文章编号:1000－2367(2015)06－0146－06 DOI:10.16366/j.cnki.1000－2367.2015.06.027

基于博弈论及惩罚机制的多 Agent 协作控制算法

郑延斌[a,b], 陶雪丽[a,b], 段领玉[a], 李 波[a]

(河南师范大学 a.计算机与信息工程学院;b.智慧商务与物联网技术河南省工程实验室,河南 新乡 453007)

摘 要:针对协作过程中自利的 Agent 选择回报更高的任务去执行,导致当前任务不能正常执行的问题,提出一种基于博弈论及惩罚机制的协作控制方法.在形成协作时优先选择信誉值较高的 Agent,在协作执行过程中利用惩罚机制来约束退出协作的 Agent 的行为.仿真结果表明,该算法能有效地避免 Agent 在协作中随意的退出,确保协作任务的顺利执行,提高协作成功率.

关键词:博弈论;协作;惩罚机制;信誉值;Multi-Agent

中图分类号:TP18 **文献标志码**:A

多 Agent 协作是指多个 Agent 之间进行交互,有效的协调合作,共享资源,共同解决整体问题的过程,协作问题是多 Agent 系统研究的核心问题之一.许多学者提出了一些不同的方法,如:Kim[1]等对 Markov 模型进行了拓展,研究了一种基于动作重复过程的多智能体 Q 学习方法,并讨论算法的收敛性.Takahashi[2]等提出通过系统动态环境的反馈来分配 Agent 角色的概念,框架中无直接合作的计划,但每个 Agent 都会根据其他 Agent 的行为所产生出来的各类环境反馈来选择自己的行为,从而实现协作的行为.由于多 Agent 系统环境的复杂多变,难以提取出有效的环境反馈定义来应用此思想.Kazunori [3]等将协作问题描述为多 Agent 马尔可夫决策过程(Multi-Agent Markov decision processes,MMDP),采用强化学习来获得 Agent 的策略,利用学习机制来减少协商过程中的不确定和不稳定因素,但由于环境的复杂性,使得学习周期变得十分长.Greenwald A[4]等用随机博弈和 Nash 均衡研究了多 Agent 协作系统.TOLEDOC B[5]等以联合意图作为智能体协作基础,建立了复杂动态环境下的协作框架.唐贤伦[6]等提出将距离因子和控制因子引入蚁群算法的多 Agent 协作策略,为解决多 Agent 系统在协作中可能出现的任务死锁及协作效率不高提供了解决途径.肖喜丽[7]把协同进化算法应用到多 Agent 协作问题中,通过个体适应度评价与其他种群协作,并把协作行为应用到目标领域后对个体进行评估,虽然能更好地协作,但是选择代表性个体以及如何减少计算量的问题有待解决.任韶萱[8]提出一种基于蚁群算法的多智能体协作算法.然而上述方法大都基于 Agent 团体理性的假设,即 Agent 本质上愿意协作的,忽略了 Agent 的个体理性的问题,或者不允许 Agent 在任务执行的过程退出.由于复杂环境中,任务是动态加入的、Agent 自身能力和偏好也会发生变化,因此 Agent 为了追求自身利益最大化,有可能退出当前正在执行的任务,去选择执行其他任务,导致当前任务不能执行,或者执行效率降低.为了使协作任务正常执行,确保协作的稳定性,本文基于博弈论方法,提出了一种惩罚机制,通过不断调整惩罚系数来实现动态的多 Agent 之间的协作稳定性(协作稳定性指一旦多个 Agent 达成了相互协作完成复杂任务的协议,则各个 Agent 在后续的任务执行过程中会遵守协议直到任务完成),Agent 通过对惩罚的大小程度判断选择继续协作或者接受惩罚并退出协作,实现均衡协作.

1 博弈论基础

博弈论(Game Theory)又称“对策论”,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

收稿日期:2014-10-19;**修回日期**:2015-06-22.

基金项目:河南省重点科技攻关项目(132102210537;13210221053)

第 1 作者简介(通信作者):郑延斌(1964－),男,河南内乡人,河南师范大学教授,博士,研究方向为虚拟现实、多智能体系统、对策论,E-mail: zybcgf@163.com.

