UAV Path Planning Based on Maximum Entropy Safe Reinforcement Learning

Feisheng Yang*, Chengliang Fang, and Ruijie Liang

School of Automation, Northwestern Polytechnical University, Xi'an 710129, China

***Abstract*—In this paper, a soft actor critic (SAC) algorithm is adopted based on maximum entropy safe reinforcement learning for the constrained unmanned aerial vehicle (UAV) path planning problem. Firstly, a reinforcement learning environment for UAV path planning in an airspace scenario is developed, which is equipped with static obstacles, dynamic obstacles, and target locations. Secondly, this paper models the path planning problem for UAVs as a constrained Markov decision process, taking into account the UAV's dynamics equations and its obstacle avoidance requirements. Finally, the corresponding reward function is designed, the SAC algorithm is used to iteratively seek the near-optimal policy for this problem, and the algorithm is analyzed with Monte Carlo tests. The results show that the UAV trained by the SAC algorithm can autonomously perform path planning in dynamic uncertain environments, superior to other strategies.**

***Keywords*—unmanned aerial vehicle, dynamic path planning, safe reinforcement learning, constrained Markov decision process, soft actor critic algorithm**

基于最大熵安全强化学习的无人机路径规划

杨飞生*　方城亮　梁睿桀

西北工业大学自动化学院, 西安 710129, 中国

摘　要　本文针对受约束的无人机动态路径规划问题，基于最大熵安全强化学习采取了一种柔性执行评价(soft actor critic, SAC)算法。首先，开发了一个空域场景下无人机路径规划的强化学习环境，其中设有静态障碍物、动态障碍物与目标地点。其次，考虑无人机的动力学方程约束及其避障需求，从安全强化学习的角度将该路径规划问题建模为约束 Markov 决策过程。最后，设计了相应的奖励函数，使用 SAC 算法迭代寻求该问题的近似最优策略，并以 Monte Carlo 测试对算法进行了分析。仿真结果表明，SAC 算法训练后的无人机能够在动态不确定的环境中自主进行路径规划，并且比其他策略更具优越性。

关键词　无人机，动态路径规划，安全强化学习，约束 Markov 决策过程，soft actor critic 算法

1．引言

随着智能化决策与空中装备性能的快速发展，无人机(unmanned aerial vehicle，UAV)由于较好的机动性和灵活性在战场打击任务和灾后救援任务等方面具有良好的发展前景。所以 UAV 需要拥有较高的路径规划能力，以应对复杂多变的飞行环境[1]。

路径规划是一个 NP 优化难题[2]，经典算法[3, 4]在复杂环境下计算复杂度会急剧上升，甚至无法求解，这也被称为维数诅咒问题。而启发式算法可以避免维数诅咒的问题，其在路径规划中的应用也越来越广泛[5]。[6]提出了一种在图形处理单元上并行实现遗传算法来解决静态环境中的路径规划问题。[7]提出了一种新的混合粒子群优化算法，该算法通过合并模拟退火算法以解决复杂环境下路径规划问题。无论是经典算法还是启发式算法，以上这些算法更适合解决静态路径规划问题。然而，对于动态路径规划问题，全局环境信息是未知的，需要实时规划路径。

近年来，深度强化学习(deep reinforcement learning, DRL)[8-10]的出现为复杂环境下的动态路径规划问题提供了一种新的解决思路。尤其是 DRL 中带约束的安全问题逐

基金项目:国家自然科学基金(编号: 62073269); 重庆市自然科学基金面上项目(编号: CSTB2022NSCQ-MSX0963); 航空科学基金(编号: 2020Z034053002); 广东省基础与应用基础研究基金自然科学基金面上项目(批准号: 2023A1515011220); 陕西省重点研发计划(编号: 2022GY-244).

