DOI:10.19651/j.cnki.emt.2211649

基于深度强化学习哈里斯鹰算法的路径规划*

曾宁坤[1] 胡 朋[2] 梁竹关[1] 丁洪伟[1] 杨志军[1,3]
(1.云南大学信息学院 昆明 650000;2.优备科技股份有限公司 昆明 650000;
3.云南师范大学教育部民族教育信息化重点实验室 昆明 650500)

摘 要:哈里斯鹰算法存在容易早熟、陷入局部最优陷阱、稳定性较差等问题。为了提升算法性能,本文提出了一种利用深度确定性策略梯度算法(DDPG)改进的哈里斯鹰算法。该改进将深度强化学习和启发式算法结合,利用深度确定性策略梯度算法训练神经网络,再通过神经网络动态地生成哈里斯鹰算法关键参数,平衡算法全局搜索和局部搜索,并赋予算法后期跳出局部最优陷阱的能力。通过函数优化和路径规划对比实验,实验结果表明,DDPGHHO 算法具有一定的泛化性和优秀的稳定性,且在不同环境下均能够搜索到更优路径。

关键词:路径规划;深度确定性策略梯度算法;哈里斯鹰算法;深度强化学习
中图分类号:TP242.6 **文献标识码**:A **国家标准学科分类代码**:510.8050

Path planning based on deep reinforcement learning Harris Hawks algorithm

Zeng Ningkun[1] Hu Peng[2] Liang Zhuguan[1] Ding Hongwei[1] Yang Zhijun[1,3]
(1. School of Information, Yunnan University, Kunming 650000, China;
2. Youbei Technology Co., Ltd., Kunming 650000, China; 3. Key Laboratory of Education Informatization for Nationalities of Ministry of Education, Yunnan Normal University, Kunming 650500, China)

Abstract: Harris Hawk algorithm has problems such as easy precocious puberty, falling into local optimal traps, and poor stability. In order to improve the performance of the algorithm, this paper proposes an improved Harris Hawk algorithm using deep deterministic policy gradient (DDPG). DDPGHHO combines deep reinforcement learning with heuristic algorithm, trains neural network by using deep deterministic policy gradient, dynamically generates key parameters of HHO through neural network, balances global search and local search, and endows the algorithm with the ability to jump out of local optimal traps in the later period. Through the comparative experiments of function optimization and path planning, the results show that the DDPGHHO has certain generalization and excellent stability, and can search the better path in different environments.

Keywords: path planning; deep deterministic policy gradient; Harris Hawks algorithm; deep reinforcement learning

0 引 言

机器人的路径规划目的是找到一条到达目标点且不会触碰到障碍物的最优路径,包括全局路径规划和局部路径规划两部分,其中全局路径规划通常是基于已知环境的地图或者感知信息来生成一条最优路径[1]。

根据全局路径规划的特性,国内外很多学者将启发式算法应用在路径规划领域。启发式算法是学者们受到自然界中个体或者群体们生活习性的启迪创造的,例如粒子群算法(particle swarm optimization, PSO),鲸鱼算法(whale optimization algorithm, WOA)[2],哈里斯鹰算法(Harris Hawk algorithm, HHO)[3]等,它们具有原理易懂、参数较少、收敛速度较快等优点。但启发式算法也普遍存在着容易早熟、陷入局部最优值、稳定性差等缺点,学者们在路径规划领域应用启发式算法往往都会进行各种改进。例如文献[4]中将协同进化框架引入到人工蜂群算法中,克服了算法的维数依赖性,提高了算法收敛速度。文献[5]中采用基于余弦规律的收敛因子平衡算法的全局搜索和局部搜索,同时利用比例权重更新种群个体位置,加快算法收敛速度。文献[6]将改进灰狼算法全局路径规划的节点作为人工势

收稿日期:2022-10-10
* 基金项目:国家自然科学基金(61461053)项目资助

场法算法的临时目标点，并改进临时目标点为临时边界，加强了算法精度。文献[7]在蚁群算法的基础上，采用多部搜索策略代替单步搜索策略，提高算法性能。文献[8]在粒子群算法中引入人类社会民主规则的概念和贪婪策略，同时在迭代中加入正弦和余弦算法，提升了算法寻优性能。文献[9]在哈里斯鹰算法中引入精英等级制度策略，利用优势种群增强种群多样性，同时，加入 Tent 混沌映射调整参数，后期使用高斯游走策略和随机游走策略防止算法陷入局部最优。

这些改进方法具有一个共同的特点，即通过人工对算法的超参数进行设置，只有经过反复尝试才能得到一个相对较优的参数组[10]。近些年来，机器学习兴起，神经网络在函数拟合方面表现出优异的能力。深度强化学习结合了神经网络和强化学习优势[11]，不需要先验知识，通过与环境交互试错就能获得最优策略，而启发式算法往往根据参数值的大小选择各种策略更新种群的位置，使得依靠强化学习训练神经网络为启发式算法设定参数，改进算法性能成为可能[10]。2016 年谷歌人工智能团队提出了深度确定性策略梯度算法[12]（deep deterministic policy gradient，DDPG），DDPG 将深度 Q-learning 算法[13]和确定性梯度算法[14]的优点结合，在连续控制领域表现出了突出的能力，因为算法的参数大多是一个连续数值区间内的随机数，DDPG 具有巨大的潜力。