作决策的问题.

定义 1　对于博弈可以用一个四元组进行表示，即

A：协作参与者 $A=\{a_1,a_2,\cdots,a_n\}$，是指协作的各方.

S：各协作参与者可行策略集 $S=\{S_1,S_2,S_3,\cdots,S_n\}$，是协作参与者可能采取的所有行为策略.

I：博弈信息，指的是参与者所拥有的信息特征.

U：收益函数是指博弈过程中参与协作博弈对象的收益，可以用 $U=\{u_1,u_2,u_3,\cdots,u_n\}$ 进行表示.

定义 2　纳什均衡：对于给定的博弈，如存在一个可行策略 $\tilde{s}=(\tilde{s}_1,\tilde{s}_1,\cdots,\tilde{s}_n)\in S$，使得：$\forall i\in A$，$\forall s_i'\in S_i$ 都有：$u(s_i',\tilde{s}_{-i})\leq u(\tilde{s}_i,\tilde{s}_{-i})$，其中 $\tilde{s}_{-i}=(\tilde{s}_1,\tilde{s}_2,\cdots,\tilde{s}_{i-1},\tilde{s}_{i+1},\cdots,\tilde{s}_n)$，则称可行策略 $\tilde{s}$ 是博弈 G 的一个纳什均衡.

纳什均衡是理性局中人之间利益冲突时达到的一种相对稳定的状态，且没有一个行为主体可以单方面改变这种状态. 因此，纳什均衡中每个成员的策略选择都是对其他成员策略选择的最佳响应.

2　多 Agent 协作控制方法

在动态复杂的环境中，随着任务的执行，可能出现新的任务，Agent 的能力、偏好也有可能发生变化. 由于 Agent 的自利性，当新任务出现时，Agent 可能放弃当前正在执行的任务，而去选择回报更高的任务，从而使得当前的协作不能完成，这实际上伤害了其他参与协作的 Agent 成员. 因此为了解决这个问题，确保协作的稳定性，本文提出了一个新的协作控制算法，首先在协作成员的选择过程中考虑 Agent 的信誉，优先选择信誉值高的 Agent；其次，为了维护系统稳定性和其他协作 Agent 的利益，给予在协作过程中退出协作的成员以惩罚；最后为了提高协作效率，对每件任务设定时限，以便当协作 Agent 数量达不到任务要求时 Agent 放弃等待，避免 Agent“死等”状态. 这样如果一个 Agent 需要退出协作时，需要判断是否接受惩罚而退出协作任务或迫于惩罚而继续协作，当然惩罚所得的报酬用于补偿参与该次协作的其他 Agent.

2.1　信誉值

为了提高协作的效率，本文提出了智能体的信誉值作为智能体选择协作智能体的一个参考因素，信誉值是用来描述 Agent 完成任务的耐心因素. 当 Agent 不能完成自己选定的任务时，其信誉值就会受损，同时也会对选择这一任务的协作 Agent 信誉值造成影响. 当 Agent 成功完成协作任务时，其信誉值也会相应增加. 其更新算法为：

$$\tau=\tau'-\frac{C_q+e(C_f+C_p)-C_r}{C_a},\tag{1}$$

其中 τ' 为 Agent 上次团队协作时的信誉值，且初始值为 1，C_q 表示 Agent 主动退出协作的次数，C_f 表示因为其他 Agent 退出当前协作而造成未能完成任务的次数，$e\in(0,0.5]$ 为调节系数. C_p 表示经过 T_l 时间的协作仍然不能完成当前协作任务，为了提高团队协作效率，由系统解散当前协作团队的次数，C_r 为成功完成任务的次数，C_a 为合作执行任务的总数.

由(1)式知，成功执行任务的次数越多，Agent 的信誉值会越大，当信誉值超过 1 时，充分说明 Agent 协作成功的次数多，可以信赖，故为了方便起见，设 τ 在[0,1]之间，当 $\tau>1$ 时，取 $\tau=1$.