*通讯作者, E-mail: yangfeisheng@nwpu.edu.cn

渐成为一个新的研究热点，即安全强化学习[11, 12]。Garca等[13]从安全的角度系统地阐述了安全强化学习的定义。安全强化学习旨在通过设定约束条件来确保智能体在学习过程中不会采纳可能导致危险行为的策略，并在保证智能体行为安全性的同时，最大限度地提升其学习到的策略质量，以达到最优或接近最优的水平。这种学习过程强调在学习和部署阶段同时实现性能的合理性和对安全约束的遵守，确保了智能体在执行任务时能够遵循既定的安全准则，从而避免潜在的灾难性后果。

因此，本文针对动态不确定环境下带约束无人机路径规划问题，考虑到 UAV 的动力学方程约束及其避障需求，从安全强化学习的角度将该问题建模为约束 Markov 决策过程(constrained Markov decision process, CMDP)[14, 15]，采用基于最大熵 DRL 的柔性执行评价算法(soft actor critic, SAC)[16]进行训练。相较先前的工作，本文的主要贡献包括以下三个方面：

1) 开发了一个空域场景下无人机路径规划的强化学习可视化仿真环境，路径规划任务考虑了无人机动力学方程约束以及避障需求。

2) 使用了最大熵 DRL 的方法处理动态路径规划问题，通过大量 UAV 飞行状态数据和地图信息进行训练，摆脱了传统方法对准确模型的依赖。

3) 从安全强化学习的角度将该无人机路径规划问题建模为约束 Markov 决策过程，采用 SAC 算法训练智能体在满足安全约束的同时到达目标地点。

2. 问题描述

2.1 受约束的无人机路径规划任务描述

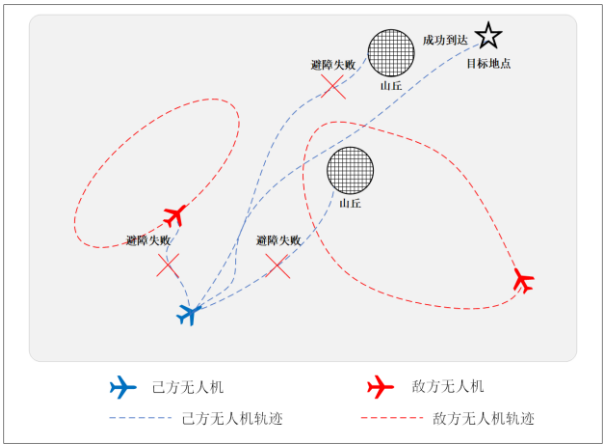

图 1 无人机路径规划任务示意图

本文的受约束无人机路径规划考虑到 UAV 动力学方程约束和飞行路径约束两部分。动力学方程约束指 UAV 的运动受到运动学方程的限制。飞行路径约束指 UAV 在飞行过程中面临对山丘等障碍物的规避，障碍物意味着 UAV 不能从其上方或内部经过，一旦发生碰撞则 UAV 损毁。己方无人机在此环境中进行路径规划的任务示意图如图 1 所示。

一架己方无人机（为图中的蓝色无人机）企图穿越战场环境避开其中的山丘和敌方无人机（为图中的红色无人机），到达右上方的目标地点（为图中的五角星）。己方无人机与敌方无人机是两种不同类型的无人机，二者拥有不同的机动性能。己方无人机有且仅有一架，它需要避开静态障碍物山丘和以各种方式运动的动态障碍物敌方无人机，并达到目标地点。由此，可以得到单无人机路径规划所需要完成的两类子任务：避开障碍物，避免发生碰撞检测从而导致己方无人机损毁；到达目标地点。

