



基于博弈论及 Q 学习的多 Agent 协作追捕算法

郑延斌^{1,2}, 樊文鑫^{1*}, 韩梦云¹, 陶雪丽¹

- (1. 河南师范大学计算机与信息工程学院, 河南 新乡 453007;)
(2. 智慧商务与物联网技术河南省工程实验室, 河南 新乡 453007)
(*通信作者电子邮箱 525845701@qq.com)

摘要: 多 Agent 协作追捕问题是多 Agent 协调与协作研究中的一个典型问题。针对具有学习能力的单逃跑者追捕问题, 提出了一种基于博弈论及 Q 学习的多 Agent 协作追捕算法。该算法首先建立协作追捕团队, 并构建协作追捕的博弈模型; 其次通过对逃跑者策略选择的学习, 建立逃跑者有限条 T 步累积奖赏的运动轨迹, 并把运动轨迹调整到追捕者的策略集中; 最后求解协作追捕博弈, 得到 Nash 均衡解, 每个 Agent 执行均衡策略完成追捕任务。同时针对在求解中可能存在多个均衡解的问题, 加入了虚拟行动行为选择算法, 选择最优的均衡策略。C# 仿真实验表明, 在同等条件下该算法能够有效的解决障碍环境中单个具有学习能力的逃跑者的追捕问题, 实验数据对比分析表明该算法的追捕效率优于其他算法。

关键词: 多 Agent; 协作追捕; 博弈论; 强化学习

中图分类号: TP24 机器人技术

文献标志码: A

Multi-Agent Collaborative Hunting Algorithm Based on Game Theory and Q-Learning

ZHENG Yanbin^{1,2}, FAN Wenxin^{1*}, HAN Mengyun¹, TAO Xueli¹

- (1. College of Computer and Information Engineering, Henan Normal University, Xinxian Henan 453007, China;)
(2. Henan Engineering Laboratory of Intellectual Business and Internet of Things Technologies, Xinxian Henan 453007, China)

Abstract: The multi-agent collaborative hunting problem is a typical problem in multi-agent coordination and collaboration research. Aiming at the single escaper problem with learning ability, a collaborative pursuit method based on game theory and Q-learning is proposed. Firstly, a cooperative hunting team is established and a game model of cooperative hunting is built. Secondly, through the learning of the escaper's strategy choice, the trajectory of the escaper's limited T-step cumulative reward is established, and the trajectory is adjusted to the pursuer's strategy set. Finally, the Nash equilibrium solution is obtained by solving the cooperative pursuit game, and each Agent executes the equilibrium strategy to complete the pursuit task. Also, in order to solve the problem that there may be multiple equilibrium solutions, it added the algorithm of virtual action behavior selection to select the optimal equilibrium strategy. C# simulation experiment shows that under the same conditions, this method can effectively solve the pursuit problem of a single runaway with learning ability in the obstacle environment, and the comparative analysis of experimental data shows that the efficiency of this method is better than other methods.

Keywords: multi-agent; collaborative pursuit; game theory; reinforcement learning

0 引言

多 Agent 协作追捕问题是多 Agent 协调与协作研究中的一个典型问题, 在军事、工业、农业等方面都有典型的应用。受到国内外研究者的广泛关注^[1-6,22]。根据逃跑者的数量, 多 Agent 协作围捕问题可以分为单逃跑者围捕和多逃跑者围捕问题, 本文关注在有障碍物条件下的单逃跑者协作围捕问题。

在单个逃跑者围捕中, 当逃避者不具备学习能力时, 围捕者可以根据逃避者的逃跑策略制定相应的围捕策略, 其追捕效率高; 当逃避者具有学习能力时, 环境中的障碍物可以同时被追捕者和逃跑者利用, 追捕者可以利用障碍物阻挡逃跑者, 逃跑者同时可以利用障碍物躲避追捕者, 同时逃跑者又可以根据围捕者的策略来改变自己的逃跑策略, 因此追捕的效率低。

收稿日期: 2019-10-20; 修回日期: 2019-12-24; 录用日期: 2019-12-30。

基金项目: 国家自然科学基金 (No. U1604156), 河南师范大学青年基金 (2017QK20)

作者简介: 郑延斌(1964-), 男, 河南内乡人, 教授, 博士, CCF 会员, 主要研究方向: 虚拟现实、多智能体系统、对策论; 樊文鑫(1994-), 男, 河南郑州人, 硕士研究生, 主要研究方向: 虚拟现实, 多智能体系统; 韩梦云(1993-), 女, 河南安阳人, 硕士研究生, 主要研究方向: 虚拟现实, 汉字识别技术; 陶雪丽(1978-), 女, 河南南乐人, 讲师, 硕士, 主要研究方向: 虚拟现实、多 Agent。



针对逃跑者具有学习能力使得追捕效率降低的问题,国内外研究者提出了许多解决可感知环境下的多 Agent 协作追捕的方法,可以分为两类:(一)利用强化学习方法探索多 Agent 协调行为,解决单逃跑者的追捕问题^[7-10]。如 Asl Z D, Derhami V^[8]等提出了一种基于强化学习的多 Agent 协作追捕方法,该方法利用 Q 学习方法建立一个共享的 Q 值表,用于记录逃避者过去的行为路线,每个追捕者在选择自己的追捕策略时,不是从固定的动作集中选择,而是从已经建立好的 Q 值表中来选择。与同类型的追捕算法相比较,追捕效率更高; Bilgin A T, Kadioglu-Urtis E^[9]使用强化学习方法对多 Agent 追捕问题进行了研究,用 Q-Learning 与资格跟踪相结合方法,首先在多 Agent 团队中使用并行学习的方式,每个 Agent 独立选择自己的行为,并收到相应的反馈信息(环境的奖励或惩罚),并利用这些反馈来更新每一个成员的 action-value 矩阵;其次为每个 Agent 存储临时的行为轨迹(存储其行为的临时记录),当资格跟踪发生错误时返回奖励或惩罚,由于过去的追捕行为会随着时间的推移而消失,因此在 Q-Learning 算法中加入衰减率。实验证明了算法的有效性,表明了在同环境下不同学习率和衰减值的差异性; Songhao Piao^[10]等提出了一种基于自组织特征映射(SOFM)和基于 agent 群角色隶属函数(AGRMF)模型的增强学习的移动多智能体追踪方法。该方法基于 SOFM 和 AGRMF 技术,促进了追求者群体的动态组织,并使追求者群体根据自己的意愿进行规避。这有助于克服在 AGRMF 模型运行过程中,当目标过于独立时,追求者不能完全重组的缺点。此外还加入了奖励功能。在群体形成后,应用强化学习得到每个 agent 的最优解。捕获过程中每一步的结果最终都会影响 AGR 隶属函数,从而加快竞争神经网络的收敛速度;(二)基于博弈论的多 Agent 协作追捕策略^[11-14]。如 Baofu F, Jinling Z 等^[12]针对多机器人协作追捕的时间会受到每个自利的机器人的动作选择的影响,提出了一种基于量子博弈的方法,将经典战略空间扩展到量子伙伴的范围,确保机器人的行为策略收敛到最优平衡点,消除随机性和盲目性;晏亚林^[13]等通过将逃跑者加入“拒捕”行为且改进了有效包围和距离影响的权重,在可感知的环境下将追捕问题转化为博弈问题,提高了追捕的效率; Raul Hakli^[14]提出了一种基于规划和博弈团队推理相结合的协同规划方法,该方法从构建一个群体计划开始,从中派生出他们的子计划,个体在群体的计划中执行他们各自的部分,适合在可以观察到彼此行动的情况下的合作,在实际情况下能够更像人类一样进行有效的联合动作。

