基于Stackelberg策略的多Agent强化学习警力巡逻路径规划

解易， 顾益军
（中国人民公安大学 网络安全保卫学院，北京 100038）

摘 要：为解决现有的巡逻路径规划算法仅仅能够处理双人博弈和忽略攻击者存在的问题，提出一种新的基于多agent的强化学习算法. 在给定攻击目标分布的情况下，规划任意多防御者和攻击者条件下的最优巡逻路径. 考虑到防御者与攻击者选择策略的非同时性，采用了Stackelberg强均衡策略作为每个agent选择策略的依据. 为了验证算法，在多个巡逻任务中进行了测试. 定量和定性的实验结果证明了算法的收敛性和有效性.

关键词：巡逻路线规划；Stackelberg强均衡策略；多agent；强化学习
中图分类号：TP 399 **文献标志码**：A **文章编号**：1001-0645(2017)01-0093-07
DOI：10.15918/j.tbit1001-0645.2017.01.019

Police Patrol Path Planning Using Stackelberg Equilibrium Based Multiagent Reinforcement Learning

XIE Yi， GU Yi-jun
(Department of Cyber Security, People's Public Security University of China, Beiing 100038, China)

Abstract: The patrol path planning has been simplified with state-of-art algorithm into two-person game in grid world, ignoring the existence of attackers. In order to deal with the problem of realistic patrol path planning, a novel multi-agent reinforcement learning algorithm was proposed. An optimum patrol path was planned in a circumstance that multiple defenders and attackers formed the multi-target configuration. Considering the asynchronism of the actions taken by many defender and attacker, a strong Stackelberg equilibrium was taken as the action selection of players in the proposed algorithm. To verify the proposed algorithm, several patrol missions were tested. The qualitative and quantitative test results prove the convergence and effectiveness of the algorithm.

Key words: patrol path planning; strong Stackelberg equilibrium; multiagent; reinforcement learning

巡逻路径规划为警力设计最优巡逻路线，以最大程度保护潜在攻击目标，使发生犯罪数量的期望或者犯罪分子通过犯罪获得的收益期望最小化. 各种巡逻路径规划研究因为侧重不同，问题具体设置也不同. 最常见的是对抗性巡逻博弈(adversarial patrolling games)[1-2]. 这类巡逻路径规划问题中有一个可见的防御者和一个隐身的攻击者. 防御者每次根据巡逻路径选择一个目标进行保护，攻击者可以等待或者对一个目标进行攻击. 攻击发生的时候，双方根据被攻击目标的防御和攻击情况得到各自收益. 这类问题将巡逻路径规划简化为一个双人博弈的过程. 因为问题的特殊设置，博弈在攻击者

收稿日期：2015-04-15
基金项目：中国人民公安大学基本科研业务费项目(2014JKF01132)
作者简介：解易(1984—)，男，博士，讲师，E-mail：breadbread1984@163.com.

发起第一次攻击时，博弈即结束. 第二种常见的问题设置侧重最小化目标的空闲时间[3-5]. 目标的空闲时间是警力最近一次经过目标到当前的时间. 这类问题中只有防御者，没有攻击者. 防御者要规划最优巡逻路径以最小化所有目标的平均空闲时间. 这两种常见的问题设置之外，还有许多更加接近真实情况的设置. 如 Reis D 等[6]中考虑到犯罪分子在犯罪过程中随着熟练程度的提高会选择犯罪收益更高的目标进行攻击. Chawathe S S 等[7]中利用地理信息系统提供的准确路径长度信息和犯罪热点信息计算出每段路径的巡逻收益和成本的比例，用该比例指导巡逻路径的规划.

本文提出了一种新的巡逻路径规划问题设置，并且通过一种新的多 agent 巡逻路径规划算法解决了这个问题. 本文的贡献主要有 3 点：① 问题设置引入了对抗巡逻博弈中的攻击者和防御者，并且把它们放到了 NetLogo[8] 采用的棋盘格空间中. 相比以往问题设置，本文问题设置在棋盘格上保留了目标分布信息的同时，没有引入过多的细节信息；② 算法采用 Stackelberg 均衡策略，有效地对防御者和攻击者之间的非对称关系进行了建模，并且可以保证纯策略均衡解一定存在；③ 算法可以规划有多个防御者与攻击者情况下的巡逻路径.