2.2 任务成败的定义

根据图 1 的无人机路径规划任务示意图，任务的成功与失败可以描述为如下。

a) 任务失败

若出现以下两种情况之一，则判定为任务失败。

情况 1（碰到障碍物，自身损毁）：如果己方无人机碰撞到山丘、敌方无人机等障碍，从而导致该己方无人机损毁，则被认为是任务失败。

情况 2（自身存活，但没在规定时间内到达目标地点）：如果己方无人机没有做出合理的路径规划决策，未能在规定时间内到达目标地点，则被认为是任务失败。

b) 任务成功

只要己方无人机能够在规定的时间内到达目标地点，并且不发生与山丘、敌方无人机等障碍物的碰撞，则被认为是任务成功，并且本次回合结束，环境重置。

2.3 评价指标

为对所提方法进行评估，本文设计任务完成率 J_{MCR}、飞行时间 J_T、飞行轨迹 J_S 和能量消耗 J_C 作为衡量算法优劣的评价指标。

根据 2.2 小节任务成败的定义，任务完成率可定义为：

$$J_{MCR} = N_C / N_T \tag{1}$$

其中 N_C 表示任务成功完成的回合数，N_T 表示进行 Monte Carlo 测试的总回合数。

飞行时间：

$$J_T = t_f \tag{2}$$

其中 t_f 为 UAV 的终端时刻。

飞行轨迹：

$$J_S = \int_0^{t_f} v dt \tag{3}$$

能量消耗：

$$J_C = \int_0^{t_f} (|u| + |\omega|)dt \tag{4}$$

3．基于安全强化学习的无人机路径规划

3.1 动态不确定无人机路径规划环境搭建

本文搭建的动态不确定环境在每次训练的回合里，所有 UAV 的初始状态——x 位置、y 位置、速度 v 和航向角 ψ 全部随机生成，目标地点、障碍物的坐标位置完全随机生成，并且是一个有界的空中领域。本文将 UAV 看成在同一高度下运动，UAV 的运动符合二维空间下的运动学方程。因此，本文的仿真实验做出了以下假设：

1) 无人机的运动只考虑平面中的运动，以二维形式做简化处理；2) 目标地点的位置信息已知，被认为由地面雷达测得并把数据告知己方无人机。

整个无人机路径规划仿真环境被定义为一个 700 像素*600 像素的二维平面(1)，该 700 像素=7000m，600 像素=6000m，环境的动画刷新频率 FPS 为 60。己方无人机是智能体，它的速度范围为 100m/s~200m/s，加速度控制输入范围为-6m/s²~6m/s²，角速度范围为-1.2rad/s~1.2rad/s；敌方无人机是动态障碍物，其速度范围为 100m/s~200m/s，加速度范围为-3m/s²~3m/s²，角速度范围为-0.6rad/s~0.6rad/s。己方无人机一旦与敌方无人机的距离过近或到达山丘所在的圆形区域内，则被认为发生碰撞检测，此时规划任务失败且本场回合结束；而己方无人机一旦到达目标地点所在的圆形区域内，即被认为成功到达目标地点且本场回合结束。

3.2 无人机路径规划 CMDP 建模

考虑到己方无人机受到动力学方程约束，以及飞行路径约束。现利用 CMDP 来描述无人机路径规划的决策模型，该模型可以定义为一个五元组 $<(S, A, R, P, C)>$。其中，S 是状态集合，A 是动作集合，R 是奖励集合，P 是状态转移概率，对智能体是不可知的，C 是约束集合。

a) 状态集 S 的设计

状态信息 s 是连续的量，可以表示为：

$$s = [x, y, v, \psi, x_G, y_G, O_{flag}] \tag{5}$$

其中 (x, y) 为己方无人机的位置坐标，v 是己方无人机的速度，ψ 是无人机航向角，(x_G, y_G) 是目标地点的位置坐标，O_{flag} 为障碍物标志位。即当己方无人机与任意一个障碍物之间的距离小于某一阈值时，则认为己方无人机附近有障碍物，此时障碍物标志位置为 1，否则置为 0。

(1) 环境：https://github.com/henbudidiao/UAV-path-planning

b) 动作集 A 的设计

己方无人机动作决策是通过选择合适的加速度和角速度执行 Δt 时间，以达到期望速度和期望航向角。UAV 的动作控制输入为一个二维向量，且动作是连续的量，可以表示为：

$$a = [u, \omega]^T \tag{6}$$

其中 u 表示己方无人机的加速度，ω 表示角速度。

c) 奖励集 R 的设计

首先，设计边界奖励 r_{edge}。当己方无人机到达边界附近则给予-2 的惩罚，否则不给任何奖惩。

$$r_{edge} = \begin{cases} -2, & \text{处于边界附近} \\ 0, & \text{否则} \end{cases} \tag{7}$$

其次，设计避障与目标奖励 r_{og}。当己方无人机与障碍物的距离小于阈值 D_2 时，此时认为己方无人机有潜在撞击障碍物的风险，给予-2 的惩罚；当己方无人机与障碍物的距离小于障碍物的半径时，此时己方无人机与障碍物相撞，无人机损毁，给予-500 的惩罚；当己方无人机与目标点的距离小于阈值 D_1 时，即认为到达目标点，给予一个 1000 的正向奖励，否则根据无人机与目标点的距离设计惩罚。

$$r_{og} = \begin{cases} 1000, & d_G < D_1 \\ -2, & d_O < D_2 \\ -500, & \text{发生碰撞} \\ k_1 d_G, & \text{否则} \end{cases} \tag{8}$$