多 Agent 追捕环境中,追捕者和逃跑者都具有学习能力,故追捕者的协作追捕行为受逃跑者的逃跑的影响,逃跑者的行为也会受到障碍物追捕者以及障碍物的影响,上述的方法在强化学习方面虽然考虑到了将逃跑者的行为策略进行学习,但是未能考虑到在动态环境中追捕双方受到的相互影响,及资源冲突的问题,在博弈论方法方面,考虑到了团队之间的协作,但纯博弈的思想会有收敛速度慢的问题。然而博弈

论为这种具有相互影响的决策性提供很好数学模型,而强化学习可以让 Agent 在特定环境中,根据当前的状态,做出行动,从而获得最大回报;另外博弈论的核心是均衡局势的问题,故为了达到均衡,追捕者和逃跑者应相互学习,从而使自身利益最大化。因此研究者提出将博弈理论与强化学习进行有效结合,考虑到在动态环境中受到的相互影响,并通过学习的方法将追捕者的策略进行迭代更新,设定出具有针对性的追捕策略,将策略作为博弈论中 Agent 可选择的动作策略,能够有效的完成多 Agent 的协作追捕任务。

本文提出了一种基于博弈论及 Q 学习的多 Agent 协作追捕算法,来解决可感知环境中,逃跑者和追捕者都具有学习能力的情况下,多 Agent 的协作追捕问题。该算法首先利用 Agent 的属性以及任务的需求,建立追捕团队,并构建博弈模型;其次根据逃跑者的运动路径做有限次的学习,得到有累积奖赏的 T 步路径轨迹,并把路径轨迹调整到追捕者可选择的可执行策略集中;最后求解协作博弈,得到 Nash 均衡,每个 Agent 选择均衡策略完成追捕任务。同时针对在求解中可能存在多个均衡解的问题,加入了虚拟行动行为选择算法,选择最优的均衡策略。研究者在实验平台中对文章提出的算法进行分析实验,验证本文算法的合理性及有效性。

1 相关基础

1.1 博弈论基础

又称“对策论”(Game Theory),它研究的是在决策者的行为之间发生相互作用时,各个决策者所作对策的问题^[15,16]。

定义 1: 博弈可以用一个三元组来描述,即 $G=\langle P, S, U \rangle$, 其中:

P 表示所有局中人的集合 $P=\{p_1, p_2, \dots, p_n\}$, S 表示局中人可行的策略集 $S=\{S_1, S_2, \dots, S_n\}$, 每个 Agent 的策略可以形式化为 $(A_1^i, A_2^i, \dots, A_n^i)$, U 表示局中人的支付函数 $U=\{U_1, U_2, \dots, U_n\}$ 。

定义 2: Nash 均衡

设 $G=\langle P, S, U \rangle$, 如果存在一个联合行为 $a^* \in S$, 满足条件:

$\forall i \in P, \forall a_i \in S, U(a_i^*, a_{i-1}^*) \geq U(a_i, a_{i-1}^*)$ 则称 a^* 为博弈 G 的 Nash 均衡 (Nash equilibrium)。

Nash 均衡是博弈的稳定解。只有当所有的居中人都预测到某一个特定的 Nash 均衡出现的情况下, Nash 均衡才会出现,当这样的一个人 Nash 均衡出现,任何一个居中人偏离这个策略组,其收益函数不会变大,因此一旦所有的局中人组成了 Nash 均衡,任何一个局中人都不会擅自偏离。

1.2 Q 学习



机器学习 (Machine Learning, ML) 是当前人工智能领域的一个热点问题, 根据数据类型不同, 以及对一个问题建模方式的不同, 将机器学习分为三种类型: 监督学习 (Supervised Learning, SL)、非监督学习和强化学习 (Reinforcement Learning, RL)。

强化学习 (Reinforcement Learning, RL) 主要强调智能体基于环境而行动, 以取得最大化的效益。即智能体在学习过程中通过环境给予的奖励或惩罚, 不断尝试, 逐步形成对刺激的预期, 从而产生能获取最大回报的策略^[17]。

强化学习中 Q-learning 是一种具有代表性的算法, 它主要有四部分组成: Q 表: $Q(s, a)$ 状态 s 下执行 a 动作的累积价值; 选择动作; 做出动作, 环境的反馈; 环境更新。在其过程中 Agent i 观察周围环境, 执行动作策略集中的动作。在 t 时刻, Agent i 执行动作 a_t , 同时反馈收益 $R(S_t, a_t)$, 更新 Q 值表, 重复上述过程, 直到任务结束。其中 $Q(S_t, a_t)$ 的值可用公式表示为:

$$Q(S_t, a_t) = R(S_t, a_t) + \gamma \max_{a'} Q(S_{t+1}, a')$$

式子中, a 为动作策略集中的某一动作, 常量参数 γ ($0 \leq \gamma \leq 1$) 称作影响因子。在 Agent i 训练学习过程中, 选择最大 Q 值的动作进行迭代训练。

2 基于博弈论及 Q 学习的协作追捕算法

2.1 追捕问题描述

假定在一个多 Agent 协作环境 X 中, 有 M 个 Agent 构成的追捕者用集合 $R = \{R_1, R_2, \dots, R_m\}$ 表示。有 N 个 Agent 构成的逃跑者, 用集合 $T = \{T_1, T_2, \dots, T_n\}$ 表示。环境内有形状和大小任意的固定障碍物, 其位置映射关系为: $m: X \rightarrow \{0, 1\}$, 指定所有 $x \in X$, $m(x)=1$ 表示位置 x 是障碍物。时间可离散化并用 $t \in T = \{1, 2, 3, \dots\}$ 表示, 规定任意 Agent 在每个时刻只能执行一个动作, 原地不动或者移动到其相邻并未被占据位置。多个追捕者形成一个协作团队 A 来完成任务 W , 完成任务后可以获得一定的效用 U 。

将逃跑者被捕获定义为:

定义 3: $Ge(t) = \{Xe(t-1)\}, t \in T$

定义表明, 当逃跑 Agent 在 t 时刻被追捕成功时, 他只能运动到 $t-1$ 时刻所在的位置中, 其中 $Xe(t-1)$ 表示其逃跑 Agent 在 $t-1$ 时刻所占据的位置, 同时若能满足以下三个条件也可以认为是被捕获的:

