-1444.
ZHAO X D, BAO F. Survey on cleaning robot path planning algorithm [J]. Journal of Mechanical & Electrical Engineering, 2013, 30(11): 1440-1444.

[2] LOPEZ A, PAREDES R, QUIROZ D, et al. R-obotman: A security robot for human-robot interaction[C]// International Conference on Advanced Robotics. Singapore Sands Expo: IEEE, 2017: 7-12.