其中 d_G 表示己方无人机与目标点的距离，d_O 表示己方无人机与障碍物的距离，k_1 为距离影响系数。

于是，可以得到己方无人机的总奖励函数

$$R = \omega_1 r_{edge} + \omega_2 r_{og} \tag{9}$$

其中 ω_1, ω_2 是每个部分的奖励权重。

d) 约束集 C 的设计

首先，无人机动力学方程约束：如下式(10)所示

$$\begin{cases} \dot{x} = v\cos\psi \\ \dot{y} = v\sin\psi \\ \dot{v} = u \\ \dot{\psi} = \omega \end{cases} s.t. \begin{cases} x_{\min} \le x \le x_{\max} \\ y_{\min} \le y \le y_{\max} \\ v_{\min} \le v \le v_{\max} \\ \psi_{\min} \le \psi \le \psi_{\max} \\ u_{\min} \le u \le u_{\max} \\ \omega_{\min} \le \omega \le \omega_{\max} \end{cases} \tag{10}$$

其中 x, y, v, ψ 为己方无人机的飞行状态量，$[u, \omega]^T$ 为控制向量，$x_{\min}, x_{\max}, y_{\min}, y_{\max}$ 为飞行边界范围，$v_{\min}, v_{\max}$ 为速度范围，$\psi_{\min}, \psi_{\max}$ 为航向角范围，$u_{\min}, u_{\max}$ 为加速

度范围，$\omega_{\min},\omega_{\max}$ 为角速度范围。

其次，无人机飞行路径约束：智能体在飞行过程中需要对山丘等障碍物进行规避，一旦发生碰撞则智能体损毁。为便于处理，所有障碍物的碰撞检测设为标准的圆形，可以描述为

$$d_O=\sqrt{\left(x-x_O\right)^2+\left(y-y_O\right)^2}\geq R_O \tag{11}$$

式中 (x_O,y_O) 为圆形障碍物的圆心坐标，R_O 为圆形障碍物的半径。

3.3 基于 SAC 算法的路径规划策略求解

SAC 算法是基于 actor-critic 架构的深度强化学习方法，可用于解决动作和状态都是连续量的情况。它以最大化 actor 网络的熵为目标，使用自适应温度系数 α 用于平衡 exploration(探索)和 exploitation(利用)之间的关系，拥有比其他 DRL 算法更强大的探索能力和鲁棒性[17]。基于 SAC 的 UAV 路径规划策略求解方法的训练过程如图 2 所示。

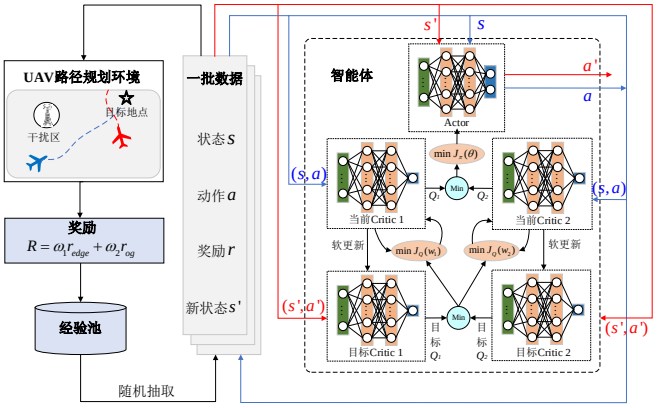

图 2　SAC 算法的训练过程

SAC 共有三个网络，分别是：参数为 θ，输入为状态 s，得到策略概率分布 $\pi(.|s_t)$ 的 actor 网络；参数为 w，输入为状态动作对 (s,a)，输出 Q 值的 critic 网络；参数为 α 的熵网络。

Actor 网络通过最小化以下 KL 散度进行更新：

$$J_\pi(\theta)=D_{KL}\left[(\pi(.|s_t)\|\exp(\frac{Q(s_t,.)}{\alpha}-\log Z(s_t))\right] \tag{12}$$

其中，$Z(s_t)$ 是用于归一化分布的函数，它只取决于状态，对策略网络的参数梯度没有影响。

对于 critic 策略网络的更新，最小化下式的损失函数：

$$J_Q(w)=\mathbb{E}_{(s_t,a_t,s_{t+1})\sim D}\left[\frac{1}{2}\left(Q(s_t,a_t)-\hat{Q}(s_{t+1},a_t)\right)^2\right] \tag{13}$$

