Advances in Applied Mathematics 应用数学进展, 2022, 11(12), 8723-8733
Published Online December 2022 in Hans. http://www.hanspub.org/journal/aam
https://doi.org/10.12677/aam.2022.1112919

Hans汉斯

基于PID控制更新的Sarsa强化学习算法及应用

吴雯珑[1]，龚谊承[1,2*]

[1]武汉科技大学理学院数学与统计系，湖北 武汉
[2]武汉科技大学，冶金工业过程系统科学湖北省重点实验室，湖北 武汉

收稿日期：2022年11月14日；录用日期：2022年12月8日；发布日期：2022年12月19日

摘 要

针对强化学习中Sarsa算法收敛速度慢且效果不稳定的问题，考虑到PID控制操作简单且鲁棒性高，提出基于PID控制优化的Sarsa算法，即Pid_Sarsa。其主要思想是将Sarsa算法中Q值的迭代方式改进为三项之和，分别对应PID控制中的比例、积分和微分，体现了对当前、过去和未来的误差进行控制的思想，理论上提高了样本利用率。为了对比Pid_Sarsa算法与Sarsa和n_Sarsa(取n = 5)两种传统算法的效果，选择悬崖寻路这一经典路径规划游戏作为算例，实验表明：Pid_Sarsa算法收敛速度更快、效果更稳定，且得到的路径安全程度比Sarsa算法高2.38%，比5步Sarsa算法高4.76%。

关键词

强化学习，Sarsa，PID控制，路径规划

Sarsa Reinforcement Learning Algorithm and Application Based on PID Control Update

Wenlong Wu[1], Yicheng Gong[1,2*]

[1]Department of Mathematics and Statistics, College of Science, Wuhan University of Science and Technology, Wuhan Hubei
[2]Hubei Province Key Laboratory of Systems Science in Metallurgical Process, Wuhan University of Science and Technology, Wuhan Hubei

Received: Nov. 14th, 2022; accepted: Dec. 8th, 2022; published: Dec. 19th, 2022

Abstract

In view of the problem of slow convergence speed and unstable effect of Sarsa algorithm in rein-

*通讯作者。

文章引用: 吴雯珑, 龚谊承. 基于 PID 控制更新的 Sarsa 强化学习算法及应用[J]. 应用数学进展, 2022, 11(12): 8723-8733.
DOI: 10.12677/aam.2022.1112919

forcement learning, considering the simple operation and high robustness of PID control, we propose a Sarsa algorithm based on PID control optimization, that is, Pid_Sarsa. The main idea is that improve the iterative way of Q values in Sarsa algorithm to the sum of three terms, corresponding to the proportions, integrals and differentiation in PID control, which reflect the idea of controlling current, past and future errors, and theoretically improves sample utilization. In order to compare the effects of the Pid_Sarsa algorithm with the two traditional algorithms of Sarsa and n_Sarsa (n = 5), the classic path planning game of cliff walking is selected as an example, and the experiments show that the Pid_Sarsa algorithm converges faster and the effect is more stable, and the obtained path security degree is 2.38% higher than that of Sarsa algorithm and 4.76% higher than that of 5-step Sarsa algorithm.

Keywords

Reinforcement Learning, Sarsa, PID Control, Path Planning

1. 引言

如今，强化学习[1]在各个领域的应用十分广泛，尤其当 AlphaGo 击败人类顶尖围棋选手之后，一股人工智能热潮更是为世界工业文明带来了新的活力。

在强化学习中，如果已知环境模型，通常使用动态规划(Dynamic Programming, DP)的方式解决问题，但实际情况往往复杂且多变，我们常常无法得知环境的状态转移函数和奖励函数的形式。由于没有具体的模型，则只能通过智能体与环境交互，不断采样的方式获得数据，从而进行策略评估和策略迭代直至得到最优策略，在强化学习中称这类算法为无模型算法。蒙特卡洛(Monte-Carlo, MC)算法是其中最为经典的算法。1988 年 Sutton 等人首次结合 MC 和 DP 算法的思想，提出时序差分(Temporal-Difference, TD)算法[2]。如何根据自己的需求，有选择地从环境中选择自己需要的数据，权衡对现在和未来时刻的利用和探索，为了更好地解决这一问题，强化学习算法又分为在线策略(on-policy)和离线策略(off-policy)算法。

