doi: 10.7690/bgzdh.2022.05.009

基于作战过程的岛礁兵力配置强化学习算法

肖　凡，乔勇军
（海军航空大学岸防兵学院，山东 烟台 264001）

摘要：针对岛礁守备作战过程中涉及的对海、对陆、对空 3 类武器，根据岛礁守备作战过程建立模型，提出一种动态动作空间方法。设置敌方武器装备、预设阵地、防守要地 3 类影响因素，利用不同的基于值函数的强化学习算法进行测试，通过测试能得到各武器装备最佳位置并判断预设阵地是否合理，通过比较可看出算法间各有优劣，适合的环境各不相同。结果表明：该方法能够运用于不同的环境，减少时空开销，提高岛礁守备决策的效率，有助于策略改进。

关键词：强化学习；值函数；岛礁守备；动态动作空间
中图分类号：TJ01　　**文献标志码：**A

Reinforcement Learning Algorithm for Forces Allocation on Islands and Reefs Based on Combat Process

Xiao Fan, Qiao Yongjun
(*School of Coast Guard*, *Naval Aviation University*, *Yantai* 264001, *China*)

Abstract: Aiming at 3 kinds of weapons involved in island and reef garrison combat process, namely sea weapons, land weapons and air weapons, a model is established according to the island and reef garrison combat process, and a method of dynamic action space is proposed. 3 kinds of influencing factors are set, including enemy weapons and equipment, preset positions, and defensive points, and different reinforcement learning algorithms based on value function are used for testing.Through the test, the best position of each weapon and equipment can be obtained and whether the preset position is reasonable or not can be judged, and the comparison shows that the algorithms have their own advantages and disadvantages, and the suitable environments are different. The results show that the method can be applied to different environments, reduce the time and space overhead, improve the efficiency of island and reef garrison decision-making, and help to improve the strategy.

Keywords: reinforcement learning; value function; island and reef defense; dynamic action space

0　引言

海洋、岛屿在国家主权、安全、军事、科技竞争中具有重要的战略地位，从作战形式上看主要由远程精确火力打击和两栖作战相结合[1]。随着现代战争信息化程度的不断提高，从战场信息的采集、传输到兵力的出动都比以往任何时候高效。提高信息系统的辅助决策效率，是信息化建设当中的重要问题之一[2]。传统的兵力配置方式已难以满足科学高效的决策需求，寻找一种新的更加适应快节奏和科学的兵力配置方法，成为一项重要课题。

近年来，随着计算能力的提高、算法的优化和数据的爆炸式增长，人工智能技术飞速发展，逐渐依靠其快速、准确、科学的优势进入了各行各业，在辅助决策方面，人工智能中的机器学习、神经网络、强化学习等技术已被广泛运用，以加快数据处理速度，提高人机交互水平，深刻地影响着军事领域[3]。随着信息化技术逐步装备部队，现代战争对战场响应速度和指挥水平提出了空前的要求，人工智能的高速计算和智能决策方面的优势渐渐让人们无法忽视。

在传统岛礁守备兵力配置问题中，由于西南沙岛礁周边自然环境复杂恶劣，交通不便，无法频繁开展演习，难以收集足够的高质量数据进行科学决策。为了改善传统决策方法时效性不高、客观性不强、准确性不够的问题，笔者利用强化学习方法，对战场环境及作战过程进行模拟[4]，最终找到合理或最佳的防守位置，同时找出当前防守阵地或防守模式的不足。

1　强化学习

包括蒙特卡罗树搜索（Monte Carlo tree search，

收稿日期：2022-01-10；**修回日期：**2022-02-18
作者简介：肖　凡（1997—），男，山东人，硕士，从事岛礁守备作战研究。E-mail: 287606177@qq.com。

MCTS)在内的强化学习(reinforcement learning，RL)算法在围棋等计算机游戏中取得了巨大成功[5]。以此为启发将其应用于兵力配置类问题，介绍强化学习的基础—马尔可夫决策过程以及 3 类常见的值函数类强化学习算法。

1.1 马尔可夫决策过程

马尔可夫(Markov)决策过程是强化学习的理论基础，是对智能体和环境交互的一个基本数学模型，其适用的系统应当具有 3 个特性：1) 状态转移无后效性；2) 状态转移可以具有不确定性；3) 智能体所处的每个状态可以被完全观察[6]。

大多数关于强化学习的研究都是建立在马尔可夫决策过程(Markov decision process，MDP)的基础上[7-8]，MDP 可表示为一个五元组 $<S,A,P,R,\gamma>$。其中：S 为状态(state)的有限集合，集合中某个状态表示为 $s\in S$；A 为动作(action)的有限集合，集合中某个动作表示为 $a\in A$；P 为状态转移方程，$P(s'|s,a)$表示在状态 s 执行动作 a 后将以 $P(s'|s,a)$的概率跳转到状态 s'；R 为奖励函数(reward function)；γ 为折损系数(discount factor)，$0\leqslant\gamma\leqslant1$。假设一个 agent 观察到自己的状态 s，此时它选择一个动作 a，会得到一个即时奖赏 $R(s,a)$，然后以 $P(s'|s,a)$的概率达到下一个状态 s'。MDP 有马尔可夫性，即系统的下个状态只与当前状态有关，与之前的状态无关。当 MDP 中作出决策时，只需考虑当前的状态，而不需要历史数据，这样大大降低了问题的复杂度。

