摘 要

随着移动机器人技术发展的愈发成熟，移动机器人在我们生活中扮演着越来越重要的角色。在移动机器人相关技术领域中，路径规划是非常重要的一项技术。在理论中，我们可以将地图数据传送给移动机器人进行分析，从而得到最佳路径，但在实际应用中，移动机器人所处的环境是部分已知的，其不可能获得全局环境信息。因此，对于在陌生环境中不具有先验知识的机器人，研究其探索和规划路径功能具有非常重要的意义。

本文以未知环境中移动机器人路径规划为背景，通过对强化学习算法Q-learning进行改进来为移动机器人寻找一条从起始点到终点的最优无碰路径。在原有强化学习算法Q-learning基础上，针对不同状态赋予不同收敛优先级，并且通过回溯思想改善原有Q-learning学习中数据传递的滞后性，提高了算法的收敛速度。仿真实验表明，改进Q-learning算法收敛速度快，能够为移动机器人寻找一条最优无碰路径。

**关键词：**Q-learning，移动机器人，路径规划，避障，强化学习，最优路径

Abstract

With the development of increasingly robotics, robots play an increasingly important role in our lives. In the related field of robotics, the path planning is a very important technology. In theory, we can transport map data to the robot for analysis to obtain the best path. However, in practice, it is impossible to know the map data in an unknown environment, geographic information. Therefore, the study which a robotic without prior knowledge explore and find the best way in an unfamiliar environment is a very important significance.

In this paper, robot path planning in unknown environment as the background, we learn reinforcement learning algorithm to achieve robot path planning. Based on the original reinforcement learning Q-learning algorithm, we propose improvements to enhance performance, complete obstacle avoidance, get the optimal path function.

**Keywords:** Q-learning, obstacle avoidance, reinforcement learning, the optimal path

目 录

[第1章 引言 1](#_Toc452301880)

[1.1 研究背景及意义 1](#_Toc452301881)

[1.2 论文结构及安排 4](#_Toc452301882)

[第2章 移动机器人路径规划及强化学习算法 5](#_Toc452301883)

[2.1 移动机器人路径规划 5](#_Toc452301884)

[2.1.1 全局路径规划技术 5](#_Toc452301885)

[2.1.2 局部路径规划 7](#_Toc452301886)

[2.2 强化学习算法 10](#_Toc452301887)

[2.2.1 强化学习基本原理 10](#_Toc452301888)

[2.2.2 Markov决策过程 13](#_Toc452301889)

[2.2.3 动态规划方法 14](#_Toc452301890)

[2.2.3.1 值迭代 14](#_Toc452301891)

[2.2.3.2 策略迭代 15](#_Toc452301892)

[2.3 本章小结 16](#_Toc452301893)

[第3章 经典强化学习算法及改进算法的提出 17](#_Toc452301894)

[3.1 Q-learning算法 17](#_Toc452301895)

[3.1.1 Q-learning算法的基本原理 17](#_Toc452301896)

[3.1.2 Q-learning算法的流程 18](#_Toc452301897)

[3.1.3 Q-learning算法的收敛性分析 19](#_Toc452301898)

[3.1.4 收敛步数 21](#_Toc452301899)

[3.2 算法简介 22](#_Toc452301900)

[3.2.1 跟踪迹的引入 22](#_Toc452301901)

[3.2.2 Q()算法流程 24](#_Toc452301902)

[3.3 改进Q-learning算法 26](#_Toc452301903)

[3.3.1 算法原理 26](#_Toc452301904)

[3.3.2 算法流程 27](#_Toc452301905)

[3.3.3 数据更新图 28](#_Toc452301906)

[3.4 本章小结 29](#_Toc452301907)

[第4章 Q-learning算法及其改进算法仿真模拟 31](#_Toc452301908)

[4.1 仿真环境介绍 31](#_Toc452301909)

[4.2 实验设计 32](#_Toc452301910)

[4.2.1 智能体路径规划系统框架 32](#_Toc452301911)

[4.2.1.1 环境模型 32](#_Toc452301912)

[4.2.1.2 机器人模型及其动作空间表示 33](#_Toc452301913)

[4.2.1.3 奖赏函数定义 33](#_Toc452301914)

[4.2.1.4 Q值表建立 34](#_Toc452301915)

[4.2.1.5 其他参数初始化 34](#_Toc452301916)

[4.2.2 路径规划策略 34](#_Toc452301917)

[4.3 仿真结果与分析 36](#_Toc452301918)

[4.4 本章小结 37](#_Toc452301919)

[第5章 结束语 39](#_Toc452301920)

[5.1本文内容 39](#_Toc452301921)

[5.2下一步学习工作方向 39](#_Toc452301922)

[参考文献 41](#_Toc452301923)

[致谢 42](#_Toc452301924)

[外文资料原文 43](#_Toc452301925)

[外文资料译文 48](#_Toc452301926)

第1章 引言

伴随着社会的发展和进步，人类的生活越来越需要机器人的辅助，因此越来越多的国家都在机器人领域投入很大的精力进行研究。

1.1 研究背景及意义

移动机器人是一种在复杂环境下工作，具有自行组织、自主运行、自主规划的智能机器人，它集中了传感器技术、信息处理、电子工程、计算机工程、自动化控制工程以及人工智能等多学科的研究成果，代表机电一体化的最高成就，是目前科学技术发展最活跃的领域之一。随着机器人性能不断地完善，移动机器人的应用范围大为扩展，不仅在工业、农业、医疗、服务等行业中得到广泛的应用，而且在城市安全、国防和空间探测领域等有害与危险场合得到很好的应用。因此，移动机器人技术已经得到世界各国的普遍关注。

移动机器人根据移动方式来分，可分为：轮式移动机器人、步行移动机器人单腿式、双腿式和多腿式、履带式移动机器人、爬行机器人、蠕动式机器人和游动式机器人等类型；按工作环境来分，可分为：室内移动机器人和室外移动机器人；按控制体系结构来分，可分为：功能式水平式结构机器人、行为式垂直式结构机器人和混合式机器人；按功能和用途来分，可分为：医疗机器人、军用机器人、助残机器人、清洁机器人等

移动机器人的研究始于60年代末期。斯坦福研究院的和Charles Rosen等人，在1966年至1972年中研发出了取名为Shakey的自主移动机器人[1]，目的是研究应用人工智能技术，在复杂环境下机器人系统的自主推理、规划和控制。我国机器人研究开始于世纪年代，至今已有多年。年，北京机械工业自动化研究所曹祥康在江苏嘉兴召开了全国性机械手技术交流大会，是我国第一个以机器人为主题的大型会议。比较有代表性的有：清华大学研制了智能移动THMR-III V型机器人；中科院自动化研究所研制出智能保安机器人，全方位移动式机器人视觉导航系统，智能移动机器人通用平台AIMR;香港城市大学研发了自动导航车及服务机器人；哈尔滨工业大学研制了导游机器人；国防科技大学研发成功了双足机器人等；“海龙2号”是我国自主研制的水下机器人，是我国目前仅有能在3500米水深、海底高温和复杂地形的特殊环境下幵展海洋调查和作业的最高精技术装备，它除了在潜水深度上的优势之外，还在国际上首次采用了一些自主研发的先进技术，包括虚拟控制系统和动力定位系统。

移动机器人导航移动机器人学的关键技术，是指按照预先给出的任务，根据已知的地图信息作出全局路径规划，并在行进过程中，不断感知周围的局部环境信息，自主地作出各种决策，随时调整位姿，引导自身安全行驶或跟踪已知路径，达到目标位姿，是移动机器人各项研究应用的基础和前提。自主导航是移动机器人导航的一项重要的基本能力，它主要解决的问题可以归纳为三个方面：“1.现在何处？2.要往哪去？3.如何去？”也就是移动机器人的地图构建、定位和路径规划题。地图构建是移动机器人在自主导航的过程中，通过传感器感知环境并建立环境模型，这是树立环境模型最常用且有效的方法；定位移动机器人在工作环境中探索所处位置的过程，也就是机器人的全局坐标和姿态[2,3]；路径规划技术[4,5]就是移动机器人在具有障碍物的环境上按照一定的评价标准（如作代价最小、行走路径最短、行走时间最短等）、寻找一条从给起点到达目标终点的无碰路径

但是不管采取何种导航方式，移动机器人上主要是完成路校规划、定位和避障等任务。路径规划问题是移动机器人自助式导航技术研究中的关键技术之，它是指移动机器人按照某一性能指标，搜索一条从起始位置到目标位置的最优或次优的无碰路径。

随着计算机、传感器及控制技术的发展，特别是各种新算法的不断涌现，移动机器人路径规划技术已经取得了丰硕的研究成果，特别是在周围环境未知或部分未知的局部路径规划（人工势场法、模糊逻辑算法、神经网络法、进化算法、强化学习等），国内外学者已经作了大量的研究和改进，下面综述中将对与本文有关的一些方法进行讨论。

1986年，Khatib首次提出用人工势场法解决机器人避障问题，在此之后许多学者不仅发展了这一方法并将其应用于机器人的实时路径规划和足球机器人的运动控制。人工势场法可以实现快速的控制，所以人工势场法丨被广泛的应用于实时的运动控制。

E.Shi等进一步对传统的人工势能场方法进行改进，他提出的方法虽然能够使移动机器人避幵障碍物，但其实施性并不强，虽然克服了零势能场（在这种情况下机器人无法到达目标位置）的情况，但又出现了很多问题，比如障碍物的影响范围、引力增益和斥力增益会影响到机器人运动过程中振动的程度、运动路径的平滑度、路径的长短及其运动过程中是否会碰撞到障碍物等，所以必须同时选择这三个系数的最优值，而在实验过程中是很难选取的，必须通过反复的实验才能选取较为合适的值；王奇志等提出了一种改进的人工势场法，通过排除一个距离机器人最远的障碍物，同时加一个同等大小反方向力来消除零势能点，从而实现消除零势能域，达到多障碍物情况下机器人运动规划的快速、实时、避障的效果，结果表明，该算法对多个障碍物和非静态的障碍物同样适用。陈华志等采用一种具有速度负反馈的基于模糊控制的移动机器人避障算法进行研究和仿真对移动机器人在实际环境中进行了实验，并证明了方法的可行性；Lee等使用模糊控制来决定移动机器人可能的十三个移动方向，使机器人在这种导航下避免机器人内部碰撞；Pradhan等在完全未知的环境中使用模糊逻辑对多达1000个机器人进行了导航实验，在试验中，对四个输入变和两个输出变量进行模糊化，最后证明了高斯隶属函数在移动机器人导航中最有效；Yang等设计了移动机器人的动态模型，根据多机器人的局部信息，建立了引力、斥力函数的模糊控制器，基于最优控制理论，来衡量系统的状态性能指标；刘利等针对海域不同的障碍环境，从运动学特性出发，提出了一种多主水航行器（AUV）编队避障策略。该策略是在领航跟随法的基础上，利用编队队形的几何关系，引入虚拟AUV，再采用模糊控制避障策略，使编队能顺利通过障碍物，到达目的地；Rigatos等研究了在包含固定障碍物和移动物体的部分未知的环境下，根据移动机器人的运动问题和控制规律，结合滑膜与模糊逻辑控制的基本原则，提出使机器人达到期望的运动效的控制器，即降低滑动模式模糊逻辑控制器的复杂度，具有稳定性和简易性；Montaner等为移动机器人导航设计了个模糊逻辑控制器，该控制器可以接收确定性和模糊的信息，并在一个由七个超声波传感器来获信息的环境中对移动机器人进行了实验研究了移动机器人在未知群集环境屮反应式导航的问题，中文把反应式导航定义为感知数据和命令的关系，建立一个反应式导航相当于对机器人提供了一个运动的地图，在此基础上提出了另一种“普遍”的逼近方法—模糊逻辑，并揭示了如何利用行为分解的方法来选择模糊规则；Tan等提出了一种可加速减速的模糊控制器模型，使移动机器人在动态环境中自主导航，包括寻找目标、躲避障碍和寻找最优路径。

Yang和Meng提出了一种离散拓扑结构的祌经网络模型，移动机器人使这种并联的祌经网络模型在离散化的完全已知环境中进行路径规划，并假定机器人的速度恒定，这在真实的环境中对移动机器人进行控制是没意义的；Zhu等提出了种基于模糊神经网络将传感器信息与机器人的移动结合起来的导航方法，这种力法可以使移动机器人充分感受周围环境、自主避开静态和动态障物，并在不同情况下避开“死循环”产生到达目标的合理的轨迹，通过仿真证明了该方法的实用性和有效性；Nishida等基于神经元系统提出了种感知预测机器人伙伴的控制方法，该预测方法对于减少计算量、提取感知信息是非常重要的；Kala等在文中提出了遗传算法在移动机器人导航中的应川，使机器人可用于业界派调查，数据采集，移动机器人可以在动态环境中避免内部碰撞证明了这种算法的有效性。