本文提出了一种基于 DDPG 优化的哈里斯鹰算法，将强化学习和启发式算法结合在一起，提前利用深度强化学习训练动作神经网络，再利用动作神经网络生成启发式算法所需参数求解函数优化问题和机器人路径规划问题。通过和其他算法进行实验对比，结果证明了 DDPGHHO 具有更强的稳定性和更高的寻优精度。

1　算法的基本原理

1.1　哈里斯鹰算法

哈里斯鹰是一种北美猛禽，受到哈里斯鹰捕猎兔子行为的启发，Heidari 等[3]在 2019 年提出了哈里斯鹰优化算法，被广泛应用于解决各种工程优化问题[15-16]。算法根据哈里斯鹰捕食的阶段性行为分为搜索阶段和围捕阶段。

1）搜索阶段

在这个阶段，哈里斯鹰根据概率 q 选取两种策略。当 $q<0.5$，哈里斯鹰靠近发现猎物的同伴，当 $q>0.5$，哈里斯鹰选择到一个随机的位置搜寻。此时哈里斯鹰的位置更新公式如下：

$$X(t+1)=\begin{cases}X_{rand}(t)-r_1\mid X_{rand}(t)-2r_2X(t)\mid, & q\geqslant 0.5\\ X_{rabbit}(t)-X_m(t)-r_3(LB-r_4(UB-LB)), & q<0.5\end{cases} \tag{1}$$

式中：$\boldsymbol{X}(t+1)$ 是鹰在下一次迭代后的位置向量，$\boldsymbol{X}(t)$ 是当前的位置向量，X_{rand} 是随机选中的一只鹰位置，X_{rabbit} 是猎物的位置，X_m 是当前种群所有个体位置的平均值。r_1,r_2,r_3,r_4 和 q 是(0,1)内的随机数，UB 和 LB 是解的上下限。

2）逃逸能量

哈里斯鹰算法根据猎物的逃逸能量 E 决定进行搜索还是围捕，当 $E\geqslant 1$ 时进行搜索，当 $E<1$ 时进行围捕，E 的更新公式如下：

$$E=2E_0\left(1-\frac{t}{T}\right) \tag{2}$$

式中：E_0 是一个(-1,1)以内的随机数，t 是当前迭代周期，T 是最大迭代周期。

3）搜索阶段

这个阶段，哈里斯鹰根据概率 r 和逃逸能量 E 采取 4 种策略。

(1) 软包围。当 $r\geqslant 0.5$，$E\geqslant 0.5$ 时，位置更新公式如下：

$$X(t+1)=\Delta X(t)-E\mid JX_{rabbit}(t)-X(t)\mid \tag{3}$$

$$\Delta X(t)=X_{rabbit}(t)-X(t) \tag{4}$$

$$J=2(1-r_5) \tag{5}$$

式中：$\Delta X(t)$ 是第 t 次迭代时个体当前位置和猎物所在位置的差，r_5 是(0,1)内的随机数，J 模拟的是猎物在被捕猎时随机跳跃的能量。

(2)硬包围。当 $r\geqslant 0.5$，$E<0.5$ 时，位置更新公式如下：

$$X(t+1)=X_{rabbit}(t)-E\mid\Delta X(t)\mid \tag{6}$$

(3) 软包围并进行快速下降。当 $r<0.5$，$E\geqslant 0.5$ 时，位置更新公式如下：

$$X(t+1)=\begin{cases}Y, & F(Y)<F(X(t))\\ Z, & F(Z)<F(X(t))\end{cases} \tag{7}$$

其中，Y 和 Z 通过以下两个公式计算：

$$\begin{aligned}&Y=X_{rabbit}(t)-E\mid JX_{rabbit}(t)-X_m(t)\mid\\ &Z=Y+S\times LF(D)\end{aligned} \tag{8}$$

式中：D 是算法解决问题的维度，LF 是 levy 飞行函数，LF 具体公式如式(9)所示。

$$LF(x)=0.01\times\frac{u\times\sigma}{\mid v\mid^{\frac{1}{\beta}}},\sigma=\left(\frac{\Gamma(1+\beta)\times\sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right)\times\beta\times 2^{\left(\frac{\beta-1}{2}\right)}}\right)^{\frac{1}{\beta}} \tag{9}$$

(4) 硬包围并进行快速下降。当 $r<0.5$，$E<0.5$ 时，以式(7)和(8)更新位置，但此时式(8)中的参数 Y 根据式(10)更新：

$$Y=X_{rabbit}(t)-E\mid JX_{rabbit}(t)-X_m(t)\mid \tag{10}$$

1.2　深度确定性策略梯度算法

在强化学习中，智能体与环境进行交互，智能体每进行一个动作，就会进入下一个状态，环境以此给出奖励。整个交互的过程可以被描述为一个马尔可夫决策过程，强化学习的目的就是通过训练智能体的行动方式，寻找一个最优