1 巡逻路径规划问题的设置

受到对抗巡逻博弈问题的启发，将对抗引入巡逻规划路径问题. 问题中存在攻击者和防御者两种 agent. 显然防御者代表进行巡逻的警力. 防御者保护潜在攻击目标不被攻击者攻击. 潜在攻击目标是一些特定坐标.

攻击者的任务是攻击目标，并由此获得收益. 攻击需要经过 3 个离散时间节点才能完成. 攻击者成功完成一次攻击后，可以继续行动. 防御者的任务是在攻击者成功攻击之前中止其攻击行为以获得收益. 防御者在中止攻击者的同时将其抓获，使其无法继续其他的行动. 在所有攻击者被抓获后，巡逻问题进入终止状态. 攻击者的状态转换关系如图 1 所示. 从图中可以看到攻击者的行为空间为 $\Omega^{\mathrm{att}}=\{$上，下，左，右，不动，攻击$\}$，状态空间为 $S^{\mathrm{att}}=XY\{$等待，阶段 1，阶段 2，阶段 3，被抓$\}$. 其中 X 和 Y 是 agent 活动空间的横纵坐标范围. 阶段 1、2、3 是攻击者攻击过程中的 3 个离散时间节点对应的状态. 攻击者只有来到潜在攻击目标位置才能选择进入攻击状态. 防御者的行为空间为 $\Omega^{\mathrm{def}}=\{$上，下，左，右，不动$\}$，状态空间为 $S^{\mathrm{def}}=XY$. 如果防御者刚好在攻击者处于攻击状态(阶段 1、2、3)时与攻击者在同一个坐标，则攻击者进入终止状态：被抓状态.

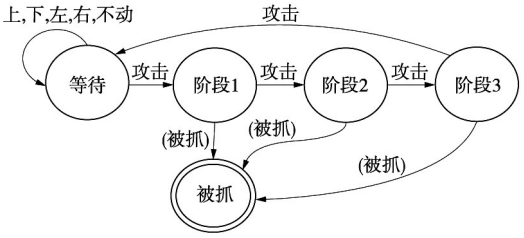


图 1 攻击者的状态转换图

Fig.1 State transition diagram of attacker

本文提出的巡逻路径规划算法在上面设置的规则下，求解最大化防御者的折扣收益和期望的巡逻路径.

2 强化学习

本文提出的巡逻路径规划算法采用强化学习算法实现. 这里首先对一些强化学习的基本概念进行介绍，以为后文算法的阐述进行铺垫.

2.1 强化学习(reinforcement Learning)[9]

在机器学习领域，强化学习算法框架往往用马尔科夫决定过程(Markov decision process，MDP)[10]对 agent 所处的环境进行建模. MDP 是一个离散时间随机控制过程. 在每一个离散时间节点，agent 可以从当前状态确定的行为集合 A 中选择一个行为 a. 到下一个离散时间节点的时候，在行为 a 的影响下，状态会随机地进入到下一个状态 s'，并且 agent 会获得一个收益 $R_{\mathrm{a}}(s,s')$. 而行为 a 影响下的状态转换概率为 $P_{\mathrm{a}}(s,s')$. 一个 MDP 可以由一个四元组$(S,A,P.(\cdot,\cdot),R.(\cdot,\cdot))$表示. 其中 S 是一个有限状态的集合，A 是一个有限的行为的集合，$P.(\cdot,\cdot)$是在不同行为条件下的状态跳转概率，$R.(\cdot,\cdot)$是在不同行为条件下的收益函数.

强化学习的任务就是让 agent 在 MDP 定义的环境中，找到一个最佳的策略函数 $\Pi^*:S\rightarrow A$，以最大化时间节点处获得折扣收益和的期望

$$V^{\pi^*}(s)=\mathrm{E}\Big\{\sum_{k=0}^{\infty}\gamma^k R_{\pi^*(s_t)}(s_t,s_{t+1})\Big\}. \quad (1)$$

式中 γ 是一个区间上的实数. 它是用来表示未来和现在收益重要性差异的折扣值. 如果 $P.(\cdot,\cdot)$完

全已知,那么公式最佳策略 $\pi^*(\cdot)$及对应的折扣收益和期望 $V^{\pi^*}(\cdot)$可以通过动态规划算法[11]计算获得.