1.2 论文结构及安排

本文的主要内容是研究和分析Q-learning强化学习算法使得移动机器人在未知环境中，通过自主学习环境信息，得出到达目标位置的最佳路径。并基于Q-learning算法，提出新的算法，使得在保证准确性的前提下性能得到优化。

本文的主要结构安排如下：

第一章主要介绍了移动机器人路径规划的背景意义及其在现实生活中的应用情况。

第二章主要介绍了移动机器人路径规划算法以及强化学习算法的研究，阐述了目前常见的移动机器人算法和强化学习算法在其领域的实现。

第三章主要分析了Q-learning强化学习算法，并在其基础之上，提出了改进的Q-learning算法。

第四章介绍了实验环境，将经典Q-learning算法和改进Q-learning算法进行了对比，并对实验结果进行了分析。

第五章为结束语，对本文工作进行总结，并对本文的不足之处进行了说明，对下一步的研究工作进行了展望。

第2章 移动机器人路径规划及强化学习算法

2.1 移动机器人路径规划

移动机器人相关学习中的一个非常重要的研究领域就是机器人导航，所有类型的机器人，在工作的时候，一般都是根据特定标准来找出一条最优或者接近最优的路径行走。这些标准常见的有：距离短、时间少、性能高、能耗少等。 在机器人寻路过程中，也会有一些限制条件，比如必须经过某一点，躲避障碍等等，我们要在这些限制条件下，对算法进行优化。来算法，使得在保证准确性的前提下性能得到优化。

在移动机器人导航中，路径规划又是它的重要任务之一。路径规划可以分类为：

1. 已知环境，静态障碍物路径规划；
2. 未知环境，静态障碍物路径规划；
3. 已知环境，动态障碍物路径规划；
4. 未知环境，动态障碍物路径规划。

除了这种分类方式意外，还可以根据对寻路环境的掌握情况，分为两类

1. 环境信息完全掌握的全局路径规划；
2. 基于传感器的局部路径规划。

本节主要介绍移动机器人全局路径规划技术及局部路径规划技术。

2.1.1 全局路径规划技术

全局路径规划，就是寻路周围的环境已知，机器人根据已知的环境制定一条从初始位置到目标位置的最优路径。其主要方法有：可视图法，自由空间法，最优控制法，栅格法，拓扑法等。

1. 可视图法[7]

可视图法视移动机器人为一点，将机器人、目标点和多边形障碍物的各顶点进行组合连接，并保证这些直线均不与障碍物相交，这就形成了一张图，称为可视图。由于任意两直线的顶点都是可见的，从起点沿着这些直线到达目标点的所有路径均是运动物体的无碰路径。搜索最优路径的问题就转化为从起点到目标点经过这些可视直线的最短距离问题。运用优化算法，可删除一些不必要的连线以简化可视图，缩短搜索时间。该法能够求得最短路径，但假设忽略移动机器人的尺寸大小，使得机器人通过障碍物顶点时离障碍物太近甚至接触，并且搜索时间长。切线图法和Voronoi图法对可视图法进行了改造。切线图用障碍物的切线表示弧，因此是从起始点到目标点的最短路径的图，即移动机器人必须几乎接近障碍物行走。其缺点是如果控制过程中产生位置误差，移动机器人碰撞的可能性会很高。Voronoi图法用尽可能远离障碍物和墙壁的路径表示弧。由此，从起始节点到目标节点的路径将会增长，但采用这种控制方式时，即使产生位置误差，移动机器人也不会碰到障碍物。

1. 拓扑法[10]

将规划空间分割成具有拓扑特征子空间，根据彼此连通性建立拓扑网络，在网络上寻找起始点到目标点的拓扑路径，最终由拓扑路径求出几何路径。拓扑法基本思想是降维法，即将在高维几何空间中求路径的问题转化为低维拓扑空间中判别连通性的问题。优点在于利用拓扑特征大大缩小了搜索空间。算法复杂性仅依赖于障碍物数目，理论上是完备的。而且拓扑法通常不需要机器人的准确位置，对于位置误差也就有了更好的鲁棒性；缺点是建立拓扑网络的过程相当复杂，特别在增加障碍物时如何有效地修正已经存在的拓扑网是有待解决的问题。

1. 栅格法[9]

将移动机器人工作环境分解成一系列具有二值信息的网格单元，多采用四叉树或八叉树表示，并通过优化算法完成路径搜索。该法以栅格为单位记录环境信息，有障碍物的地方累积值比较高，移动机器人就会采用优化算法避开。环境被量化成具有一定分辨率的栅格，栅格大小直接影响环境信息存储量大小和规划时间长短。栅格划分大了，环境信息存储量小，规划时间短，但分辨率下降，在密集环境下发现路径的能力减弱；栅格划分小了，环境分辨率高，在密集环境下发现路径的能力强，但环境信息存储量大，规划时间长。

栅格法是由w．E．Howden在1968年提出的。栅格法将机器人工作环境分解成一系列具有二值信息的网格单元，工作空间中障碍物的位置和大小一致，并且在机器人运动过程中，障碍物的位置和大小不发生变化。用尺寸相同的栅格对机器人的二维工作空间进行划分，栅格的大小以机器人自身的尺寸为准。若某个栅格范围内不含任何障碍物，则称此栅格为自由栅格；反之，称为障碍栅格。自由空间和障碍物均可表示为栅格块的集成。栅格的标识方法有两种：直角坐标法和序号法。多采用四叉树或八叉树表示工作环境，并通过优化算法完成路径搜索。该方法以栅格为单位记录环境信息，栅格粒度越小，障碍物的表示越精确，但同时会占用大量的存储空问，算法的搜索范围将按指数增加。栅格的粒度太大，规划的路径会很不精确。所以栅格粒度的大小的确定，是栅格法的主要问题。

1. 自由空间[8]

自由空间应用于移动机器人路径规划，采用预先定义的如广义锥形和凸多边形等基本形状构造自由空间，并将自由空间表示为连通图，通过搜索连通图来进行路径规划。自由空间的构造方法是：从障碍物的一个顶点开始，依次作其它顶点的链接线，删除不必要的链接线，使得链接线与障碍物边界所围成的每一个自由空间都是面积最大的凸多边形：连接各链接线的中点形成的网络图即为机器人可自由运动的路线。其优点是比较灵活，起始点和目标点的改变不会造成连通图的重构，缺点是复杂程度与障碍物的多少成正比，且有时无法获得最短路径用栅格法建模受到了空间分辨率和内存容量的矛盾限制。而自由空间法建模，解决了这一矛盾。但自由空间法的分割需构造想象边界，想象边界本身具有任意性，于是导致路径的不确定性。

1. 最优控制法[9]

在确定的空间里，二维平面上的一条边界可由方程来描述。那么，机器人在运动过程中，从起点到终点的众多路径里，有障碍物的路径是不允许机器人通过的。这些路径可以作为约束条件，由数学表达式表示。非完整移动机器人通过适当的变换，可将其转化为链式形式。因此，通过选择适当的控制量就可以驱使机器人从一个位置运动到另一个位置。

2.1.2 局部路径规划

(1) 人工势场法[7]

人工势场法是Khatib提出的一种虚拟力法。其基本思想是将移动机器人在环境中的运动视为一种虚拟人工受力场中的运动。障碍物对移动机器人产生斥力，目标点产生引力，引力和斥力周围由一定的算法产生相应的势，机器人在势场中受到抽象力作用，抽象力使得机器人绕过障碍物。该法结构简单，便于低层的实时控制，在实时避障和平滑的轨迹控制方面，得到了广泛应用，其不足在于存在局部最优解，容易产生死锁现象，因而可能使移动机器人在到达目标点之前就停留在局部最优点。

人工势场法的主要缺陷有：

1)陷阱区域；

2)在相近的障碍物之间不能发现路径；

3)在障碍物前振荡；

4)在狭窄通道中摆动。

针对这些缺陷，提出了一些改进办法。针对人工势场法存在“机器人在到达目标位置前由于陷入局部极小点而无法到达目标位置”的问题，解决的方法有：重新定义势函数，使之没有或有更少的局部极小点；利用搜索算法跳出局部极小点。还可以利用模拟退火算法使势函数跳出局部极小点，到达机器人的目标位置。

(2) 模糊逻辑算法[4]

模糊逻辑算法基于对驾驶员的工作过程观察研究得出。驾驶员避碰动作并非对环境信息精确计算完成的，而是根据模糊的环境信息，通过查表得到规划出的信息，完成局部路径规划。优点是克服了势场法易产生的局部极小问题，对处理未知环境下的规划问题显示出很大优越性，对于解决用通常的定量方法来说是很复杂的问题或当外界只能提供定性近似的、不确定信息数据时非常有效。假设检测的是障碍物与机器人的距离和障碍物的运动信息，输出机器人速度变化和转角变化。

模糊控制算法有诸多优点，但也有固有缺陷：人的经验不一定完备；输入量增多时，推理规则或模糊表会急剧膨胀。由于模糊隶属度函数的设计、模糊控制规则的制定主要靠人的经验和试凑，总结模糊控制规则时比较困难，而且，控制规则一旦确定，在线调整困难，无法很好地适应情况的变化。因此，如何得到最优的隶属度函数、控制规则以及对控制规则进行在线调整是该方法最大的问题。

(3) 模拟退火算法[7]

模拟退火算法由Kirkpatrick S于1983年提出，源于物理退火过程。基本思想是利用随机优化问题求解过程与统计力学中热平衡问题的相似性，通过设定初温、初态和降温率控制温度的不断下降，结合概率突跳特性，利用解空间的邻域结构进行随机搜索。模拟退火算法用于路径规划可避免局部极值，但其理论收敛条件过于苛刻，在实际应用中往往无法满足。在有限计算量条件下的收敛性能依赖于自身参数，这使得参数设定成为算法应用过程中的一个关键环节。

(4) 神经网络法[5]

路径规划是感知空间到行为空间的一种映射。映射关系可用不同方法实现，很难用精确数学方程表示，但采用神经网络易于表示。将传感器数据作为网络输入，由人给定相应场合下期望运动方向角增量作为网络输出，由多个选定位姿下的一组数据构成原始样本集，经过剔除重复或冲突样本等加工处理，得到最终样本集。将神经网络和模糊数学结合也可实现移动机器人局部路径规划。先对机器人传感器信息进行模糊处理，总结人的经验形成模糊规则。再把模糊规则作用于样本，对神经网络进行训练。通过学习典型样本，把规则融会贯通，整体体现出一定智能。实际中允许输入值偏离学习样本，只要输入接近一个学习样本的输入模式，则输出也就接近该样本输出模式。该性质使得神经网络可以模仿人脑在丢失部分信息时仍具有对事物正确的辨识力。

(5) 遗传算法[6]

遗传算法以自然遗传机制和自然选择等生物进化理论为基础，构造了一类随机化搜索算法。利用选择、交叉和变异编制控制机构的计算程序，在某种程度上对生物进化过程作数学方式的模拟。只要求适应度函数为正，不要求可导或连续，同时作为并行算法，其隐并行性适用于全局搜索。多数优化算法都是单点搜索，易于陷入局部最优，而遗传算法却是一种多点搜索算法，故更有可能搜索到全局最优解。遗传算法的整体搜索策略和优化计算不依赖于梯度信息，解决了一些其它优化算法无法解决的问题。遗传算法运算速度不快，进化众多的规划要占据较大存储空间和运算时间，优点是克服了势场法的局部极小值问题，计算量不大，易做到边规划边跟踪，适用于时变未知环境的路径规划，实时性较好。遗传算法运用于移动机器人路径规划的研究近来取得了许多成果，其基本思想：将路径个体表达为路径中一系列中途点，并转换为二进制串。首先初始化路径群体，然后进行遗传操作，如选择、交叉、复制、变异。经过若干代进化以后，停止进化，输出当前最优个体。

(6) 蚁群算法[7]

蚁群优化算法是由意大利学者Dorigo等人在2O世纪9O年代从蚁群觅食行为受到启发，通过模拟自然界蚂蚁寻径的行为，提出的一种全新的模拟进化算法。蚁群优化算法在并行运行环境(如网格环境)下可以同步寻优，加快了寻优速度。另外，它是一种通用性强的算法，稍加修改便可用于其他优化问题。但计算量较大，搜索时间较长，易于陷入局部最优解。

(7) 粒子群算法[7]