行为策略使智能体在环境中获得最大奖励,数学描述为式(11)。

$$R_i = \sum_{t=i}^{T} \gamma^{(t-i)} r(s_t, a_t) \tag{11}$$

式中:R_i 为智能体在整个过程中获得的总奖励,S_t 为 t 时刻的状态,a_t 为 t 时刻智能体的动作,γ 为折扣因子,长期收益会根据折扣因子衰减。

强化学习通常基于策略尺度或者价值尺度来评估每个状态和动作的价值并进行训练,定义了状态价值函数和动作价值函数两种价值函数。

DDPG 算法属于深度强化学习,利用神经网络来拟合强化学习的价值函数。DDPG 采用了演员-评论家(actor-critic)形式的神经网络,同时具有基于策略尺度和基于价值尺度的两种神经网络,分别为 $Q(st, at \mid \theta^Q)$ 和 $\mu(st \mid \theta^\mu), \theta^Q$ 和 θ^μ 代表网络参数。Actor 网络 $\mu(st \mid \theta^\mu)$ 通过输入智能体当前的状态根据确定性的策略输出一个最优动作,critic 网络 $Q(st, at \mid \theta^Q)$ 根据 Actor 网络输出的动作和智能体的状态评价动作的价值,Actor 网络再根据评价更新自己的策略,由此循环。其中 critic 网络通过最小化当前网络损失函数进行更新,损失函数公式为:

$$L = \frac{1}{N}\sum_{t=1}^{N}(r_t + \gamma Q'(s_{t+1}, \mu'(s_{t+1} \mid \theta^{\mu'}) \mid \theta^{Q'}) - Q(s_t, a_t \mid \theta^Q))^2 \tag{12}$$

式中:神经网络 Q' 为 critic 网络的目标网络,它的网络参数 θ' 定期从 θ 计算得到。更新 critic 网络后再通过神经网络的梯度反向传播来为更新 actor 网络的参数 θ^Q,更新方向为:

$$\nabla_{\theta^\mu} \approx \frac{1}{N}\sum_{t=1}^{N} \nabla_a Q(s, a \mid \theta^Q) \mid_{s=s_t, a=\mu(s_t)} \nabla_{\theta^\mu} \mu(s \mid \theta^\mu) \mid s_t \tag{13}$$

2 基于深度确定性策略梯度的哈里斯鹰算法

在标准的哈里斯鹰算法中,当算法处于搜索阶段,根据式(1),算法会根据概率 q 决定扩大全局搜索范围还是靠近当前最优值。同时,根据式(2),算法会根据逃逸能量来确定是进行全局搜索还是局部搜索,当 $E<1$ 时,种群只会采取局部搜索策略,而如图 1 所示,逃逸能量 E 在不断迭代之后,会不断减小,并且在一定迭代次数之后 E 总是小于 1 的,使算法失去全局探索能力,这也导致了算法后期有可能陷入局部最优陷阱。

本文利用深度强化学习动态调整哈里斯鹰算法参数,考虑到哈里斯鹰算法的参数都是在一个连续的数值区间内选取,例如(0,1),所以该算法改进选取的是 DDPG 算法,DDPG 算法在连续动作空间内有着优秀的表现。

DDPG 算法的目的是训练一个智能体,该智能体的状态和哈里斯鹰算法的适应度值直接关联,根据适应度的变化是否符合本文的需要给出奖励,训练出合适的神经网络。

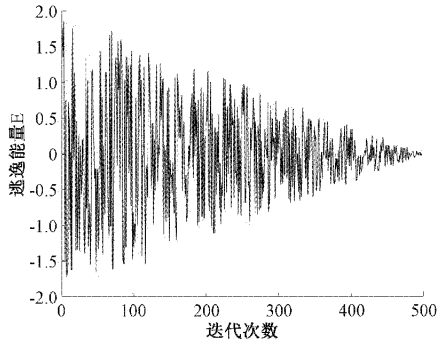

图 1 逃逸能量变化曲线

训练后的智能体在得到哈里斯鹰算法的状态后根据强化学习的动作神经网络输出对应的最优动作,动作经过转换后生成哈里斯鹰算法对应的参数,即概率 q 和逃逸能量 E。

2.1 DDPG 参数设定

使用 DDPG 训练需要考虑到训练目标的状态 s、动作 a、奖励 r 的定义。

DDPG 算法的状态空间设定为 5 维,前 4 个维度代表的是哈里斯鹰算法在当前迭代和过去 3 个迭代周期的平均适应度值,第 5 维代表的是当前的迭代进程。具体公式如下:

$$s_i = \begin{cases} \dfrac{fitnessm_{t+i-4}}{5\ 000}, & 1 \leqslant i \leqslant 4 \\ t/T, & i = 5 \end{cases} \tag{14}$$