Sarsa [3]作为 TD 算法中基于无模型的典型在线策略学习方法，大量学者对此都有所探讨。基于 Sarsa 算法收敛效果不稳定的缺点，VanS 等人将下一状态 Q 值的期望作为目标值的估计，提出方差更小、估值更加稳定的 Expected Sarsa 算法[4]。Robards 等人考虑使用无参函数逼近函数的核方法，将一种有界的基于核的感知器与 Sarsa 算法相结合，提出基于核的 Sarsa(λ)算法[5]，使得算法更加高效；在此基础上，朱海军从稀疏化算法的角度出发，提出基于聚类的选择性核 Sarsa(λ)算法[6]，都有效缓解了强化学习算法中收敛精度低、收敛速度慢的问题。De A.等人在全采样和非采样之间，将经典 Sarsa 和 Expected Sarsa 算法结合，提出用两者估值方程的凸组合进行估值的 $Q(\sigma)$ 算法，在实验效果中有着更优的表现[7]；同年，杨瑞分析了 $Q(\sigma)$ 算法的收敛条件[8]，为其提供了理论基础。这几种方法虽然能一定程度上缓解 Sarsa 算法收敛速度慢、精度低、效果不稳定的问题，但仍然存在各自的局限性。

随着信息技术的不断发展，不仅无模型控制是自动控制领域中的一个重要发展方向，在复杂系统和复杂性科学面前，如何找到简单有效的控制方式也是目前研究的重难点。而 PID (Proportion Integration Differentiation)控制作为系统控制经典方法的代表，有着结构简单、稳定性好、可靠性高等优势，在工业控制中有着广泛的应用。在无法得知精确的数学模型，或者涉及的问题过于复杂的情况下，PID 控制技

术可以仅仅根据系统的误差，利用比例、积分、微分计算出控制量，从而“稳定、快速、准确”地逼近目标值。

在系统与控制科学不断进步的当今，各种复杂系统的精准有效调控显得尤为重要[9] [10]，而 PID 控制与强化学习方向的有效结合，更是有望将问题化繁为简，使得算法更加高效稳定。部分学者对此也有所研究。陈学松等人将强化学习中执行器－评价器(Actor-Critic, AC)学习算法与 PID 控制结合，利用 AC 算法的无模型在线学习能力，对 PID 参数进行自适应调整，提出 AC-PID 控制器设计方法，使得控制器的响应速度提高，自适应能力增强[11]；在此基础上，段友祥等人利用多线程异步学习特性，提出基于异步优势执行器－评价器的 PID 控制器[12]。在连续型环境下，程丽梅等人以一阶倒立摆系统为例，将 PID 控制和强化学习算法效果进行对比分析[13]。甄岩等人将深度强化学习算法应用于 PID 控制的参数整定，研发出更加智能的复杂飞行控制器[14]。

Sarsa 算法作为一种在线策略算法，一定程度解决了数据的利用和探索困境，但探索的程度也会影响算法收敛速度以及稳定程度。而 PID 控制在无模型的情况下，不仅能够迅速稳定地逼近目标，并且通过比例、积分、微分部分的参数调整，对现在、过去、未来的误差进行有效地控制和预防，对数据利用与探索的权衡方面也有着很大地启发。因此，本文将 PID 控制与强化学习 Sarsa 算法中 Q 值的更新方式相结合，讨论这种基于 PID 迭代更新算法的优点和适用性，以期提高算法的收敛速度和稳定程度，使其具有更高的理论和实践意义。

2. 主要知识简介

2.1. 强化学习的 Sarsa 学习简介

2.1.1. 强化学习基本原理

在强化学习中，智能体通过不断地与环境进行交互，得到反馈，在试错中学习，以此来调整优化自身的状态信息，其目的是为了找到最优策略或是最大奖励。

在现实中的强化学习环境中，一般是基于马尔可夫决策过程(MDP)的相关理论，将问题抽象为五元组$\langle S, A, P, R, \gamma \rangle$，其中 S 表示环境的有限状态集，A 表示智能体的有限动作集，$P: S \times A \times S' \to [0,1]$ 表示状态转移函数，R 表示奖励函数，γ 表示折扣因子。需要特别注意的是，智能体从当前状态 s，根据某一策略 π 选择动作 a，到达下一个状态 s'的概率是满足马尔可夫性质的，如(1)式所示。

$$P(s' \mid s, a) = P(S_{t+1} = s' \mid S_0, A_0, S_1, \cdots, S_t, A_t) = P(s' \mid S_t = s, A_t = a) \tag{1}$$

(1)式表明：在给定当前状态时，智能体的未来状态与过去无关，它们之间是条件独立的。

在强化学习中，我们不但要考虑算法的计算量，也要考虑我们产生数据所消耗的成本。如何控制成本提高数据效率，即对应着强化学习中经典的利用探索困境。强化学习的目标是“尽量找出奖励期望值最大的动作”，但由于未来的不确定性，无法真正确定奖励的真值。如何在最大化利用数据的同时又不失对数据的探索，这二者之间的平衡是提高数据效率的关键所在。这一关键不仅运用于选取动作的探索算法中，在对动作价值函数的迭代更新中也有所体现。Sarsa 算法、n 步 Sarsa 算法、Q-learning 算法都是强化学习中的常用方法，下面仅对 Sarsa 算法的更新过程做简要介绍。