粒子群算法是由Kennedy博士和Eberhart博士于1995年从鸟类的捕食行为中受到启发提出的一种基于群体的智能随机优化算法。粒子群算法具有收敛速度快、算法简单、容易编程实现和鲁棒性强等特点，但是，粒子群算法也有一些缺陷，一是容易陷入局部极值点，导致得不到全局最优解；二是粒子群算法本身的参数设置，若参数选择不当，会导致寻优过程中粒子的多样性迅速消失，造成算法“早熟收敛”。

(8) 启发式搜索方法[7]

启发式方法的最初代表是A\*算法 ，其新发展是D\*和Focussed D\*。这2种由Stentz A提出的增量式图搜索算法的产生。D\*算法可以理解为动态的最短路径算法，而Focussed D\*算法则利用A\*算法的主要优点即使用启发式估价函数，2种方法都能根据机器人在移动中探测到的环境信息快速修正和规划出最优路径，减少了局部规划的时间，对于在线的实时路径规划有很好的效果。此外，还出现了一些基于A\*的改进算法，它们一般都是通过修改A\*算法中的估价函数和图搜索方向来实现的，可以较大地提高路径规划的速度，具有一定的复杂环境自适应能力。

2.2 强化学习算法

在机器学习领域内，随着反馈的不同，可以分为：监督学习（supervised learning），非监督学习（Unsupervised learning）和强化学习（Reinforcement）三类。本文要讲的强化学习，是以环境反馈作为输入信息，不断适应环境的方法。

2.2.1 强化学习基本原理

对于强化学习，我们可以把他看是一个不断“试探”环境的过程，并且实时“评价”自己的试探:首先，智能体（也就是学习系统）感受周围的环境“状态”，然后对环境施加一个“动作”，在动作之后，环境会产生改变，同时生成一个奖励值(可能是正，也可能是负)反馈给智能体（即学习系统），最后智能体凭借强化信号以及当前的环境状态作出评估，再进行下一个动作，选择动作的基本原则就是使得即将收到的奖励概率变大。

强化学习系统应该具有的特点：

1. 适应性，智能体不仅仅适用于某种特定环境，而是可以根据环境的反馈而改善性能；
2. 反应性，智能体可以根据以往的经验来判断对当前状态应采取什么行为；
3. 对教师信号依赖少，智能体从内置的强化规则生成强化信号，再根据强化信号进行学习。

强化学习系统的基本结构，由“环境”和“智能体(Agent)”组成。智能体通过接收外部信号（比如通过传感器）来感知所处环境，然后对环境采取相应行为。这里的环境，指的是，所有与该智能体产生交互行为的物体。

在与环境交互过程中，智能体通过反复“试探”环境来学习合适的行为，这就是强化学习。它不同于监督式学习和非监督式学习，而是介于他们之间，是和策略紧密联系的学习，通过不断“试探”，从而学习到最优的策略。

强化信号（可能为奖励或者惩罚）是环境在被智能体的某个行为影响了之后反馈给智能体的。当奖赏值为正，那么智能体在之后会更加倾向于采取此行为；反之，如果奖赏值为负，那么智能体在此之后会不倾向于采取此行为



智能体处在环境中，并与之交互，每一个交互周期都会发生如下事件：

图 2 - 1 强化学习模型

1. 智能体在*t*时刻接受环境信息，得到环境状态*s(t)*；
2. 根据当前状态和回报*r(t)*选择动作*a(t)*，并执行；
3. 智能体做出的动作，对环境产生影响，环境状态发生变化；
   1. 环境状态更新到新的状态*s(t+1)*；
   2. 给出回报*r(t)*,有可能为正，也有可能为负。
4. 回报*r(t)*传递给智能体，*t←t+1*；
5. 如果新的状态为结束状态，则停止循环，否则转第二步。

其中，环境状态*s(t)*和智能体的行为*a(t)*共同决定了回报*r(t)*。

用一个四元数组来描述强化学习系统

|  |  |  |
| --- | --- | --- |
|  |  | (2 - 1) |

其中，*S=(,…,)*为智能体遇到的环境状态的集合；*A=(,…,)*为智能体的动作的合集；*W*表示环境状态转移:*W:S×A→S*; *R=(,…,)*表示环境因不同行为对智能体产生的强化信号（也就是奖赏值），在每一个交互周期内，环境和智能体交互都会获得环境状态，智能体根据获得的环境状态来选择一个行为，在*t+1*时刻，环境状态更新为，智能体会根据更新的环境状态获得一个奖赏值，强化学习最终会找到一个最优策略，使得智能体得到的奖励值总和最大。

上述内容可以表示为：四个元素来表示一个强化学习系统：环境模型，奖赏函数，状态值函数以及控制策略

图 2 - 2 强化学习系统

1. 控制策略（Policy，又称决策函数）：是强化学习的核心,指的是映射（状态到动作），规定在状态*s*下智能体应该选择动作*a*。策略的好坏决定了智能体的行为以及系统的整体性能。
2. 奖赏函数（Reward Function，即强化信号）:是智能体将行为作用在环境上，环境对智能体的该行为产生的反馈信号，也就是奖赏信号*R*，奖赏信号可正可负，如果为正，那么表示动作为好；如果为负，那么表示动作并不好，并且负值越大，说明此行为越差
3. 状态值函数（Value Function，又称评价函数）:是评价“状态-动作”对的指标
4. 环境模型（Environment Model）：外界环境经过一定处理后，一种能被智能体识别的形式，就建立了环境模型。环境模型会因为智能体选择的行为而改变，环境模型也会给出下一状态和奖赏信号的预测值。这样，智能体就可以根据环境模型对行为的选择进行预测，从而选出比较好的行为。

在早期，强化学习可以称为“试错（Trial-and-Error）”的方式，自从将环境模型和规划引入之后，强化学习就更倾向于动态规划。试错和规划都是强化学习中智能体获得经验的一个过程。

2.2.2 Markov决策过程

在路径规划中，强化学习算法的智能体的每一步都是离散的，因此，我们可以对强化学习中的智能体进行数学建模。马尔科夫决策过程[12]（Markov Decision Process，MDPs）是很多强化学习算法的基础，因为马尔科夫决策过程就是有限状态的，离散的。常见的马尔科夫模型有：

|  |  |  |  |
| --- | --- | --- | --- |
| 马氏模型 | | 是否智能系统行为控制环境状态转移 | |
| 否 | 是 |
| 是否环境为部分可感知 | 否 | 马尔科夫链 | 马氏决策过程 |
| 是 | 隐马尔科夫模型 | 部分感知马氏决策过程 |

在本文讨论范围内，环境模型对于智能体来说是未知的，并且是可以被感知的，所以我们可以认为环境模型是马尔科夫型的。所以我们可以用马氏决策过程（Markov Decision Process，MDP）来给强化学习问题建模。

我们用四元组*<S,A,R,W>*来定义马氏决策过程。*S*是环境状态集合，*A*是系统行为集合，状态转移函数*W: S×A→S*和奖赏函数*R:S×A→R*。我们把系统在状态通过使用行为来转移到状态获得的瞬时奖赏值记为，系统在状态采用使环境状态转移到的概率定义为。

马氏决策过程的实质就是：智能体在当前状态下，转移到下一个状态的概率和因此产生的奖励值，仅仅由目前状态和选择的行为决定，和以前的状态和行为无关。所以在指导*W*和*R*的基础上，可以采用动态规划技术来得到最好的策略。而强化学习则是侧重在W和R未知的情况下，智能体如何自己学习并得到最好的策略。

怎么定义环境模型是学习系统面临的一个重要问题，因为事先并不知道*W*和*R*，智能体只能够依赖在每一个“试探”周期获得的奖励值来选择合适的策略。在选择行为的时候，为了解决环境模型的波动性和目标的长远性，要在奖励值和行为策略之间构造函数，也就是评价值函数，用评价值函数来进行策略的选择

|  |  |  |
| --- | --- | --- |
|  |  | (2 - 2) |
|  |  | (2 - 3) |

利用一式我们可以得到一个返回函数，这个返回函数是用来得出在一次学习周期中，在策略下，从状态之后获得的所有奖赏值得累计折扣和。因为环境是未知的，所以每一次学习循环中，在策略下，系统的奖赏值累计和有可能是不一样的。所以利用第二个式子，来考虑不同学习周期中所有返回函数的期望:在某策略下，智能体所获得的期望的累计奖赏折扣。

根据Bellman公式，在我们选定的最优策略下，并且在状态*s*下，值函数的定义为：

|  |  |  |
| --- | --- | --- |
|  |  | (2 - 4) |

如果*P*和*R*已知的情况下，从随意设置的策略出发，为了逼近最优的和，我们可以采用策略迭代的方案：

|  |  |  |
| --- | --- | --- |
|  |  | (2 - 5) |
|  |  | (2 - 6) |

因为在强化学习中，转移函数*P*和奖赏函数*R*是不知道的，所以学习系统不能通过上两个式子进行计算值函数。实际中，如果要进行值函数的估计，常用的方法是采用逼近的方法，比如Monte Carlo采样。

2.2.3 动态规划方法

动态规划方法，又称之为分阶段决策。也就是说能够在做出下一个决策之前，预测这个决策产生的结果(并不能完全预测，有误差)。动态规划方法就是强化学习中最重要的技术。动态规划有两个常用的主要方法，值迭代和策略迭代。虽然是两种方法，但是都是利用动态规划求解，不过一个求最优策略，另一个求最优值。目前比较常用的方法就是值迭代。

2.2.3.1 值迭代

研究值迭代的目的并不是为了找到最优策略，而是找出在当前学习周期内，最好的一个值函数所采取的行为。定义回报折扣和为

|  |  |  |
| --- | --- | --- |
|  |  | (2 - 7) |

这个回报折扣和表示：在当前状态下，采取行为和后续策略得到的值。时，可以得到与之相应得最优评价函数，则Bellman方程可以写作：

|  |  |  |
| --- | --- | --- |
|  |  | (2 - 8) |

值迭代算法具体流程如下：

1. 对于所有的，选择随意的初始值；
2. 对于所有的，计算公式为

|  |  |  |
| --- | --- | --- |
|  |  | (2 - 9) |
|  |  | (2 - 10) |

1. 一直进行前一步操作，直到

我们这里设为一个特别小的值，这样就可以使得充分接近也就是最优解。

2.2.3.2 策略迭代

策略迭代的目的就是找到最优策略，也就是通过不断迭代从而不断提高策略。

从任意策略开始，对所有状态，使用下列式子来计算当前状态的策略

|  |  |  |
| --- | --- | --- |
|  |  | (2 - 11) |

一直重复2、3步，直到策略值不再发生改变。

2.3 本章小结

本章主要是介绍了机器人路径规划的方法以及强化学习在机器人路径规划中的应用、研究现状以及发展，解释了马尔科夫决策过程，分析了强化学习系统模型以及其中的基本元素，为下一章学习Q-learning算法提供了基础。

# 第3章 经典强化学习算法及改进算法的提出

本章节主要研究经典Q-learning强化学习算法分析，以及在此基础上提出改进算法。

3.1 Q-learning算法

Q-learning[11]算法是强化学习中比较常见的一种无模型的迭代算法，1989年，第一次被Watkins提出。该算法第一步就是设置一个Q值表，在智能体与环境交互过程中，通过环境信息的反馈，对智能体的“状态-动作”对形成或正或负的奖励，一个接近最佳路径的行为，对应的Q值会不断增加，偏离最佳路径的行为，对应的Q值会不断减小，通过对Q值的选择，使得智能体的选择不断接近最优行为。在马尔科夫尼科夫[12]规则下，该算法的Q值是可以收敛的。

3.1.1 Q-learning算法的基本原理

Q-learning算法常常采用数值迭代计算来逼近最优值，若移动机器人所处的复杂环境是一个有限状态的Markov过程，当机器人在动作集合中选择某动作后，环境接收该动作并发出状态转移，同时给出奖赏*r*。如果其动作导致环境正奖赏（*+r*），则移动机器人此后产该动作的趋势就会增强；反之，如果导致环境负奖赏（*-r*），则会减弱。设在*t*时刻，机器人选取动作，环境从状态转移到时，所给出的奖赏为，显然

|  |  |  |
| --- | --- | --- |
|  |  | (3 - 1) |

与的概率分布由和所决定，且为*t*时刻幵始时状态-动作对的佔计值，在Q学习过程中，以通过优化迭代计算的*Q(s,a)*函数值来逼近最优值函数，基本更新规则如下：

|  |  |  |
| --- | --- | --- |
|  |  | (3 - 2) |
|  |  | (3 - 3) |

所以Q-learning算法的学习过程可以总结如下，智能体在*t*时刻的学习过程可以描述为：

1. 观察当前状态；
2. 选取并执行动作；
3. 观察下一个状态；
4. 生成一个强化信号；
5. 根据强化信号更新Q值函数。

|  |  |  |
| --- | --- | --- |
|  |  | (3 - 4) |

其中：

为折扣因子；

*A*为动作集合，元素个数*n*；

*S*状态集合，也是环境单元集合，元素个数为*m*；

*Q，q*值表集合，元素个数是m\*n，表示*t*时刻，由和确定的q值。表示*t+1*时刻，由和确定的*q*值，也就是*q*的更新值。表示执行动作后，下一个状态的所对应的*Q*值，表示这些Q值表里面的最大值。