其中,$fitnessmt$ 代表第 t 轮迭代哈里斯鹰算法的平均适应度值,整个训练过程中适应度最大值未超过 100,将适应度值除以 5 000 的目的是对数据进行归一化处理,加快训练速度。

本次改进需要的是算法能够动态生成概率 q 和逃逸能量 E,所以算法的动作空间设定为二维空间,记为[$a1$,$a2$],由于神经网络的输出值范围为(−1,1),根据式(1)和(2)规定的参数取值范围将输出动作值进行变换到合适的大小。

奖励设置中,为了让算法进行更多的有效探索,并且更快的收敛到最优值,将奖励和适应度的变化直接挂钩,鼓励算法更快的改变,具体公式如下:

$$r_i = \begin{cases} 9 \times (1 - \dfrac{bestRabbit}{bestRabbit'}), & bestRabbit < bestRabbit' \\ 0, & bestRabbit \geqslant bestRabbit' \end{cases} \tag{15}$$

式中:$bestRabbit'$ 是历史最佳的适应度值,$bestRabbit$ 是当前迭代周期内的最佳适应度值。根据公式,当前迭代周期最佳适应度比历史最佳适应度小的时候,智能体就能获得正向奖励,且小的越多奖励越大,这能鼓励智能体朝着加速收敛的方向前进。同时考虑算法在后期逐渐收敛,适应度变化较小,正向收益不足,此时采用负收益算法可能会选择停滞以防止负收益,导致算法收敛速度变慢。所以适应度未变优的情况下奖励设置为 0 而不是负收益。

2.2　DDPG 参数设定

DDPG 训练神经网络时需要智能体执行和环境进行交互，执行动作，转到新状态，得到奖励这一流程。因此使用 DDPG 训练 HHO 的参数控制器需要加入 HHO 的算法优化过程，具体的训练流程如图 2 所示。

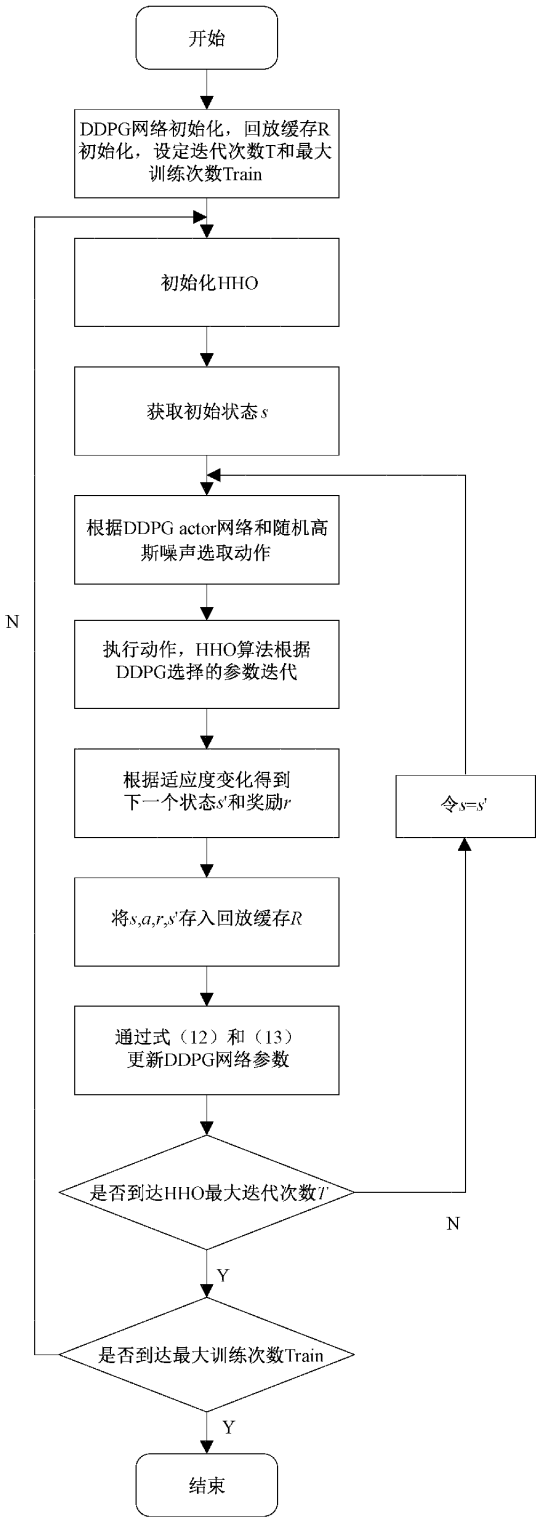

图 2　DDPGHHO 训练流程

训练 actor 神经网络和 critic 神经网络所用的网络结构分别如表 1 和 2 所示。

表 1　actor 网络结构

层	输出维度	输入
输入层	5(状态向量)	
L0	128	输入层
L1	128	L0
输出层	2(动作向量)	L1

表 2　critic 网络结构

层	输出维度	输入
输入层	5,2(状态向量,动作向量)	
L0	128	输入层
L1	128	L0
输出层	1(动作价值)	L1

经过 50 次 epoch(每一次 epoch 都是一次完整的 HHO 迭代过程)后，价值函数逐渐收敛，如图 3 所示。这表明在最初的训练中，由于神经网络未能理解任务，选取的参数不够好，智能体得到的奖励值并不高。但在得到足够多的训练之后，DDPG 已经能够理解任务的需求，使 HHO 算法能够更有效率地往本文需要的方向迭代，智能体所获得的奖励也不断提高。

