多智能体学习中的博弈和近似均衡

高 阳 教授，博导

南京大学计算机软件新技术国家重点实验室

合作者：胡裕靖(南京大学)，安波(南洋理工大学)

2015年8月11日

报告提纲

- 大数据背景下的多智能体决策
- 延迟反馈下的在线学习
- 非共享支付矩阵的多智能体博弈
- 近似博弈的均衡迁移
- 稀疏交互下的知识迁移和博弈约简

多智能体学习中的博弈和近似均衡

大数据背景下的多智能体决策

技术发展

数据规模

- ✓ 社交网络、传感器（RFID、GPS等）、科学研究
- ✓ 2013年底，全世界数据量已达到ZB级别

国家战略

- ✓ 美国OSTP：“大数据研究计划”（Big Data R&D Initiative）
- ✓ 中国科技部：“十二五科技计划信息技术领域”把大数据研究列在首位
- ✓ 中国自然科学基金委员会：2014年度设立“大数据”重点项目群

学术发展

- ✓ 2013年度，创办IEEE Big Data 国际会议
- ✓ 2014年度，创办IEEE Transactions on Big Data (IEEE TBD)

大数据特点和关键技术

大数据的4V特征

✓海量(Volume)、快速多变(Velocity)、多样性(Variety)、不精准性(Veracity)

适应大数据特征的数据分析技术

关键技术

表示技术　　学习技术　　推理技术

存在问题

面向大数据的高效知识表示

面向大数据的在线学习技术

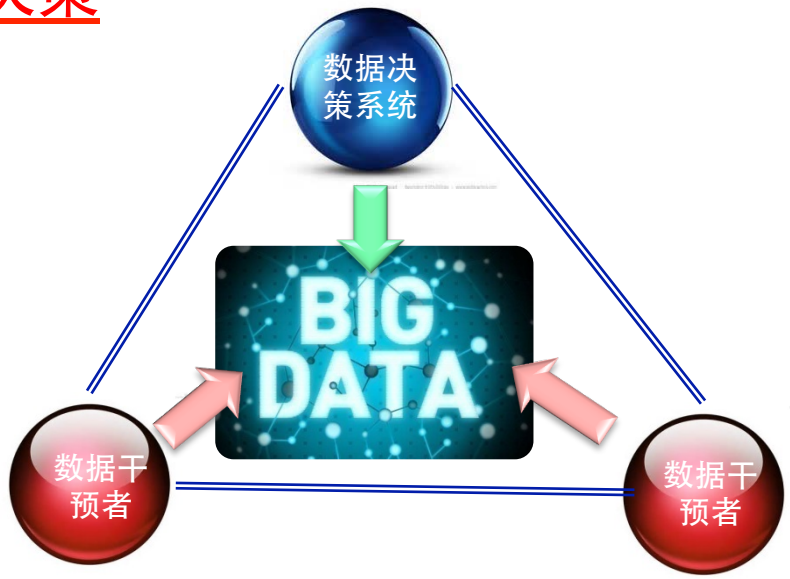
面向大数据的动态推理技术

博弈推理技术

动态、不确定导致的大数据复杂决策

多个运营商的多种推荐

用户复杂决策



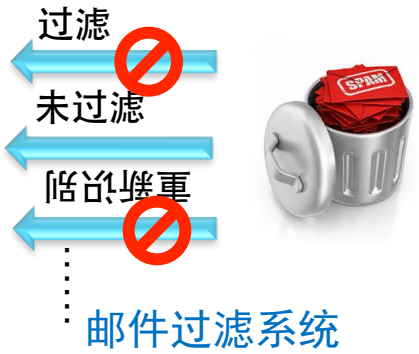
博弈推理复杂

大数据环境下面向多个行为实体复杂决策的博弈推理技术

博弈推理技术

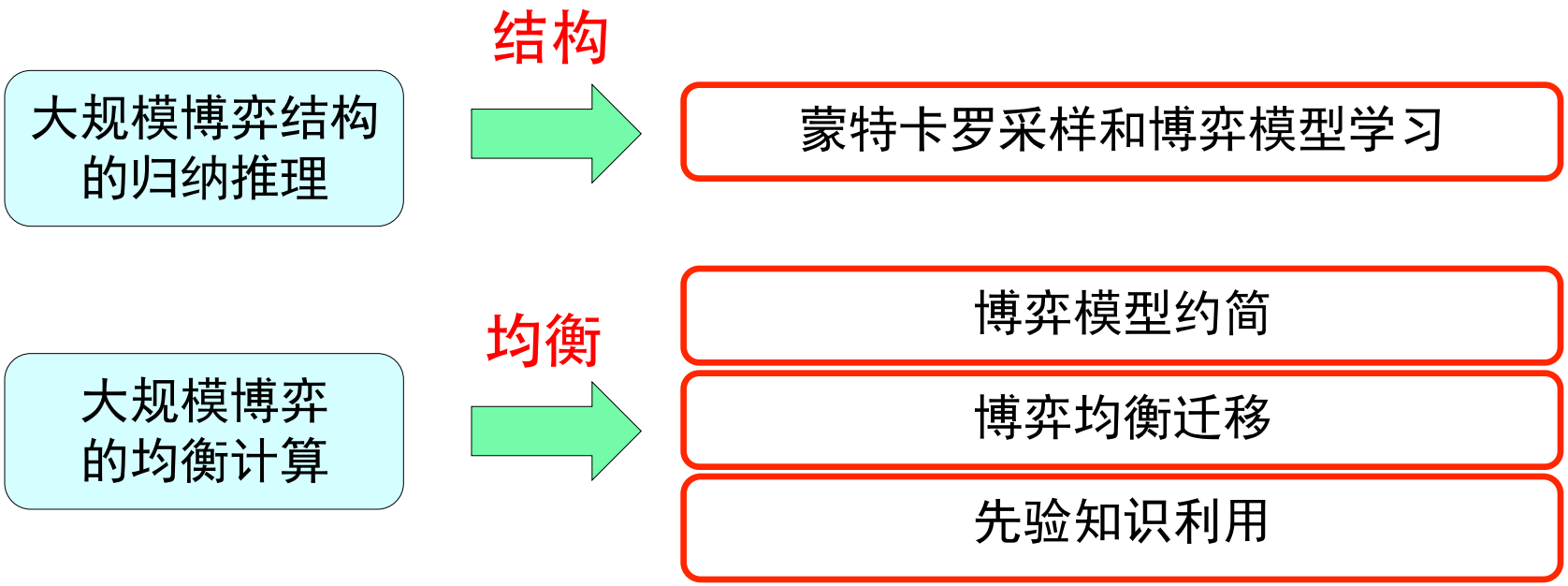
- 已有推理技术
  - 基于逻辑的演绎推理方法 ⇒ 适用于小规模问题
  - 结合统计的归纳推理方法 ⇒ 适用于大数据场景