为学习率，他控制着学习的速度，越大，则收敛速度越快，但是过大的可能会引起不成熟的收敛，所以选择合适的学习率，对于学习速度起着重要的作用。下面将会证明:当时，经过无数次迭代，Q-learning算法一定收敛于最优解。所以，Q-learning算法不是去估计环境模型，而是直接优化一个可以迭代计算的Q值函数。

3.1.2 Q-learning算法的流程

根据Q-learning算法的原理，总结其算法流程如下表

|  |
| --- |
| Initialize *Q(s,a)* arbitrarily  Repeat (for each episode)  Initialize *s*  Repeat (for each step of episode)  Choose *a* from *s* using policy derived from *Q*  Take action *a,* observer *r , s*      Until *s* is terminal |

表 3 - 1 Q-learning算法

在上表中，机器人的每一次尝试都是完成由起始状态到达目标状态的整个任务。

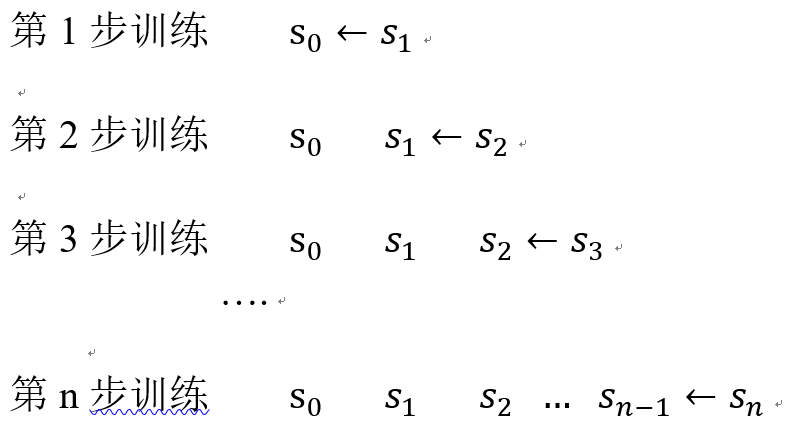
 机器人的每一次尝试中的迭代过程可以用下图说明， 图 3 - 1中表示机器人的起始状态，表示机器人经历的第一个状态,表示机器人经历的第二个状态，表示机器人经历的第*n*个状态，箭头表示状态数据传递的方向，所传递的数据包括奖励值*r*和某状态占用的所有Q值。

图 3 - 1 Q-learning 数据更新图

由于该算法每一步训练只传递一次数据，所以常常被称为一步迭代Q-learning算法。该算法的特点是：所用的内存空间大，并且学习实践长，收敛速度慢。

3.1.3 Q-learning算法的收敛性分析

Q学习算法通常在一张搜索表中存储与每个状态-动作对相关的Q值，由上一节公式我们知道状态对应的Q值要想达到最终值必要条件是状态对应的恒定，为一个常熟；否则状态对应的Q值将随着对应Q值得变化而变化。同理在状态的Q值变化过程中，其前面所有指向该状态的动作对应的状态都不能达到稳定值。

假设状态稳定，为一个常数，即恒定，为求的稳定状态，我们通过公式(3.3)不停迭代并总结得知：

一次迭代

二次迭代

三次迭代

……

n次迭代

其中，易知当有足够迭代次数n时候，可以得到：

此时收敛。

通过以上证明可以得出结论：状态不稳定，其初始值为，指向的下一个状态，则若收敛，且经过足够多的迭代次数，将收敛。

Q-learning算法的收敛和动作的选取都是通过Q值得变化和更新来决定的，其值的变化和更新说明如下。

Q值得最终态：

传统的Q-learning通过(3.3)式不断叠加，某一状态的数据最终都会到达一个极值。根据公式(3.3)获得：

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | (3 - 5) |
|  | |  | (3 - 6) |

Q值通过不断的迭代，当时，将以概率1收敛于最优值。此时，

|  |  |  |
| --- | --- | --- |
|  |  | (3 - 7) |

在此基础上继续叠加将不该变Q值。

数据之间的反馈作用：

传统的Q-learning算法，不同状态Q值之间的数据都具有单级反馈。根据公式(3.4)，我们知道，要更新的Q值是根据原来的Q值和下一个状态的Q值共同决定的，也就是说前一个状态的Q由后一个状态的Q反馈形成新的Q值。

数据传递滞后性：

传统Q-learning算法的Q值数据传递具有一定的滞后性。在传统Q-learning训练循环中，假设有任意相连的三种状态，状态想要获得状态的信息反馈我们需要2此重复训练循环，即：

第一次循环：

第二次循环：

传统算法，每进行一次更新状态的Q都需要从头训练一次，更新两次状态对应的Q值，需要训练两次，此时的才具有的信息。通过堆到指导传统算法的数据传递具有一定的滞后性，第一个Q值要获得相隔n个状态的Q值的反馈需要迭代n次，也就是重复训练n此。

当后续状态的动作决策出现了错误，对前面状态的动作决策，需要经过许多次重复迭代才能有影响，而这部分需要的总迭代数目是庞大的，这也就是限制强化学习算法收敛速度的一个重要原因。

收敛性：

不同状态收敛优先级不同，由于后一个状态的收敛是前一个状态收敛的必要条件，我们知道不同状态的收敛优先级是不同的，越靠近理想状态Q值收敛越早，学习的过程也就是Q值从目标状态依次向起始点收敛的过程，我们通过不断迭代，被训练的状态点Q值有朝最终值靠近的趋势，经过足够多的训练次数最终将达到理想值。

3.1.4 收敛步数

通过上面对Q-learning算法收敛性的分析，我们知道状态收敛的必要条件是后续状态的收敛，首先是目标状态收敛，然后是接近于目标位置的状态收敛，进而状态链中的其他状态在不同时间下依次收敛。也就是说距离目标越远的状态对应的Q值收敛越迟，已经收敛的状态有助于后续学习做出正确的动作选择，不能收敛的Q值对于机器人来说没有意义。

如果状态的后续状态对应的Q值已经收敛，那么对应的Q值收敛的必要条件是经过足够训练更新，现假设状态收敛时需要更多的更新次数最少为M，则对于，使得：

|  |  |  |
| --- | --- | --- |
|  |  | (3 - 8) |

则满足：

|  |  |  |
| --- | --- | --- |
|  |  | (3 - 9) |

定义：

|  |  |  |
| --- | --- | --- |
|  |  | (3 - 10) |

这里的M就是状态收敛时所需要的最少更新次数。

3.2 算法简介

近年来，由于单步的Q-learning算法结构简单且易于扩展，已成为强化学习中的一个研究热点，国内外学者结合Q-learning算法，对强化学习算法展开了深入的研究。通过对Q-learning算法数据流动特性的分析，可以看出要提高Q-learning算法的收敛速度，其一可以通过打破数据传递的滞后特性，让后续状态的动作决策可以快速的影响到前面状态的动作选择，其二可以通过加快目标状态附近状态对应的状态动作对的收敛，为前面状态动作对的收敛提供必要条件，其三可以通过估计，利用相似的先验知识，估计状态-动作对。

Peng等通过对Q学习算法进行改进,提出了多步的算法，算法是借鉴算法而产生的，通过回溯的思想[13]，使得后续产生的数据能够及时反馈回来。其实际意义为：我们通过数据的不断传递，使得某一状态的动作决策受到其后续状态的影响。如果未来某一决策*a*，是一个失败的决策，那么当前的决策也要承担相应的责任，也会把这种影响追加到当前决策上来；如果未来某一决策*a*，是一个正确的决策，那么当前的决策也获得相应的奖励，同样也要把这种影响追加到当前决策上来。

3.2.1 跟踪迹的引入

多步的算法能够提高算法的收敛速度，满足在线学习的实用性，关键在于引入了跟踪迹，跟踪迹最初是由Klopf从认知科学的知度出发提出的一种记忆机制[14]，Sutton和Singh把跟踪迹用于强化学习，使其成为一种重要的强化学习基本机制。实际上跟踪迹的思想比较简单，每当一个状态被访问时，该状态就会被短暂的记忆下来，随着时问的推移，状态的跟踪将会衰减，这个迹标志这个状态对学习足有资格的，该过程称为渐进遗忘。跟踪迹不为零时，若一个状态发生了，那么该状态就会随机赋予一定的信度，所以跟踪迹可以用于延时强化学习的信度分配问题。

通常，渐进遗忘的过程里现指数衰减状态。一般令遗忘衰减系数和折扣系数的乘积指数衰减，由此来展现跟踪状态的渐进遗忘过程。利用有效的跟踪以解决强化学习中的效用分配问题。

跟踪迹一般可以分为累积迹和替换迹两种，传统计算跟踪迹的方法累枳法，当一个状态被访时，其迹就已经建立起来了，累积迹的计算方法可以通过公式(3.11)得到：

|  |  |  |
| --- | --- | --- |
|  |  | (3 - 11) |

其中表示状态*s*在*t*时刻的迹，相应的替换迹可表示为

|  |  |  |
| --- | --- | --- |
|  |  | (3 - 12) |

在本文中，我们将跟踪迹的数学模型定义为

|  |  |  |
| --- | --- | --- |
|  |  | (3 - 13) |

具体控制过程可描述为：当一个状态位被重复的访问且选择了一个新的动作，那么这个动作的跟踪将被重新设置为1。随着时间推移，这个跟踪的信号将会根据(3.13)呈现指数衰减，最后趋近于零。

我们知道当跟踪信号经过一定时间，将会趋近于零，失去效用，现研究其失效时间。设某一状态动作对在被访问一次后，然后连续n次没被访问时，则该状态

动作对被遗忘时

|  |  |  |
| --- | --- | --- |
|  |  | (3 - 14) |

为一个极小值。由于，则当

|  |  |  |
| --- | --- | --- |
|  |  | (3 - 15) |

时，相应的状态-动作对的跟踪将会被遗忘。这种跟踪迹在某种情况下比原始的替代迹要好，因为一旦选择了正确的动作，那么剩下错误的动作就没有迹了。

3.2.2 Q()算法流程

算法是在Q-learning算法原理的基础上进行改进的，与Q-learning算法相比，算法加入了跟踪迹，使其具有一定的记忆性，这样机器人不但能够预测下一步旳行为，而且能够控制其行为，使错误的行为对应的值在更新的过程中逐渐被遗忘，具体算法流程如表3-2所示。

|  |
| --- |
| Initialize Q(s) arbitrarily and e(s,a)=0 for all  Repeat(for each episode)  Initialize *s*  Repeat (for each step of episode)  Choose *a* from *s* using policy derived from *Q*  Take action *a* , observer *r ,s*`        for all s      Until *s* is terminal |

表 3 - 2 Q()算法

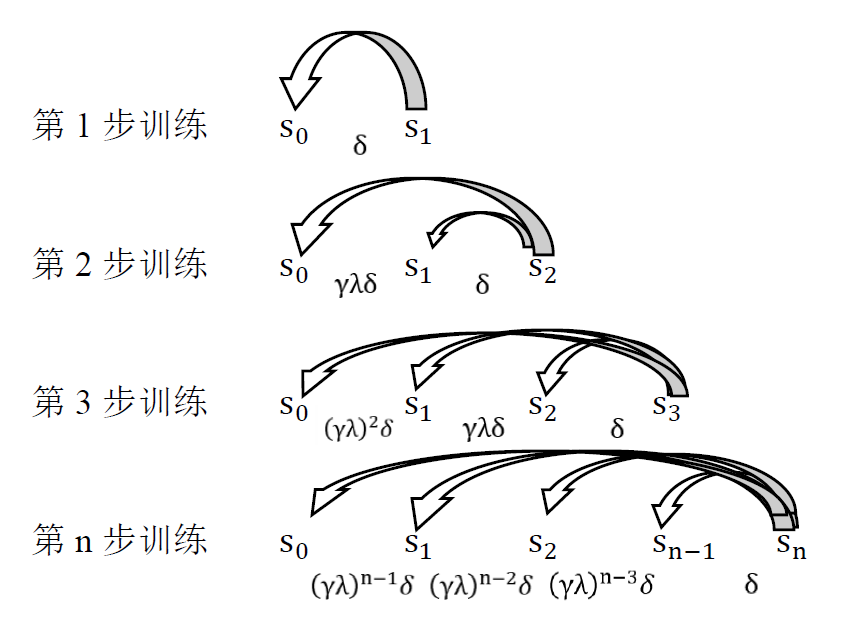
在表3-2中，机器人每一次从起始位置到目标位置的尝试迭代过程可用下图进行说明

图 3 - 2 算法的数据更新

从图 3 - 2可以看出机器人每一步的训练，，都会并行更新所有经历过的状态-动作对，因此常常被称为多步的Q-learning算法。

迭代过程可用图 3 - 2说明，图中表示起始状态，表示第一个经历的状态，表示第二个经历的状态，表示第n个经历的状态，箭头表示数据传递方向，也是跟踪过程中数据的回馈，所传递的数据是由末尾两个状态，即状态*s*和下一个状态*s`*共同决定的，根据渐进跟踪遗忘的原理，由公式(3 - 16)求得，传递而来的数据将会有一定的衰减，第i个状态-动作对，经历了n-i的时间单位后，传递回来的数据即为。