2.2　惩罚

协作的稳定性是确保协作任务顺利完成的关键. 由于 Agent 的自利性，为了追求自身利益的最大化，可能随着协作的进行，参与 Agent 不满足于对现阶段的协作，或者经过长时间的协作仍然不能完成任务，参与 Agent 可能中途退出并参与收益比较大的任务，从而导致协作团队效率低下.

本文允许协作者动态加入或退出系统以改善协作团队的协作效率，但如果 Agent 在发现自己可能获得更大利益后就可以不加任何条件的退出当前协作而加入新协作任务，反而会破坏协作的稳定性，使当前所在的协作的其他 Agent 蒙受损失，这样将会形成恶性循环，导致协作团队的整体性能下降. 所以在改善协作效率的同时，还应当保证协作的相对稳定性. 为此，引入“惩罚”的概念，以维护协作任务的正常完成. 退出当前协作的 Agent 必须接受“惩罚”以弥补其他协作 Agent 的损失. 如果某个 Agent 认为自己退出当前协作加入

其他协作而获取的收益,即使在接受惩罚的条件下,仍多于当前协作可能获得的收益,这时 Agent 会选择退出当前协作并接受惩罚. 如果 Agent 发现退出后的收益小于当前收益,就选择留在当前系统继续协作. 惩罚机制由下式确定:

$$P(n_i)=\begin{cases} ku_i+\tau^{\Delta t}\delta, \Delta t \le T, \\ ku_i, \Delta t > T, \end{cases} \tag{2}$$

其中 $k \in [0,1]$ 为惩罚系数,$\delta \ge 0$ 为最低惩罚值,τ 为信誉值,$\Delta t = t - t_0$ 为协作持续的时间,t 为当前时间,t_0 为协作初始时间,T 为任务预期完成时间,u_i 为完成任务后将获得的利益.

2.3 基于博弈论的多 Agent 协作控制方法

Agent 的自利性决定了多个 Agent 参与协作任务的执行过程是多个 Agent 相互博弈的结果,一旦协作团队形成,在静态环境中 Nash 均衡是确保该协作稳定的基础,Agent 不能通过偏离 Nash 均衡而获得更多的回报. 然而在动态的环境中,新任务可能随时加入到环境中,协作团队中的 Agent 可能受到利益的诱惑而放弃当前正在执行的任务,从而去选择获利更高的任务,导致现有的任务不能执行,同时对参与该次协作的其他 Agent 也是一种伤害. 惩罚机制的引入使得 Agent 不能随意地退出正在协作的团队,故惩罚机制是动态任务环境中确保协作稳定性的基础. 图 1 给出了基于该思想的个体 Agent 结构模型,个体 Agent 通过环境感知模块感知外部环境,通过计算模块来实现信誉值及惩罚等的计算,通过推理决策模块做出决策并能通过通讯模块与其他 Agent 进行交互.

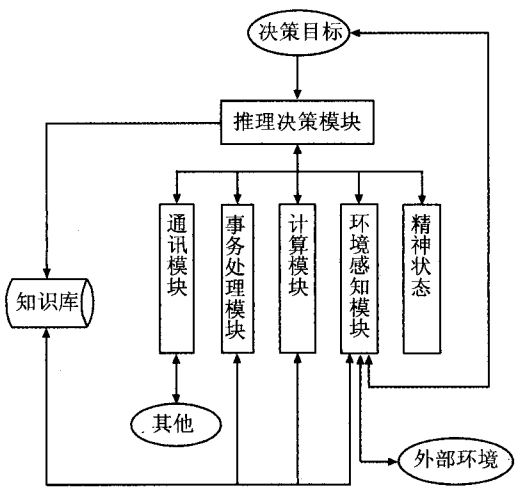

图1 Agent结构模型

本文提出的协作算法分为两个部分,第一部分为协作形成方法,第二部分为协作维持方法. 基本过程是,当 Agent 发现任务后,首先向其他 Agent 发出协作请求,在 t 时隙内等待回应,若超出 t 则放弃,否则 Agent 根据协作 Agent 的距离及信誉值来选择协作团队,团队形成之后各成员相互协作完成任务. 为了提高任务完成效率,Agent 可以根据具体情况做出自己的决策,但为了保持系统的稳定性采用惩罚机制约束 Agent 的形为.