- 面向大数据复杂决策的动态推理技术
  - 多行为智能体的交互［Battista, SMC011］
  - 贝叶斯博弈、Stackelberg博弈［Bruckner, KDD2011］



大规模博弈推理

- 存在问题和研究思路



多智能体学习中的博弈和近似均衡

2 延迟反馈下的在线学习

网络广告中的延迟反馈

排序＝点击率＊竞价

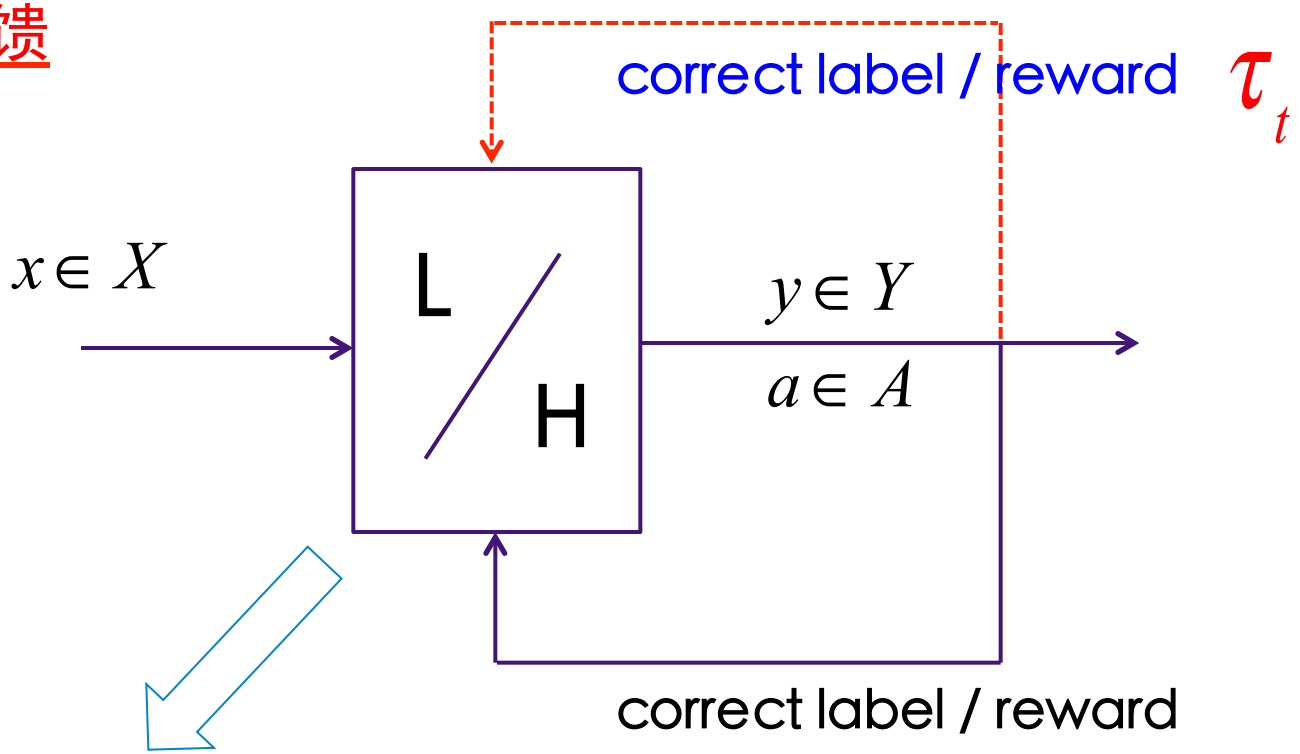
投放广告的商家

? ⇩ ⇧ 延时获得Reward（真实点击率）

sina 新浪网 sina.com.cn　腾讯网 QQ.com　網易 NETEASE www·163·com　搜狐 SOHU.com

在线学习范型

✧ 延迟反馈



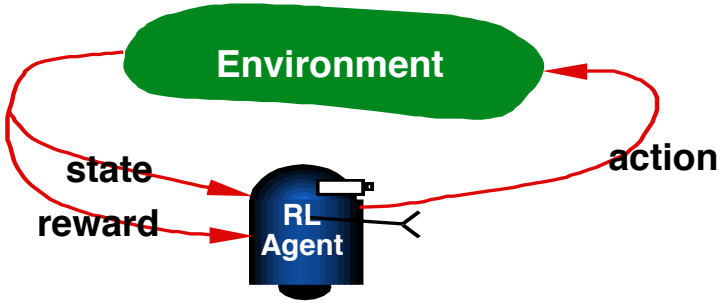
- ✓ 学习器从X集合中得到输入样本x；
- ✓ 学习器输出样本的预测标记；
- ✓ τ_t时刻后，获得样本的正确标记。

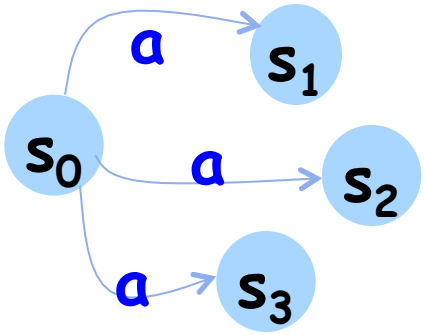
强化学习范型

✧ 延迟反馈 ✧ (环境)交互 ✧ 顺序决策

- 马尔科夫决策过程(Markov Decision Process)
  - 状态集合: S
  - 动作集合: A
  - 奖赏函数: $r: S \times A \to \Re$