1) 在没有障碍物的情况下:

假设逃跑 Agent 在 $t-1$ 时刻运动, 其周围的呈三角形位置中已经被其它追捕的 Agent 占据, 且相邻两个追捕 Agent 的距离小于两个身长的长度。如图 1 所示

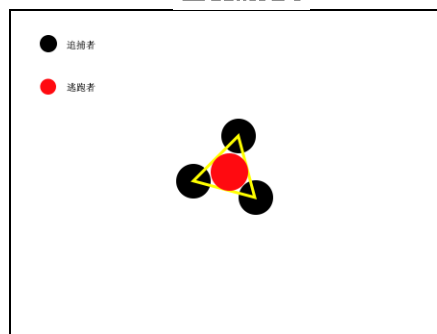


图 1 没有障碍下追捕成功

Fig 1 Successful pursuit without obstacles

2) 在有障碍物的情况下:

假设逃跑 Agent 在 $t-1$ 时刻, 逃跑 Agent 的某一个或者不多于四个方向都存在障碍物, 此时追捕 Agent 占据的其它可移动方向的位置。如图 2 所示

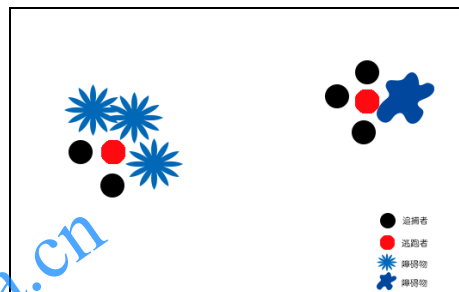


图 2 障碍下追捕成功

Fig 2 Successful pursuit under obstacles

3) 在无法挣脱的情况下:

假设逃跑 Agent 在 $t-1$ 时刻, 已经被团队形成围捕之势, 其活动空间存在, 但其运动的路径已经无法挣脱围捕圈, 这种情况也可判定为已被抓捕。

2.2 障碍物问题描述

实验平台中会设定一些大小, 位置不一的障碍物, 用于模拟真实的环境。下面设定障碍物的一些属性:

- 1) 由于实验环境设定的比较大, 在环境中设定的障碍物不能占满整个环境。
- 2) 环境中设定的障碍物不能全部聚集在一起。

由于障碍物没有布满整个环境, 在没有障碍物阻挡的一些地方就会形成无障碍的状态, 在无障碍环境下, 追捕者只能靠自己运动才能将逃跑者围捕, 其追捕难度就会比较大; 相反在有障碍的环境下, 追捕者既可以利用障碍物将逃跑者进行围捕, 逃跑者也可以利用障碍物躲避追捕者, 因此平台中障碍物的大小, 数量以及分布会对追捕者和逃跑者的运动产生影响, 这种影响就会降低追捕的难度。在实验平台中障碍物的数量越多, 逃跑者在某一特定的位置可选择的方向就越少, 此时追捕者利用障碍物将逃跑者围捕的概率就变大; 然而障碍物比较分散的情况, 追捕者与逃跑者的可选择方向都会减少, 此时对两者都是有影响的。



2.3 虚拟管理者

在整个追捕环境 X 中已经设立了 M 个追捕者和 N 个逃跑者, 为了使追捕的环境趋于真实化, 就需要有管理者同意安排并初始化障碍物的位置和大小, 以及在追捕者团队中担任指挥的角色。

在多 Agent 系统中建立一个虚拟管理者, 此管理者不参与任何追捕活动, 虚拟管理者的任务负责确定有 N 个逃跑者, 有 M 个追捕成员, 记录所有 Agent 从开始到结束的轨迹路线, 当追捕团队有多个均衡解时, 虚拟管理者作为居中调度选择最优解。

2.4 追捕团队的形成

在多 Agent 中选择一个 Agent 来管理完成任务的分配, 管理 Agent 掌握所有 Agent 的位置, 能量, 偏好等属性信息, 但由于 Agent 的自利性, 了解所有 Agent 的全部信息是不现实的, 故采用基于拍卖的方式实现任务分配, 管理 Agent 了解所有的任务信息, 负责发布任务, 其它 Agent 根据自己的能量, 自身属性和已经发布的任务需求进行投标。具体算法如下:

算法 1:

步骤 1: 虚拟管理者对环境中的所有 Agent 初始化为逃跑者与追捕者;

步骤 2: 虚拟管理者发布, 并利用广播的方式告知未分配的任务 W_i 的信息, 以及该项任务完成所能给予的报酬 U_{wi} ;

步骤 3: 追捕者接收到虚拟管理者的广播, 预估该任务可能消耗的能量和所获得的报酬 U_{wi} , 根据自己剩余的能量, 以及偏好等属性选择愿意承担的任务来投标, 并把投标的信息广播给虚拟管理者;

步骤 4: 虚拟管理者等待追捕者提交投标信息, 若无投标信息, 转向步骤 8;

步骤 5: 虚拟管理者设定投标价格为 FP , 并将每个任务中 m 个追捕者的投标从高到低进行排序;

步骤 6: 虚拟管理者选择 $n(n \leq m)$ 个追捕者分配给该任务, 并将中标信息广播给追捕者, 该追捕者的标志 $busy=1$, 不允许再投标其它任务;

步骤 7: 如果同一个 Agent 向多个任务提交了投标, 虚拟管理着计算 $U_{wi}-UP_{wi}$, 选择能获得的最大效用的任务分配给该追捕者;

步骤 8: 若所有的任务分配完毕, 则转向步骤 9; 否则对未分配的任务进行分解, 转步骤 1;

步骤 9: 结束。

2.5 协作追捕算法

2.5.1 博弈模型的构建

在追捕环境中通过团队之间协作完成任务的追捕者, 团队中一个追捕者的行为会受到其它追捕者的影响, 同时逃跑者的行为也会受到追捕者的行为决策的影响。而博弈论为这种相互影响的决策行为给出了很好数学模型^[21]。

定义 4: 多 Agent 协作追捕博弈模型, $G=\langle P, S, U \rangle$

其中 P 为追捕团队中追捕者的集合, $P=\{1, 2, 3, \dots, n\}$, S 为追捕者可能执行的策略集合, S_i 是团队中每个追捕者的动作策略, 每个追捕者根据当前自己周围的环境和其它追捕者周围的环境做出相应的动作。每个 Agent 的策略可以形式化为 $(A'_1, A'_2, \dots, A'_n)$, U 是支付函数, 表示执行策略之后的得失情况。

设多 Agent 系统所处的环境为 X , X_t 表示多 Agent 系统在时刻 t 所处的环境。

设多 Agent 系统所处的环境为 X , X 表示多 Agent 系统在时刻 t 所处的环境。

设 Agent 可观测到的环境状态的集合为 S_t , $S'_t = f(E^t)$, $S^t = (S'_1, S'_2, \dots, S'_n)$ 为系统中在 t 时刻的联合观测, 记为: $S = \prod_{i \in N} S_i$ 。在动态复杂的环境下多 Agent 系统中的 Agent 所获取的信息可能是完全的, 也可能是不完全的。