2.2 Q-Learning[12]

因为在现实的强化学习应用中,agent 所处环境的 $P.(\cdot,\cdot)$是未知的. 所以这些情况下,无法通过动态规划算法计算最优策略 $\pi^*(\cdot)$. Watkins 在文献[12]中提出一种完全无需了解环境中状态转换概率,就可以得到最优策略函数的方法. 这种方法定义了一种行为-收益(action-value)函数

$$Q^{\pi}(s,a)=\sum_{s'}P_a(s,s')[R_a(s,s')+\gamma V^{\pi}(s')]. \tag{2}$$

该函数表示在状态 s 下,采用行为 a 在策略 π 下的折扣收益和的期望. 该函数可以通过

$$Q_{t+1}(s_t,a_t)=Q_T(s_t,a_t)+\alpha(R_{a_t}(s_t,s_{t+1})+\gamma\max_a Q_t(s_{t+1},a)-Q_t(s_t,a_t)). \tag{3}$$

迭代获得. 其中 α 是在区间上的实数,用做调节学习率之用. 而最优策略可以通过

$$\pi(s)=\max_a Q(s,a). \tag{4}$$

计算获得. 从式(3)可以看到,行为-收益函数的计算不需要预先知道状态转换概率 $P.(\cdot,\cdot)$. Q-Learning 算法在文献[13]中被证明是确定收敛的. 该算法的出现极大地方便了强化学习在实际问题中的应用.

2.3 Nash Q-Learning[14]

在多 agent 强化学习算法中,如果多个 agent 之间是对抗关系,那么可以结合 Q-Learning 算法与 Nash 均衡策略[15]. Nash 均衡策略是一种多人博弈的解决方案. 如果在博弈过程中每个参与者 i 都有一组可选的策略方案集合 $\Omega_i=\{\pi_i\}$,那么策略组合 $\langle\pi_1^*,\pi_2^*,\cdots,\pi_n^*\rangle$是一个 Nash 均衡策略,当且仅当

$$\forall_i,\pi_i\in\Omega_i,$$

$$R^i_{\langle\pi_1^*,\pi_2^*,\cdots,\pi_i^*,\cdots,\pi_n^*\rangle}\geqslant R^i_{\langle\pi_1^*,\pi_2^*,\cdots,\pi_i^*,\cdots,\pi_n^*\rangle}. \tag{5}$$

成立. 其中 $R^i_{\langle\cdot,\cdots,\cdot\rangle}$是在某策略组合下第 i 个参与者的收益. 在 Nash 均衡策略下,没有任何参与者可以在别人保持策略的情况下,通过调整自己的策略来获得更高的收益. 所以 Nash 均衡策略对对抗中的每一个人参与者来说是最优的策略. 借助这个思想,可以像 Q-Learning 算法一样,对博弈中的每一个参与者 i 通过

$$Q^i_{t+1}(s_t,\langle a_t^1,a_t^2,\cdots,a_t^n\rangle)=Q^i_t(s_t,\langle a_t^1,a_t^2,\cdots,a_t^n\rangle)+\alpha(R^i_{\langle a_t^1,a_t^2,\cdots,a_t^n\rangle}(s_t,s_{t+1})+\gamma\max_{\langle a^1,a^2,\cdots,a^n\rangle}Q^i_t(s_{t+1},\langle a^1,a^2,\cdots,a^n\rangle)-Q^i_t(s_t,\langle a_t^1,a_t^2,\cdots,a_t^n\rangle)). \tag{6}$$

训练一个行为-收益函数 $Q^i(s\langle a^1,a^2,\cdots,a^n\rangle)$. 然后对每个状态 s,根据所有参与者的行为-收益函数在这个状态的函数值,计算该状态的 Nash 均衡策略. Nash 均衡策略中第 i 个参与者的行为就是参与者 i 在状态 s 下的最优策略.

3 采用 Stackelberg 均衡进行巡逻规划

虽然 Nash Q-Learning 对巡逻问题提供了一种解决方案. 但是在攻击者-防御者这种对抗性任务中,应用 Nash 均衡策略存在两个问题. 一方面,因为巡逻路径规划任务需要规划确定的路径,所以训练得到的策略是纯策略,也就是在一定的状态下,以概率 1 选择某种行为. 而只有混合 Nash 均衡策略才保证存在. 一些情况下,Nash Q-Learning 可能找不到纯 Nash 均衡策略. 另一方面,在真实的攻击者-防御者对抗中,双方不会像玩石头剪刀布一样同时采取行动. 防御者往往首先采取行动,然后攻击者根据防御者的行为选择最有利自己的行为,如此交替进行. 用 Nash 均衡策略无法解决这种处于非对称关系的博弈问题.