    以及
  - 状态转移函数: $P: S \times A \to S$





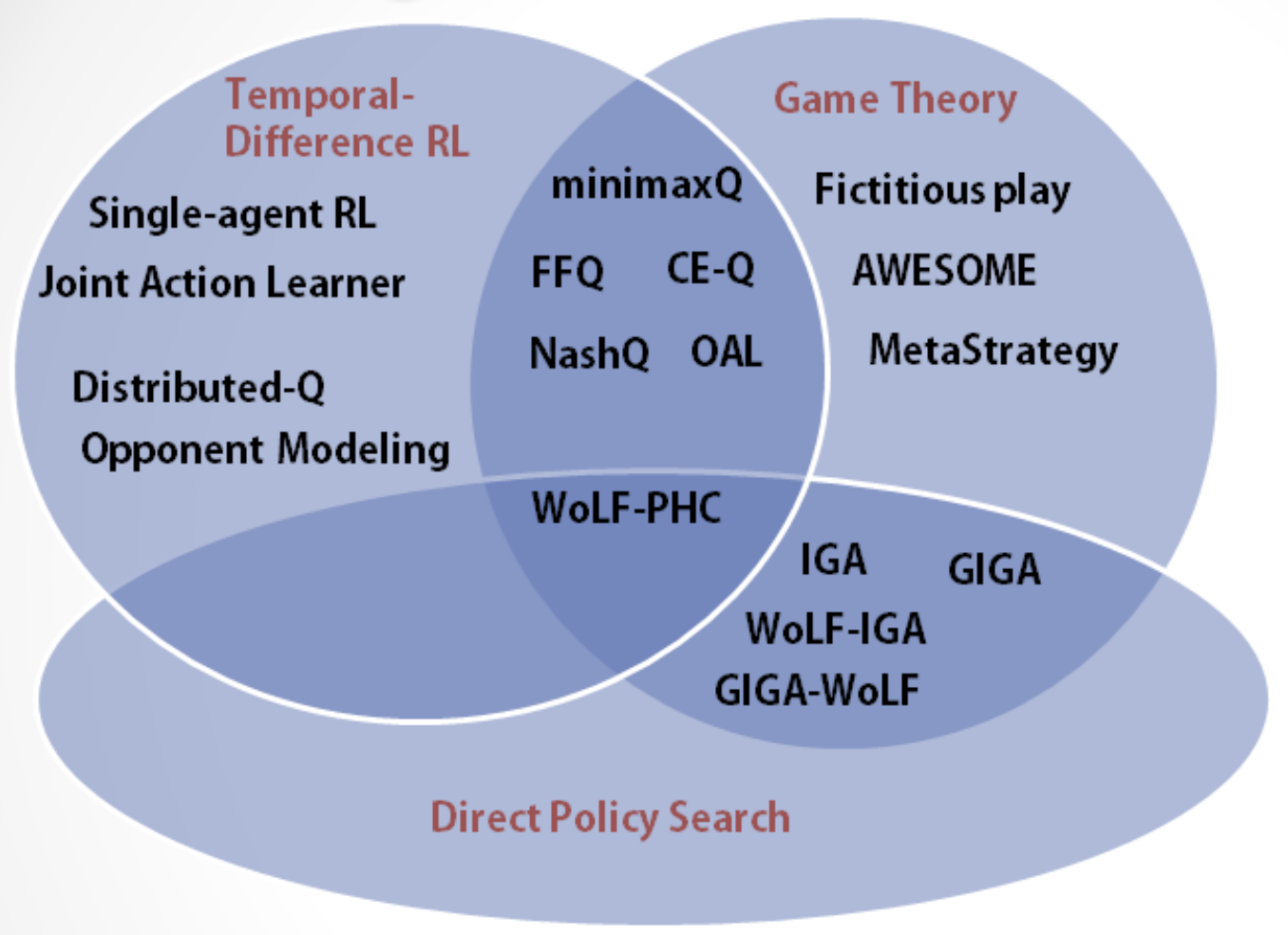
多智能体学习范型

✧ 延迟反馈　✧ 环境交互　✧ 顺序决策　✧ 多智能体博弈

马尔科夫博弈

- 马尔科夫博弈(Markov Game)
  - Agent集合：N
  - 状态集合：S
  - 联合动作集合：$A = A_1 \times A_2 \times \cdots \times A_n$
  - 奖赏函数：$r_i : S \times A \to \Re_i$
  - 状态转移函数：$P : S \times A \times S \to [0,1]$
- 学习方法和思路
  - 单Agent学习，或者多Agent联合学习
  - 对抗学习
  - 基于博弈均衡的学习

多智能体强化学习算法家族



[Lucian, TSMC-C 2008]Lucian Buşoniu, Robert Babuška, Bart De Schutter, A Comprehensive Survey of Multi-Agent Reinforcement Learning, IEEE Transactions on System, Man, and Cybernetics, Part C: Applications and Reviews, vol. 38, no. 2, pp. 156-172, 2008.