当机器人处于t时刻的状态时，在t+n时刻，其被记忆的状态=动作对相应的Q值经过每一个时刻的累计更新为公式(3 - 19)

|  |  |  |
| --- | --- | --- |
|  |  | (3 - 16) |
|  |  | (3 - 17) |
|  |  | (3 - 18) |
|  |  | (3 - 19) |

3.3 改进Q-learning算法

3.3.1 算法原理

改进之后的Q-learning算法和算法原理类似，都是基于回溯的思想。该算法的设计初衷是为了提高强化学习算法的收敛速度，根据不同状态具有不同收敛优先级的特点，让距离目标状态近的状态区域提前收敛，为后续状态的收敛提供必要条件，并且通过回溯的思想改善Q-learning学习中数据传递的滞后性，使当前状态的动作决策能够快速的受到后续动作决策的影响。

改进的Q-learning算法把一次路径规划尝试（机器人从初始状态到目标状态的一次循环叫做一次尝试）看作是一系列数据的传递，我们用一条链来形象比喻这一系列的数据传递。该循环中每一步搜索的状态可以是相同状态，也可以是不同状态，每一步状态中依次排列，前一步状态指向下步状态，形成一个从起始状态始逐级相连，最后指向当前状态的单条状态链。

本文通过公式递归回溯，使得后续产生的数据能够及时反馈回来。其实际意义为：通过数据的不断传递，使得某一状态的动作决策受到其后续状态的影响。如果未来某一决策*a*，是一个失败的决策，那么当前的决策也要承担相应的责任，要把这种影响追加到当前决策上来；如果未来某一决策*a*，是一个正确的决策，那么当前的决策也获得相应的奖励，也要把这种影响追加到当前决策上来。

改进的Q-learning算法就是利用单链和回溯的思想，把一次路径规划尝试（机器人从初始状态到达目标状态的一次循环叫做一次尝试）看作是一系列数据的传递，我们用一条链来形象比喻这一系列的数据传递，该循环中经历过的每个状态中都封装在一起，形成一个状态链，每一个状态形成链中的环。训练循环中的每一步，由当前状态跳到下一个状态，通过公式(3-3)更新当前状态对应的值的同时，通过重复循环，使链中环与环之问的数据直接地简单迭代计算，一环接一环，由现在向过去，迭代更新前面经历过状态的Q值。

这种数据传递方式一方面使得靠近目标状态的区域更快收敛，为其他区域状态的收敛提供必要条件；另一方面使后续产生的评估他能够及时反馈上来，从而使某一状态的动作决策受其后续状态的影响，及早地识别错误动作决策，避免无意义的搜索，通过回溯的思想改善Q-learning学习中数据传递的滞后性，使收敛速度加快。

3.3.2 算法流程

该算法的核心是引入了记忆功能。通过记录相关参数，利用Q-learning算法迭代公式，重复地迭代被记录的状态-动作对，从而一步到位更新所有经历过的状态-动作对对应的Q值，使得某一状态的动作决策能收到后续动作决策的影响。

首先定义记忆矩阵：

|  |  |  |
| --- | --- | --- |
|  |  | (3 - 20) |

在Q-learning算法学习过程中，使用记忆矩阵来记录每一次使用过的参数。其中，用来表示Q值更新链中状态需要更新的Q值在Q值表中的索引，表征奖励值,表征学习效率。

在新算法中，利用记忆矩阵获取状态链，根据回溯的思想通过Q-learning算法的迭代公式(3-3)进行一次迭代变化。

|  |  |  |
| --- | --- | --- |
|  | *for k=t-1,t-2,……,2,1*  *Until k=1 end* | (3 - 21) |

所以新算法可总结如下表

|  |
| --- |
| Initialize *Q(s,a)* arbitrarily and *M*=[] for all  Repeat (for each episode)  Initialize  Repeat (for each step of episode)  Select and perform and action    Observe the subsequent state  Receive an immediate reward  Add a new row for memory matrix  For k=t-1 to 1 do  Update Q-value along the state chain recorded in  Until *k* is 1  Until *s* is terminal |

表 3 - 3 新Q-learning算法

3.3.3 数据更新图

在不断迭代过程中，t+1时刻更新最新状态对应的Q值的同时，单链中状态前面的状态，也进一步更新，这个过程称为单步重复迭代。

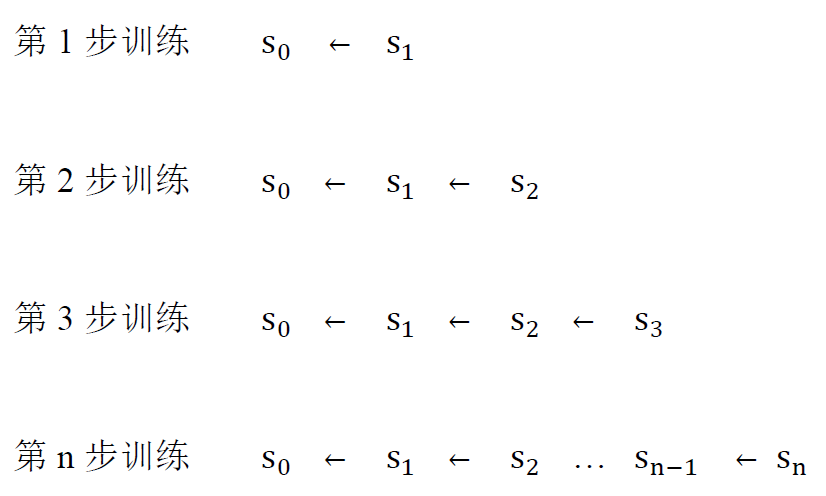
新算法的迭代过程可用3-3说明，图中表示起始状态，表示第一个经历的状态，表示第二个经历的状态，表示第n个经历的状态，箭头表示数据传递方向，也是跟踪过程中数据的回馈，所传递的数据包括奖励值和某状态所占有的Q值。

图 3 - 3 新算法中的数据更新图

根据新算法的算法流程和数据更新图，对于t时刻的状态，在t+n时刻时，其被记忆的状态-动作对对应的Q值同故宫单链回溯更新如下

与单步的Q-learning算法相比，Q值更新速度变快，也就是说学习收敛需要的步数相应的减少了，收敛速度变快。

3.4 本章小结

本章详细地解释并分析了Q-learning算法，算法的基本原理和算法流程，在此基础上提出了改进Q-learning算法。

第4章 Q-learning算法及其改进算法仿真模拟

4.1 仿真环境介绍

MATLAB（矩阵实验室）是MATrix LABoratory的缩写，是一款由美国The MathWorks公司出品的商业数学软件。MATLAB是一种用于算法开发、数据可视化、数据分析以及数值计算的高级技术计算语言和交互式环境。除了矩阵运算、绘制函数/数据图像等常用功能外，MATLAB还可以用来创建用户界面及与调用其它语言（包括C,C++,Java,Python和FORTRAN）编写的程序。

MATLAB 产品家族是美国 MathWorks 公司开发的用于概念设计,算法开发,建模仿真,实时实现的理想的集成环境。自1980 年问世以来，由于其完整的专业体系和先进的设计开发思路，使得 MATLAB 在多种领域都有广阔的应用空间，特别是在 MATLAB 的主要应用方向— 科学计算、建模仿真以及信息工程系统的设计开发上已经成为行业内的首选设计工具，已广泛应用于生物医学工程、图像信号处理、语言信号处理、信号分析、电信、时间序列分析、控制论和系统论等各个领域。

由于使用 MATLAB 编程运算与人进行科学计算的思路和表达方式完全一致，所以学习MATLAB 不象学习其它高级语言--如Basic、Fortran 和C 等那样难于掌握。因而，MATLAB 具有用法简易、运用灵活、程序结构性强且兼具延展性等特点。

MATLAB 的含义是矩阵实验室（MATRIX LABORATORY），其名字来自MATrix 和LABoratory 两个词的前三个字母的组合。主要用于矩阵的方便存取，其基本元素是无须定义维数的矩阵。MATLAB 自问世以来,就是以数值计算称雄。MATLAB 具有很强的数值运算功能，在MATLAB 环境中，有超过500 种数学、统计、科学及工程方面的函数可使用。MATLAB 进行数值计算的基本单位是复数数组（或称阵列），这使得MATLAB 高度“向量化”。经过十几年的完善和扩充，MATLAB 现已发展成为线性代数课程的标准工具。由于它不需定义数组的维数，并给出了矩阵函数、特殊矩阵专门的库函数，使之在求解诸如信号处理、建模、系统识别、控制、优化等领域的问题时，显得大为简捷、高效、方便，这是其它高级语言所无法比拟的。

Matlab主要特点：

可用于技术计算的高级语言

可对代码、文件和数据进行管理的开发环境

可以按迭代的方式探查、设计及求解问题的交互式工具

可用于线性代数、统计、傅立叶分析、筛选、优化以及数值积分等的数学函数

可用于可视化数据的二维和三维图形函数

可用于构建自定义的图形用户界面的各种工具

可将基于MATLAB的算法与外部应用程序和语言（如C、C++、Fortran、Java、COM以及Microsoft Excel）集成的各种函数

4.2 实验设计

本文使用Matlab，利用改进算法对迷宫寻路问题进行实验模拟。为了更好的对改进算法进行评估，智能体将会分别利用传统Q-learning算法和改进之后的Q-learning算法在选取多种大小不同、复杂程度不同的环境下进行学习和路径规划，通过比较两种算法在多种环境下的收敛性指标进行评价。

4.2.1 智能体路径规划系统框架

4.2.1.1 环境模型

本文采用栅格法在坐标中建立栅格的环境模型，在坐标轴上等距离的整数格子上画网格，每一个网络交汇处作为智能体的一个位置。每一个位置都是ws矩阵的一个元素。在本环境中，障碍物和墙体用黑色方格表示，可以通过黑色方格累积成不同障碍物；白色块为自由区域，也就是该位置没有障碍物；绿色块为智能体出发地点；红色块为智能体的目标点，示例如图4-1。在环境矩阵*WS*中，位置的状态可以表示为，也就是说，如果=0表示该位置为自由区域，如果=1则表示该位置为障碍区域。

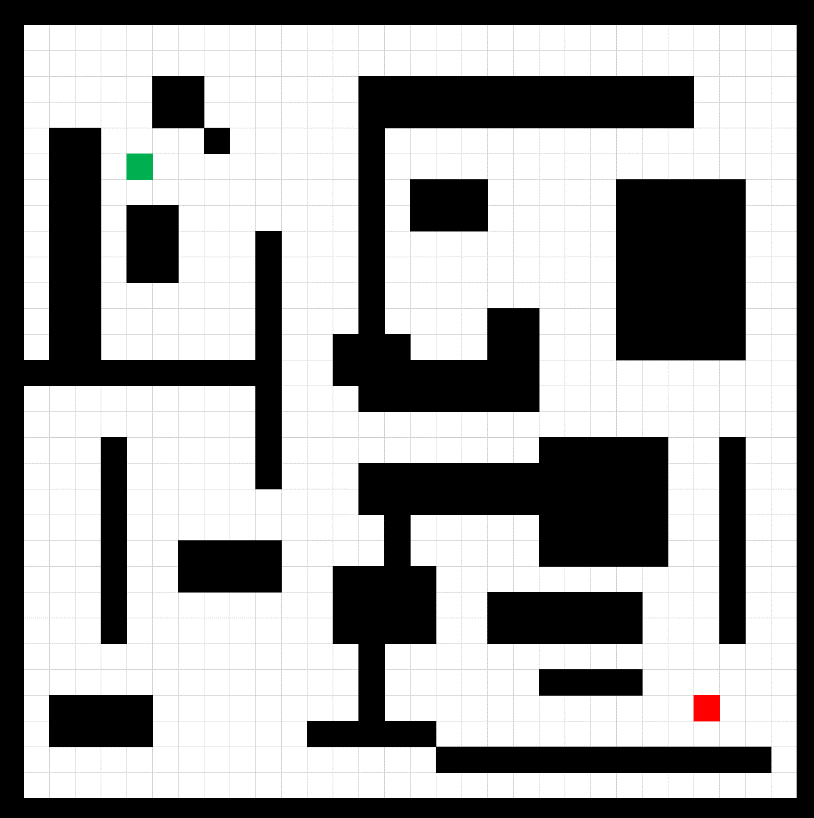
4.2.1.2 机器人模型及其动作空间表示

图 4 - 1 迷宫环境模型示例图

本文将智能体简化成一个只占一个方格位置的点，因此在路径规划中不必考虑半径问题。

本文将赋予智能体4个动作，即上下左右，单步长，在环境中，机器人将沿着方格行走。我们用Act表示离散动作矩阵，定义*Act*=[1,0;0,1;-1,0;0,-1]用于表示五个动作