2.3.1 协作团队形成方法

设环境中任务描述为:$T(u(t),s,z(t))$,其中 $u(t)$ 是完成任务后获得效益,t 是博弈的次数,s 为执行任务需要的 Agent 个数,$z(t)$ 为任务的特征信息,如位置、类型等信息. 整个系统中由 N 个理性 Agent 组成,即 $P=\{Ag_1,Ag_2,Ag_3,\cdots,Ag_n\}$,下面是基于博弈论和信誉值的团队形成算法.

(1)初始阶段:$t=1$,对于所有的 Ag_i 初始阶段在系统中随机找寻未完成的任务 Ta_j,用 SN 来标记 Ag_i 是否找到 Ta_j,若 SN 值为真则 Ag_i 在系统中发出协作请求;设 Agent x 发现未完成的任务;

(2)Agent x 向系统中所有其他 Agent 发送任务协作信息 $T(u(t),s,z(t))$;

(3)博弈开始:每个 Agent 根据自己掌握的信息 I,即自己对任务和其他 Agent 行为选择的了解,求出自己加入团队可能得到的回报 u_i,选择自己的行为策略;

(4)若达到 Nash 均衡,则博弈结束,转(9);

(5)Agent x 观察其他成员的选择,当选择加入团队的成员数量 $m>=s$ 时,博弈过程结束,转(9);

(6)$t=t+1$;

(7)如果 $t>T_0$(T_0 为一个常量,预先设定的博弈次数),则团队形成失败,转(11);

(8)进行任务效益的更新,转(2);

(9)Agent x 对选择加入团队的成员按照信誉值由高到低进行排序,选择前 s 个成员组成团队,并向这些成员发送成功加入团队的信息,向没有被选择的成员发送拒绝加入团队信息;

(10)团队形成,进入协作任务执行阶段.

(11)结束.

2.3.2 基于惩罚机制的协作维持算法

多 Agent 间的协作是开放式的，合理的协作团队应当具有足够的灵活性适应系统的动态变化. 如在协作执行的过程中，若某个 Agent 发现收益更大的新任务，允许 Agent 脱离当前协作团队，加入获利更大的团队，特别是对一些经过长时间合作仍不能完成的困难任务，随着协作团队代价的增加，放弃就会逐渐成为更优的选择. 但是，若团队中的 Agent 可以不加任何条件的退出当前协作团队，会使团队中的其它协作成员蒙受损失. 因此，本文提出惩罚机制对选择退出当前协作团队的 Agent 给予一定的惩罚，以补偿团队中其它协作成员蒙受的损失，保证协作团队的稳定性. 算法描述如下.

(1)判断团队是否形成，若没有形成则转(10)；

(2)若协作 Agent 需要退出当前协作，转(3)，否则转(4)；

(3)计算执行当前任务获得效益，执行新任务获得的效益－惩罚效益，Agent 判断是否退出当前协作，若确定退出，转(9)；

(4)各 Agent 保持协作状态，判断协作时间是否在 T_l 内，若不在 T_l 内，转(8)；

(5)计算 Nash 均衡策略，每个 Agent 按照均衡策略行动；

(6)判断协作任务是否完成，若没有完成，则转(2)；

(7)各 Agent 得到收益，更新 τ，转(10)；

(8)合作失败，更新 τ，转(10)

(9)惩罚处理，支付惩罚值，团队中其他 Agent 得到补偿，更新 τ；

(10)结束.