2.1.2. Sarsa 算法简介

Sarsa 算法由其迭代过程$\langle s, a, r, s', a' \rangle$得名，是强化学习中经典的无模型方法。

在强化学习中，对目标量的估计更新形式如(2)式所示。

$$\begin{aligned} &\text{NewEstimate} \leftarrow \text{OldEstimate} + \text{StepSize} \\ &(\text{Target-OldEstimate}) \end{aligned} \tag{2}$$

(2)式中 Target-OldEstimate 表示估计的误差，也对应着更新的方向，StepSize 表示更新步长。

时序差分算法不同于蒙特卡洛和基于动态规划的算法，其无需知道具体的状态转移函数 P 和奖励函数 R，而是直接通过在环境中采样即可得到数据。基于贝尔曼方程的思想，它采用当前获得的奖励加上下一个状态的价值函数来当作在当前状态所获得的回报，其中常数值 α 为更新步长，γ 为折扣因子，更新形式如(3)式。

$$V(S_t) \leftarrow V(S_t) + \alpha\left[R_t + \gamma \times V(S_{t+1}) - V(S_t)\right] \tag{3}$$

(3)式中 $R_t + \gamma \times V(S_{t+1}) - V(S_t)$ 被称为时序差分误差。

Sarsa 算法基于时序差分的思想，直接更新动作价值函数 $Q(S_t, A_t)$，再根据 ε-greedy 策略 $\pi(a|s)$ 选择相应的动作，更新方式如(5)式。

$$\pi(a|s) = \begin{cases} 1-\varepsilon & a_t = \arg\max_{a \in A} Q(s_t, a_t) \\ \varepsilon & a_t \text{ is a random action} \end{cases} \tag{4}$$

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\left[R_t + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)\right] \tag{5}$$

不同于 Sarsa 算法只利用一步奖励和下一个状态的价值估计去估计目标值，n 步 Sarsa 算法利用了 n 步的奖励和第 n 步时的动作价值函数估计，如(6)式。

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\left[R_t + \gamma R_{t+1} + \cdots + \gamma^n Q(S_{t+n}, A_{t+n}) - Q(S_t, A_t)\right] \tag{6}$$

Sarsa 算法的收敛性与 ε-greedy 策略有关，智能体以 $1-\varepsilon$ 的概率选择 Q 值最大的动作，再以 ε 的概率从动作集中随机选择动作。由于 ε 的取值较小，所以可以保证算法以概率 1 收敛到最佳策略，但也会导致更新过程中对所有 Q 值的访问时间较长，收敛速度变慢，并且也有陷入局部最优的风险。

针对 Sarsa 算法收敛较慢、收敛效果不稳定、容易陷入局部最优的问题，本文拟利用 PID 控制的思想来改进 Sarsa 算法。

2.2. PID 控制简介

2.2.1. PID 控制系统

经典的 PID 控制系统如图 1 所示，其中输入给定目标值，输出实际值，$e(t)$是目标值与实际值之间的偏差，而 $u(t)$是控制量，控制系统主要有被控对象和控制器组成，其中控制器是关于偏差 $e(t)$的比例、积分、微分三部分的线性组合[15]。

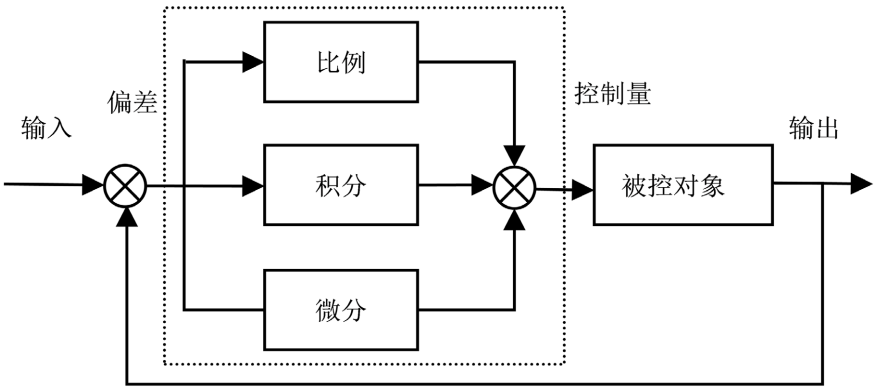

Figure 1. Flowchart of a classic PID control system
图 1. 经典 PID 控制系统流程图

PID 控制公式可写为：

$$u(t)=k_P*e(t)+\frac{k_P}{T_i}*\int_0^t e(t)\mathrm{d}t+k_P\tau*\frac{\mathrm{d}e(t)}{\mathrm{d}t} \tag{7}$$