多智能体强化学习算法分类

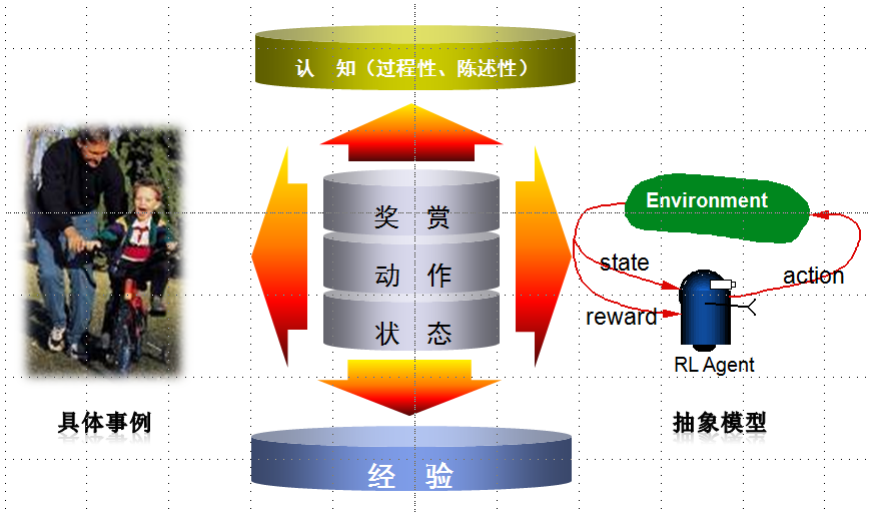

- ✓ 类型：合作型多智能体强化学习
- ✓ 问题空间：分布、同构、合作型多智能体系统
- ✓ 学习准则：提高学习速度

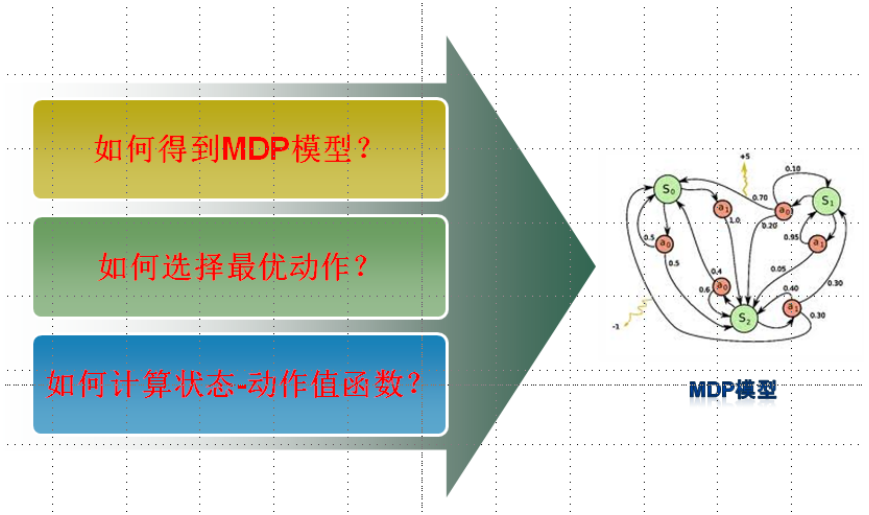

- ✓ 类型：竞争型多智能体强化学习
- ✓ 问题空间：异构多智能体系统
- ✓ 学习准则：不遗憾(No-regret)


- ✓ 类型：博弈型多智能体强化学习
- ✓ 问题空间：异构或同构多智能体系统
- ✓ 学习准则：收敛、均衡解

Yang Gao, Hao Wang, Ruili Wang, Three Perspectives on Multi-agent Reinforcement Learning. . In: Architectural Design of Multi-Agent Systems: Technologies and Techniques. IGI Global, pp:234-246, 2007.

(单智能体)强化学习算法设计





算法构造思路

- ✓ 根据先验得到初始认知(值函数)
- ✓ 根据认知选择动作(伴随一定的随机性)
- ✓ 获得经验
- ✓ 根据反馈，修改认知
- ✓ 根据延迟的反馈，回退修改历史认知

Initialize $Q(s,a)$ arbitrarily and $e(s,a)=0$, for all s,a
Repeat (for each episode):
  Initialize s, a
  Repeat (for each step of episode):
    Take action a, observe r, s'
    Choose a' from s' using policy derived from Q (e.g., ε-greedy)
    $a^* \leftarrow \arg\max_b Q(s',b)$ (if a' ties for the max, then $a^* \leftarrow a'$)
    $\delta \leftarrow r + \gamma Q(s',a^*) - Q(s,a)$
    $e(s,a) \leftarrow e(s,a) + 1$
    For all s,a:
      $Q(s,a) \leftarrow Q(s,a) + \alpha\delta e(s,a)$
      If $a' = a^*$, then $e(s,a) \leftarrow \gamma\lambda e(s,a)$
                    else $e(s,a) \leftarrow 0$
    $s \leftarrow s'$; $a \leftarrow a'$
  until s is terminal

基于博弈均衡的多智能体强化学习

学习最优联合动作并执行

S_1

$agent_1$ / $agent_2$	a_1	a_2
a_1	(1,2)	(3,7)
a_2	(7,2)	(4,8)

S_2

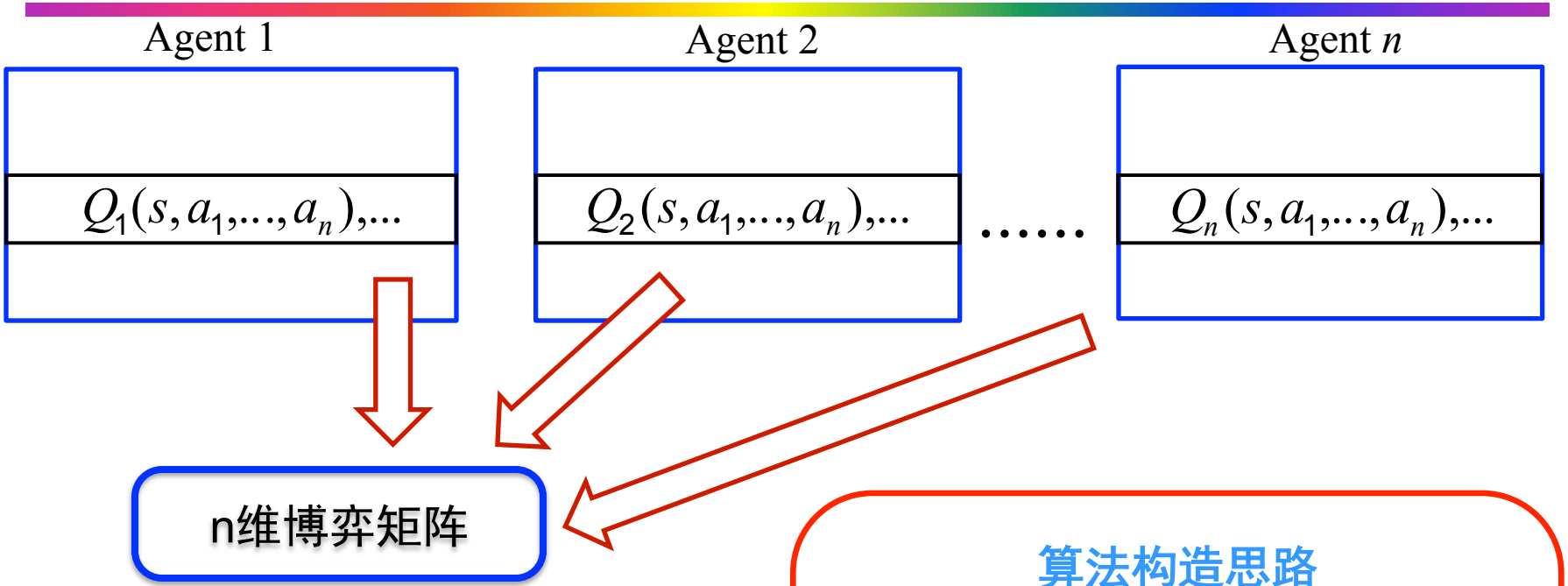
$agent_1$ / $agent_2$	a_1	a_2
a_1	(4,2)	(5,7)
a_2	(3,6)	(6,4)