这种博弈方式刚好与 Stackelberg 博弈[16]中的领导者-跟从者(leader-follower)模型是一样的. 防御者和攻击者在博弈中分别扮演领导者和跟从者的角色. 所以,采用 Stackelberg 均衡策略来处理警力巡逻路径规划问题.

下面介绍下 Stackelberg 均衡策略. 如果领导者选择行为 $d\in\Omega^{\text{leader}}$,那么定义跟从者的最优响应函数为

$$BR(d)=\arg\max_{a\in\Omega}{}_{\text{follower}}\ R^{\text{follower}}_{\langle d,a\rangle}, \tag{7}$$

式中 $R^{\text{follower}}_{\langle d,a\rangle}$为在领导者与跟从者策略组合为$\langle d,a\rangle$时,跟从者的收益值. 则$\langle d^*,BR(d^*)\rangle$是一个 Stackelberg 均衡策略,当且仅当

$$\forall d\in\Omega^{\text{leader}},$$

$$R^{\text{follower}}_{\langle d^*,BR(d^*)\rangle}\geqslant R^{\text{follower}}_{\langle d,BR(d)\rangle},\text{及}\ R^{\text{leader}}_{\langle d^*,BR(d^*)\rangle}\geqslant R^{\text{leader}}_{\langle d,BR(d)\rangle}. \tag{8}$$

成立. 在 Stackelberg 均衡策略中,因为博弈双方不是同时选择策略,所以领导者可以在选择策略时候预测跟从方会选择什么策略以最大化跟从方收益. 如果领导者选择一个双方收益都可以最大化的策

略,双方都不能通过单方调整自己的策略以获得更高收益. 因为在一些情况下最优响应策略可能不唯一,所以需要在 Stackelberg 均衡策略基础上再计算出一个 Stackelberg 强均衡策略(strong Stackelberg equilibrium). $\langle d^*, p^* \rangle$ ($p^* \in BR(d^*)$)是一个 Stackelberg 强均衡策略当且仅当

$$\forall d \in \Omega^{\text{leader}}, \quad \forall p \in BR(d^*)$$
$$\begin{cases} R^{\text{follower}}_{\langle d^*, p^* \rangle} \geqslant \max\limits_{q \in \Omega^{\text{follower}}} R^{\text{follower}}_{\langle d, p \rangle} \\ R^{\text{leader}}_{\langle d^*, p^* \rangle} \geqslant \max\limits_{q \in BR(d)} R^{\text{leader}}_{\langle d, p \rangle} \\ R^{\text{leader}}_{\langle d^*, p^* \rangle} \geqslant R^{\text{leader}}_{\langle d^*, p \rangle} \end{cases}. \tag{9}$$

成立. Stackelberg 强均衡策略是所有 Stackelberg 均衡策略中最大化领导者收益的策略.

基于 Stackelberg 强均衡的巡逻路径规划算法,首先通过式(6)迭代得到防御者和攻击者的行为-收益函数. 然后对每个状态 s 通过

$$BR(\langle d_1, d_2, \cdots, d_n^{\text{def}} \rangle) = \underset{\langle a_1, a_2, \cdots, a_n^{\text{att}} \rangle \in \Omega^{\text{att}}}{\operatorname{argmax}} Q^{\text{att}} \times (s, \langle d_1, d_2, \cdots, d_n^{\text{def}}, a_1, a_2, \cdots, a_n^{\text{att}} \rangle). \tag{10}$$

计算防御者每种行为对应的攻击者最优响应行为集合. 然后,通过

$$BR^*(\langle d_1, d_2, \cdots, d_n^{\text{def}} \rangle) = \underset{\langle a_1, a_2, \cdots, a_{n^{\text{att}}} \rangle \in BR(\langle d_1, d_2, \cdots, d_n^{\text{def}} \rangle)}{\operatorname{argmax}} Q^{\text{def}} \times (s, \langle d_1, d_2, d_n^{\text{def}}, a_1, a_2, \cdots, a_{n^{\text{att}}} \rangle). \tag{11}$$