其中为 k_P 为比例系数，T_i 为积分时间常数，τ 为微分时间常数。

在经典 PID 控制系统的前提下，离散状态为相应的位置式 PID 控制，控制公式如(8)式。

$$u(t)=k_P*e(t)+k_I*\sum_{i=0}^{t}e(i)+k_D\left[e(t)-e(t-1)\right] \tag{8}$$

其中 k_P,k_I,k_D 分别为比例、积分、微分系数。

2.2.2. PID 学习律

基于迭代学习的思想，在系统控制理论下，学习控制的目的是找到一个合适的学习律，当迭代学习中序列 $u_k(t)$ 一致收敛于理想控制输入时，被控系统的实际输出值 $y_k(t)$ 尽可能接近目标值 $y_0(t)$，其中 k 为迭代次数，$\Gamma_p,\Gamma_i,\Gamma_d$ 为迭代学习增益矩阵。PID 学习律的形式如(9)式。

$$u_{k+1}(t)=u_k(t)+\left(\Gamma_p+\Gamma_i*\int\mathrm{d}t+\Gamma_d*\frac{\mathrm{d}}{\mathrm{d}t}\right)e_{k+1}(t) \tag{9}$$

而在非线性系统的 PID 学习律的收敛性在文献[16]中已得到证明。

正是考虑到 PID 控制能够“稳定、快速、准确”地逼近目标值，本文提出 Pid_Sarsa 算法，以期解决 Sarsa 算法收敛较慢、收敛效果不稳定、容易陷入局部最优的问题。

3. 基于 PID 更新的 Pid_Sarsa 算法

在工程控制中，通常根据系统是否有反馈，将控制分为开环控制和闭环控制，而 PID 控制在闭环系统中有着独特的优势。在强化学习中智能体与环境的交互过程中，智能体每执行一个动作，都会到达下一个状态，获得一个及时奖励，如图 2 所示。

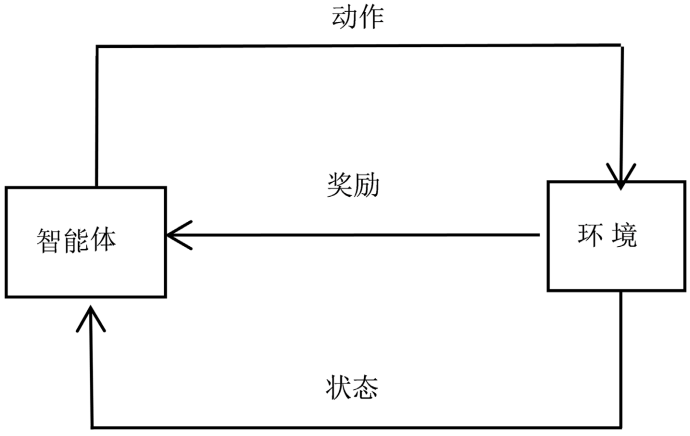

Figure 2. Flowchart of reinforcement learning
图 2. 强化学习流程图

另一方面，开环控制只关注输入，而不关注输出结果精度，输出结果是不确定且没有目标的，不会反馈回输入参与控制。而强化学习以找到最优策略、或者找到最大奖励为目标，且拥有及时反馈，所以将 PID 控制应用于强化学习环境是可行的。

在强化学习中，Sarsa 算法具有收敛较慢、收敛效果不稳定且容易陷入局部最优的缺点，由于 Q 值的迭代方式决定了该算法的收敛效果，所以考虑对 Sarsa 算法中的迭代方式进行细化。而 PID 控制能够“稳定、快速、准确”地逼近目标值，在控制误差方面有着十分优良的特性，所以我们将 Sarsa 算法中 Q 值的迭代更新方式改进为三项之和，分别对应 PID 控制中的比例、积分和微分，体现了对当前、过去和未来的误差进行控制的思想。

用误差的比例部分来控制当前误差的变化，比例系数越大，消除能力也就越强。但由于系统存在惯性，比例作用太强不仅不能完全消除误差，还会导致系统的不稳定。由于比例部分不能完全消除误差，总是存在静态误差，所以对过去的误差进行积分直至误差为零，达到无差调节的效果。

误差关于时间的微分体现变化趋势，比例和积分部分都是对当前及过去产生的误差进行调节消除，而微分部分则关注误差产生的趋势，通过调节微分系数，对误差进行超前调节，以达到预防控制的效果。由于微分控制的及时性，可以最大程度地减少动态误差。但微分作用太强也会导致系统的不稳定。

基于 PID 控制对误差更新调节，本文提出 Pid_Sarsa 算法。在 Sarsa 算法中，将时序差分误差作为实际值与目标值之间的偏差 $e(t)=R_t+\gamma*Q(S_{t+1},A_{t+1})-Q(S_t,A_t)$，引入比例、积分、微分部分，来控制误差的大小和变化趋势，其中自然地将比例系数 α_p 取为步长 α 的值，令 α_i，α_d 分别为积分、微分系数，得到以下对动作价值函数的更新方式：

$$Q(S_t,A_t)\leftarrow Q(S_t,A_t)+\alpha_p*e(t)+\alpha_i*\sum_{i=0}^{t}e(i)+\alpha_d*\left[e(t)-e(t-1)\right] \tag{10}$$