设 Agent 动作集合为 A , A_i 用于表示 Agent i 的动作集合, Agent 的动作集合用 A 表示, $A = \prod_{i \in N} A_i$, 每个 Agent

在时刻 t 所观测的环境采取的动作 $A'_i \in A_i$ 对环境产生的影响看, 多个追捕者之间的联合行动 $(A'_1, A'_2, \dots, A'_n)$ 也会对当前所处的环境的状态产生影响。

设状态转移函数为 T , $T_i: S \times A \rightarrow S$, 表示在某一特定的环境下, 某个追捕者与其他追捕者之间通过协作对环境可能产生的影响。

设 Agent 支付函数为 U , $U_i: S \times A \rightarrow U$, 表示 Agent i 在多 Agent 系统中为了完成任务所采取的行为后的收益情况。

Agent 的目标集合 $G = \{G_1, G_2, \dots, G_n\}$, G_i 表示多 Agent 系统中每个 Agent 的目标, 通常可以利用支付函数 U 来表示, 每个 Agent i 之间的目标可能存在多种关系, 当目标一致时, Agent 之间目标的完成是相互促进的, 当目标冲突时, 就会产生利益资源的冲突。

2.5.2 基于 Q 学习的可执行的策略集

在单猎物追捕问题中, 需要在环境中随机生成 M 个追捕者, 由他们组成一只协作追捕团队 A 。由于单猎物以是否具有学习能力分为智能化和非智能化。非智能化的逃跑者其逃跑运动轨迹一般比较固定, 智能化的逃跑者运动轨迹不确定,



会根据自身对环境的状态进行选择路径。因此在其逃跑的策略集中加入强化学习算法, 给其动作选择的策略集进行改进。

在 t 时刻可以进行移动的方向称为 Agent 的策略集, 其策略集就是其能进行决策的集合。 t 时刻逃跑者的策略集表示为 St

$$\begin{aligned} St \text{ 包含 } \{ & X_a, X_b, X_{at}, X_{bt} \} \\ X_{at} = & X_a + \cos(Dir * PI / 180) * V \\ X_{bt} = & X_b + \sin(Dir * PI / 180) * V \end{aligned}$$

其中 $0^\circ \leq Dir \leq 360^\circ$, 供其选择的方向有 360° , PI 表示 π 值, V 表示速度。

由于追捕者无法获知逃跑者的状态-动作值 Q , 也就无法准确的找到适合的策略应对, 其原有的策略集就显得比较宽泛, 因此加入 Q 学习的方法将逃跑者的动作状态值进行学习, 调整为具有针对性的追捕策略集。

由于逃跑者的逃跑策略都是未知的, 因此使用强化学习的方法, 对逃跑者的逃跑策略进行学习, 制定适合追捕者追捕策略。首先在此算法中加入 T 步累积奖赏的学习任务, 从逃跑者的初始状态出发, 使追捕者经过有限次的学习获得一条具有 T 步的逃跑者的逃跑轨迹,

$$\langle x0, a0, r1, x1, a1, r2, \dots, xT-1, aT-1, rT, xT \rangle$$

其次记录轨迹中每一对状态-动作 Q 的累计奖赏之和, 作为一次关于逃跑者累积奖赏采样值, 当对逃跑者进行多次采样得到多条逃跑轨迹后, 将对多次获得的累积奖赏采样值利用公式(1)求取平均, 得到 Q 值的估计。

$$\begin{aligned} Q_n(k) &= \frac{1}{n} ((n-1) \times Q_{n-1}(k) + u_n) \\ &= Q_{n-1}(k) + \frac{1}{n} (u_n - Q_{n-1}(k)) \end{aligned} \quad (1)$$

由于要得到较好的动作-状态值函数的估计, 就需要产生多条不同的轨迹, 然而逃跑者选择的策略有可能是固定的, 经过采样会导致追捕者得到的路线都是一致的。为了得到最优的策略, 引入 ϵ -贪心算法, 以 ϵ 的概率从所有的动作中均匀地随机选择一个动作, 以 $1-\epsilon$ 的概率选取当前最优动作, 将已经确定的策略标记为“原始策略”, 在原始策略中使用了 ϵ -贪心算法的策略记为(2):

$$\pi^\epsilon(x) = \begin{cases} \pi(x) & \text{以概率 } 1-\epsilon \\ A \text{ 中以均匀概率选取的动作, 以概率 } \epsilon \end{cases} \quad (2)$$

对于原始策略 $\pi = \operatorname{argmax}_a Q(x, a)$, 其中 ϵ -贪心算法 π^ϵ 中当前最优动作被选中的概率是 $1-\epsilon + \epsilon/|A|$, 而每个非最优动作被选中的概率是 $\epsilon/|A|$, 因此逃跑者在运动中的每个动作都有可能取代, 这样就能保证经过多次采样将会产生不同的路径。算法中奖赏均值采用增量式计算方式, 每获得一条新的轨迹, 就会将该轨迹的 Q 值进行更新, 将其调整到建立的博弈方法的追捕策略中。

2.5.3 追捕团队成员避障策略

追捕团队成员在环境中进行抓捕以及逃跑者在进行逃避的过程中, 在 t 时刻其要运动到的位置可能被其它物体占领, 这就有可能会发生碰撞, 因此就需要根据所处环境的约束, 进行实时有效的避障。传统的人工势场法的基本思想就是将追捕团队所处的环境充斥着混合势力场, 环境中的逃跑者充斥着引力势场, 方向有追捕者指向逃跑者; 环境中的障碍物以及各个追捕者充斥着斥力势场, 方向是由障碍物指向追捕者及逃跑者。分析传统人工势场法易出现局部极小点和目标不可达的原因, 文献[19,20]给出了一种改进后的合力公式(3)如下:

$$\vec{F}_{all} = \frac{\vec{F}_g}{|\vec{F}_g|} + \alpha \cdot \overline{derc} + \beta \cdot \sum_{i=1}^n \vec{F}_{oi} \quad (3)$$

式中: \vec{F}_{all} 为合力, \vec{F}_g 为虚拟目标点对 Agent 的引力, α 为方向向量的增益系数, \overline{derc} 为单位方向向量, β 为斥力增益系数, \vec{F}_{oi} 为障碍物点 i 对 Agent 的斥力。

这样既能保证追捕者趋向于逃跑者, 又能避免环境中的所有 Agent 与障碍物发生碰撞以及追捕者之间发生碰撞。

2.5.4 支付函数

追捕过程中, 双方需要一个标准来评估自己选择的策略的优劣, 博弈论中用支付函数这一功能实现这一功能, 追捕者的目标是: 1、判断逃跑者下一个时刻 t 的逃跑方向, 2、预测下一点的位置中是否存在障碍, 3、先将逃跑者在最短的时间内围住。而逃跑者的目标是有多条路径供其选择逃跑。双方具有不同的目标, 且双方在一方收到利益损害时一方并不一定有收益, 因此可以认为追捕者与逃跑者之间博弈为协作博弈。