从攻击者最优响应行为集合中找一个能够最大化防御者收益的攻击者行为. 最后,通过

$$\pi^{\text{def}}(s) = \underset{\langle d_1, d_2, \cdots, d_{n^{\text{def}}} \rangle \in \Omega^{\text{def}}}{\operatorname{argmax}} Q^{\text{def}} \times (s, \langle d_1, d_2, \cdots, d_{n^{\text{def}}} \rangle \times BR^*(\langle d_1, d_2, \cdots, d_{n^{\text{def}}} \rangle)). \tag{12}$$

$$\pi^{\text{att}}(s) = BR^*(\pi^{\text{def}}(s)). \tag{13}$$

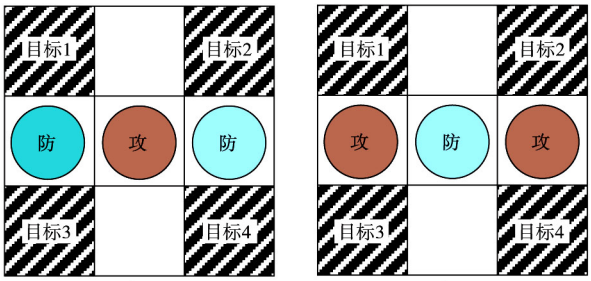
确定状态 s 下防御者与攻击者的最优策略. 用最优策略函数 $\Pi^{\text{def}}(s)$指导防御者行为,即可得到一条期望收益和最优的巡逻路径.

采用 Stackelberg 算法对巡逻路径进行规划,解决了 Nash Q-Learning 的两个问题. 首先,文献[16]中已经证明过纯 Stackelberg 均衡策略总是存在,不会出现像 Nash 均衡策略一样的无解的可能. 另外,攻击者和防御者选择行为的时机与真实的非对称关系更加相符,不再需要同时选择策略. 除此之外,本文提出的算法不再像对抗巡逻博弈一样,仅仅考虑双人博弈问题. 在算法中允许若干攻击者和若干防御者进行对抗. 组内因为共享一个行为-收益函数,所以组内的 agent 是相互协作的. 而组间因为采用不同的行为-收益函数,所以属于不同组的 agent 之间是对抗的关系. 这有效地模拟了现实巡逻问题中,警力与犯罪分子之间的关系.

4 实验结果

为了验证本文提出的基于 Stackelberg 均衡的多 agent 强化学习警力巡逻路径规划算法的效果. 利用 C++语言借助多 agent 模拟开发套件 Repast HPC,实现了实验采用的模拟程序. 通过模拟程序的结果,从 3 方面对本文提出的算法进行评价. 首先,通过训练获得算法收敛的情况;其次,通过对比本文提出的 Stackelberg 算法与 Nash Q-Learning 的收益值,说明利用 Stackelberg 均衡策略能够得到更优的路径规划策略;最后,多 agent 在棋盘空间中的行动路线展示出来,通过直观的方式验证本文的算法.

在如图 2 所示的两种模拟场景中测试了本文算法. 模拟环境大小 3×3. 4 个潜在攻击目标分别在空间的 4 个角位置. 带"攻"和"防"两字的圆圈分别代表攻击者与防御者. 它们在两个场景中的初始位置如图 2 所示. 如果强化学习算法最终收敛,初始状态对最终的行为-收益函数的影响非常小. 所以如果可以证明算法最终收敛,则在固定防御者与攻击者数量情况下,无需对其他初始状态进行测试. 巡逻中,所有防御者共享一个收益,所有攻击者共享另一个收益. 两种 agent 的收益函数设置如表 1 所示.



(a) 场景1 (b) 场景2

图 2 测试的两种模拟场景设置

Fig.2 Two scenarios for testing

表 1 Agent 收益设置

Tab.1 Reward setting for agent

Agent 类型	收益值	条件
防御者	10	抓获一个正在攻击状态的攻击者
攻击者	1	攻击者进入攻击阶段 1、2 或 3
攻击者	10	攻击者完成一次攻击(也就是从攻击阶段 3 到等待状态)

这个巡逻任务中，防御者的状态数为 $|S^{def}|=3\times3$，攻击者状态数为 $|S^{att}|=3\times3\times5$. 防御者的行为数为 $|\Omega^{def}|=5$，攻击者的行为数为 $|\Omega^{att}|=6$. 所以图 3(a)场景的行为-收益函数的定义域的势为 $9^2\times45\times5^2\times6=546\ 750$. 图 2(b)场景的行为-收益函数定义域的势为 $9\times45^2\times5\times6^2=3\ 280\ 500$. 实验采用各项参数如表 2 所示.