4.2.1.3 奖赏函数定义

奖赏函数，也就是强化信号，如何设置将直接影响学习速率。本文采用非线性的函数表示立即奖励函数，设计如下

|  |  |  |
| --- | --- | --- |
|  |  | (4 - 1) |

由上式可以看出，当机器人没有碰到障碍物，则获得-2的立即将上；当机器人碰到障碍物时，则获得-50奖赏；当机器人到达目标点时，获得100奖赏。

4.2.1.4 Q值表建立

强化学习算法的关键之处在于Q值表的建立，我们通过环境坐标和动作来建立Q值表。在栅格环境模型中，机器人在每一个栅格处可到达的位置对应4个动作，我们给每一个动作分配一个Q值，比如使用12\*12的坐标系，内部有1000个位置可以到达，则拥有400个Q值，我们初始化Q值表为1\*400的零矩阵。

我们通过建立Q值表，并让机器人遍乂遍的从起点出发到达目标点，不停的完成训练循环，并不停的修改Q值表，当智能体找到了一条最优路径后结束机器人探索。每次的训练循环中，机器人从起始点出发，不停的训练，当智能体在动作集合中选择某动作后，环境接收该动作并发出状态转移，同吋给出奖赏。如果其动作获得环境正的奖赏(*+r*)，则移动机器人此后产该动作的趋势就会增强；反之，如果获得环境负的奖赏(*-r*)，则会减弱。最终通过足够多次的训练次数，Q值表会出现相对收敛的情况，机器人将会在Q值表中找到一条最优的路径。

4.2.1.5 其他参数初始化

对于文中其他参数的设计定义如下：折扣因子，学习效率;最大训练循环次数根据迷宫大小而不同；设置最低尝试次数Avgtrials=10。

4.2.2 路径规划策略

移动机器人路径规划步骤可以分为以下几步：

1. 初始化各项数据，开始从起点到终点的尝试（训练循环）；
2. 机器人从起始点出发；
3. 获取当前坐标的四个动作状态对应的四个Q值，通过贪婪决策，找出最大Q值对应的动作，记录原坐标。根据算法要求记录相应数据；
4. 获取下个坐标位置和奖励值，如果下一个坐标上有障碍物，则机器人下一个坐标为原来坐标，奖励值为-50；如果下一个坐标上没有障碍物，则返回奖励值为-2，进入下一个坐标；如果下一个坐标是终点，则返回奖励值为100，进入下一个坐标。机器人通过算法更新相应的Q值；
5. 判断当前坐标是否为终点，若是终点，进入6)，否则进入3)；
6. 判断智能体尝试次数是否超过最低尝试次数，如果超过则进入7)，否则进入2)；
7. 判断Q值是否收敛，是则进入8)，否则进入2)；
8. 结束。

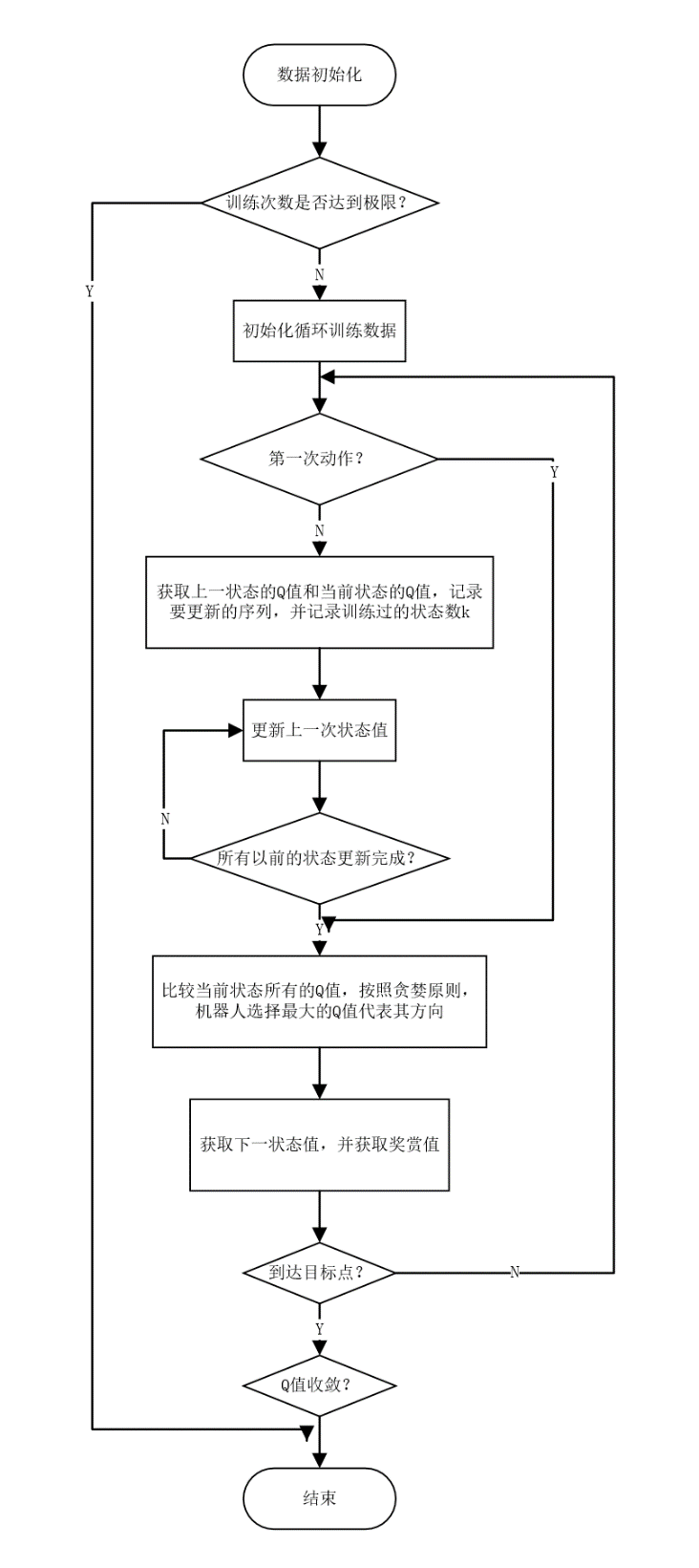
单个移动机器人在静态未知环境下的路径规划用流程图的形式给出如下图。

图 4 - 2 路径规划策略流程图

4.3 仿真结果与分析

本次实验结果采用Q-learning算法和改进后的算法对比展示。两种算法在迷宫环境，学习率，折扣率等设置方面完全相同。

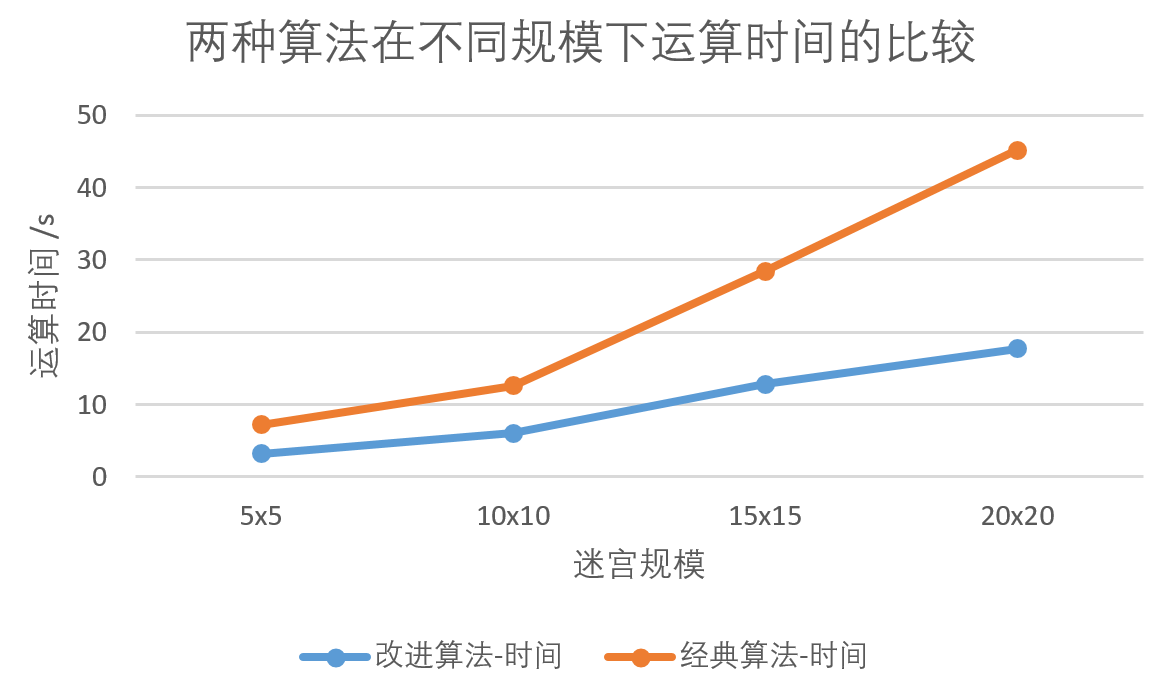
为全面展示两种算法，防止小概率事件发生，仿真在5\*5，10\*10，15\*15，20\*20的迷宫规模下进行，并且每一种规模将随机生成3个迷宫环境，取其平均值进行展示。

图 4 - 3 不同迷宫规模学习总步数比较

如图4-3所示，在不同迷宫规模下，经典Q-learning算法和改进之后的Q-learning算法进行比较，可以明显看出，改进之后的算法在学习步数和运算时间上有了很大进步

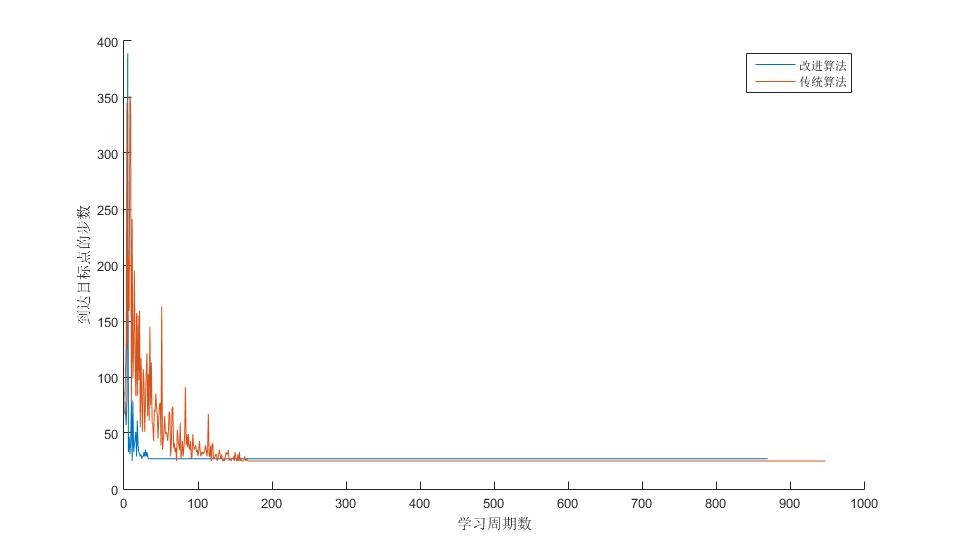
 这里我们用15\*15的迷宫规模来比较两种算法具体的收敛情况，如图4-4所示。

图 4 - 4 在迷宫规模为15\*15下两种算法的收敛情况比较

从上图中可以看出，在改进Q-learning算法较经典Q-learning算法，收敛速度明显加快。

两种算法的比较如下：

每一次回溯迭代的公式和Q-learning大体一致，但也有一定的差别：改进算法在t+1时刻，更新状态-动作所用的便是最纯粹的Q-learning。传统Q-learning的两大要素是采用贪婪决策和公式（3-3）。改进算法在更新t时刻前所有的状态-动作对时候，和Q-learning算法使用了同样的公式（3-3），但并没有采用贪婪决策，这边是最主要的区别。