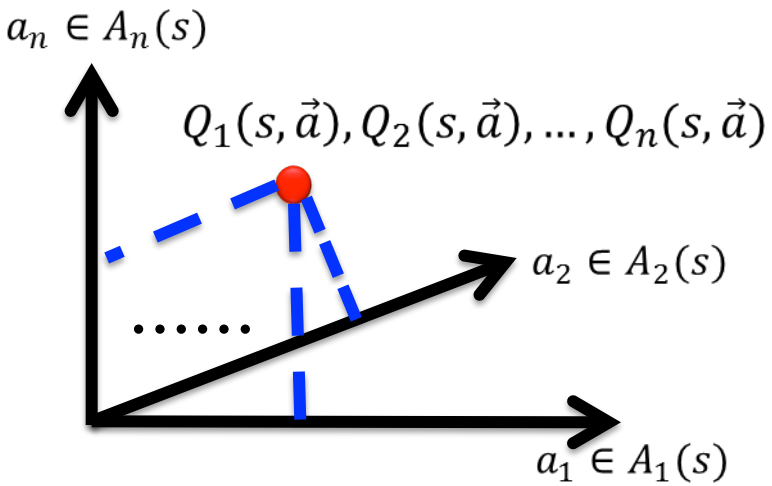
学习最优联合动作并执行

S_3

$agent_1$ / $agent_2$	b_1	b_2	b_3
b_1	(1,1)	(0,8)	(2,6)
b_2	(3,5)	(0,0)	(5,3)
b_3	(4,4)	(4,6)	(6,4)

基于博弈均衡的多智能体强化学习





算法构造思路

- ✓ 综合每个Agent的初始认知(值函数)
- ✓ 在每个状态构成n维的博弈矩阵
- ✓ 根据博弈选择均衡动作(伴随一定的随机性)
- ✓ 获得经验和反馈，修改认知
- ✓ 根据延迟的反馈，回退修改历史认知

基于博弈均衡的多智能体强化学习

Algorithm 1: The general framework of equilibrium-based MARL

Input: Learning rate α, discount rate γ, exploration factor ε
Initialization. $\forall s \in S, \forall i \in N, \forall \vec{a}, Q_i(s, \vec{a}) \leftarrow 0$;
foreach *episode* **do**
  Initialize state s;
  foreach *step* **do**
    $\vec{a} \leftarrow \Omega(Q_1(s), \ldots, Q_n(s))$ with ε-greedy policy;
    /* Ω is for computing an equilibrium */
    foreach *agent* $i \in N$ **do**
      Receive the experience $(s, \vec{a}, r_i, s')$;
      $Q_i(s, \vec{a}) \leftarrow (1-\alpha) Q_i(s, \vec{a}) + \alpha(r_i + \gamma \Phi_i(s'))$;
    /* Φ_i is the expected value of the equilibrium in the game occurring in state s' for agent i */
    $s \leftarrow s'$;

- ✓ Ω：根据当前认知，求解当前状态的博弈均衡
- ✓ Φ_i：计算智能体*i*的预测期望均衡值

基于博弈均衡的多智能体强化学习

算 法	值函数更新	动作选择
Q-learning	$Q(s,a) \leftarrow (1-\alpha)Q(s,a)+\alpha[r+\gamma \max_{a'} Q(s',a')]$	ε-贪心策略，或基于布尔兹曼分布的动作选择
MinimaxQ	$Q(s,a,o) \leftarrow (1-\alpha)Q(s,a,o)+\alpha[r+\gamma V(s')]$ $V(s') = \max_{\pi} \min_{o} \sum_{a} Q(s',a,o)\pi(s',a)$	基于当前学习到的策略π选择动作
NashQ	$Q_i(s,\vec{a}) \leftarrow (1-\alpha)Q_i(s,\vec{a})+\alpha[r_i+\gamma NashQ_i(s')]$ $NashQ_i(s') = \pi_1(s')\cdots\pi_n(s')Q_i(s')$	基于当前博弈的纳什均衡选择动作
FFQ	$Q_i(s,\vec{a}) \leftarrow (1-\alpha)Q_i(s,\vec{a})+\alpha[r_i+\gamma NashQ_i(s')]$	基于当前博弈的纳什均衡选择动作
CE-Q	$Q_i(s,\vec{a}) \leftarrow (1-\alpha)Q_i(s,\vec{a})+\alpha[r_i+\gamma CEQ_i(s')]$	基于当前博弈的相关均衡选择动作

多智能体学习中的博弈和近似均衡

3 非共享支付矩阵的多智能体博弈

基于博弈均衡的多智能体强化学习算法存在问题

(A,B)	b_1	b_2
a_1	(0, 0)	(1, 1)
a_2	(2, 2)	(3, 3)

Original Game Matrix

值表在多智能体间共享，为公共知识

✓透露太多信息，不安全

✓适用范围有限，分布决策环境中难适用

✓空间复杂度高

信息分布环境下的博弈形式

Distributed Game Matrix

(A,B)	b_1	b_2
a_1	(0, ?)	(1, ?)
a_2	(2, ?)	(3, ?)