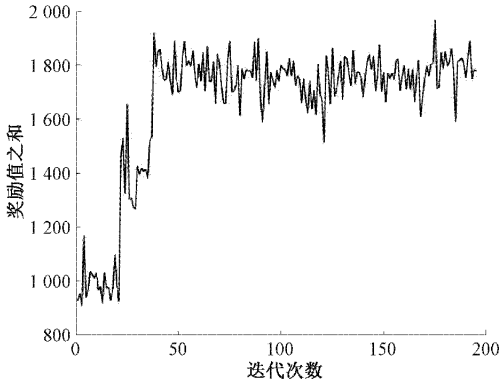

图 3　DDPG 训练奖励曲线

2.3　DDPGHHO 优化过程

1) 初始化种群 X_i ($i=1,2,\cdots,N$)，当前迭代次数 $t=0$，最大迭代次数 $T=500$

2) 根据适应度函数计算每一个代理的初始适应度值

3) 找出初始适应度值最符合期望的代理，并将其标记 X_{rabbit}

4) while ($t<T$)

5) for (每一个代理) do:

6) 将当前状态输入 DDPG 训练好的神经网络，神经网络根据状态输出动作

7) 根据动作网络输出的动作得到概率 q 和逃逸能量 E;

8) IF ($|E| \geqslant 1$), 根据式(1)更新位置

9) IF ($|E| < 1$)):

if $r \geqslant 0.5$ and $|E| \geqslant 0.5$) then

根据式(3)~(5)更新位置

else if ($r \geqslant 0.5$ and $|E| < 0.5$) then

根据式(6)更新位置

else if ($r < 0.5$ and $|E| \geqslant 0.5$) then

根据式(7)~(9)更新位置

else if ($r < 0.5$ and $|E| < 0.5$) then

根据式(7)~(10)更新位置

10) 根据适应度函数再次计算每一个代理的适应度值,选取最优适应度值更新 X_{rabbit}

11) $t = t + 1$

12) RETURN X_{rabbit}

3 实验仿真对比

3.1 函数对比实验

在进行路径规划实验之前,通过 8 组函数实验,将 DDPGHHO 算法同自适应粒子群算法[17](APSO),改进收敛因子和比例权重的灰狼优化算法[5](CGWO),经典哈里斯鹰算法(HHO)以及鲸鱼算法(WOA)进行对比。这组实验的目的是考虑到训练动作网络需要一定的时间成本,为了证明通过单一适应度函数训练的 DDPGHHO 动作网络在运算其他适应度函数时具有一定的泛化性,即通过训练解决特定函数问题的强化学习模型也能运用在解决其他函数问题上,节省训练时间成本。实验条件是训练模型时 HHO 算法所选用的函数问题是 Sphere 函数,但进行对比实验会用到包括训练函数在内的 8 个常用的基准测试函数(F1~F8)如表 3 所示。对于所有的算法和测试函数,最大迭代次数都设置为 500,种群数设置为 50,重复计算 30 次,通过每个算法的最优值,平均值以及标准差来评价算法的性能。

表 4 是 5 种算法在 8 个测试函数的实验结果对比,加粗项为最优结果。从实验结果中可以看出,DDPGHHO 除了训练动作网络用的 F1 函数之外,在其他 F2~F7 函数中也能够搜索到最优值,只有在 F8 函数中排第 2,略次于

表 3 测试函数

名称	测试函数	公式(目标都是求最小值)	维度	取值范围
F1	Sphere	$F_1(x) = \sum_{i=1}^{n} x_i^2$	30	[−100,100]
F2	Schwefel 2.22	$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	[−10,10]
F3	Schwefel 1.2	$F_3(x) = \sum_{i-1}^{n} (\sum_{j-1}^{i} x_j)^2$	30	[−100,100]
F4	Schwefel 2.21	$F_4(x) = \max_i \{\lvert x_i \rvert, 1 \leqslant i \leqslant n\}$	30	[−100,100]
F5	Schwefel	$F_5(x) = -\sum_{i=1}^{n} (x_i \sin(\sqrt{\lvert x_i \rvert}))$	30	[−500,500]
F6	Rastrigin	$F_6(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$	30	[−500,500]
F7	Griewank	$F_7(x) = \frac{1}{4\,000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$	30	[−600,600]
F8	Penalized	$F_8(x) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{n} (x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	[−50,50]

APSO 算法。另一点是 DDPGHHO 算法在稳定性的表现十分优秀,除去训练用的 F1 函数标准差为 0 之外,其他 7 个函数的标准差都是 4 个算法中的最低值或者并列最低值。函数对比实验表明 DDPGHHO 对于不同适应度函数拥有泛化性和突出的稳定性。