其中 D 是经验池，负责收集训练数据，能打破样本之间的关联性。

SAC 的熵网络具有自动调整温度系数 α 的能力，熵网络在时刻 t 的损失函数为:

$$J(\alpha)=\mathbb{E}_{a_t\sim\pi_s s_t\sim D}\left[-\alpha\log\pi\left(a_t|\ s_t\right)-\alpha\mathcal{H}_0\right] \tag{14}$$

其中，$\mathcal{H}_0$ 是预定义的最小策略熵阈值（原始论文[16]推荐 $\mathcal{H}_0=-dim(A)$；例如，二维连续动作空间环境 $\mathcal{H}_0$ 为-2）。

基于SAC的受约束无人机路径规划策略训练算法步骤如算法 1 所示，采用高斯噪声作为策略网络的探索噪声。

算法 1　基于 SAC 的无人机路径规划训练算法

初始化 actor 网络、critic 网络和熵网络
初始化经验池 D 与高斯噪声
for episode=1，Maxepisode **do**
　环境初始化并得到当前时刻的状态 s
　for step=1，Maxstep **do**
　　使用高斯噪声进行动作选择，输入状态 s，得到动作 $a=[u,\omega]$
　　动作限幅至[-1,1]
　　根据动作 a，执行 step 函数，环境返回奖励 r，下一时刻状态 s'，回合结束标志位 done
　　存储样本 (s,a,r,s') 于 D
　　if 经验池 D 存满 **then**
　　　从 D 中随机采样一批数据，根据公式(12)执行 actor 网络的梯度下降
　　　根据公式(13)执行 critic 网络的梯度下降
　　　根据公式(14)执行熵网络的梯度下降
　　　软更新 critic 网络的权重参数，硬更新 actor 网络与熵网络的权重参数
　　end if
　　更新状态 $s\leftarrow s'$
　　if 回合结束标志位 done 为真 **then**
　　　break
　　end if
　end for
end for

4. 仿真验证

4.1 参数及版本信息

在搭建无人机路径规划仿真环境的过程中，为确保仿真环境的可重复性，给出提及的环境参数如下：$R_O=20$，$D_1=40$，$D_2=40$，$k_1=-0.001$，$\omega_1=1$，$\omega_2=1$。

为了演示算法的细节，表 1 列出了 SAC 算法的所有参数和超参数。需要注意的是，我们仅在前 50 回合加入了高斯噪声，这是为了让智能体能探索尽可能多的状态。一些主要安装包的版本信息：pygame 版本：2.1.2；gym 版本：0.19.0；pytorch 版本：1.10.0+cu113；numpy：1.23.1。

表 1　SAC 的参数和超参数设置

SAC 参数	SAC 超参数	actor 网络结构	critic 网络结构
actor 网络学习率 $1e^{-3}$	Maxestep 1000	输入节点 7	输入节点 7+2
critic 网络学习率 $3e^{-3}$	一批大小 128	隐藏层 (256,256)	隐藏层 (256,256)
熵网络学习率 $3e^{-4}$	Maxepisode 500	输出节点 2	输出节点 1
软更新率 $1e^{-2}$	经验池 20000	激活函数 relu, tanh	激活函数 relu
衰减因子 0.9	优化器 adam	权重初始(0,0.1)正态分布	权重初始(0,0.1)正态分布

4.2　训练过程

为了加速训练过程，我们将时间步长 Δt 设置为 1 以节省现实时间花费。整个仿真实验在具有 16Gb RAM(RTX-3050 显卡)的标准英特尔酷睿 i5-11260H 上运行。在智能体的训练阶段，将空域场景中山丘的数量设置为 1，敌方无人机的数量设置为 1。如果下述两种情况出现其一，情况 1：碰到障碍物，自身损毁；情况 2：自身存活，但没有在规定的 1000 个时间步内到达目标地点。那么这一次的训练会被重置，这被称为一个回合，并继续开启下一回合的训练，训练的总回合数设置为 500。我们累加计算了智能体每个回合内所取得的单步奖励，然后画出 500 个回合 SAC 算法训练的总奖励曲线图，并重复多次 500 个回合的训练以计算平均值与方差。