3 仿真实验

追捕问题涉及 Agent 之间的协作，通过多个移动 Agent(追捕者)之间的协作和协调完成追捕多个逃跑者的任务. 在追捕问题中，逃跑者需要多个追捕者协作才能捕获，不同的猎物通常需要不同数量的追捕 Agent 合作才能捕获. 发现逃跑目标后，对其类型及行为参数进行识别、分析，追捕 Agent 与合适的同伴形成协作追捕，追捕问题实现简单规则约束少，能够使许多复杂的问题简单化，在问题的解决上很具有代表性，因此本文使用追捕问题进行仿真实验. 为了检验所提方法的可行性和有效性，进行如下仿真实验：假设选择在 500×500 大小的矩形场地内，按 10×10 的大小划分成数量为 50×50 的栅格，障碍物是由系统随机生成的矩形实心区域，追捕者与猎物均为半径为 5 的圆形智能体，初始位置在每次实验开始前随机生成，追捕者和逃跑者的速度、视野和运动机会等各种条件都相同，如图 2，场地内分布着能力相同的 $n=14$ 个同质 Agent，图中红色圆点表示；分布着难度系数不同的 $m=4$ 个动态猎物，图中蓝色圆点表示；黑色不规则形状为障碍物. 当多个 Agent 形成的协作团队成功追捕猎物后，系统会随机产生不同难度系数的新猎物，使猎物总数保持不变，猎物自身带有自己的类型 R_{Ty}，且 $y\in m$，R_{Ty} 指定捕获猎物需要的最少追捕 Agent 数量，多 Agent 协作追捕猎物根据相应猎物的价值获得不同的收益. 系统所产生的动态猎物最少由 4 个 Agent 协作完成追捕，最多由 7 个 Agent 协作完成追捕，不考虑其他情况. Agent 在协作追捕过程当中能够动态的加入或退出.

图 3 给出的状态为 Agent 形成 3 个协作团队，并有一个猎物被捕获，猎物 1 由 4 个 Agent 协作所捕获，当猎物 1 被捕获后就从所在位置消失，下一时刻由系统随机产生不同难度系数的新猎物.

图 4 中系统随机产生新猎物，此时捕获成功的 4 个 Agent 协作捕获新的猎物，并发出协作请求，Agent9 放弃等待猎物 2 的协作请求，并同意新猎物 1 的协作请求，Agent11 退出猎物 4 的协作团队并接受新猎物 1 更大收益的协作请求.

为测试不同信誉值对追捕团队整体性能的影响，分别选用信誉值 τ 初值从 0 按步长为 0.1 增加到 1.0，同时分别进行 N={30,50,70,100}次实验. 在允许动态退出当前协作团队的情况下，选择任务完成情况作为性能评价指标，其中用任务完成率 Completion rate(Agent 参加并成功完成协作次数与 Agent 参加的协作总数的比率)来衡量任务的完成情况. 图 5 给出了不同信誉值下的任务完成率.

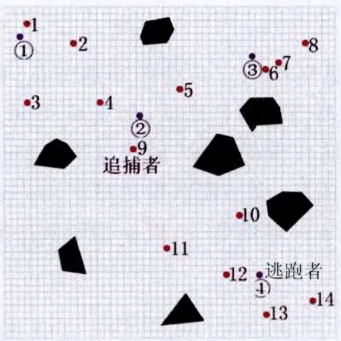

图2 追捕初始阶段状态

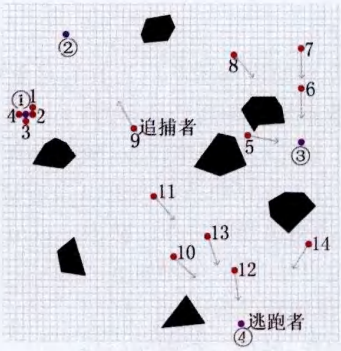

图3 Agent协作完成追捕

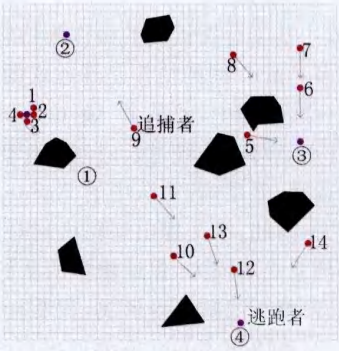

图4 Agent接受惩罚并退出

仿真结果表明,当信誉值 $\tau=0$ 时,此时追捕团队的协作性能最差,这是由于追捕者都是理性的自利 Agent,由于其短视行为,受其他协作团队的利益诱惑,为了争取更大的局部利益而不能完成当前团队协作,结果导致协作团队的整体性能严重下降. 完成率随着 τ 的增长而明显提高,说明多 Agent 的完成率随 Agent 的 τ 增长而得到了提高改进. 当 $\tau>0.8$ 时,其完成率接近为 1,说明随着自身信誉值的提高,Agent 的协作可靠性增强,因而追捕者完成当前协作团队的概率增大,使得协作团队的整体性能有所改善.