由于追捕者对于逃跑者的威胁程度主要体现在距离的远近, 包围圈的好坏上, 因此在支付函数中包含以下三个影响系数:

1) 距离影响系数 K_d

当追捕者距离逃跑者的距离越近, 那么它对逃跑者的威胁系数就越大, 反之则越小。以此来定义距离影响系数(4):

$$K_d = n \times d_0 / \sum_{j=1}^n D_{p_j Et} \quad (4)$$

其中, n 为追捕环境中追捕者的数量, $D_{p_j Et}$ 表示 t 时刻第 j 个追捕者与逃跑者之间的距离。

2) 有效包围系数 K_c

由于成功追捕的条件是逃跑者在其周围已经没有其选择的逃跑方向, 并且其可能的走步数在逐渐减少, 此时的状态是追捕者以及在逐步的接近逃跑者, 并且已经在形成围捕的局势。以此来定义有效包围系数(5):

$$K_c = \sum_{j=1}^n D_{p_j Et} / n \times Dir \quad (5)$$

式子中 Dir 为可供其选择的方向。



3) 速度变化系数 K_v

在环境中者的运动由于受到人工势场的影响, 其会受到合力 \vec{F}_{all} 的作用, 其次追捕者的初始位置是随机的, 有的追捕者就可能会出现在距离逃跑者较远的位置, 为了保证能快速形成围捕的趋势, 因此需要根据受到的合力 \vec{F}_{all} 以及距离逃跑者的距离 S 及时的调整速度, 其公式定义如下(3,6,7)。

$$\vec{F}_{all} = \frac{\vec{F}_g}{|\vec{F}_g|} + \alpha \cdot \vec{derc} + \beta \cdot \sum_{i=1}^n \vec{F}_{oi} \quad (3)$$

$$S = \sqrt{(tx - dx)^2 + (ty - dy)^2} \quad (6)$$

$$K_v = \alpha * S + \beta * F \quad (7)$$

式子中 S 表示追捕者与逃跑者之间的距离, tx, ty 为当前逃跑 Agent 的坐标位置, dx, dy 为当前追捕者的坐标位置。 α, β 为影响速度 v 的权重值, 在不同的局势下, 追捕的侧重点不同。例如在距离较远的情况下, 采取的策略为加速靠近为主; 在距离缩小到一定的范围之后, 采取的策略应该是形成包围圈为主。因此针对不同的情况设定不同的权重值。

定义支付函数 U :

$$U = \lambda d * K_D + \lambda c * K_C + \lambda v * K_V \quad (8)$$

式子中 λd 与 $\lambda c, \lambda v$ 分别代表不同的权重值, 其中 $\lambda d + \lambda c + \lambda v = 1$, 权重的大小要根据在环境中遇到的情况进行调整。

2.5.5 逃跑 Agent 的追捕策略选择算法

追捕与逃跑者在某 t 时刻选择各自的走步策略, 追捕者根据定义的支付函数可以分别计算出追捕双方在不同的策略选择下追捕者的支付矩阵 U_t 。将 t 时刻的支付矩阵表示如下(7):

$$U_t = \begin{bmatrix} U_{t11} & U_{t12} & \dots & U_{t1n} \\ U_{t21} & U_{t22} & \dots & U_{t2n} \\ \vdots & \vdots & \ddots & \vdots \\ U_{tm1} & U_{tm2} & \dots & U_{tmn} \end{bmatrix} \quad (9)$$

根据博弈论中矩阵博弈的基本定理, 一定存在混合策略意义下的解, 通过排除法求解此矩阵, 可得到 t 时刻局中人的最优策略。

2.5.6 基于虚拟行动的 Agent 行为选择算法

由上述的多 Agent 协作追捕方法中, 多个 Agent 通过学习收敛到纳什均衡 Q 值。然而在协作追捕的博弈模型 G 的环境中, 由 Nash 定理可知, 博弈 G 至少存在一个 Nash 均衡解, 因此每个 Agent 通过学习都可能存在有多个纳什均衡的情况, 当多个 Agent 存在多个纳什均衡解时, 就需要每个 Agent 都会选择同一个纳什均衡解。

本节要考虑的问题是当博弈中存在多个纳什均衡解时, Agent 如何通过合适的策略保证最终选择同一个均衡解。再

此引入博弈学习中虚拟行动过程的概念, 这种学习模型将有利于解决存在多个均衡解的问题。

虚拟行动模型中, 多个 Agent 处于有限重复博弈中, 每个 Agent 都会根据对手 Agent 的历史行为, 对在当前阶段对手 Agent 行动的概率分布进行预测和评估, 并且会选择一个最优化其预测支付的行动。其评估的特定形式如下:

Agent i 有一个初始的加权函数, $k_0^i: S^{-i} \rightarrow R+$, 每次当对手 Agent 选择策略 S^{-i} 时, 通过给每个对手相应的策略权重加 1 对该函数进行调整, 即:

$$k_t^i(s^{-i}) = k_{t-1}^i(s^{-i}) + \begin{cases} 1 & s_{t-1}^{-i} = s^{-i} \\ 0 & s_{t-1}^{-i} \neq s^{-i} \end{cases} \quad (10)$$

在阶段 t , Agent 赋予其他 Agent 采取策略 S^{-i} 的概率为:

$$P_t^i(S^{-i}) = \frac{k_t^i(s^{-i})}{\sum_{\tilde{s}^{-i} \in S^{-i}} k_t^i(\tilde{s}^{-i})} \quad (11)$$

在虚拟行动中, Agent 仅仅跟踪对手 Agent 的行动频率是不行的, 还需要学习到这些概率分布, 因此 Agent 应该渐进的获取概率分布时相应的效用 U 。用 D_t^{-i} 表示 Agent i 的对手的行動的经验分布。

$$\hat{U}_t^i = \max_{\sigma^i} u^i(\sigma^i, D_t^{-i}) \quad (12)$$

定义在虚拟行动中某一时刻 t , Agent 对其所评估的对手 Agent 的行为策略而言最优的行动集合

$$BR_t^i = \max U_i(s^i, s_t^{-i}) \quad (13)$$

在重复博弈的过程中, 每个 Agent 相信对手 Agent 的行为是一个未知的但固定概率分布的多重随机变量序列, 这种序列他们可以通过公式 10, 11 从行为历史中学到。Agent i 在学习时刻 t 实际行为选择是他在 t 时刻关于对手 Agent 行为策略的最优行动如公式 13。

算法 2: 基于虚拟行动的 Agent 行为选择算法

do {

给定博弈具有有限的策略 S_1, S_2, \dots 和支付函数 u_1, u_2, \dots ;

$T = t_i$;

在 t 时刻: Agent i 观察其他对手 Agent 的历史行为, 利用公式 10, 11 计算相应的行为概率;

利用公式 12 计算在该时刻 Agent 的效用值;