基于 PID 控制更新的 Pid_Sarsa 算法流程如下[17] [18]：

1) 初始化状态 $S_t=s$；
2) 用 ε-greedy 策略选择动作 $A_t=a$；
3) 采取动作 a 得到下一个状态 S_{t+1} 和奖励 R_t；
4) 用 ε-greedy 策略根据 S_{t+1} 求得动作 A_{t+1}，得偏差 $e(t)=R_t+\gamma*Q(S_{t+1},A_{t+1})-Q(S_t,A_t)$
5) $Q(S_t,A_t)\leftarrow Q(S_t,A_t)+\alpha_p*e(t)+\alpha_i*\sum_{i=0}^{t}e(i)+\alpha_d*\left[e(t)-e(t-1)\right]$
6) $S_t\leftarrow S_{t+1}$；$A_t\leftarrow A_{t+1}$
7) 直到到达终止状态(到达目标状态或者掉入悬崖)。

4. 实验及结果分析

4.1. 实验环境

路径规划下的悬崖寻路(Cliff Walking)是强化学习的经典环境，属于一种特殊的网格世界游戏。

在这个游戏环境中，每一个网格是一个状态。如图 3 所示，起点在网格左下角，目标在右下角，其中网格世界底部有一段悬崖。智能体到达目标状态或掉入悬崖都会结束并回到起点，即目标状态和悬崖都称为终止状态。

智能体在每一个状态都可以采取上下左右 4 种动作。如果采取动作后触碰到网格边界则状态不发生改变，否则就会相应到达下一个状态。每走一步或者掉入悬崖，智能体都会获得相应的奖励值。目标是找到一条避开悬崖到达目标位置的最短路径，使得预期得到的奖励值最大。

为了便于说明，如图 3 所示建立坐标系，并将悬崖寻路游戏抽象成数学模型进行描述。

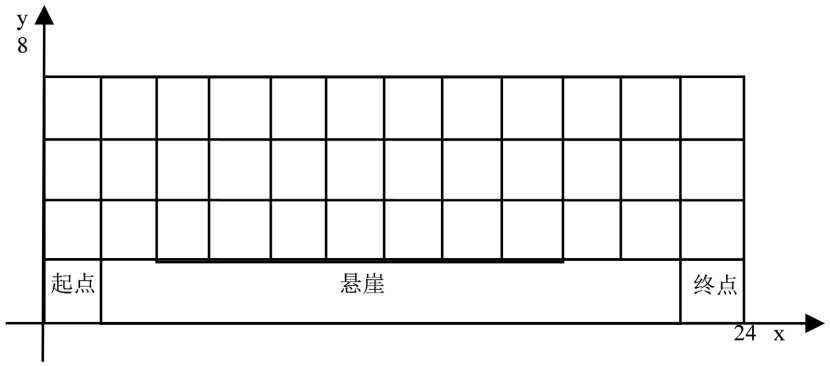

Figure 3. Schematic of cliff walking environment
图 3. 悬崖寻路环境示意图

游戏环境由 4 × 12 的网格组成，每个网格的长度与宽度都是 2 个单位。状态空间 S_t 表示如(11)式。

$$S_t = \left\{ \left(x_t, y_t \right) | \, x_t = 2n+1, y_t = 2n+1, x_t \in \left[0,24 \right], y_t \in \left[0,8 \right], n \in Z \right\} \tag{11}$$

$$\left\{ \left(x_t, y_t \right) | \left(x_t, y_t \right) \in S_t, x_t \in \left[0,2 \right], y_t \in \left[0,2 \right] \right\} \tag{12}$$

$$\left\{ \left(x_t, y_t \right) | \left(x_t, y_t \right) \in S_t, x_t \in \left[22,24 \right], y_t \in \left[0,2 \right] \right\} \tag{13}$$

$$\left\{ \left(x_t, y_t \right) | \left(x_t, y_t \right) \in S_t, x_t \in \left[2,22 \right], y_t \in \left[0,2 \right] \right\} \tag{14}$$

类似(11)式，(12)式、(13)式、(14)式分别表示悬崖寻路环境的起点状态、目标状态及悬崖。动作空间 A_t 表示如(15)式。

$$A_t = \left\{ a_t \, | \, a_t = 1,2,3,4 \right\}$$

$$a_t = \begin{cases} 1 & \left\{ y_t = y_{t+1} + 1 | \, x_t = x_{t+1} \right\} \\ 2 & \left\{ y_t = y_{t+1} - 1 | \, x_t = x_{t+1} \right\} \\ 3 & \left\{ x_t = x_{t+1} - 1 | \, y_t = y_{t+1} \right\} \\ 4 & \left\{ x_t = x_{t+1} + 1 | \, y_t = y_{t+1} \right\} \end{cases} \tag{15}$$