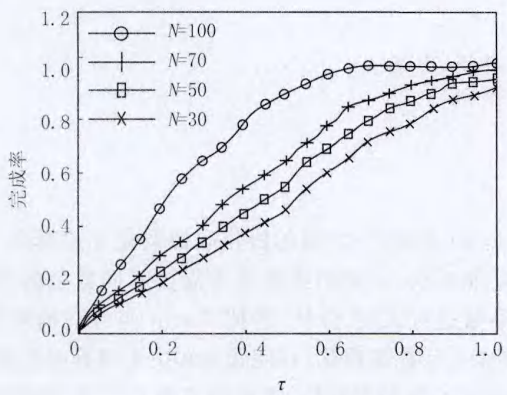

图5 信誉值对任务完成率的影响

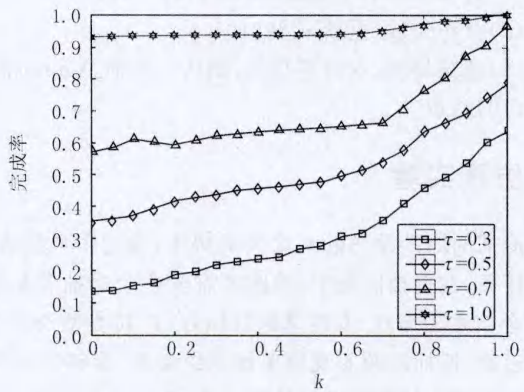

图6 不同惩罚系数下Agent的协作追捕完成率

图 6 为实验仿真 100 次不同惩罚系数 k 下的协作追捕完成率情况,从图 6 中可以看出,对于不同的惩罚系数,Agent 会选择不同的行为. 在惩罚系数 $k=0$ 时,由于参加协作的 Agent 都是理性 Agent,为了争取自身利益最大化自由选择退出当前协作;随着惩罚系数的不断增长,完成率有明显的提高,说明随着对协作 Agent 选择退出行为的严厉惩罚,个体 Agent 退出协作的代价会增大,选择新任务消耗的代价也会随之增大,即选择新的任务执行而获得的效用值 u 就会减少,当(u-完成当前任务获取的效用值)小于 Agent 期望的数值时,Agent 迫于惩罚的威慑会主动选择继续协作,提高了协作的稳定性.

对比图 5、图 6,图 6 表明,随着信誉值 τ 的增加,单纯依靠惩罚机制的协作促进效果将显著削弱. 对比图 5 不同信誉值 τ 下的完成情况,当 $\tau=1$ 时,不同惩罚值下的任务完成率趋近于 1,上述观察有力地说明:Agent 的信誉值 τ,对协作团队的整体性能的影响非常显著. 而当 Agent 的信誉值 τ 较低时,惩罚机制能有效的提高协作的整体性能.

为了验证本文提出的基于惩罚机制的协作算法在多智能体追捕实验中的有效性,分别用强化学习算法、改进的合同网协议算法与本文算法(分别取 k 为 0.75,τ 为 0.85)进行搬运完成率对比仿真实验,实验结果见表 1. 分别对 3 种算法进行 4 组实验,每组各进行 100 次实验,并对实验结果进行比较,经过 4 组实验结果的比较可知,强化学习算法的协作完成率为 51.75%,改进的合同网协议算法的协作完成率为 56.25%,而本文算法的协作完成率为 76.25%,较前两种算法有大幅提高,说明本文提出的引入信誉值与惩罚机制的协作算法明显提高了协作追捕的成功率,提高协作性能及协作效率.