表 2　实验参数表

Tab.2　Experiment parameters

参数符号	数值	含义
α	0.10	Q-Learning 的学习率，见式(6)
γ	0.98	Q-Learning 的折扣率，见式(1)
ε	0.10	ε-greedy 选择随机行为的概率
λ	0.90	适合度轨迹(eligibility traces)的轨迹弱化参数(trace-decay parameter)[17]

4.1　算法的收敛

巡逻路径规划算法需要通过 Q-Learning 算法训练行为-收益函数. 为了能够快速得到收敛的行为-收益函数，实现中采用了适合度轨迹(eligibility traces)技术[17]. 对每个场景进行10 000次模拟. 场景 1 的训练在一台 CPU 为四核酷睿 i5，内存为 4 GB 的计算机上运行了 10 h 左右. 场景 2 在同样的计算机上运行了 3 d 左右时间. 每次模拟通过 ε-greedy 算法进行随机状态跳转 1 000 次，并在跳转的过程中更新行为-收益函数. 为了能够衡量行为-收益函数的收敛情况. 在每次模拟结束的时候，计算当前行为-收益函数与上次模拟结果的欧式距离. 场景 1 的行为-收益函数的收敛情况如图 3 和图 4 所示.

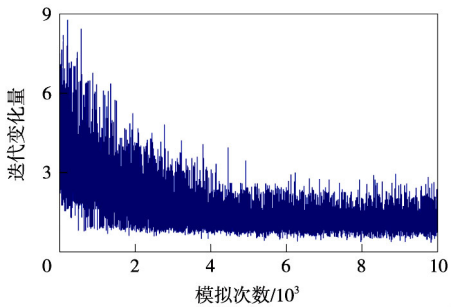


图 3　场景 1 防御者行为-收益函数收敛情况

Fig.3　Convergence of the action-reward function of the defender in scenario 1

可以看到防御者和攻击者的行为-收益函数的迭代误差. 在 10 000 次模拟结束时，维度为 546 750 的向量迭代误差被控制在 1 左右. 场景 2 训练的行为-收益函数收敛情况在图 5 和图 6 中. 从图中

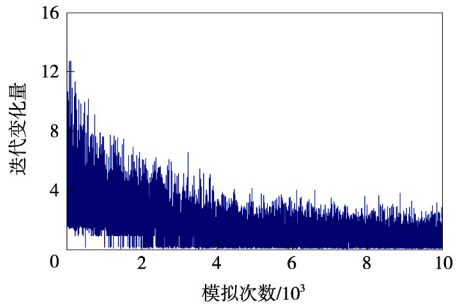


图 4　场景 1 攻击者行为-收益函数收敛情况

Fig.4　Convergence of the action-reward function of the attacker in scenario 1

可以看到防御者和攻击者的行为-收益函数的迭代误差. 在训练结束的时候，维度为3 280 500的向量的迭代误差被控制在 2 左右. 从误差折线图可以看到，本文提出的算法可以在有限次模拟后达到收敛. 这也证明了算法的初始状态不会影响最终的结果.

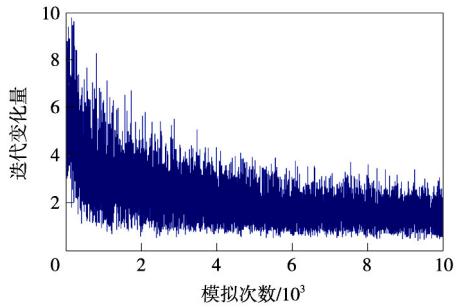


图 5　场景 2 防御者行为-收益函数收敛情况

Fig.5　Convergence of the action-reward function of the defender in scenario 2

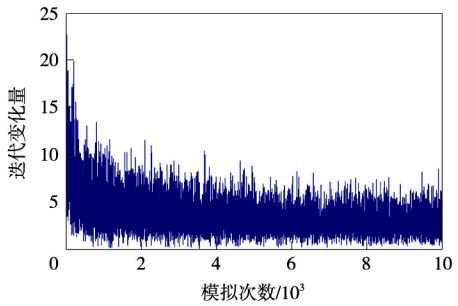


图 6　场景 2 攻击者行为-收益函数收敛情况

Fig.6　Convergence of the action-reward function of the attacker in scenario 2

4.2　Stackelberg 与 Nash 策略的收益对比

为了证明采用 Stackelberg 均衡策略的必要性. 这里通过模拟实验计算 Stackelberg 均衡策略的巡逻路径规划算法与 Nash 均衡策略的巡逻路径规划算法收益，并对它们的结果进行对比. 在实现 Nash 策略的巡逻路径规划算法时候可以利用 Stackelberg 训练得到的行为-收益函数，无需重新

训练. 因为在第3节中已经讨论过，纯 Nash 均衡策略可能不存在，所以在实现的时候，如果找不到纯 Nash 均衡策略，就采用一个随机的策略代替.