强化学习需要 agent 学习到一个策略 $\pi:S\times A\rightarrow[0,1]$，通过 $\pi(s,a)$的值来指导 agent 进行动作的选择。给定一个策略 π 和一个状态 s，V_s^{π} 表示从状态 s 开始，按照策略 π 进行选择可得到的期望累积奖赏。将 V 称作值函数(value function)，其具体的数学定义为 $V^{\pi}(s)=E\{r_t+\gamma r_{t+1}+\gamma^2 r_{t+2}+\cdots+\gamma^n r_{t+n}\,|\,s=s_t,\pi\}$。强化学习的目标是学到一个最优的策略 π^*，最大化每一个状态下的 V 值，此时的最优值函数记作 V^*。

同理，通过学习到一个最优的 Q 函数 Q^*，使 agent 可以直接通过 Q 函数来选择当前状态下应该执行的动作。

经过多年的研究，已出现一些算法致力于解决传统的强化学习问题，比如 Q-learning、蒙特卡洛方法(Monte Carlo learning)、时序差分方法(temporal-difference learning)等。

1.2 基于价值的强化学习

当智能体的学习进行到某一状态 s 时，基于价值的方法会根据奖励 r 的高低对动作进行选取，此时智能体将屏蔽各种动作发生的概率，典型的 Value-Based RL 有 Q-learning、Sarsa 和 Deep Q Network（DQN）。

1.2.1 Q-learning

Q-learning 是一种典型的无模型(model-free)强化学习算法，是一种由 Watkins 等在 1992 年提出的类似于动态规划算法的一种激励学习方法，目前已经被广泛应用于各个领域[9-10]。在 Q-learning 中，智能体通过与环境的不断交互来进行学习，具体过程为智能体在环境中做出动作 a，获得奖励 r，更新 Q 表，不断重复上述过程直到 Q 表收敛或达到最大循环次数。学习过程如图 1 所示。

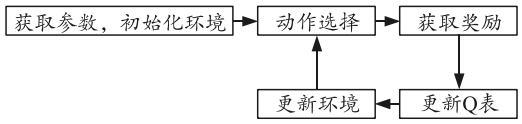


图 1 Q-learning 学习过程

Q-learning 算法的奖励函数通过 Q 表来进行描述，Q 表的每一项都由状态 s 和动作 a 唯一确定，每一项的值称为 Q 值，在 Q 值的计算上 Q-learning 采用了一种可迭代的计算方式[11]：

$$Q(s,a)\leftarrow Q(s,a)+\alpha[r(s,a)+\gamma\max Q(s+1,A_{s+1})-Q(s,a)]。\tag{1}$$

其中：等式右边 $Q(s,a)$由 Q 表得来。由于 Q 表中的值由前几次学习得来，所以该值表示的是由前几次学习得来的 $Q(s,a)$的估计值，即 $Q_{估计}=Q(s,a)$；$\max Q(s+1,\ A_{s+1})$表示智能体在进行到状态 s 选择动作 a 时下一个状态 $s+1$ 中可选择的所有动作集 A_{s+1} 中对应的所有 Q 的最大值，在此基础上乘一个表示未来的预期对当前 Q 值影响程度的抑制的衰减因子 γ，再加上状态 s 下选择动作 a 的奖励值 $r(s,a)$，就构成了 $Q(s,a)$的现实值，即：$Q_{现实}=r(s,a)+\gamma\max Q(s+1,A_{s+1})$。如果 Q-learning 的每一步都采取最大化 Q 值的 action，那么 Q-learning 就转化成了利用蒙特卡洛采样实现贝尔曼最优方程的过程，但实际上 Q-learning 在 $st+1$ 时未必会采取最大化 Q 值的 action，这样可让整个学习过程不至于太快或总是掉入局部最优值的陷阱；二者的差值乘以学习因子 α 即智能体在状态 s 下执行 a 动作后的学习值，即：

$Q_{学习}=\alpha[r(s,a)+\gamma\max Q(s+1,A_{s+1})-Q(s,a)]$。

将学习值与原本的估计值相加，即 $Q(s,a)$更新后的估计值。

1.2.2　Sarsa

Sarsa 算法与 Q–learning 的基本原理类似，也是仅通过智能体与环境的交互进行学习，是典型的无模型学习算法。与 Q–learning 类似，Sarsa 算法也是通过不断更新 Q 表直到其基本收敛来达到学习核决策的目的，但与 Q–learning 不同的是，Sarsa 算法在更新 Q 表时采用的具体方法与 Q–learning 有所区别，更新的公式如下[12]：

$$Q(s,a)=Q(s,a)+\alpha[r(s,a)+\gamma Q(s+1,a+1)-Q(s,a)]。\quad(2)$$

可以看出，Sarsa 算法与 Q–learning 唯一的区别在于 Sarsa 算法中 $Q_{现实}$的计算用到的是下一个状态 s+1 和在下一个状态下选择的动作 a+1，即：$Q_{现实}=r(s,a)+\gamma Q(s+1,a+1)$，此处，Sarsa 算法在更新 Q 表时用到了在下一个状态 s+1 实际选择的动作 a+1，也就是说当智能体处于状态 s 选择动作 a 之后，就确定了在下一个状态要选择的动作，这是与 Q–learning 最大的不同之处。