表 1　3 种算法实验结果比较

算法	100 次实验成功次数				成功率/%
	第 1 组	第 2 组	第 3 组	第 4 组	
强化学习算法[10]	53	49	50	55	51.75
改进的合同网协议算法[11]	58	55	52	60	56.25
本文算法	73	79	76	77	76.25

4　结　语

动态环境中，任务会不断地加入到环境中，自利 Agent 为了追求更大的利益回报，可能选择放弃当前正在执行的任务，从而出现当前任务不能执行，或者执行效率降低的问题. 针对该问题，本文提出了一种基于博弈论及惩罚机制的多 Agent 协作控制算法. 与已有的工作相比，该方法充分考虑了协作 Agent 的理性，对长时间不能完成的协作任务允许动态的退出当前协作，以提高协作任务的效率，同时，为了保证协作的正常执行，对于随意退出当前协作的 Agent 给予惩罚的威慑来敦促其完成协作，并分析了信誉值、惩罚系数对协作的影响，仿真结果表明，通过合理选择惩罚系数及提高协作 Agent 的信誉值，可以有效提高协作团队的整体性能.

参　考　文　献

[1] Kim H E, Ahn H S. Convergence of multiagent Q-learning: Multi action replay process approach[C]. Proceedings of the IEEE International Symposium on Intelligent Control Part of Multi-Conference on Systems and Control, Yokohama, 2010.

[2] TAKAHASHI Y, TAMURA T, ASADA M. Cooperation via environmental dynamics caused by multi-robots in a hostile environment [C]. The Fourth IFAC Symposium on Intelligent Autonomous Vehicles, Sapporo, 2001.

[3] Kazunori I, Kazushi I, Hideake S. A Statistical property of multi-agent learning based on Markov decision process[J]. IEEE Transactions on Neural Networs, 2006, 17(4): 829-942.

[4] Greenwald A, Keith Hall. Correlated Q-Learning [C]. Proc of the 20th Int Conf on Machine Learning, Washington, 2003.

[5] TOLEDO C B, JENNINGS N R. Learning when and how to Coordinate [J]. International Journal of Web Intelligence and Agent Systems, 2003, 1(3/4): 203-218.

[6] 唐贤伦，李亚楠，樊　峥. 未知环境中多 Agent 自主协作规划策略[J]. 系统工程与电子技术，2013，35(2)：345-349.

[7] 肖喜丽. 基于协进化的多智能体协作研究[D]. 南京：南京邮电大学，2012.

[8] 任韶萱. 蚁群算法在多机器人协作中的应用[J]. 沈阳理工大学学报，2011，30(05)：49-53.

[9] 施锡全. 博弈论[M]. 上海：上海财经大学出版社，2000.

[10] 李　珺. 基于强化学习的多机器人追捕问题研究[D]. 哈尔滨：哈尔滨工业大学，2010.

[11] 林　琳，刘　峰. 基于改进合同网协议的多 Agent 协作模型[J]. 计算机研究与发展，2010，20(3)：70-75.

The Algorithm for Mulit-agents Cooperation Controling Based on Game Theory and Punishment Mechanism

ZHENG Yanbin[a,b], TAO Xueli[a,b], DUAN Lingyu[a], LI Bo[a]

(a. College of Computer and Information Technology; b. Engineering Laboratory of Intellectual Business and Internet of Things Technologies, Henan Normal University, Xinxiang 453007, China)

Abstract: For the problem of currents task cannot be normally performed when selfish agents would select higher utility tasks during collaborative process, a cooperative controlling method based on game theory and punishment mechanism was proposed. In phase of cooperative teamforming, selected agents which have higher credibility, during taskexecuting, used punishment mechanism to constrain the behaviors of agent for exiting. Simulation results showed that the algorithm can effectively avoid agent arbitrary exiting collaboration, ensured the smooth implementation of collaborative tasks and improve the success rate of collaboration.

Keywords: game theory; cooperation; punishment mechanism; credit value; multi-Agent