其中 a_t 为一个分段函数，对应上下左右四个动作。设奖励值为 R_t，每走一步的奖励 $R_t = -1$，掉入悬崖的奖励 $R_t = 100$。

为了定量地刻画实验结果的安全性，本文引入了三个相关的定义：1) 在 y 轴方向上，路线里每个状态中 y_t 距离悬崖的长度称为安全距离；2) 将路径所占的网格数量，即状态个数，称为路径长度；3) 将安全距离与最安全距离值的比值称为安全程度。可知最安全路径是指最远离悬崖的路线，安全距离为 84；最优路径是指不掉入悬崖的最近路线，安全距离为 24，也最为冒险。

在强化学习环境基础上，可知上述游戏为一个马尔可夫决策过程，且该状态空间和动作空间都是有限且离散的。已知当前状态 S_t 和动作 a_t，到达下一个状态 S_{t+1} 的转移概率为 $P\left(S_{t+1} | \, S_t = s_t, A_t = a_t \right)$，而由此得到的奖励 $R_t\left(s_t, a_t \right)$ 也是随机变量。智能体的目标是找到一条避开悬崖到达目标位置的最短路径，即寻找一个策略 $\pi\left(a_t | \, s_t \right)$，使得预期得到的奖励值最大。

4.2. 实验结果与分析

由于 PID 控制的积分部分是为了消除静差，且是根据系统环境进行具体调节的，而在悬崖寻路的实验环境下，积分部分会使得系统大幅度振荡且掉入悬崖，所以在实验过程中，本文根据实际情况采用 PD

控制，PD 型学习律的迭代控制结果也是收敛的[19] [20]。设置积分部分的系数 $k_I = 0$，再此基础上调节比例系数 k_P 和微分系数 k_D 的大小。

在悬崖寻路的环境下，首先采用强化学习中经典 Sarsa 算法和 n 步 Sarsa 算法(取 $n = 5$)求在多次实验下智能体通过悬崖的累计回报 Returns，如图 4、图 5 所示。

Figure 4. Cliff walking experiment based on Sarsa algorithm
图 4. 基于 Sarsa 算法的悬崖寻路实验

Figure 5. Cliff walking experiment based on 5_Sarsa algorithm
图 5. 基于 5 步 Sarsa 算法的悬崖寻路实验

由图 4 和图 5 可知，经过 1000 次实验，智能体的累积回报都不断提高，最终收敛在−20 左右。但两种经典算法相比，在初始状态，5 步 Sarsa 算法比 Sarsa 算法的累计回报较低，但收敛速度比 Sarsa 算法更快。

再采用基于 PID 控制更新的 Pid_Sarsa 算法进行实验，调整参数 $k_P = 0.3, k_D = 0.1$，得到结果，如图 6。

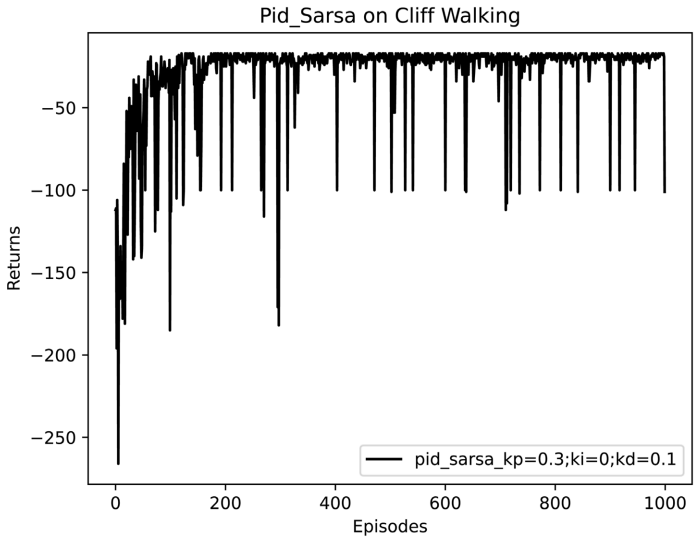

Figure 6. Cliff walking experiment based on Pid_Sarsa algorithm
图 6. 基于 Pid_Sarsa 算法的悬崖寻路实验

从图 6 可以看出，在悬崖寻路实验中，Pid_Sarsa 算法也能使累计回报收敛至−20 左右。为了避免一些偶然因素对实验结果产生影响，纵坐标改用累计回报的滑动平均值，将三种算法进行比较，如图 7、图 8 所示。

从图 7 可知，Pid_Sarsa 算法与 Sarsa 算法相比更快收敛。而从图 8 中可以看出，5 步 Sarsa 算法虽然收敛速度更快，但前 50 幕的累计回报更低，且在 700 幕左右出现十分小的累计回报，波动较大，所以 Pid_Sarsa 算法与其相比更加稳定。

根据经典的 Sarsa 和 5 步 Sarsa，以及本文提出的基于 PID 控制的 Pid_Sarsa 三种算法，在游戏环境中得到相应的路径图，如图 9 所示。

从图 9 可知，三种算法得到的最优路径都是远离悬崖一侧，较为安全保守。在路径长度相同的情况下，Sarsa 算法得到的结果在路线的开头会接近悬崖，而 5 步 Sarsa 算法在路线末端靠近悬崖，都存在一定的风险。