利用公式 13 计算 Agent 行为的最佳集合。从该集合中选择一个行为;

$t = t + 1$;

}while(直到收敛)

将上述的方法进行整合, 用一个完整的算法 3 表示整个多 Agent 协作追捕算法的过程。如下所示

算法 3:

步骤 1: 初始化生成逃跑者和追捕者; 令其动作-状态 $Q(x, a) = 0$, $\text{count}(x, a) = 0$, $\pi(x, a) = 1/|A|$ 。

步骤 2: 根据算法 1 建立追捕团队, 随即构建博弈模型;



- 步骤 3: 根据成功捕获的条件判断逃跑者是否被捕获, 如果已经被捕获, 则结束任务; 如果不是, 则进行第 4 步;
- 步骤 4: 利用公式(4)(5)(7)控制速度变化, 并向逃跑者的位置进行移动;
- 步骤 5: 执行策略 π , 产生轨迹 $\langle x_0, a_0, r_1, x_1, a_1, r_2, \dots, x_{T-1}, a_{T-1}, r_T, x_T \rangle$
- 步骤 6: $\text{count}(x_t, a_t) = \text{count}(x_t, a_t) + 1$, 利用公式(1) $Q(x, a)$ 策略进行更新;
- 步骤 7: 是否形成完整 T 步运动轨迹, 否则继续进行 4, 5, 6, 7 步;
- 步骤 8: 对所有可见的状态 x 利用值函数公式(2)得到完整策略;
- 步骤 9: 根据第 8 步获得的学习策略得到支付值, 形成支付矩阵 P_t ;
- 步骤 10: 由第 9 步得到的 t 时刻的支付矩阵 U_t 求取其纳什均衡解, 得到 t 时刻的较优走步策略;
- 步骤 11: 执行算法 2 虚拟行动方法找到协作追捕的最优解;
- 步骤 12: 执行走步策略, 追捕成功, 返回第 3 步。

3 实验仿真

为了充分验证本文方法的有效性和合理性, 进行三种方法的仿真实验的对比。实验的仿真环境为一个具有多处不同大小障碍物(房屋, 人, 山, 河流等)的仿真实环境平台, 环境中三个追捕者, 一个逃跑者。

假设所有的 Agent 具有以下特质:

1. 追捕过程中所有的 Agent 均只能活动在具有边界的地形中。其运动可选择的方向为 360° 。
2. 所有的 Agent 对环境中的障碍物以及每个 Agent 等位置信息已知, 并且所有的 Agent 在环境中初始位置随机, 初始化所有参与追捕者的速度值均为 2cm/s , 追捕者在速度上比逃跑者的速度要大, 其追捕者的速度变化要根据速度影响系数 K_v 变化。
3. 假设当前时刻 t 是追捕者的所占据的位置 $X_{a,b}(0 < a < 720\text{m}, 0 < b < 720\text{m})$, 追捕者与逃跑者每次移动一个身位(自身的宽度 20cm)* V (速度), 其移动的方向选择为 360° , 用 $H_{i(t)}$ 表示此事 $X_{a,b}$ 处的 Agent 可以运动到的位置, 有公式(14):

$$H_i(t) = \begin{cases} X_{a,b} \\ X_a + \cos(\text{Dir} * \pi / 180) * 20 * V, X_b + \sin(\text{Dir} * \pi / 180) * 20 * V \end{cases} \quad (14)$$

其中 Dir 为的转向度数, 并且规定转向度数一次为 5° , π 为 π , 20 为追捕者以及逃跑者的身长, V 为速度值。

在上述同样的环境中进行基于博弈方法的多 Agent 追捕, 基于强化学习的多 Agent 追捕, 以及本文提出的基于博弈论及 Q 学习的多 Agent 协作追捕算法的研究。每种算法进行试验 50 次, 整理结果数据如表 1 所示, 表中数字代表追捕者从开始追捕逃跑者开始, 到完成追捕所用的时间, 从中可以看出本文算法的协作追捕效率更高。

在图 4(a, b)中具有障碍物的环境中, 多 Agent 追捕团队从初始位置开始, 在人工势场合力的影响下, 经过学习逃跑者的逃跑路径形成 T 步策略, 求解协作追捕博弈, 得到 Nash 均衡解, Agent 选择均衡策略进行围捕。从整个追捕的时间上可以得出本文的算法较优, 且较为稳定。图 4(a)中可知, 环境中的障碍物可以作为围捕逃跑者的有力条件, 图 4(b)中为追捕团队不利用障碍物进行的围捕。

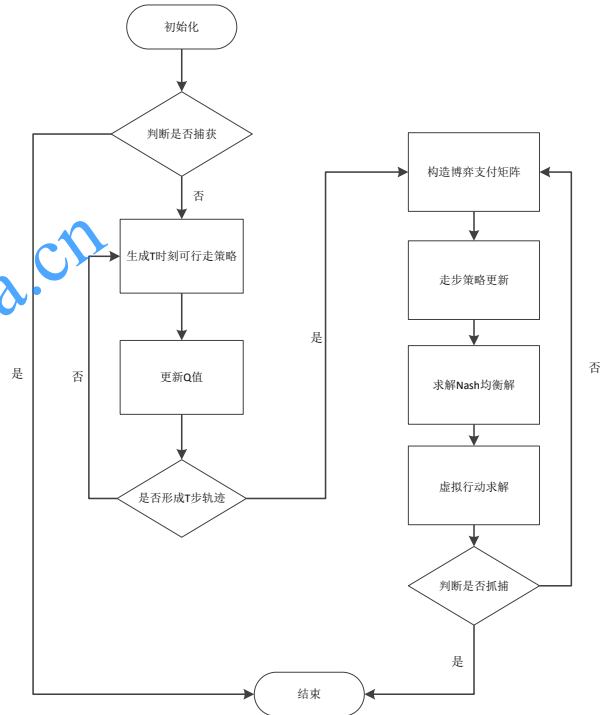
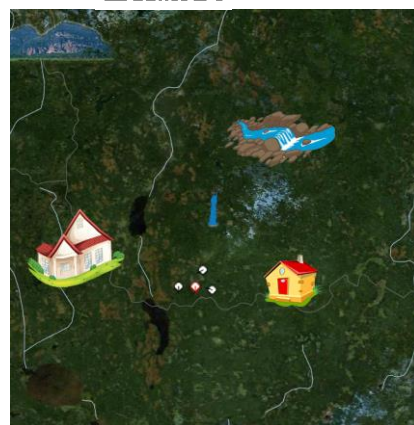


图 3 多 Agent 追捕单目标猎物流程图
Fig 3 Multi-Agent Hunting Single Target Prey Flowchart



(a)



(b)