1.2.3　Deep Q Network (DQN)

传统的 Q–learning 在问题复杂度越来越高时，会产生 Q–table 过大的问题，对其存储所带来的空间开销和搜索带来的时间开销逐渐成为了强化学习过程中无法承受的负担，于是考虑能否采用一个模型来表示状态、动作和值函数之间的关系。而深度强化学习结合了深度学习的结构和强化学习的思想，对于高维状态动作空间任务问题具有良好的适应性[13–15]。

我们令状态为 $s\in S$，动作为 $a\in A$，引入一个状态价值函数，用于计算给定状态动作下的 Q 值。

神经网络在 DQN 中有 2 种形式：1) 将状态和动作当作输入，神经网络输出动作的奖励值；2) 将状态当作输入，神经网络输出该状态下所有动作的奖励值，如图 2—3 所示。

由于 Q–learning 要以状态为输入计算该状态下所有动作的 Q 值，所以网络 2 更加符合这样的特点，其代替 Q–table 的方法和训练过程如图 4 所示。

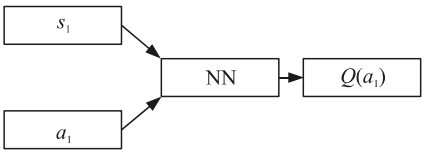


图 2　神经网络形式 1

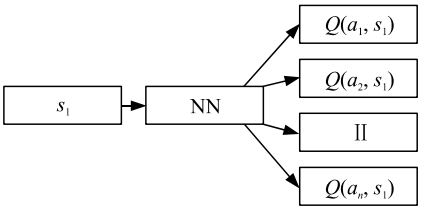


图 3　神经网络形式 2

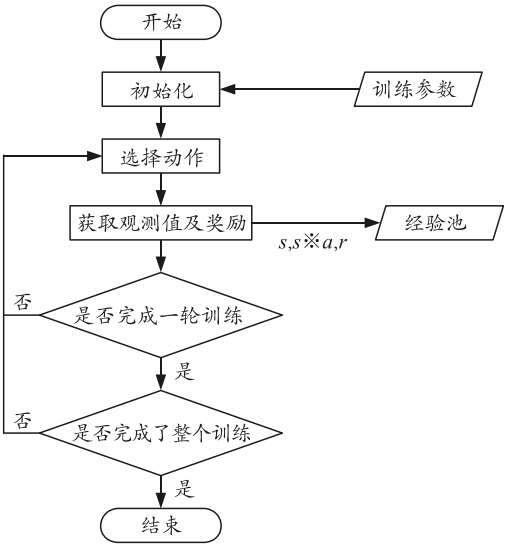


图 4　DQN 算法流程

其中预测网络用于预测当前状态下的各个动作对应的 Q 值，目标网络用于计算下一个状态的目标 Q 值。

对比 Q–learning 和 Sarsa 算法，DQN 有 2 个明显的优势：1) 利用经验池进行经验重放(experience replay)；2) 固定 Q–目标(fixed Q–target)，将原本在网络中同步变化的 Q–target 和 Q–eval 拆解开来，利用变化的 Q–eval 逼近固定住的 Q–target，将问题转化为一个回归问题。这也是 Google DeepMind 团队得以运用 DQN 实现 Alpha Go 的重要原因。

在 DQN 中，神经网络的训练过程如图 5 所示。

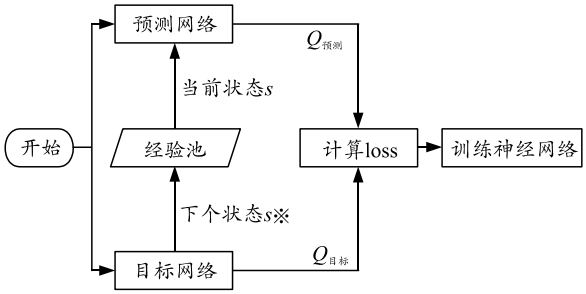


图 5　DQN 神经网络训练流程

基于价值的强化学习的好处在于无需知道在各个状态下各个状态出现的概率，也不需要计算策略梯度，而且在价值迭代的过程中，决策是非常“肯定”的，学习给出的决策就能够找到价值最大的策

略。对比基于策略的强化学习算法来说，其效率是很高的。

2 问题描述

2.1 任务描述

笔者提出的方法基于岛礁守备作战过程，即通过模拟敌方来袭的过程动态寻找防守最佳位置，并根据结果判断当前防守的预设阵地及防守武器装备是否合理。设岛礁兵力配置需求场景为一长 H、宽 W 的矩形区域，环境内存在我方武器装备、敌方目标、我方防守要地和预设阵地，研究对象为我方武器装备在给定敌方目标、防守要地和预设阵地情况下的最佳防守位置。

岛礁守备作战基本流程：我方武器装备前出寻找最佳攻击位置，同时敌方武器装备向其攻击目标前进，当敌方目标进入我方攻击范围内认为可以进行打击，是否打击视情况而定，当毁伤所有敌方目标时认为作战结束，我方胜利；当敌方目标距离我方要地小于其攻击范围时，认为我方要地被攻击，此时认为作战结束，我方失败。