(A,B)	b_1	b_2
a_1	(?, 0)	(?, 1)
a_2	(?, 2)	(?, 3)

基于博弈均衡的多智能体强化学习算法存在问题

经典的博弈均衡解概念：纳什均衡

$$U_1(\pi_1^*, \pi_2^*) \geq U_1(\pi_1', \pi_2^*)$$

$$U_2(\pi_1^*, \pi_2^*) \geq U_2(\pi_1^*, \pi_2')$$

囚徒困境博弈

(A,B)	Confess	Deny
Confess	(-9,-9)	(0,-10)
Deny	(-10,0)	(-1,-1)

分布式决策下的博弈，绝对理性的纳什均衡未必最适用

智能体

A	AConfess	Deny
Confess	-9	0
Deny	-10	-1

智能体

B	BConfess	Deny
Confess	-9	-10
Deny	0	-1

分布式决策下的新均衡解概念

(D,D)比(C,C)更有利

智能体A

A	Confess	Deny
Confess	-9	0
Deny	-10	-1

(D,D)比(C,C)更有利

智能体B

B	Confess	Deny
Confess	-9	-10
Deny	0	-1

策略组(*D*,*D*) 帕里托优超(Pareto dominates)于纳什均衡策略(*C*,*C*)。

分布式决策下的新均衡解概念

一个有趣的博弈

(A,B)	b_1	b_2	b_3
a_1	**(20,40)**	(4,22)	**(29,30)**
a_2	(18,9)	**(36,19)**	(7,4)
a_3	(17,26)	(15,38)	**(27,38)**

(a_1, b_3): (29, 30)　　(a_3, b_3): (27, 38)

另一种形式的优超

✓纳什均衡策略(a_1,b_1)和(a_2,b_2)。

✓(a_1,b_3)和(a_3,b_3)分别局部优超于纳什均衡策略(a_1,b_1)和(a_2,b_2)。

分布式决策下的新均衡解概念

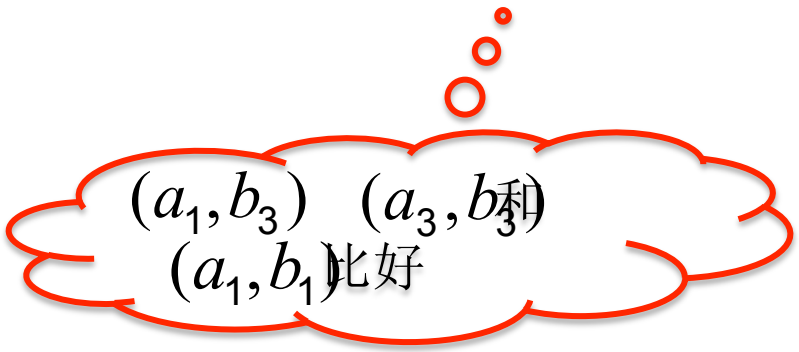
分布式决策下的博弈矩阵

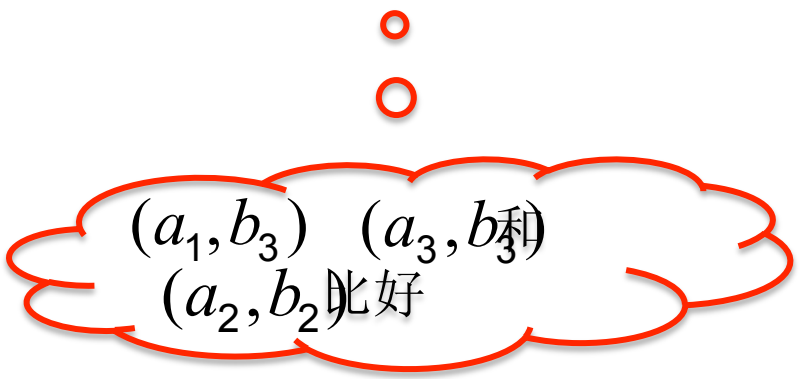
A	b_1	b_2	b_3
a_1	20	4	29
a_2	18	36	7
a_3	17	15	27

B	b_1	b_2	b_3
a_1	40	22	30
a_2	9	19	4
a_3	26	38	38

智能体A

智能体B





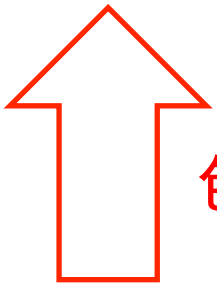
非严格均衡优超策略

帕里托优超于纳什均衡

定义1 均衡优超策略［Equilibrium-Dominating Strategy Profile (EDSP)］：在一般式博弈中，策略组$\vec{a}$被称为均衡优超策略当且仅当存在纯策略纳什均衡$\vec{e}$，使得策略组$\vec{a}$帕里托优超于$\vec{e}$，即对于主体$i = 1,2,\ldots,n$来讲，均有

$$U_i(\vec{a}) \geq U_i(\vec{e}).$$

部分优超于（多个）纳什均衡



包含

定义2 非严格均衡优超策略［Non-Strict Equilibrium-Dominating Strategy Profile］：在一般式博弈中，策略组$\vec{a}$被称为非严格均衡优超策略当且仅当对于主体$i = 1,2,\ldots,n$来讲，均存在纯策略纳什均衡$\vec{e}_i$，均满足

$$U_i(\vec{a}) \geq U_i(\vec{e}_i).$$

分布决策下的均衡解概念及计算

多智能体学习均衡解

- ✓ 纯策略纳什均衡
- ✓ 均衡优超策略
- ✓ 非严格均衡优超策略

非共享值函数下的均衡计算

- ✓ 协商
- ✓ 智能体间的信息交互
- ✓ 智能体的策略偏好

A	Confess	Deny
Confess	-9	0
Deny	-10	-1

信息交换

B	Confess	Deny
Confess	-9	-10
Deny	0	-1

策略偏好矩阵

(A,B)	Confess	Deny
Confess	(Y,Y)	(Y,N)
Deny	(N,Y)	(Y,Y)

只有(Y,Y)的策略组才能被双方接受

求解均衡的分布式协商机制

分布式协商机制的总体框架

1. 协商求解纯策略纳什均衡的集合：J_{NE}
2. 协商求解非严格均衡优超策略的集合：J_{NS}
3. 从集合中 $J = J_{NE} \bigcup J_{NS}$ 选择一个联合动作（帕里托优或者社会福利最大化）