Q-learning算法中任意状态的动作选择是通过比较状态对应的所有Q值，其最大的Q值对应的动作即为的动作选择，如果它指向,则通过的Q值更新的Q值中最大的值.整个更新过程可以理解为，为选择要更新的Q值，再通过该Q值找到相应的动作，然后由下一个状态的Q值和返回的奖励值r，更新需要更新的Q值。这里面动作的选择属于一个动作决策—贪婪决策，即动作的选择为最大的Q值对应的动作。

而在改进算法回溯迭代部分的公式中，动作决策由E记录的动作参数来决定，即在状态中不再通过最大的Q值来选择动作。当上一个状态若转为，则在该训练循环中，以后状态的动作必然是指向，并且要更新的Q值（这里的Q不一定是中最大的Q值）也是由记忆矩阵M中的数据来指定的。数据的传递是固定的，从状态链的末端不断往前传递，整个更新过程可以变相理解为先选择指定的动作，通过动作确定状态中更新指定的Q值，然后由下一个状态的Q值和返回的奖励值r，更新需要更新状态的中指定的Q值。

4.4 本章小结

本章主要目的是对前一章节的理论内容进行实验验证。首先介绍了实验环境，以及相应的参数设置，详细地介绍了算法原理，算法实现过程，以及实验的流程。最后对实验结果进行了简单的展示，证明改进的算法切实可行。

第5章 结束语

5.1本文内容

移动机器人在环境探索过程中，由于环境的复杂性，不可避免的会遇到各式各样的障碍物，如何识别避开障碍物，如何尽快的到达目标位置，这是最基本的问题，也是衡量移动机器人路径规划性能的重要指标。

本文针对单个移动机器人在静态环境下的路径规划问题进行了研究，采用强化学习算法来实现移动机器人在未知环境下的路径规划，在阅读大量文献研究路径规划算法的基础上，提出了改进算法，仿真实验结果证明了算法的有效性和可行性，达到了预期目标。总结起来，本文的主要研究成果和贡献如下：

1) 回顾和总结了目前国内外在移动机器人路径规划领域的发展和研究现状；详细介绍了强化学习算法的相关内容，并对其他常用的路径规划算法做了简要介绍。

2) 针对移动在未知环境下的路径规划问题，机器人如何从起始点，通过一定的策略，规划出一条无碰撞的最优或次最优的可行路径，本文首先对Q-learning算法和算法进行了研究，发现移动机器人在原有的强化学习算法上需要很长的时间才能够找到最优或次最优的路径，在学习的过程中收敛速度很慢，为解决这一问题，提出了一种改进算法，通过实验证明了该算法能够使机器人通过快速的学习找到最优路径，为移动机器人路径规划问题提供了一种崭新的方法。

5.2下一步学习工作方向

移动机器人在未知环境下的路径规划问题足一项比较复杂的题，这领域还处于研究阶段。本文虽然在路径规划算法研究中取得了一定的进展，提出了改进算法应用到了单个移动机器人的路径规划中，但仍然存在很多问题值得深入研究和探讨。主要包括以下儿点：

1) 本文仅进行了基于强化学习算法和改进强化学习算法的研究和实验，没有应用到移动机器人平台上运行。接下来可以根据提出的算法，在移动机器人平台上进行实验，进一步验证算法的切实应用效果。

2) 在未知环境下的路径规划问题中，本文仅研究了在静态障碍物环境的规划，而对于在动态障碍物、不规则障碍物的环境中，如何实现移动机器人的路径规划仍需要深入研究。

参考文献

[1] 王志文，郭戈. 移动机器人导航技术现状与展望[J]. 机器人，2003.25(5):193-197

[2] Salichs. M. A, Moreno. L. Navigation of Mobile Robot: Open Questions[J]. Robotica, 2000,18:227-234

[3] 李熊明，熊蓉，褚键. 室内自主移动机器人定位方法研究综述[J]. 机器人，2003,25(6):561-567

[4] 王奇志，给予改进人工势场法的多障碍机器人运动控制，2003年中国智能自动化会议论文集[J], 2003:656-659

[5] Lee M, Tarokh M, Cross M. Fuzzy logic decision making for multi-robot security systems[J]. Artificial Inteligence Review, 2010(257): 33-52

[6] A. Zhu and S. X. Yang. Perceptual control based on prediction for natural communication of a partner robot[J]. IEEE Trans.Ind.Electron.2007,54(2):866-877

[7] 郭锐，吴敏，彭军，彭娇，曹卫. 一种新的多智能体Q学习算法[J]. 自动化学报,2007,4:367-372

[8] 李磊，叶涛等. 移动机器人技术研究现状与未来[J]. 机器人，2002,24(5):20-24

[9] Rolf L., Jan L.L., Sun, X. Y., Diedrich w. Geometric robot mapping[J]. Lecture Notes in Computer Science, 2005,34(29):11-22

[10] 艾海舟，张跋. 基于拓扑的路径规划问题的图形解法[J]. 机器人，1999,12(5):20-24

[11] 黄炳强，曹光益，王古全. 强化学习算法原理及应用[J].河北工业大学学报，2006,35(6):34-38

[12] 焦宝聪，陈兰平. 运筹学的思想方法及应用[M]，北京：北京大学出版社，2008

[13] 余涛，胡细兵，刘婧. 基于多步回溯学习算法的多目标最优潮流计算[J]. 华南理工大学学报，2010,38(10):139-145

[14] Singh S, Sutton R S. Reinforcement learning with replacing eligibility traces[J]. Machine Learning,1996,22:123-158

致谢

本论文的工作是在我的导师屈鸿老师悉心指导下完成的，屈鸿老师和蔼的脾气和细心的品质一直在端正着我的态度，求实的精神深深感染和激励着我，让我终身受益，也是我今后努力学习的楷模。在论文开始撰写时期，遇到了很多问题，感谢黄利伟学姐的耐心讲解，让我少走了很多弯路，没有她，这篇论文也不会写的很顺利。最后感谢我的室友，遇到了很多格式问题，都是大家一起解决的。

谨以此文献给所有关心和帮助过我的人！

外文资料原文

Application of single agent Q-learning for light exploration

SECTION IV. Single Agent Q-learning

Learning is needed for forming a map/ function between the state (sensor information) and action (actuator commands). Supervised learning approaches need the model of good behaviour by a teacher. Reinforcement learning is one of the widely used online learning methods in robotics. It is sometimes called learning with the critic that gives scalar reward or punishment based on behaviors. The robot learns during an action and also acts/ responds during the learning, which is neglected in supervised learning methods. The Reinforcement learning method used in this paper is single agent based Q-learning as shown in Figure 1. In single agent based Q-learning algorithm the external world is modeled as Markov Decision Process with discrete states and action spaces. It is flexible in the sense that it can learn from actions which it/ programmer does not suggest. This ability is often called exploration - insensitivity. After each step an agent (single) observes the state vector , chooses and applies an action . The system passes in state and the agent receives a reinforcement . The goal of the learning is to find a policy which maximizes the sum of the future reinforcements. For a given policy *π*, it is to be noted that , the chosen action. The evaluation function *π*, designated as , is given by:

|  |  |  |
| --- | --- | --- |
|  |  | (1-1) |

The discounting factor *γ*∈[0,1] smaller than 1, ensures convergence of the sum. The optimal evaluation function , is defined as follows

|  |  |  |
| --- | --- | --- |
|  |  | (1-2) |

Where *β* is a learning parameter which must tend towards 0 when t tends towards infinity.

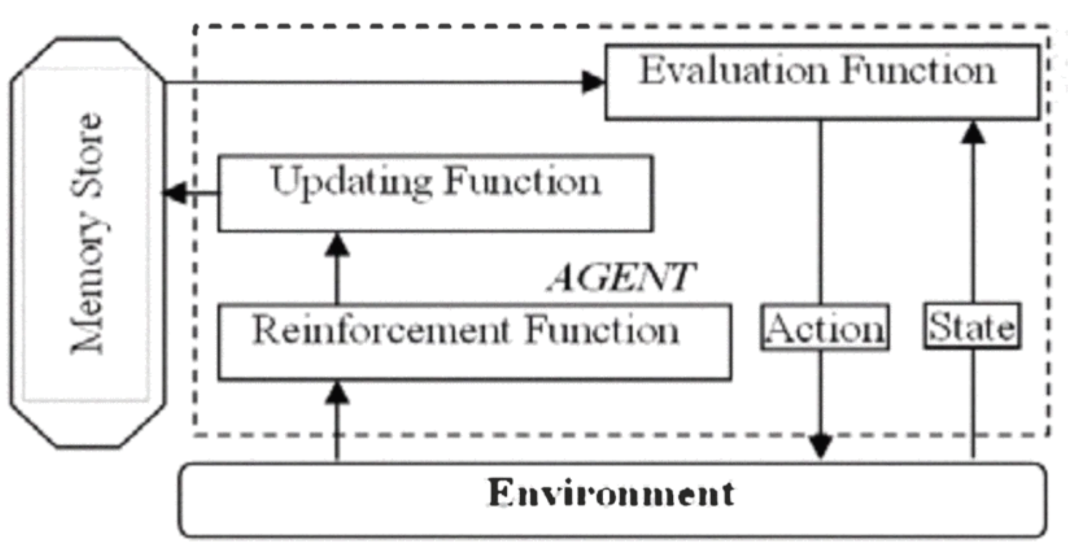
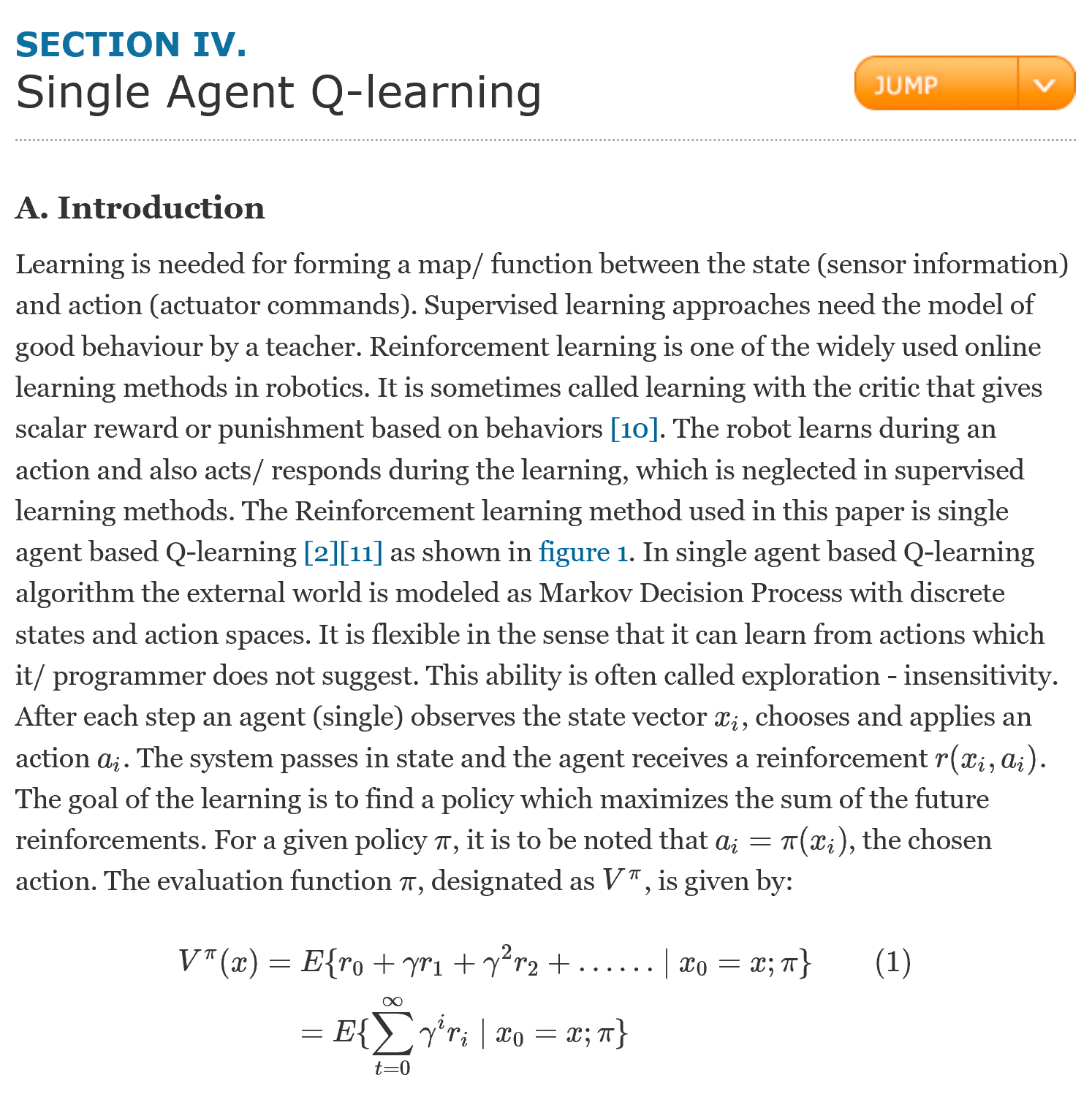
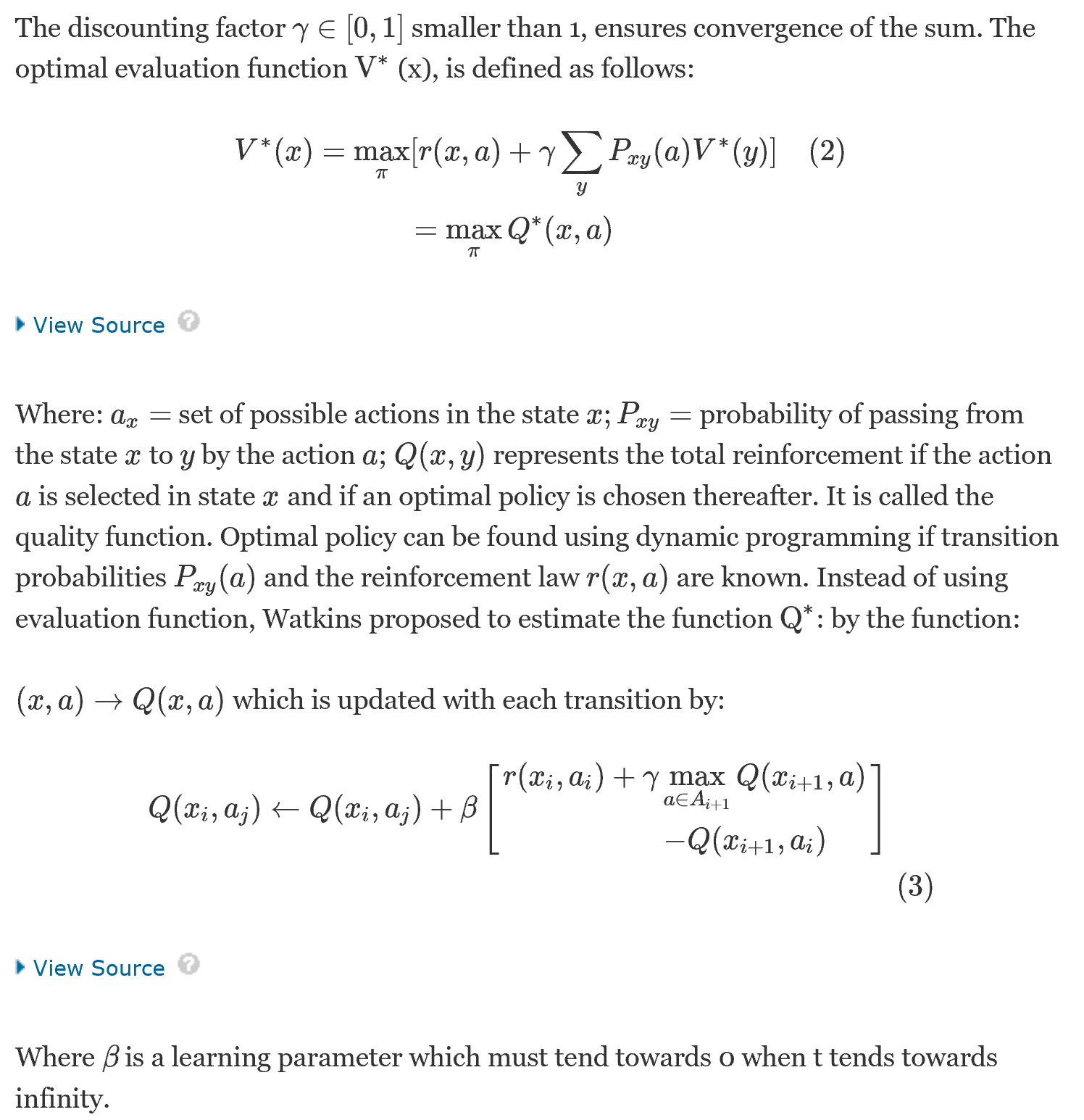