为研究方便给出下列假设：

1) 对最佳位置的确定只针对打击而言，即寻找对我方最有利的打击位置，不考虑探测、指挥、协同等其他环节。

2) 敌方武器在给定高度作匀速直线运动。

3) 当我方武器装备对给定的敌方目标满足攻击条件时认为命中率一定且命中即毁伤，即毁伤概率为 100%。

4) 环境中的每一个位置带其他属性，不对智能体起任何作用，仅位置信息本身作用于智能体。

5) 我方和敌方的行动均沿上下左右 4 个基本方向，其他方向均由上下左右 4 个方向按比例组合而成。

2.2 马尔科夫模型设计

2.2.1 状态空间

在本文中，智能体的状态即智能体的位置，用 2 维向量进行描述：$\boldsymbol{S}=\{(x,y)\,|\,x\in(0,W),y\in(0,H)\}$。由于对每个状态而言，真正对强化学习过程有用的是智能体的横纵坐标，故智能体其他属性不作为强化学习状态的特征参与学习过程，且学习的最终目的是得到奖励值最大的位置，故只将智能体的横纵坐标作为状态空间进行学习。在深度强化学习的算法中，为了使网络更加简单，可将 2 维状态转化为 1 维进行训练，这样网络的输入层仅需 1 维即可。

2.2.2 动作空间

根据前文所述，动作空间应当包含 2 个要素：1) 向某个方向前进；2) 是否对敌方目标进行打击，则正常情况下动作空间应为二者笛卡尔积，即 $A=\{(a_1,a_2)\,|\,a_1\in A_1,a_2\in A_2\}$，其中 $A_1=\{u,d,l,r\}$，即上下左右 4 个前进方向，A_2={h,n}，h=*hit*，n=*no action*，即攻击或不作任何动作，此时动作空间也可以写成：$A=\{(u,h),(u,n),(d,h),(d,n),(l,h),(l,n),(r,h),(r,n)\}$。

由于在实际情况中，只有当智能体满足一定条件的情况下才能够对目标发起攻击，且状态空间很大的情况下满足条件的状态并不多，也就是说 a_2 在较大概率下选择不作任何动作，所以实际上并不需要 2 维动作空间，故笔者提出动态动作空间的方法，引入可用动作空间的概念，整个强化学习过程中，动作空间为 $A=\{u,d,l,r,hit\}$，当满足智能体攻击条件时，可用动作空间 $A=\{u,d,l,r,hit\}$，当不满足智能体攻击条件时，可用动作空间 $A=\{u,d,l,r\}$。设某状态下智能体状态为 $s=(x_t,y_t)$，将 2 维动作空间压缩至 1 维，在本文中对比传统值函数强化学习算法可将 Q 表大小减为原来的 5/8，同时减少了搜索时间，在深度强化学习算法中减少了网络的维数，节约了动作选择的时间。

在其他的强化学习案例中，随着环境及智能体复杂程度的提高，动作空间维数及每一个维度中动作数量也随之增多，带来的是 Q 表占用内存量的爆炸式增长和搜索最大 Q 值时间的大幅延长。为解决这一问题，使得强化学习的算法在计算能力较低的情况下同样适用，在合适的情况下采用上述动态动作空间的方法降低动作空间维度，能够大大减少由动作空间维数增长所带来的时空开销。

2.2.3 奖励函数

在实际问题的求解中，reward 的计算是一个十分重要的环节，reward 的设计越符合实际，强化学习的效果就与实际效果越接近。在本文中，reward 由成本和价值 2 部分构成。

1) 成本计算。

设 i 为我方防御武器批号，$i=1,2,3,\cdots$，C_i 为使用该武器需要的成本，则本回合的总成本为：

$$C=\sum\nolimits_i C_i\ 。$$

成本 C 由以下因素确定：

人员成本 $C_1=\sum_i C_1^i$，C_1^i 为第 i 批武器装备出动时消耗的人员成本。

武器出动成本：C_2 为出动的武器装备成本累加和，$C_2=\sum_i C_2^i$；C_2^i 为第 i 批武器装备出动的成本。

路径成本：C_3 为优化过程中每一步消耗的累加和，$C_3=\sum_i C_3^i$；C_3^i 为第 i 批武器在寻径过程中消耗的成本，用于使智能体通过更短的路径找到最合适的位置。

弹药成本：C_4 为每次范围外攻击或未命中攻击的损耗和，$C_4=\sum_i C_4^i$；C_4^i 为第 i 批武器的弹药损耗。

2) 价值计算。

设 j 为目标的批号，$j=1,2,3,\cdots$。W_j 表示毁伤第 j 批目标的价值，本回合内通过毁伤敌方所获得的价值 $W'=\sum_j W_j$，毁伤第 j 批目标的价值由以下因素确定：

目标价值本身价值：W_1 为敌方目标价值，$W_1=\sum_j W_1^j$；W_1^j 为第 j 批目标由型号确定的价值。

根据敌情紧急程度给出的价值：W_2 为毁伤目标时，根据敌情紧急程度获得的价值，$W_2=\sum_j W_2^j$。通常来说，该防守要地与目标间的距离越小则认为敌情越紧急，此时越应毁伤该敌方目标，但如果防守要地与目标间距离小于敌方攻击范围，则视为我方防守要地被毁，任务失败，W_2^j 为第 j 批目标由距离给出的价值，$W_2^j=\begin{cases}\text{任务失败}, s_2\leqslant S_2\\ W_j^2\left(S_2^2/s_2^2\right), s_2>S_2\end{cases}$。式中：$s_2$ 为敌方与我方防守要地之间的距离；S_2 为敌攻击范围；W_j^2 为目标奖励系数，对应的是实际情况中敌方对我方的威胁系数，是对第 j 批目标对我方威胁程度的反映，也同样是对敌方该批目标作战能力的反映。