3.2 路径规划实验

为了验证 DDPGHHO 算法在求解路径规划问题的有效性和能力,本文使用 MATLAB2022a 平台进行编程仿真。实验电脑配置为:处理器 AMD 5800H,32 G 内存,显卡 Nvidia RTX3060。

仿真程序通过栅格地图对现实场景进行建模,即通过对地图进行单元化分区,将地图看作一个二维矩阵,黑色方格代表障碍物,白色方格代表可通行路径,起点用圆点表示,终点用方块表示,小车看作一个质点。实验一共采

表 4　函数对比实验结果

函数		APSO	WOA	CGWO	HHO	DDPGHHO
F1	最优值	2.51×10^{8}	1.48×10^{85}	1.01×10^{147}	1.55×10^{119}	$\mathbf{5.29\times10^{-313}}$
	平均值	3.04×10^{-6}	3.35×10^{-73}	2.53×10^{-143}	1.31×10^{-98}	2.49×10^{-283}
	标准差	4.79×10^{-6}	1.73×10^{-72}	4.89×10^{-143}	7.04×10^{-98}	0.00×10^{0}
F2	最优值	5.27×10^{-12}	7.82×10^{-57}	9.42×10^{-76}	2.92×10^{-60}	$\mathbf{7.85\times10^{-91}}$
	平均值	9.14×10^{-6}	7.01×10^{-51}	1.49×10^{-74}	7.66×10^{-53}	2.99×10^{-79}
	标准差	3.22×10^{-5}	4.60×10^{-50}	2.33×10^{-74}	3.42×10^{-52}	1.53×10^{-78}
F3	最优值	9.46×10^{0}	6.57×10^{3}	2.71×10^{-111}	9.36×10^{-102}	$\mathbf{1.72\times10^{-166}}$
	平均值	2.86×10^{1}	2.77×10^{4}	2.53×10^{107}	7.84×10^{80}	1.56×10^{139}
	标准差	1.77×10^{1}	1.02×10^{4}	5.34×10^{-107}	4.20×10^{-79}	8.42×10^{-139}
F4	最优值	4.67×10^{-1}	8.09×10^{-2}	8.46×10^{-65}	6.88×10^{-59}	$\mathbf{1.63\times10^{-154}}$
	平均值	1.53×10^{0}	4.31×10^{1}	3.76×10^{64}	3.73×10^{53}	1.25×10^{144}
	标准差	6.05×10^{-1}	2.90×10^{1}	2.82×10^{-64}	1.23×10^{-52}	4.25×10^{-144}
F5	最优值	−11 706.84	−12 569.33	−5 146.169 6	−12 569.49	**−12 569.49**
	平均值	−9 468.64	−10 780.43	−4 437.698 6	−12 569.22	−12 569.17
	标准差	1.24×10^{3}	1.71×10^{3}	3.87×10^{3}	6.06×10^{-1}	5.11×10^{-1}
F6	最优值	1.99×10^{0}	0.00	0.00	0.00	**0.00**
	平均值	7.93×10^{0}	3.92×10^{-15}	0.00	0.00	0.00
	标准差	3.65×10^{0}	2.11×10^{11}	0.00	0.00	0.00
F7	最优值	0.00	0.00	0.00	0.00	**0.00**
	平均值	0.00	2.17×10^{2}	0.00	0.00	0.00
	标准差	0.00	6.12×10^{-2}	0.00	0.00	0.00
F8	最优值	$\mathbf{1.46\times10^{-20}}$	8.37×10^{-4}	1.01×10^{-1}	4.29×10^{-8}	9.49×10^{-11}
	平均值	3.57×10^{-3}	1.69×10^{-2}	3.39×10^{-1}	4.96×10^{-6}	1.84×10^{-6}
	标准差	1.92×10^{-2}	4.16×10^{-2}	1.71×10^{-1}	$5.292\,8\times10^{-6}$	$3.571\,71\times10^{-6}$

用了大小为 20×20 和 30×30 且复杂度不同的两张地图代表两个不同的环境。实验中每种算法独立进行 10 次求解，20×20 地图中各算法迭代次数设置为 300，种群数为 50，解的维度为 18，30×30 地图中迭代次数设置为 500，种群数为 100，解的维度为 28. 适应度函数设置为路径的长度，长度越短代表解越优秀，最后通过每种算法的最优值、平均值、最劣值、标准差来进行对比。

图 4 和 5 代表 5 种算法在两种不同环境下 10 次求解中寻得的最优路径，表 5 表明了每种算法解的最优值、平均值、标准差，都保存到小数点后两位。首先观察环境 1 路径规划图，在路径规划初始部分路径重合，路径后半段可以明显地看出，在环境 1 中，相较于另外两种算法，HHO 和 DDPGHHO 寻得的路径较短，且 DDPGHHO 寻得的路径是最短的。同时，通过表 5 观察平均值、最劣值和标准差，在障碍较为简单的环境 1 中，尽管 HHO 算法最优值和改进后的 DDPGHHO 算法差距不大，但是原始的 HHO 各项对比指标比起优化后的 DDPGHHO 算法大的较多，尤其 DDPGHHO 标准差的表现具有明显的优势。