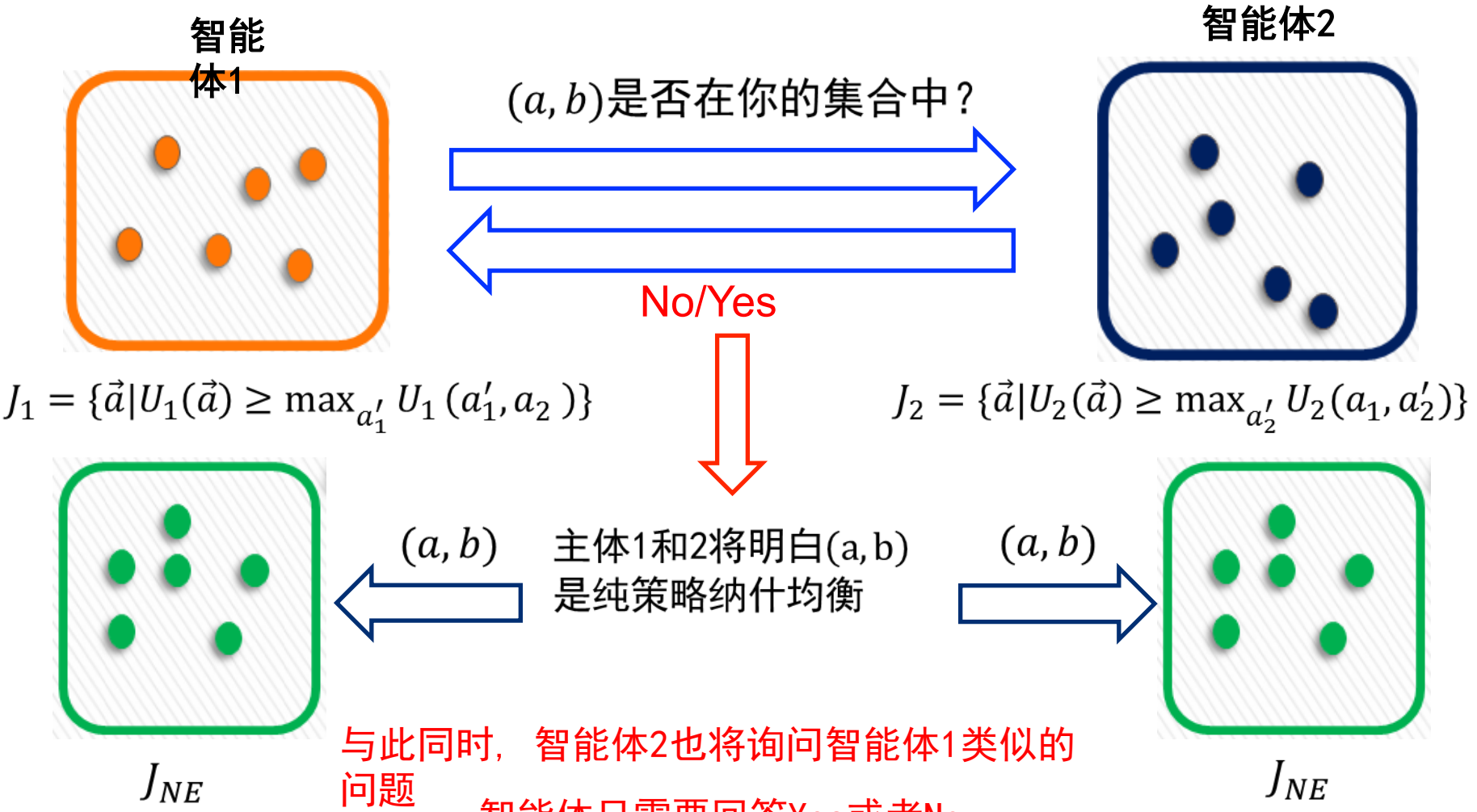
求解均衡的分布式协商机制

步骤一：求纯策略纳什均衡集合 J_{NE}

$$J_1 = \{\vec{a} | U_1(\vec{a}) \geq \max_{a_1'} U_1(a_1', a_2)\} \qquad J_2 = \{\vec{a} | U_2(\vec{a}) \geq \max_{a_2'} U_2(a_1, a_2')\}$$