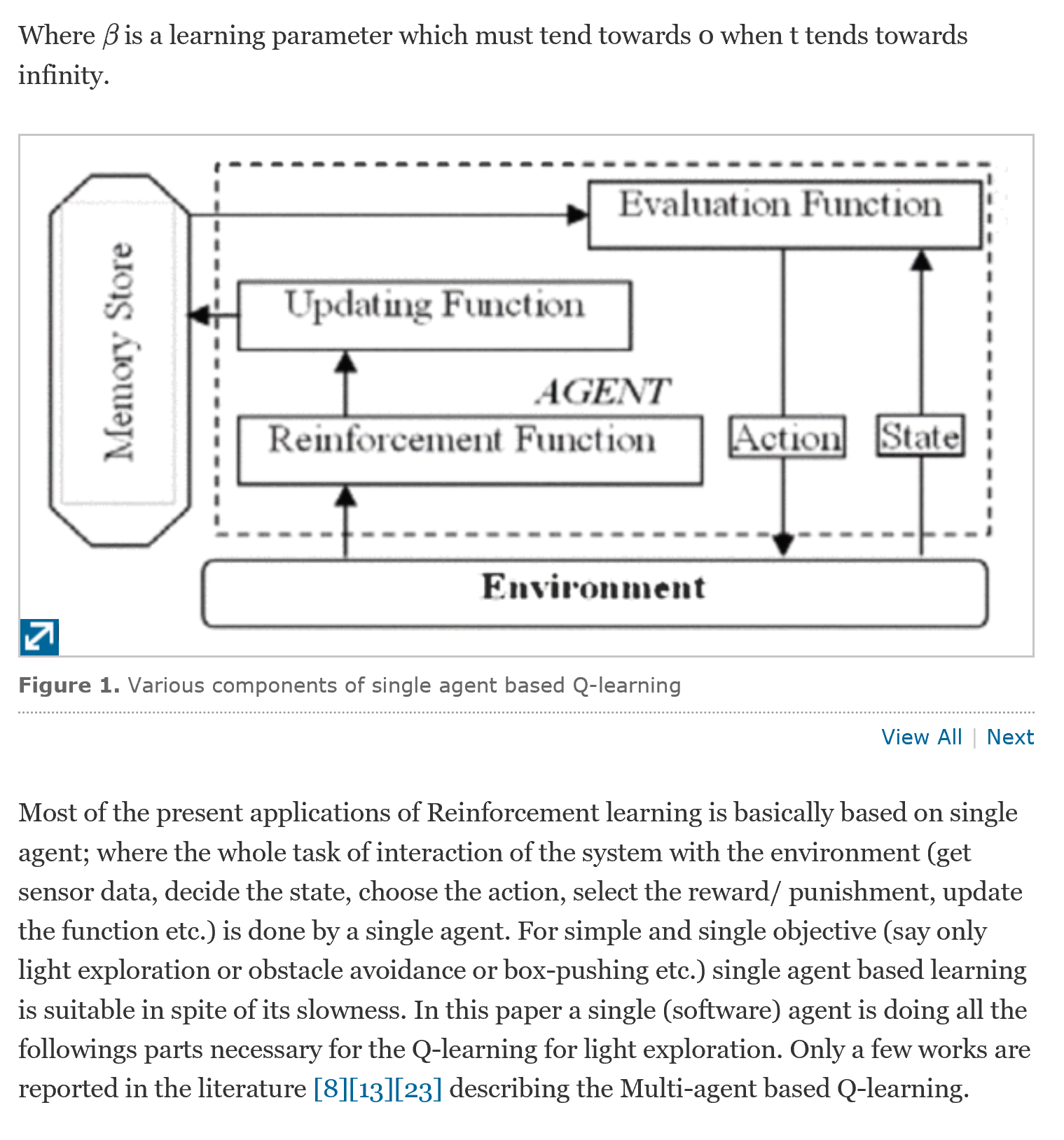
Figure Single Agent

Most of the present applications of Reinforcement learning is basically based on single agent; where the whole task of interaction of the system with the environment (get sensor data, decide the state, choose the action, select the reward/ punishment, update the function etc.) is done by a single agent. For simple and single objective (say only light exploration or obstacle avoidance or box-pushing etc.) single agent based learning is suitable in spite of its slowness.

英文原文截图：







外文资料译文

Q-learning算法在单一机器人寻光中的应用

第四节 单一机器人Q-learning

需要形成状态（传感器信息）和行动（行为选择器的命令）之间的映射/函数学习。监督学习方法意思是：在不断接近目标的过程中，需要一个“老师”的指导。强化学习是在机器人学习领域广泛使用的在线学习方法之一。它有时被称为基于行为的标量奖励或惩罚的批评学习。机器人在作出行为和环境相应之间进行学习，这一点恰恰是监督式学习中所缺乏的一点。基于强化学习算法的单一机器人应用，参见图一。基于强化学习算法的单一机器人应用，周围的环境模型可以视为马尔科夫决策过程，拥有状态和行为值。它的灵活性就体现在，在没有程序人员建议的情况下，可以自己从行为中学习。这种能力通常被称为试探--不敏感。单一机器人在每一步学习之后，都会观察状态向量，以此来选择并且应用行为。机器人在强化学习系统通过函数来传递和接受状态值。强化学习的目标就是找到一个策略，使得奖励值最大。我们给定一个策略，使用这个策略来选择行为的方式定义为：。这个最优的策略，我们设为,由下式给出：

|  |  |  |
| --- | --- | --- |
|  |  | (1-1) |

这里的为折扣函数，小于1，作用是使得总和收敛。最优评测函数，定义式为：

|  |  |  |
| --- | --- | --- |
|  |  | (1-2) |

这里的是在状态x下，所有可能的行为的集合；是状态由x转移到y，采用行为a的可能性；Q(x,y)代表着采取行为a之后，得出的最优的强化值。如果转移概率和奖励值已知，最优策略也可以用动态更新的方法来获得。Watkins提出了建立一个公式1-3，来建立，状态-行为对更新Q值更新方法如公式1-3

|  |  |  |
| --- | --- | --- |
|  |  | (1-1) |

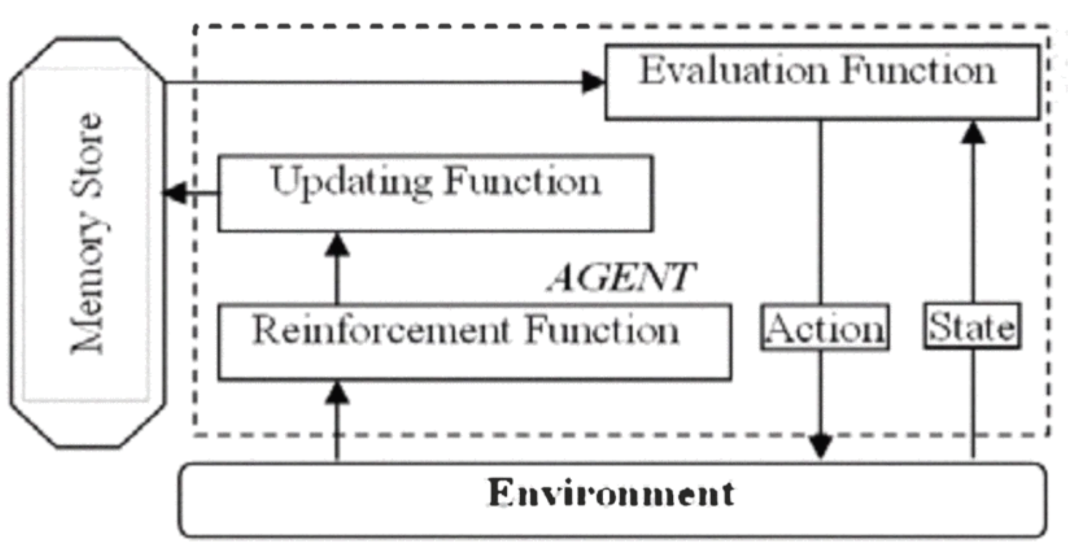
在这里是学习函数，当t趋向于无穷时，它必须趋近于0

图 1 - 1 单机器人强化学习

目前，绝大多数强化学习算法的应用都是基于单机器人的，也就是说与环境交互（包括获取传感器信息，决定状态，选择策略，得到奖赏惩罚值，更新函数等）的机器人只有一个。