其次，根据图 5 所示，在环境较为复杂的环境 2 中，HHO 明显陷入了局部最优陷阱，并不能找到最优路径，而

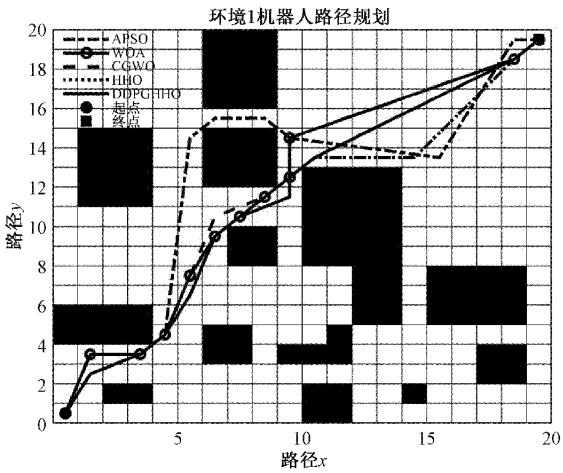

图 4　环境 1 路径规划

改进后的算法找到了 4 种算法中最好的路径，并且平均值等各项指标也是最优秀的。这表明了改进后的 DDPGHHO 通过利用 DDPG 算法提前训练的动作网络，在 HHO 算法进行迭代时动态地为 HHO 算法提供必要的参数，使算法在迭代后期依然拥有全局搜索的能力，克服

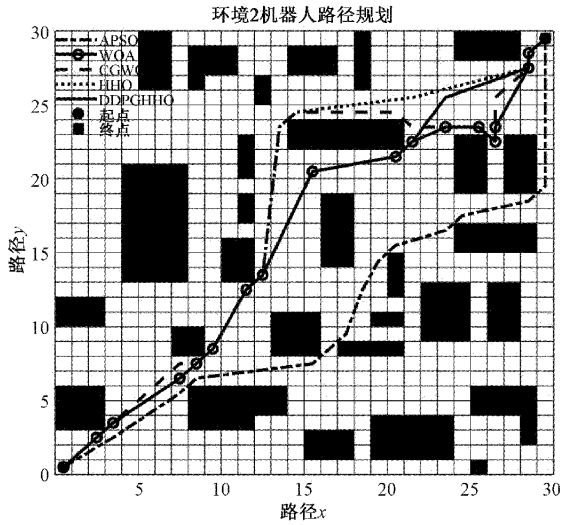

图 5 环境 2 路径规划

表 5 路径规划对比实验结果

环境名		APSO	WOA	CGWO	HHO	DDPGHHO
环境 1	最优值	30.2	29.05	29.09	28.75	**28.19**
	最劣值	33.53	32.49	31.59	33.45	30.77
	平均值	31.3	30.84	30.80	30.49	29.09
	标准差	1.12	0.94	0.68	1.42	0.67
环境 2	最优值	48.13	45.62	47.31	46.18	**43.49**
	最劣值	59.51	56.82	57.84	54.84	51.78
	平均值	55.6	52.24	53.76	49.94	48.50
	标准差	3.18	3.25	3.37	3.40	2.06

了 HHO 算法在进行求解时容易陷入局部最优的陷阱,稳定性较差的缺点。

为了验证 DDPGHHO 的收敛速度改进,通过图 6 和 7 观察适应度迭代曲线。在环境 1 中,虽然 PSO 和 WOA 更快收敛到了它们的最优值,但根据适应度大小和路径规划图,明显能看出它们陷入了局部最优陷阱。DDPGHHO 和 HHO 收敛到了更优值,而且 DDPGHHO 明显比 HHO