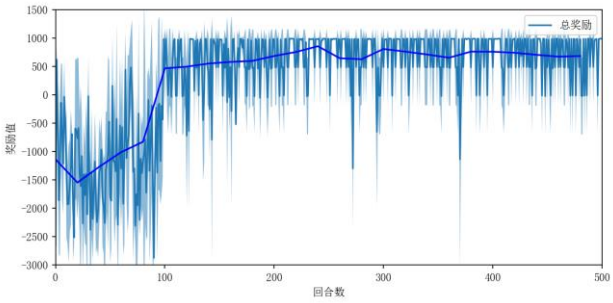

图 3　SAC 算法 500 回合训练下总奖励曲线图

图 3 所示的总奖励在 50 多回合时约为-1500，此时智能体在高斯噪声的作用下随机的探索环境。在 50 回合后，高斯噪声取消，奖励逐渐上升，最终总奖励收敛到了 1000 左右。尤其是在 100 回合之后，可以看到大部分总奖励都到达了 1000，这说明智能体已经找到了近似最优策略，可以到达目标地点，总奖励收敛。

4.3　测试过程与结果分析

a) 障碍物数量对 SAC 算法的影响分析

在测试阶段，为了清晰的观察 UAV 的飞行路径，我们将时间步长 Δt 设置为 0.1。我们使用 4.2 小节已经训练好的神经网络权重参数进行仿真测试，并将障碍物的数量扩展为山丘数量：3，敌机数量：2；山丘：3，敌机：3；山丘：3，敌机：4；山丘：3，敌机：5；分别进行 100 次 Monte Carlo 测试，测试的数据结果如表 2 所示。

表 2　不同障碍物数量下评价指标

障碍物数量	J_{MCR}	J_T	J_S	J_C
山丘:1, 敌机:1	89.0%	185.72	157.73	182.21
山丘:3, 敌机:2	69.0%	182.33	137.01	196.08
山丘:3, 敌机:3	66.0%	185.57	143.27	212.90
山丘:3, 敌机:4	62.0%	189.77	140.49	224.93
山丘:3, 敌机:5	59.0%	193.67	139.26	233.26

从表 2 可见，当山丘数量为 1，敌机数量为 1 时，己方无人机的路径规划任务完成率达到最高，为 89%。而随着障碍物数量的增加，无人机的路径规划任务完成率逐渐减小。这是因为障碍物数量越多的环境，无人机就越容易与障碍物相撞而导致任务失败。尤其是山丘数量为 3，敌机数量为 5 时，任务完成率仅为 59%。显然，障碍物数量对无人机路径规划有一定影响。山丘数量相同的情况下，无人机的平均飞行时间和平均能量消耗会随着敌机数量的增加而增大。当山丘数量为 3，敌机数量为 5 时，此时无人机所用的平均飞行时间最长，为 193.67，平均能量消耗最大，为 233.26。

b) 不同策略与 SAC 算法的对比分析

现在将 SAC 算法与随机策略、深度确定性策略梯度算法 (deep deterministic policy gradient, DDPG) [18]进行对比。在与其他策略的对比测试中，我们选取山丘数量为 3，敌机数量为 5，分别进行 100 次 Monte Carlo 测试。

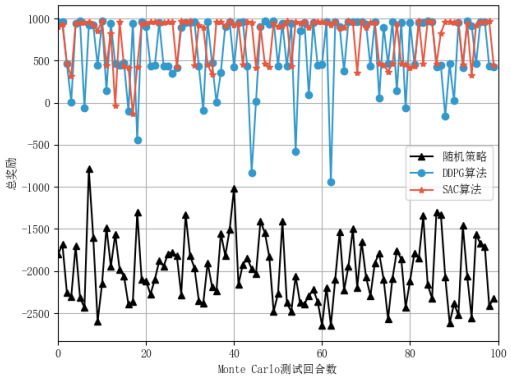

图 4　不同算法 Monte Carlo 测试下总奖励曲线图

如图 4 所示，描绘了 100 次 Monte Carlo 测试下每种算法所取得的总奖励，SAC 算法的总奖励值较高于 DDPG 算法，完全高于随机策略。表 3 可以看出，SAC 算法的任务

完成率总是最高的，平均飞行时间、飞行轨迹、能量消耗也是最少的；而随机策略的任务完成率总是最低的，平均飞行时间、飞行轨迹、能量消耗也是最多的。图 4 和表 3 说明了在相同的障碍物数量下，SAC 与 DDPG 都能使无人机学会自主路径规划，而 SAC 算法的效果要优于 DDPG 算法与随机策略。