设 k 为要地编号，则本回合内通过保卫要地成功获得的奖励为 W_3，$W_3=\sum_j W_3^k$，W_3^k 为保卫第 k 个要地成功时获得的奖励。当毁伤所有敌方目标时，认为防御作战成功，获得奖励 W_3，并结束该回合的学习；当敌我之间距离小于敌攻击范围时则认为我方会遭到攻击，此时损失掉第 k 个要地的保卫奖励 W_3^k，学习继续，若所有要地均被攻击，学习结束。

到达预设阵地奖励：W_4 为我方武器距离预设阵地越近时，获得奖励越大，达到预设阵地时获取最大奖励，$W_4^j=W_j^4(1-S_4^2/s_4^2)$；$S_4$ 为当前我方武器距离预设阵地的距离，s_4 为初始状态下我方武器距离预设阵地的距离。

实际上在获得毁伤奖励前需判断是否命中，若命中，则获取奖励；若未命中，则无法获取。

由以上可以计算出本回合智能体获得的总奖励值 $R=W-C$。

3　模型建立与测试

3.1　环境模型建立

根据问题描述，分别对岛礁守备任务环境中的 4 类实体进行建模。

3.1.1　敌方武器装备

敌方武器装备分为海陆空 3 类目标，每类目标的固有属性有目标类别、初始位置、速度、高度(仅空中目标)、目标价值、攻击范围、威胁能力和存活标志位，具体如下：

1) 目标类别：1 代表空中目标，2 代表陆上目标，3 代表海上目标。

2) 初始位置：目标初始位置坐标，不包含高度。

3) 速度：分为 x 轴和 y 轴 2 个方向的速度，在强化学习的每个 step 中，敌方目标在 x 轴和 y 轴上移动的距离。

4) 高度：目标初始高度，默认不变，若速度中包含 z 轴速度，则每个 step 敌方目标在 z 轴变化相应高度。

5) 目标价值：用于计算 W_1。

6) 攻击范围：攻击远界，当敌方目标与我方要地之间的距离小于攻击远界时，认为要地被攻击，该防守任务失败。

7) 存活标志位：用于在强化学习过程中判断目标是否存活，置 1 表示目标存活，置 0 表示目标已被击毁。

8) 威胁能力：用于计算 W_2^j 的系数，即 W_j^2，表示目标对我方要地的能力或其打击能力。

3.1.2　我方武器装备

我方武器装备对应敌方武器装备分为对陆、对海、对空 3 类，每类武器的固有属性有打击目标类别、初始位置、攻击上下界(仅对空中目标)、攻击

远近界、攻击命中率、存活标志位、单位弹药成本、使用弹药量、出动武器的人员成本，武器出动成本、寻径成本、速度，具体如下：

1) 打击目标类型：我方武器装备能够打击的目标类型，对应目标的海陆空，1 表示能够打击空中目标，2 表示能够打击陆上目标，3 表示能够打击海上目标。

2) 初始位置：我方武器装备初始位置坐标，不包含高度。

3) 攻击上下界：若敌方目标高度超出此范围则认为无法攻击，此属性仅针对空中目标有效。

4) 攻击远近界：若敌方目标距我方武器装备距离超出此范围则认为无法攻击，此属性针对所有类型目标有效。

5) 攻击命中率：当所有攻击条件均符合的情况下，系统认为我方武器装备对敌方目标进行攻击时的命中率，不同武器针对不同目标有不同的命中率，以字典形式存储。

6) 存活标志位：用于在强化学习过程中判断目标是否存活，置 1 表示目标存活，置 0 表示目标已被击毁。

7) 出动武器的人员成本：在该回合中出动武器装备的人员成本，用于计算 C_1。

8) 武器出动成本：出动该武器的损耗成本，用于计算 C_2。

9) 寻径成本：我方武器装备每前进一步的成本，用于计算 C_3。

10) 单位弹药成本：每枚弹药的成本，用于计算 C_4。

11) 使用弹药量：在一个回合中使用的弹药总量，用于计算 C_4。

12) 速度：用于计算我方到达最终位置的时间。

3.1.3 预设阵地

一般情况下，在应对特定敌情的条件下，每类武器都提前设有预设阵地，作为该类武器在无其他因素干预的条件下人为认定的最佳位置。其固有属性为：

1) 预设位置：人为认定的最佳位置或区域，为固定值。

2) 预设阵地奖励：到达指定位置或根据到指定位置的距离获得的奖励，用于计算 W_4。

3.1.4 我方要地

我方需要保卫的目标，当我方要地与敌方目标距离小于敌方武器装备攻击范围时认为我方要地被攻击，任务失败回合结束。其固有属性为：

1) 要地位置：我方防守要地所在位置，为固定值。

2) 存活标志位：用于在强化学习过程中判断要地是否存活，置 1 表示目标存活，置 0 表示目标已被击毁。

3) 要地价值：成功保卫时获得，要地被攻击时损失，用于计算 W_3。

3.2 模型测试

3.2.1 环境模型参数

测试中敌方目标选取海、陆、空类型目标各 1 个，我方武器装备选择对海、对陆、对空武器装备各 1 个，保卫要地 1 个，预设阵地 3 个。各要素初始属性如表 1—4 所示。