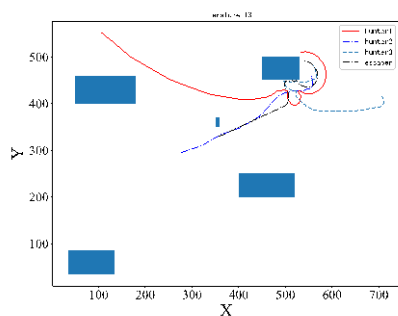
图 4 多 Agent 追捕环境

Fig 4 Multi-Agent Hunting Environment

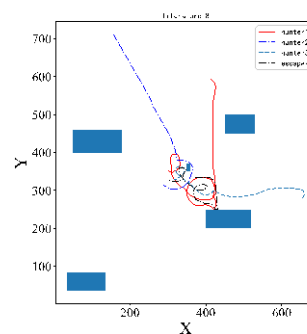
表 1 五十次实验数据表

Table 1 Fifty experimental data sheets

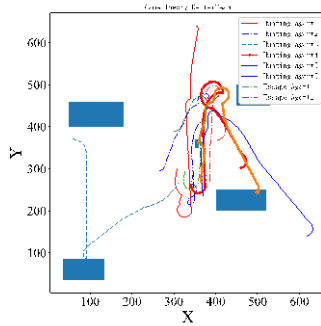
文献	50 次实验数据	平均/s
文献 13	57.4, 50.9, 49.1, 46.2, 45.6, 39.1, 36.1, 35.9, 33.5, 31.1, 30.2, 29.8, 29.8, 29.0, 28.6 28.6, 28.5, 27.8, 27.2, 26.4, 26.2, 25.9, 25.3, 25.1, 24.3, 23.6, 23.5, 23.2, 23.2 22.5, 22.1, 22.1, 21.9, 21.5, 21.0, 20.9, 19.2, 18.4, 17.9, 17.5, 17.5, 17.1, 17.0, 16.3 16.3, 15.9, 15.7, 12.1, 11.4	26.414
文献 8	80.0, 68.3, 63.3, 61.8, 60.8, 55.6, 49.5, 49.5, 45.2, 41.2, 38.7, 37.9, 37.5, 36.9, 36.2 35.6, 33.9, 33.9, 30.8, 30.1, 29.6, 29.6, 28.9, 28.9, 28.5, 27.8, 26.8, 26.2, 25.8, 24.0 24.0, 23.5, 22.7, 22.7, 22.5, 22.2, 21.6, 19.4, 19.1, 18.5, 17.1, 16.8, 15.9, 14.9, 14.8 14.4, 13.9, 13.6, 12.0, 10.2	31.252
文献 22	76.2, 68.7, 52.9, 52.5, 47.3, 44.0, 40.1, 38.2, 35.3, 34.0, 33.9, 31.9, 31.8, 30.1, 28.4 28.2, 27.9, 27.8, 25.0, 24.6, 23.8, 23.3, 23.2, 23.0, 22.3, 21.7, 20.7, 20.4, 20.4, 20.2 20.0, 19.9, 18.9, 18.8, 18.8, 18.2, 18.0, 17.5, 17.0, 15.7, 15.0, 14.5, 14.3, 13.6, 13.5 12.8, 11.9, 11.0, 10.4, 7.3	26.098
本文算法	62.7, 41.7, 41.2, 39.9, 39.5, 36.2, 36.0, 35.3, 34.5, 32.8, 32.3, 31.6, 31.1, 29.9, 29.4 29.3, 28.4, 26.7, 26.7, 26.6, 26.3, 24.9, 24.7, 23.0, 22.6, 21.9, 21.7, 21.7, 20.8, 19.7 19.4, 19.4, 19.2, 19.1, 19.0, 17.4, 16.6, 16.4, 16.0, 14.7, 14.4, 14.2, 13.8, 13.6, 12.8 10.0, 9.9, 8.5, 8.4, 8.2	24.202



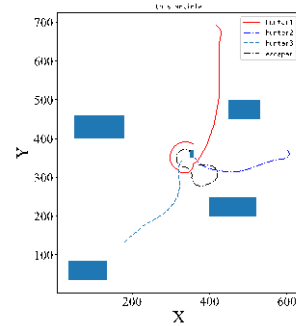
(a) 文献 13



(b) 文献 8



(c) 文献 10



(d) 本文

图 5 模拟实验追捕交互路径图

Fig 5 Interactive path diagram of simulation hunting

图 5 中(a)是利用文献[13]中博弈论的方法在本平台中进行的追捕,图 5(b)是利用文献[8]的方法在本平台中完成的追捕,图 5(c)是利用文献[10]的方法在本平台中完成的追捕,图 5(d)是所提出的方法在本平台中完成的追捕。在图 5 中可以看出,追捕者在前半段发现逃跑者之后都尽可能的去贴近逃跑者,速度变化为会根据公式(7)动态改变,在后半段中其包围系数 K_c 会变得比较重要,可以直观的看出(d)中后半段的追捕完成的效率较高。

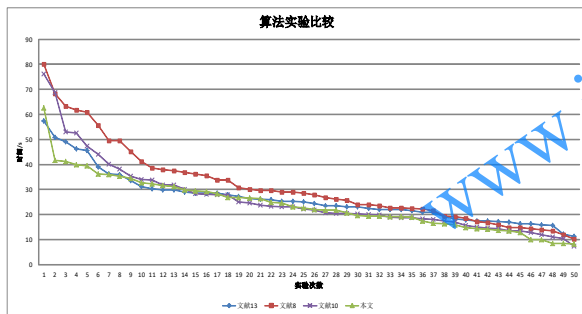


图 6 三种算法 50 次实验对比

Fig 6 Comparison of three algorithms in 50 experiments

为了验证算法 3 的最优,使用随机的策略进行了 50 次实验,对三种算法进行对比,如图 6 所示。从数据中可以看出算法 3 具有较高的适应性。

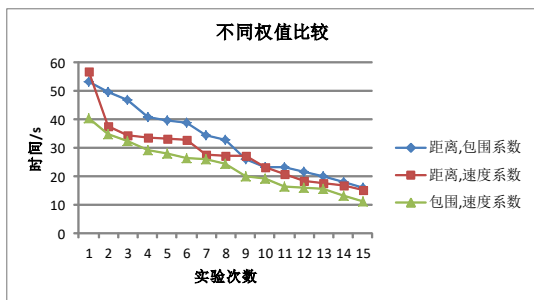


图 7 同等条件下不同权值 15 次实验对比

Fig 7 Comparison of 15 different experimental weights under the same conditions

图 7 实验验证算法 3 在同等环境下,由于支付函数权重值的不同,对追捕效率的影响。使用随机的策略进行了 15 次实验,分别每次改变两种参数。根据图 5 中由各算法在本平台中进行的追捕模拟运动轨迹,在前半段中进行的贴近逃跑者运动中,速度影响的因素 K_v 极为重要,在形成围捕之势时,包围系数 K_c 的重要性就显示出来。如图 7 所示,从结果中可以看出在同等环境下距离,包围程度,以及速度的权重不同,会使追捕的效率产生较大差异。