计算两种算法中防御者与攻击者按照两种策略进行巡逻获得的折扣收益和. 场景1的结果如表3所示；场景2的结果如表4所示.

表3 场景1中两种 agent 在两种策略下收益对比

Tab.3 Comparison between the rewards of the two strategies in scenario 1

采用策略	防御者	攻击者
Stackelberg	9.411 9	0.960 4
Nash	9.223 7	1.901 6

表4 场景2中两种 agent 在两种策略下收益对比

Tab.4 Comparison between the rewards of the two strategies in scenario 2

采用策略	防御者	攻击者
Stackelberg	17.419 2	13.403 4
Nash	16.609 4	23.650 7

从结果可以看到，采用 Stackelberg 策略的巡逻路径算法，防御者折扣收益和比 Nash 的要高，而攻击者则相反. 结果显示采用 Stackelberg 策略效果在这个巡逻任务中更有利于防御者.

4.3 基于 Stackelberg 均衡策略的巡逻路径

这节通过实验结果图，直观地展示本文算法得到的巡逻路径. 每个离散时间节点内，防御者首先选择策略，然后是攻击者选择策略. 截图展示的状态是在两者都完成策略选择后的状态. 图7与图8分别是场景1和场景2的结果. 图中红色框表示潜在目标所在的位置. 蓝色圆形表示防御者位置，红色圆形表示攻击者位置. 加粗的红框表示对应的目标正在遭到攻击. 红色的叉子表示所在位置的攻击者被抓获.