表 1 敌方目标

目标编号	目标类别	初始位置	速度	高度	目标价值	攻击范围	存活标志位	威胁能力
1	陆	(10, 15)	(−0.03, −0.05)	0	300	0.36	1	50
2	海	(300, 200)	(−0.5, −0.3)	0	3 000	10 000	1	1 000
3	空	(200, 200)	(−1, −1)	3 000	1 000	2 500	1	500

表 2 我方武器装备

武器装备编号	打击目标类型	初始位置	攻击上下界	攻击远近界	攻击命中率	存活标志位	出动武器的人员成本	武器出动成本	寻径成本	单位弹药成本	使用弹药量	速度
1	对陆	(0, 0)	0	(100, 2 000)	0.5	1	1	1	0.1	5	0	0.01
2	对海	(0, 0)	0	(2000, 10 000)	0.3	1	10	100	1	200	0	1
3	对空	(0, 0)	(500, 5 000)	(15, 3 500)	0.7	1	5	100	1	100	0	1

表 3 预设阵地

预设阵地标号	预设位置	预设阵地奖励
1	(30, 30)	1 000
2	(40, 40)	1 000
3	(60, 10)	1 000

表 4 防守要地

要地编号	打击目标类型	速度
1	对陆	0.01

默认没有任何干扰设备及防空火力，敌方在我

方攻击范围内且速度符合我方攻击条件，认为此时我方单发命中率可以维持在 70%。

3.2.2　测试环境及算法参数

测试环境：

语言：python；

python 版本：3.6；

TensorFlow 版本：1.0；

处理器：Intel(R) Core(TM) i7−10870H CPU @ 2.20 GHz 2.21 GHz；

系统类型：64 位操作系统，基于 x64 的处理器；

RAM：16.0 GB（15.8 GB 可用）。

在算法选择上，由于笔者研究的问题是离散条件下的强化学习，故采用值函数法，分别利用了 *Q*−learning、Sarsa 和 DQN 3 种算法进行学习。3 种算法参数如表 5 所示。

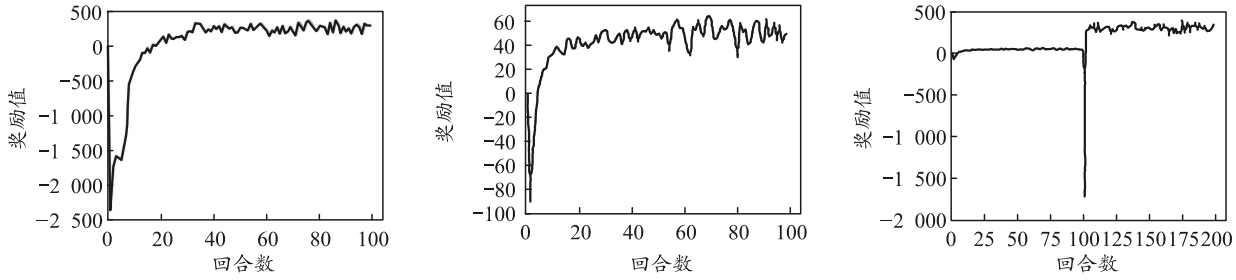
表 5　算法参数

算法	学习率	衰减因子	ε	回合数	经验池大小	单次替换样本数
Q−learning	0.10	0.9	0.9	50 000/50 000/100 000	/	/
Sarsa	0.10	0.9	0.9	50 000/100 000/150 000	/	/
DQN	0.01	0.9	0.9	200 000/200 000/100 000	2 000	200

3.2.3　测试结果及结论

分别利用 *Q*−learning、Sarsa 和 DQN 3 种算法对本章建立的模型进行测试，结果为 3 种武器在 3 种算法情况下在环境中与敌方武器的对抗情况。最佳位置的合理性、奖励值和损失值作为算法好坏的评价标准。为减小误差并使实验结果更加清晰，每 *N* 回合将结果取平均值，结果如图 6—9 所示。

图 6 为 *Q*−learning 算法在环境中获得的奖励值随回合数变化的情况。*N*=500，分别为海陆空 3 类武器的奖励值图像。



(a) 海军武器　(b) 陆军武器　(c) 空军武器

图 6　Q−learning 算法下防守奖励

从图中可以看出，我方的 3 类武器在经过一定回合数的学习之后奖励值逐渐趋于平稳，图 6(c)中回合数为 100 附近奖励值产生较大波动，其原因是智能体在随机性选择的过程中跳出了局部最优值，转而选择奖励值更大的状态。可以看出在该环境下 *Q*−learning 算法是基本有效的。

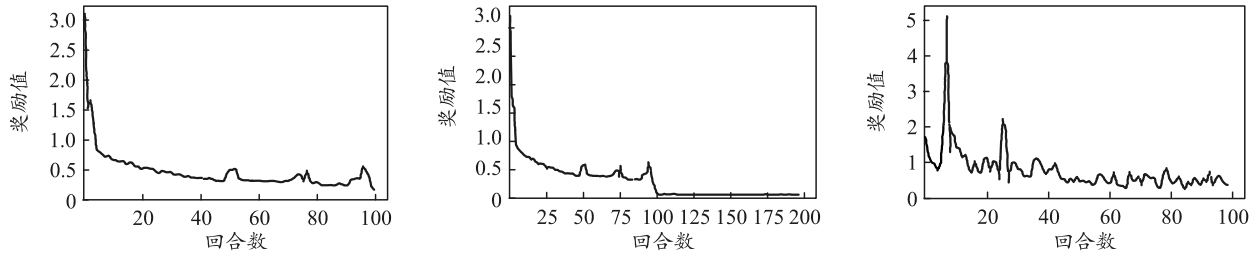
图 7 为 Sarsa 算法在环境中获得的奖励值随回合数变化的变化情况。*N*=500，分别为海陆空 3 类武器的奖励值图像。