4 结语

本文提出了一种基于博弈论及 Q 学习的多 Agent 协作追捕算法,考虑到 Agent 具有的学习能力,在追捕过程追捕者与逃跑者会因为策略选择相互受到影响。首先建立协作追捕团队,并构建协作追捕的博弈模型;其次根据逃跑者的运动路径进行有限次的学习,每次学习都去更新其动作状态值,建立逃跑者的具有累积奖赏的 T 步路径轨迹,调整为追捕者可供选择的可执行策略集;最后求解协作博弈模型,得到 Nash 均衡解,追捕者选择较优的追捕行动策略并完成追捕,增强了博弈方法的环境适应性,同时研究者加入了虚拟行动行为选择算法,再出现有多个 Nash 均衡的情况下,使得 Agent 能够选择最优的均衡策略。仿真实验,验证了该方法既能快速捕获逃跑者和避开障碍物,又能适应当前的环境。

参考文献

- [1] Tuyen LP, Viet HH, et al Univector field method-based multi-agent navigation for pursuit problem in obstacle environments[J].Central South University Press and Springer-Verlag Berlin Heidelberg 2017,24(04):1002-1012.
- [2] Souidi, M E, Piao, S, A New Decentralized Approach of Multi-agent Cooperative Pursuit Based on the Iterated Elimination of Dominated Strategies Model[J], Mathematical Problems in Engineering,2016
- [3] Souidi M E, Songhao P, Guo L, et al. Multi-agent cooperation pursuit based on an extension of AALAADIN organizational model[J]. Journal of Experimental & Theoretical Artificial Intelligence, 2016, 28(6):14.



- [4] Pei, HQ, Chen, SM, Lai, Q, Multi-target consensus circle pursuit for multi-agent systems via a distributed multi-flocking method[J], International Journal of Systems Science, 2016, 47(16):3741-3748.
- [5] Bhadauria, D, Klein K, Isler V, et al. Capturing an evader in polygonal environments with obstacles: The full visibility case[J]. The International Journal of Robotics Research, 2012, 31(10):1176-1189.
- [6] Souidi, M E, Siam, A, Pei, ZY, Multi-agent pursuit coalition formation based on a limited overlapping of the dynamic groups[J], Journal of Intelligent & Fuzzy Systems, 2019, 36(6):5617-5629
- [7] 李珣, 潘启树, 周浦城, 洪炳溶. 未知环境下多机器人协作追捕算法[J]. 电子学报, 2011, 39(3): 568-574 (LI J etc. Multi-Robot Cooperative Pursuit Algorithm for in An Unknown Environment [J]. Acta Electronica Sinica, 2011, 39(3):568-574)
- [8] Asl Z D, Derhami V, Yazdian-Dehkordi M. A new approach on multi-agent Multi-Objective Reinforcement Learning based on agents' preferences[C]// 2017 Artificial Intelligence and Signal Processing Conference (AISP). IEEE, 2017.
- [9] Bilgin A T, Kadioglu-Urtis E. An approach to multi-agent pursuit-evasion games using reinforcement learning[C]// 2015 International Conference on Advanced Robotics (ICAR). IEEE, 2015.
- [10] Zuhair Q M, PIAO S H, JIANG H Y, et al. A novel approach for multi-agent cooperative pursuit to capture grouped evaders[J]. The Journal of Supercomputing, 2018.
- [11] 郑延斌, 陶雪丽, 段领玉, 李波. 基于博弈论及惩罚机制的多 Agent 协作控制算法 [J]. 河南师范大学学报 (自然科学版), 2015, 43(06):146-151. (Zheng Y B, Tao X L, etc. The Algorithm for Multi-agents Cooperation Controlling Based on Game Theory and Punishment Mechanism [J]. Journal of Henan Normal University (Natural Science Edition), 2015, 43(06):146-151.)
- [12] FANG B L, ZHU J L, et al. Multi Self-interested robot pursuit based on quantum game theory[C]// 2017 Chinese Automation Congress (CAC). IEEE, 2017.
- [13] 晏亚林. 基于博弈论的多机器人追捕问题的研究[D]. 哈尔滨: 哈尔滨工程大学, 2014. (Yan Y L. Research on multi-robot hunting based on game theory[D]. Harbin: Harbin Engineering University 2014)
- [14] Hakli, R. Cooperative Human-Robot Planning with Team Reasoning [J]. International Journal of Social Robotics, 2017, 9(5):643-658
- [15] ZHANG C Y, LI C Y, Zhu Y Y, ZHANG J L, Dynamics of Task Allocation Based on Game Theory in Multi-Agent Systems[J], IEEE Transactions on Circuits and Systems—II: Express Briefs, 2019, 66(6):1068-1072
- [16] 现代应用数学手册编委会. 现代应用数学手册, 运筹学与最优化理论卷[M]. 清华大学出版社, 1998. (Editorial Board of Modern Applied Mathematics Handbook. Handbook of Modern Applied Mathematics, Operations Research and Optimization Theory Volume [M]. Tsinghua University Press, 1998.)
- [17] Mnih, V, Kavukcuoglu, K, Silver, D, et al Human-level control through deep reinforcement learning [J]. Nature, 2015, 518(7540):529-533.
- [18] YUAN Y L, YU Z L, GU Z H et al., A novel multi-step Q-learning method to improve data efficiency for deep reinforcement learning[J], Knowledge-Based Systems 2019, 175:107-117
- [19] 程志, 张志安, 李金芝, 江涛. 改进人工势场法的移动机器人路径规划 [J/OL]. 计算机工程与应用: 1-6 [2019-06-29]. (Cheng Z, Zhang Z A, etc. Mobile Robots Path Planning Based on Improved Artificial Potential Field [J/OL]. Computer Engineering and Applications: 1-6 [2019-06-29].)
- [20] JIA Y W, YIN G Q et al. Path planning for mobile robot using the novel repulsive force algorithm [J]. IOP Conference Series: Earth and Environmental Science, 2018, 108(5).
- [21] Lavalle S M. Robot Motion Planning: A Game-Theoretic Foundation [J]. Algorithmica, 2000, 26(3-4):430-465.
- [22] 肖文雅, 尚艳玲. 一种基于多 Agent 的有效负载均衡的 WebGIS 体系模型 [J]. 河南师范大学学报 (自然科学版), 2015, 43(04):151-156. (XIAO W Y, SHANG Y L. A Loading-Balancing Framework for Distributed WebGIS Based on Multi-Agent [J]. Journal of Henan Normal University (Natural Science Edition) 2015, 43(04):151-156)

This work is partially supported by National Natural Science Foundation of China (No.U1604156), Henan Normal University Youth Fund (2017QK20).

ZHENG Yanbin, born in 1964, Ph. D., professor. CCF member His research interests include virtual reality, multi-agent system, and game theory.

FAN Wenxin, born in 1994, M. S. candidate. His research interests include virtual reality and multi-agent system.

HAN Mengyun, born in 1993, M. S. candidate. Her research interests include virtual reality and Chinese character recognition technology.

TAO Xueli, born in 1978, M. S. instructor, His research interests include virtual reality and multi-agent system.