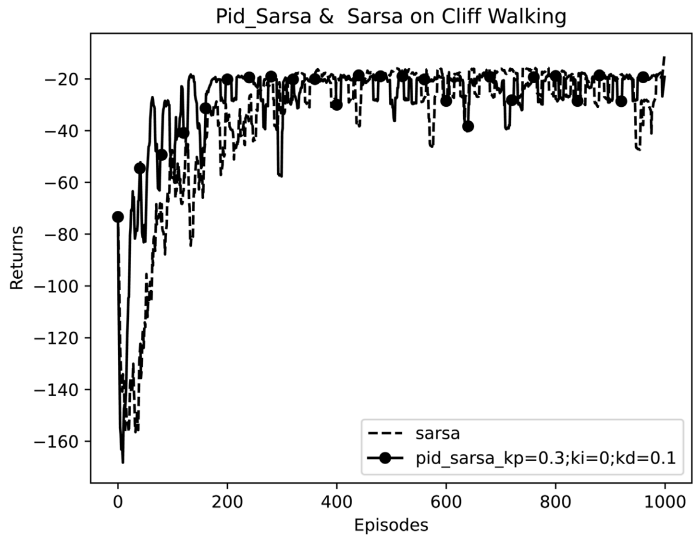

Figure 7. Comparison of Sarsa algorithm and Pid_Sarsa algorithm in cliff walking experiment
图 7. 悬崖寻路实验中 Sarsa 与 Pid_Sarsa 算法对比

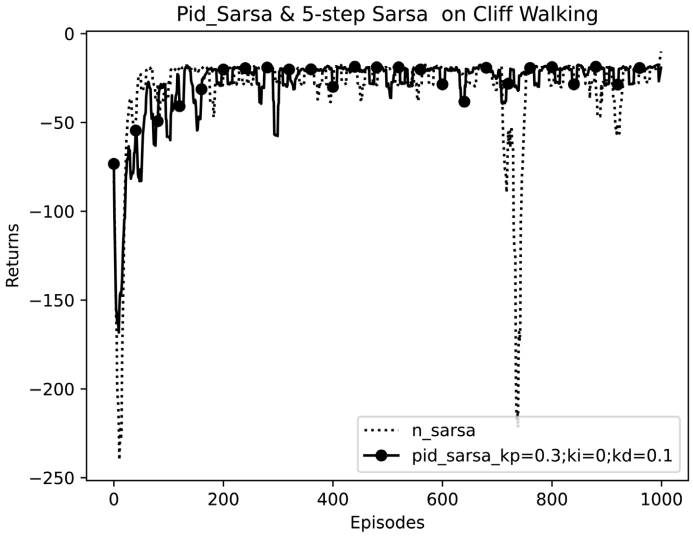

Figure 8. Comparison of 5_Sarsa algorithm and Pid_Sarsa algorithm in cliff walking experiment
图 8. 悬崖寻路实验中 5 步 Sarsa 与 Pid_Sarsa 算法对比

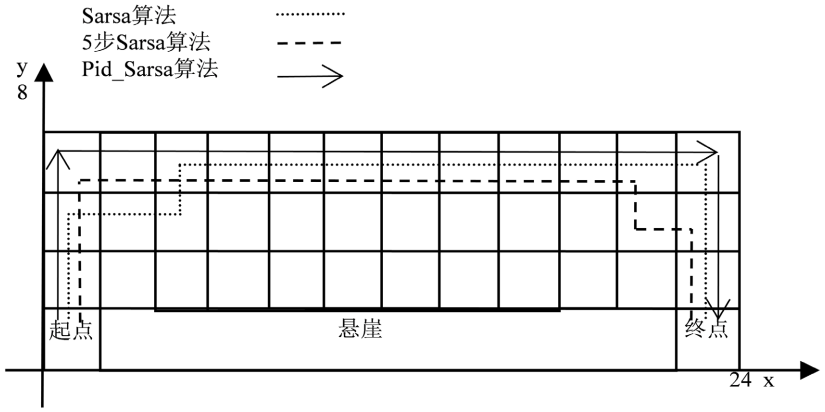

Figure 9. Comparison of Cliff Walking paths of three algorithms
图 9. 三种算法的悬崖寻路路径对比图

由于不同的算法会得到不同的路径，根据上文定义的三个安全性度量，分别得到不同路径的安全距离和安全程度，如表 1 所示。

Table 1. Cliff walking path comparison table of three algorithms
表 1. 三种算法的悬崖寻路路径对比表

算法	路径长度	安全长度	安全程度
Sarsa	18	80	95.24%
5 步 Sarsa	18	82	97.62%
Pid_Sarsa	18	84	100%

由表 1 可知，不同于其它两种经典算法，在路径长度相同的前提下，基于 PID 控制的 Pid_Sarsa 算法不仅收敛速度更快更稳，而且从路径的安全系数考量，得到的结果也是最为安全的，安全程度比 Sarsa 算法高 2.38%，比 5 步 Sarsa 算法高 4.76%。