(a) 海军武器　(b) 陆军武器　(c) 空军武器

图 7　Sarsa 算法下防守奖励

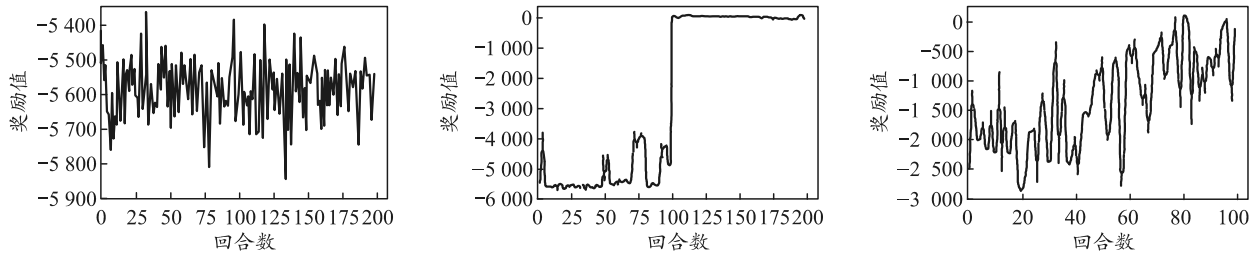
与 Q-learning 算法结果相比，Sarsa 算法在环境中寻找结果的速度是相对慢的，图 7(c)中 200 回合左右奖励值突然下降的原因与图 6(c)相同。在这里智能体也用了比 Q-learning 算法更长的时间重新寻找最佳值，都证明了 Sarsa 算法较为保守，相比于寻找最大奖励来说更看重躲避坏值的特点；但由于本实验中环境内没有设置障碍，智能体不存在避险的问题，故对比而言 Q-learning 更适合本文中环境。

图 8 显示的是 DQN 算法中网络损失值随回合数的变化情况。N=1 000，虽然 DQN 算法回合数与传统强化学习方法相比相同甚至更多，但其学习时间比传统方法少很多，原因在于节省了在 Q 表中搜索最大 Q 值的时间。这种效率上的优势在大状态空间下尤为明显。



(a) 海军武器 (b) 陆军武器 (c) 空军武器

图 8 DQN 损失



(a) 海军武器 (b) 陆军武器 (c) 空军武器

图 9 DQN 算法下防守奖励

由图可见，在一定回合数之后网络已基本收敛，但图 9(a)、(c)中奖励值并不稳定，原因在于能够获得击毁奖励的状态太多，智能体在刚能获取奖励时便开始获取奖励，难以达到最佳位置再进行攻击，且满足攻击条件的状态之间奖励值差距很大，使得图 9(b)中奖励值最终趋于稳定，也正是由于能获得击毁奖励的状态较少，智能体的动作选择在存在一定随机性的情况下，基本能找到最佳攻击位置或最佳攻击区域，使最终奖励值趋于稳定。

从以上测试结果可看出：对于笔者所研究的环境来看，由于没有设置障碍，Q-learning 算法从学习速度上要比 Sarsa 算法快很多，二者准确性和稳定性相差无几。DQN 从学习速度上来看要比传统强化学习方法快很多，但其认为的最佳位置实际上是能将敌方武器装备击毁的位置，在大状态空间下的结果比较粗糙，如果满足条件的状态奖励值相差很大，就会出现奖励值不稳定的问题，想要更加准确的结果需要更加精确的约束条件或改变奖励规则。

4 结束语

笔者基于 Q-learning、Sarsa 和 DQN 3 种强化学习中基于值函数的算法，对给定条件下的岛礁守备兵力配置问题进行了研究。根据岛礁守备作战过程和值函数基本原理设计了基于值函数的强化学习算法、奖励函数，并进行了建模及测试。针对多维动作空间的时空开销随维数增长而爆炸式增长的问题，提出了动态动作空间的解决办法，通过压缩动作空间，减少了时空开销。该方法可运用于不同的环境，应用潜力巨大。

经过学习，智能体能够根据环境较快地找到合适的防守位置，提高了岛礁守备决策的效率，并且在学习过程中能够发现当前防守策略或预设防守位置的不足，有助于防守策略的改进。为了方便起见，本文中的算法是在较为简单、理想的情况下进行的测试，但同样适用于复杂环境及条件。

参考文献：

[1] 赵国艳, 邹伟, 金山. 体系作战条件下岛礁作战中辅助决策问题研究[J]. 航天电子对抗, 2019, 35(4): 40-42, 47.

[2] 张浩. 基于决策支持的防空兵兵力配置系统研究与设计[D]. 重庆: 重庆大学, 2006.