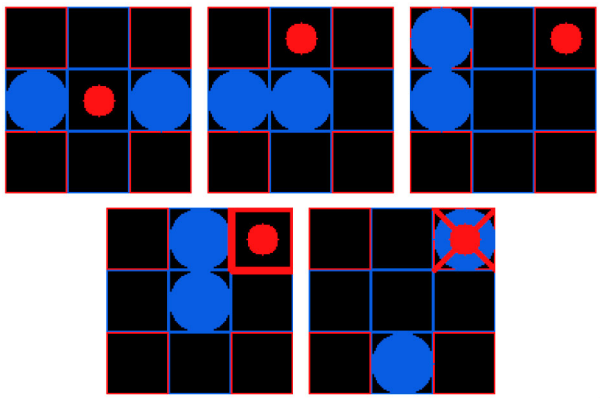


图7 场景1的巡逻路径

Fig.7 Patrol path of scenario 1

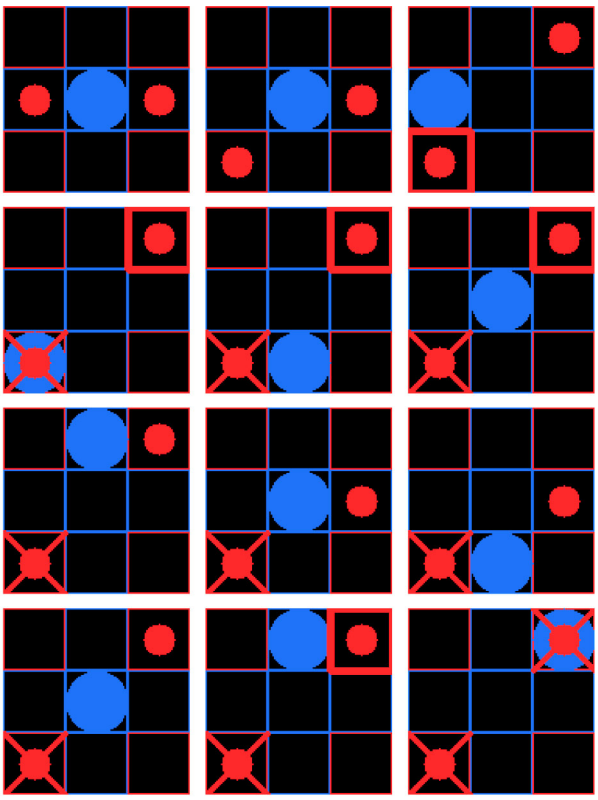


图8 场景2的巡逻路径

Fig.8 Patrol path of scenario 2

从实验结果可以看到，本文算法规划的巡逻路径，能使防御者迅速发现警情，并将攻击者抓获. 在场景1中，攻击者甚至没有机会进入到攻击状态1以后的状态就被抓获. 场景2中，两个攻击者仅仅成功实施一次攻击后就都被抓获. 实验结果可以证明，本文算法规划的巡逻路径有效地保护了潜在攻击目标的安全.

5 结 论

通过多 agent 强化学习方法解决巡逻路径规划问题，在路径规划算法中创新地采用了 Stackelberg 策略，解决了之前 Nash Q-Learning 算法可能得不到 Nash 均衡策略和不能有效模拟攻击者-防御者对抗问题中双方行为的两个问题，并通过实验验证了本文提出的算法的收敛性和有效性.

在未来的研究工作中，将继续研究如何通过分布式算法优化行为-收益函数的训练效率. 通过改进算法，使本文提出的算法能够在更短的时间解决更大规模的问题.

参考文献:

[1] Agmon N, Sadov V, Kaminka G A, et al. The impact of adversarial knowledge on adversarial planning in perimeter patrol[C]// Proceedings of the Autonomous Agents and Multiagent Systems. Richland, SC, USA: [s.n.], 2008(1):55-62.

[2] Basilico N, Gatti N, Amigoni F. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies[C]// Proceedings of the Autonomous Agents and Multiagent Systems. Richland, SC, USA: [s.n.], 2009:57-64.

[3] Santana H, Corruble V, Ratitch B. Multi-agent patrolling with reinforcement learning[C]// Proceedings of Autonomous Agents and Multiagent Systems. New York, USA:[s.n.], 2004(3):1122-1129.

[4] Almeida A, Ramalho G, Santana H, et al. Recent advances on multi-agent patrolling[C]// Proceedings of Symposium on Artificial Intelligence. Brazil: [s.n.], 2004:474-483.

[5] Marier J S, Besse C, Chaib-draa B. A markov model for multiagent patrolling in continuous time [C] // Proceedings of 16th International Conference on Neural Information Processing. Bangkok, Thailand: [s.n.], 2009:648-650.

[6] Reis D, Melo A, Coelho A L V, et al. Towards optimal police patrol routes with genetic algorithms[C]// Proceedings of IEEE International Conference on Intelligence and Security Informatics. San Diego, USA: [s.n.], 2006:485-491.

[7] Chawathe S S. Organizing hot-spot police patrol routes [C] // Proceedings of IEEE International Conference on Intelligence and Security Informatics. New Brunswick, NJ, Canada: [s.n.], 2007:79-86.

[8] Kornhauser Daniel, Rand William, Wilensky Uri. Visualization tools for agent-based modeling in netLogo[C] // Proceedings of Agent 2007. Chicago, IL, USA: [s.n.], 2007:1-12.

[9] Sutton R S. Temporal credit assignment in reinforcement learning[D]. Amherst, MA: University of Massachusetts, 1984.

[10] Bellman R. A Markovian decision process[J]. Journal of Mathematics and Mechanics, 1957,4:679-684.

[11] Howard R A. Dynamic programming and markov processes[M]. USA: The M.I.T. Press, 1960.

[12] Watkins, Dayan C J C H. Learning from delayed rewards [D]. Cambridge, England: Cambridge University, 1989.

[13] Watkins, Dayan C J C H. Q-learning[J]. Machine Learning, 1992,8(3-4):279-292.

[14] Hu Junling, Wellman Michael P. Nash Q-Learning for general-sum stochastic games[J]. Journal of Machine Learning Research, 2003,4(11):1039-1069.

[15] Nash John F. Non-cooperative games[J]. Annals of Mathematics, 1951,54(2):286-295.

[16] Stackelberg H Von. Market structure and equilibrium [M]. 1st edition. Bazin, Urch & Hill: Springer, 2011.

[17] Sutton R, Barto A. Reinforcement learning: an introduction[M]. Cambridge, USA: [s.n.], 1998.

(责任编辑:刘芳)