5. 结语

本文在基于经典强化学习 Sarsa 算法，提出用 PID 控制对算法进行优化。根据悬崖寻路的系统环境，在对 Q 值的迭代更新过程中，引入 PID 控制中的比例部分和微分部分，通过对比例系数 K_p 和微分系数 K_d 的参数调优，将优化过后的 Pid_Sarsa 算法与经典的 Sarsa 算法和多步 Sarsa 算法分别进行比较。实验结果表明，在路径长度相同的前提下，经 PID 优化的 Pid_Sarsa 算法，收敛速度更快，并且也更加稳定，得到的悬崖寻路的路径也更加安全。适用于风险较大、安全性较高的路径规划问题，如车辆驾驶、灾难避险等。但在不同的环境中，算法涉及到 PID 控制系统的参数整定，会更加复杂，将更具针对性，也是接下来相关研究的重点。

基金项目

武汉科技大学教学研究项目(Yjg202116)。

参考文献

[1] 袁唯淋, 罗俊仁, 陆丽娜, 等. 智能博弈对抗方法: 博弈论与强化学习综合视角对比分析[J]. 计算机科学, 2022, 49(8): 191-204.

[2] Sutton, R. (1988) Learning to Predict by the Methods of Temporal Differences. *Machine Learning*, **3**, 9-44. https://doi.org/10.1007/BF00115009

[3] Tommi, J., Michael, I. and Satinder, P. (1994) Convergence of Stochastic Iterative Dynamic Programming Algorithms. *Advances in Neural Information Processing Systems*, Denver, 28 November-1 December 1994, 703-710.

[4] Robards, M., Sunehag, P., Sanner, S., *et al*. (2011) Sparse Kernel-SARSA(λ) with an Eligibility Trace. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, Berlin, 1-17. https://doi.org/10.1007/978-3-642-23808-6_1

[5] 朱海军. 基于核方法的近似强化学习的研究[D]: [硕士学位论文]. 苏州: 苏州大学,2017.

[6] Van, S., Van, H. and Whiteson, S. (2009) A Theoretical and Empirical Analysis of Expected Sarsa. *Proceedings of the IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, Nashville, 30 March-2 April 2009, 177-184.

[7] De, A., Hernandez-Garcia, J. and Holland, G. (2018) Multi-Step Reinforcement Learning: A Unifying Algorithm. *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, New Orleans, 2-7 February 2018, 2902-2909.

[8] 杨瑞. 多步强化学习算法的理论研究[D]: [硕士学位论文]. 天津: 天津大学, 2018.

[9] 郭雷. 不确定性动态系统的估计、控制与博弈[J]. 中国科学: 信息科学, 2020, 50(9): 1327-1344.

[10] 王芳, 郭雷. 人机融合社会中的系统调控[J]. 系统工程理论与实践, 2020, 40(8): 1935-1944.

[11] 陈学松, 杨宜民. 基于执行器-评价器学习的自适应 PID 控制[J]. 控制理论与应用, 2011, 28(8): 1187-1192.

[12] 段友祥, 任辉, 孙歧峰, 闫亚男. 基于异步优势执行器评价器的自适应 PID 控制[J]. 计算机测量与控制, 2019, 27(2): 70-73+78.

[13] 程丽梅, 贾文川. 连续型强化学习与 PID 控制的应用对比分析: 以一阶倒立摆系统为例[J]. 工业控制计算机, 2021, 34(10): 20-22.

[14] 甄岩, 郝明瑞. 基于深度强化学习的智能 PID 控制方法研究[J]. 战术导弹技术, 2019(5): 37-43.

[15] 孙波, 张伟, 杨青, 辛晨. 继电自整定 PID 控制算法比较研究[J]. 信息技术与信息化, 2021(2): 42-43+46.

[16] 李玉忍, 杨金孝, 张兴国, 齐蓉, 林辉. 基于迭代学习的 PID 控制研究[J]. 计算机工程与科学, 2007(4): 98-100.

[17] 吴少波, 杨薛钰. 基于 Sarsa 算法的交通信号灯控制方法[J]. 信息与电脑(理论版), 2021, 33(6): 49-51.

[18] 刘全, 翟建伟, 章宗长, 钟珊, 周倩, 章鹏, 徐进. 深度强化学习综述[J]. 计算机学报, 2018, 41(1): 1-27.

[19] 李必文. 线性广义系统的 P 型、PD 型和 PID 型迭代学习控制[J]. 数学杂志, 2008(6): 667-672.

[20] 代明光, 齐蓉, 李兵强, 赵逸云. 具有自适应非线性增益的开环 PD 型迭代学习控制[J]. 系统工程与电子技术, 2020, 42(3): 660-666.