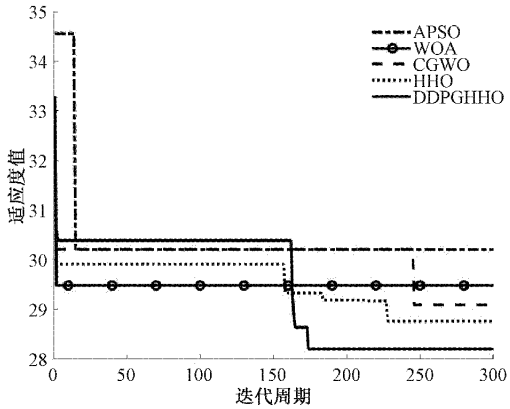

图 6 环境 1 适应度迭代曲线

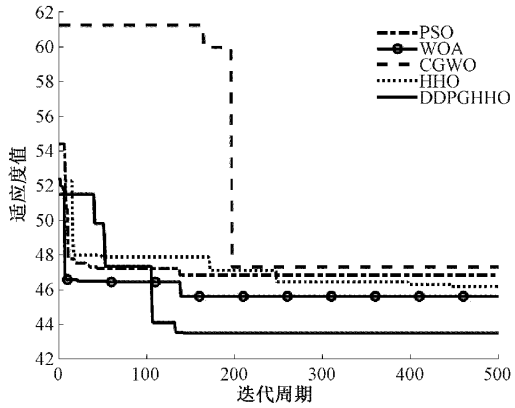

图 7 环境 2 适应度迭代曲线

更快地收敛到了最优值,这个效果在图 7 环境 2 的迭代曲线中表现得更加明显,在环境更为复杂的情况下,改进后的 DDPGHHO 不但收敛到了最优值,而且是四种算法种最快的。实验表明了通过式(15)的奖励设置,DDPG 训练出来的神经网络提供给 HHO 算法的参数会鼓励算法朝着更加快速收敛的方向前进,通过强化学习使算法加速收敛是具有一定效果的。

4 结 论

为了克服哈里斯鹰算法在解决路径规划问题时容易陷入局部最优陷阱,且稳定性较差的缺点,本文提出了一种基于深度确定性策略性梯度算法改进的哈里斯鹰算法(DDPGHHO),利用深度强化学习训练神经网络,再通过神经网络动态产生哈里斯鹰算法参数解决优化问题。通过 8 组函数对比实验以及两个不同环境下路径规划对比实验,实验结果证明了 DDPGHHO 算法具有一定的泛化性,且相较于原始的 HHO 算法拥有更快的收敛速度,更高的精度以及更优秀的稳定性,提高了算法的性能。

参考文献

[1] MAC T T, COPOT C, TRAN D T, et al. Heuristic approaches in robot path planning: A survey[J]. Robotics and Autonomous Systems, 2016, 86: 13-28.

[2] MIRJALILI S, LEWIS A. The whale optimization algorithm[J]. Advances in Engineering Software, 2016, 95: 51-67.

[3] HEIDARI A A, MIRJALILI S, FARIS H, et al. Harris Hawks optimization: Algorithm and applications[J]. Future Generation Computer Systems, 2019, 97: 849-872.

[4] XU F, LI H, PUN C M, et al. A new global best guided artificial bee colony algorithm with application in robot path planning[J]. Applied Soft Computing, 2020, 88: 106037.

[5] 王秋萍,王梦娜,王晓峰. 改进收敛因子和比例权重的

灰狼优化算法[J]. 计算机工程与应用,2019,55(21),55(21):60-65,98.
[6] 音凌一,向凤红. 融合改进灰狼优化算法和人工势场法的路径规划[J]. 电子测量技术,2022,45(3):43-53.
[7] XUE T, LI L, SHUANG L, et al. Path planning of mobile robot based on improved ant colony algorithm for logistics [J]. Mathematical Biosciences and Engineering, 2021, 18(4): 3034-3045.
[8] PAIKRAY H K, DAS P K, PANDA S. Optimal multi-robot path planning using particle swarm optimization algorithm improved by sine and cosine algorithms [J]. Arabian Journal for Science and Engineering, 2021, 46(4): 3357-3381.
[9] 汤安迪,韩统,徐登武,等. 混沌精英哈里斯鹰优化算法[J]. 计算机应用,2021,41(8): 2265-2272.
[10] 鲁华祥,尹世远,龚国良,等. 基于深度确定性策略梯度的粒子群算法[J]. 电子科技大学学报,2021,50(2):199-206.
[11] LOPES D S J. NASCIMENTO C L. Gait synthesis of a hybrid legged robot using reinforcement learning [C]. Proc. of the Annual IEEE Systems Conference, 2015: 439-444.
[12] LILLICRAP T P, HUNT J J, PRITZEL A, et al. Continuous control with deep reinforcement learning[J]. Computer Science, 2016, 8(6): A187.
[13] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529-533.
[14] SILVER D, LEVER G, HEESS N, et al. Deterministic policy gradient algorithms [C]. International Conference on Machine Learning, PMLR, 2014: 387-395.
[15] PENCHALAIAH G, RAMYA R. Stability enhancement of power system based on HHO-TSA control scheme [J]. International Journal of Electronics, 2022, 109(12): 2108-2134.
[16] 苗振腾,王威,王俊鹏. 基于神经网络改进的 HHO 算法 AGV 路径规划[J]. 组合机床与自动化加工技术,2022(9):20-23,28.
[17] 敖永才,师奕兵,张伟. 自适应惯性权重的改进粒子群算法[J]. 电子科技大学学报,2014,43(6): 874-880.

作者简介

曾宁坤,硕士研究生,主要研究方向为启发式算法、强化学习、机器人路径规划。
E-mail:458637541@qq.com

胡朋,本科,主要研究方向为机器人路径规划。

梁竹关,副教授,主要研究方向为强化学习、机器人路径规划。

丁洪伟(通信作者),教授,博士生导师,主要研究方向为启发式算法、强化学习、机器人路径规划。
E-mail:nkz513@mail.ynu.edu.cn

杨志军,教授,博士生导师,主要研究方向为启发式算法。