表 3　SAC 算法与其他策略的对比

策略	J_{MCR}	J_T	J_S	J_C
随机策略	5.0%	788.36	593.59	789.49
DDPG 算法	41.0%	274.65	250.16	266.22
SAC 算法	**59.0%**	**193.67**	**139.26**	**233.26**

5．结论

本文采用安全强化学习以保证在满足约束的前提下优化策略，提出了一种基于最大熵深度强化学习的 SAC 算法来解决动态不确定环境下无人机路径规划问题。结果表明，训练后的无人机能够在受到动力学方程约束和安全避障约束的情形下到达目标地点，SAC 算法不仅可以适应于不同数量的障碍物，还比随机策略、DDPG 算法具有优越性。在未来的工作中，我们将考虑无人机之间的协同作用，研究多无人机协同路径规划问题。

参考文献

[1] 舒健生, 周于翔, 郑晓龙等. 基于深度强化学习的无人机实时航迹规划. *火力与指挥控制*, vol. 48, no.12, pp. 133-141, 2023.

[2] R.K. Dewangan, A. Shukla, and W.W. Godfrey, “Three dimensional path planning using grey wolf optimizer for UAVs, ” *Applied Intelligence*, vol. 49, pp. 2201-2217, 2019.

[3] P.E. Hart, N.J. Nilsson, B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100-107, 1968.

[4] A. Stentz, “Optimal and efficient path planning for partially-known environments,” in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, 1994, pp. 3310-3317.

[5] Z. Han, M. Chen, S. Shao, et al, “Path planning of unmanned autonomous helicopter based on hybrid satisficing decision-enhanced swarm intelligence algorithm,” *IEEE Transactions on Cognitive and Developmental Systems*, vol. 15, no. 3, pp. 1371-1385, 2023.

[6] V. Roberge, M. Tarbouchi, and G. Labonté, “Fast genetic algorithm path planner for fixed-wing military UAV using GPU,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 5, pp. 2105-2117, 2018.

[7] Z. Yu, Z. Si, X. Li, et al, “A novel hybrid particle swarm optimization algorithm for path planning of UAVs,” *IEEE Internet of Things Journal*, vol. 9, no. 22, pp. 22547-22558, 2022.

[8] R.S. Sutton, A.G. Barto, *Reinforcement Learning: An Introduction (Second Edition)*. Cambridge, Massachusetts: The MIT Press, 2018.

[9] V. Mnih, K. Kavukcuoglu, D. Silver, et al, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529-533, 2015.

[10] J. Seo, S. Kim, A. Jalalvand, et al, “Avoiding fusion plasma tearing instability with deep reinforcement learning,” *Nature*, vol. 626, pp. 746-751, 2024.

[11] S. Shalev-Shwartz, S. Shammah, and A. Shashua, “Safe, multi-agent, reinforcement learning for autonomous driving,” *arXiv preprint arXiv*:1610.03295, 2016.

[12] D. Ding, X. Wei, Z. Yang, et al, “Provably efficient generalized lagrangian policy optimization for safe multi-agent reinforcement learning,” in *Learning for Dynamics and Control Conference*, 2023, pp. 315-332.

[13] J. Garcıa, F. Fernández, “A comprehensive survey on safe reinforcement learning,” J*ournal of Machine Learning Research*, vol. 16, no. 1, pp. 1437-1480, 2015.

[14] E. Altman, *Constrained Markov Decision Processes*. New York: Routledge, 1999.

[15] Q. Yang, T.D. Simão, S.H. Tindemans, et al, “WCSAC: worst-case soft actor critic for safety-constrained reinforcement learning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, pp. 10639-10646.

[16] T. Haarnoja, A. Zhou, P. Abbeel, et al, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor,” in *International Conference on Machine Learning*, 2018, pp. 1861-1870.

[17] T. Haarnoja, H. Tang, P. Abbeel, et al, “Reinforcement learning with deep energy-based policies,” in *International Conference on Machine Learning*, 2017, pp. 1352-1361.

[18] T.P. Lillicrap, J.J. Hunt, A. Pritzel, et al, “Continuous control with deep reinforcement learning,” in *International Conference on Learning Representations*, 2016.