[3] 张智敏, 石飞飞, 万月亮, 等. 人工智能在军事对抗中的应用进展[J]. 工程科学学报, 2020, 42(9): 1106−1118.
[4] 陆志沣, 洪泽华, 张励, 等. 武器装备体系对抗仿真技术研究[J]. 上海航天, 2019, 36(4): 42−50.
[5] LOEFFLER T D, BANIK S, PATRA T K, et al. Reinforcement learning in discrete action space applied to inverse defect design[J]. Journal of Physics Communications, 2021, 5(3): 031001.
[6] 范长杰. 基于马尔可夫决策理论的规划问题的研究[D]. 合肥: 中国科学技术大学, 2008.
[7] 秦智慧, 李宁, 刘晓彤, 等. 无模型强化学习研究综述[J]. 计算机科学, 2021, 48(3): 180−187.
[8] 马骋乾, 谢伟, 孙伟杰. 强化学习研究综述[J]. 指挥控制与仿真, 2018, 40(6): 68−72.
[9] WATKINS C J C H, DAYAN P. Technical Note: Q-Learning[J]. Machine Learning, 1992, 8(3−4): 279−292.
[10] 金则灵, 武晓春. 基于 Q 学习算法的城轨列车智能控制策略[J/OL]. 铁道标准设计: 1−7[2021−06−08]. https://doi. org/10. 13238/j. issn. 1004−2954. 202011100008.
[11] KE H C, WANG H, ZHAO H W, et al. Deep reinforcement learning-based computation offloading and resource allocation in security-aware mobile edge computing[J]. Wireless Networks, 2021(prepublish).
[12] 徐帷, 卢山. 基于 Sarsa(λ)强化学习的空间机械臂路径规划研究[J]. 宇航学报, 2019, 40(4): 435−443.
[13] 赵星宇, 丁世飞. 深度强化学习研究综述[J]. 计算机科学, 2018, 45(7): 1−6.
[14] 杨思明, 单征, 丁煜, 等. 深度强化学习研究现状及展望[J/OL]. 计算机工程: 1−18[2021−06−08]. https: //doi. org/10. 19678/j. issn. 1000−3428. 0061116.
[15] BOUKTIF S, CHENIKI A, OUNI A. Traffic Signal Control Using Hybrid Action Space Deep Reinforcement Learning[J]. Sensors (Basel, Switzerland), 2021, 21(7): 2302.

****

（上接第 35 页）

4.4　设计寿命

影响机器人使用寿命的有以下因素：

1) 零配件的损耗：该机器人重要零配件为机身铝型材及碳纤维框架，减轻自身重量的同时防止水质侵蚀带来的锈腐蚀情况；

2) 电源的供给：本机器人电源由 14.8 V 锂电池提供，另外在船体顶端加装太阳能电池板，太阳能电池将光能转化为电能储存在蓄电池中，工作时蓄电池直接供能，供能稳定，节约能源[10]，保证机器人长时间工作运行。

3) 主板内置芯片的错误叠加：由于机器人长时间运行，有可能导致单片机内置程序混乱，也可能是硬件发生损坏，但一般的使用寿命在 10 年以上。

综合以上因素，机器人预期使用寿命在 4 年以上。

5　结束语

笔者设计的机器人主要针对当前市面上小型水域漂浮垃圾的清理设备的空缺，对于公园或景区小面积水域以及不适合大型垃圾收集船工作的水面(如小型水库、港区、岔口等)垃圾清理问题尤为合适。该机器人可打捞大部分水面漂浮垃圾，如树叶、树枝、包装物、塑料垃圾等，基于无线电遥控技术，操控简单、流动使用、方便快捷、机动性好。使用时只需人员在岸上遥控机器，完成清理工作后再利用人工辅助回收即可。使用结果表明：该水面垃圾清理机器人以其性价比高、操作便捷、运行可靠、体积小巧等优点，解决了传统垃圾收集的诸多弊端，有较大的市场推广空间与应用潜力。

参考文献：

[1] 卢思雨, 张建伟, 王稳. 一种水车式湖面垃圾清理及水体增氧双体船结构设计[J]. 机械工程师, 2020(9): 48−50.
[2] 徐启明, 杨晨, 林超伦, 等. 一种微型水面垃圾清理机器人的设计[J]. 电子世界, 2020(14): 50−51.
[3] 陈华勇, 方鼎, 洪锟, 等. 水上垃圾清理机器人[J]. 兵工自动化, 2018, 37(11): 89−92, 96.
[4] 王法鑫, 倪云林, 骆文彬. 节能环保型水面垃圾清理船的设计与实现[J]. 机械工程师, 2019(9): 25−27.
[5] 张凯淇. 一种节能环保型水面垃圾清理船设计[J]. 现代制造技术与装备, 2020(3): 15−17.
[6] 李嘉琪, 邓彦松, 余国娟. 基于 Python 视觉识别的双模双动力水面垃圾清理船[J]. 兵工自动化, 2020, 39(2): 93−96.
[7] 张国洲, 朱晨炜, 卢加津, 等. 一种水面垃圾清理机器人[J]. 兵工自动化, 2020, 39(3): 90−92, 96.
[8] 陈玲, 高洁. 一种新型水面垃圾清理分拣船的设计[J]. 船舶工程, 2020, 42(2): 39−43.
[9] 高晓红. 一种小型水面垃圾清理装置的研究与设计[J]. 海南大学学报(自然科学版), 2019, 37(3): 254−260.
[10] 赵阳, 赵飞, 李依帆, 等. 一种水面清洁机器人及其系统设计[J]. 科技风, 2019(32): 28.