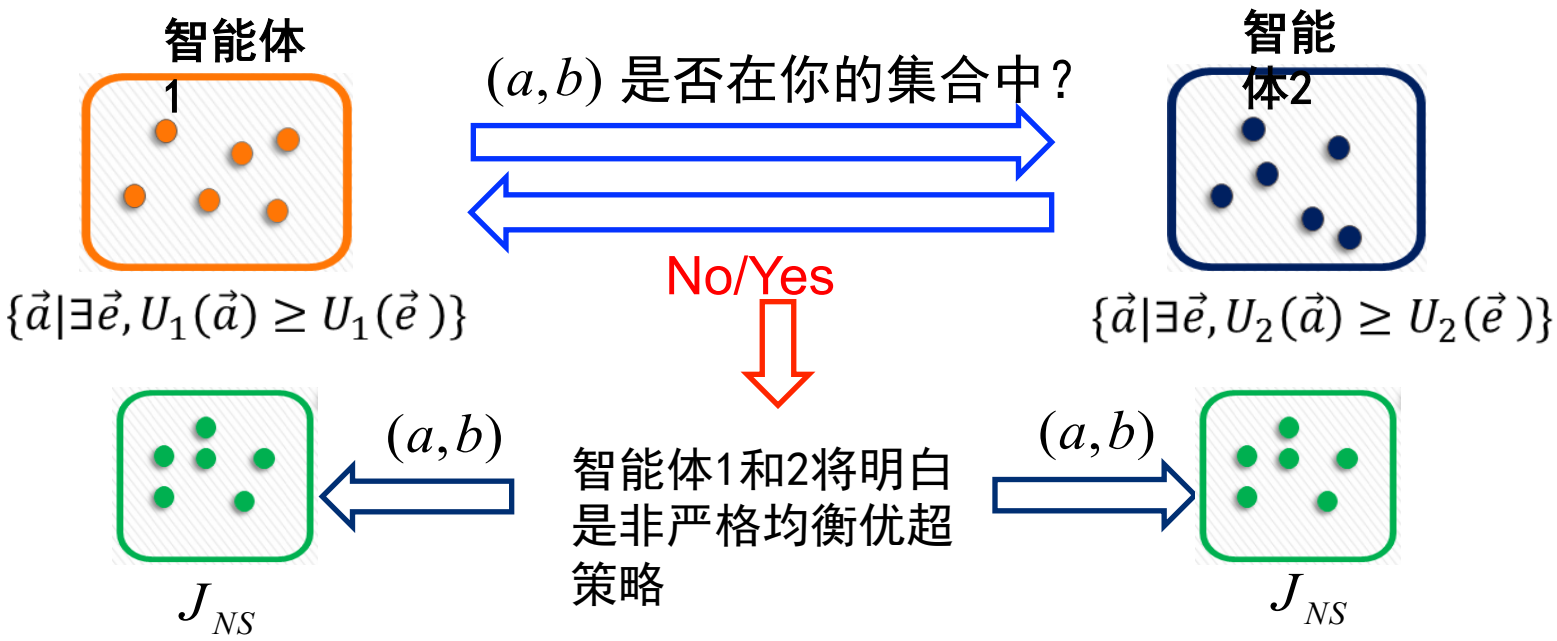
$$J_{NE} = J_1 \cap J_2$$

求解均衡的分布式协商机制



求解均衡的分布式协商机制

步骤二：类似地，非严格均衡优超策略集J_{NS}也可以通过这种方法得到



多智能体学习中的博弈和近似均衡

近似博弈的均衡迁移

基于博弈均衡的多智能体强化学习

Algorithm 1: The general framework of equilibrium-based MARL

Input: Learning rate α, discount rate γ, exploration factor ε
Initialization. $\forall s \in S, \forall i \in N, \forall \vec{a}, Q_i(s, \vec{a}) \leftarrow 0$;
foreach *episode* **do**
  Initialize state s;
  foreach *step* **do**
    $\vec{a} \leftarrow \Omega(Q_1(s), \ldots, Q_n(s))$ with ε-greedy policy;
    /* Ω is for computing an equilibrium */
    foreach *agent* $i \in N$ **do**
      Receive the experience $(s, \vec{a}, r_i, s')$;
      $Q_i(s, \vec{a}) \leftarrow (1-\alpha) Q_i(s, \vec{a}) + \alpha(r_i + \gamma \Phi_i(s'))$;
    /* Φ_i is the expected value of the equilibrium in the game occurring in state s' for agent i */
    $s \leftarrow s'$;

- ✓ Ω：根据当前认知，求解当前状态的博弈均衡
- ✓ Φ_i：计算智能体*i*的预测期望均衡值

值函数更新存在的问题

每一次经验获得后，仅涉及一个状态动作对（state-action pair）值函数的更新，并且每次的改变都很微小。

Algorithm 1: The general framework of equilibrium-based MARL

Input: Learning rate α, discount rate γ, exploration factor ε

Initialization. $\forall s \in S, \forall i \in N, \forall \vec{a}, Q_i(s, \vec{a}) \leftarrow 0$;

foreach *episode* **do**

  Initialize state s;

  foreach *step* **do**

    $\vec{a} \leftarrow \Omega(Q_1(s), \ldots, Q_n(s))$ with ε-greedy policy;

    /* Ω is for computing an equilibrium */

    foreach *agent* $i \in N$ **do**

      Receive the experience $(s, \vec{a}, r_i, s')$;

      $Q_i(s, \vec{a}) \leftarrow (1-\alpha) Q_i(s, \vec{a}) + \alpha(r_i + \gamma \Phi_i(s'))$;

    /* Φ_i is the expected value of the equilibrium in the game occurring in state s' for agent i */

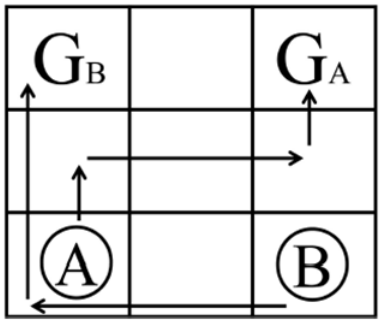
    $s \leftarrow s'$;

问题1：是否意味着值函数的改变很缓慢？

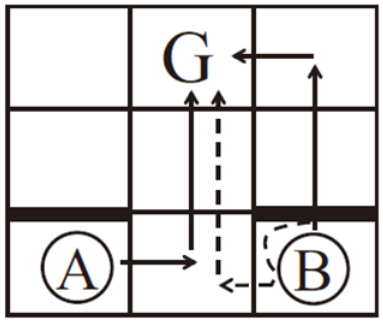
问题2：对每个状态的博弈均衡的变化产生什么影响？

问题3：是否意味着学习过程中存在很多相似博弈？

实验观察



GW1



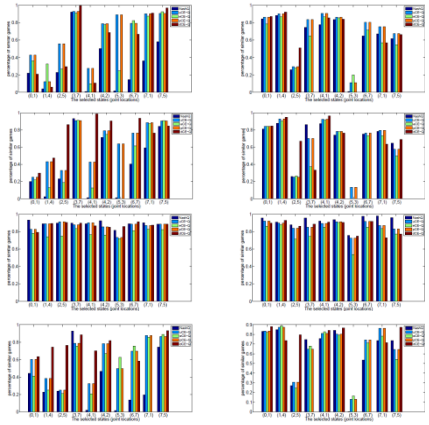
GW2

网格世界实验

- ✓ 5种学习算法各运行500次
- ✓ 记录相似博弈的比例
- ✓ 相似博弈：均衡解欧式距离小于0. 01

PS	Learning Rate	Discount Rate	Exploration Factor	Reward Function
PS1	0.2	0.9	0.02	100 for reaching a goal, −10 for collision
PS2	0.9	0.5	0.02	100 for reaching a goal, −10 for collision
PS3	0.9	0.9	1.0	100 for reaching a goal, −10 for collision
PS4	0.9	0.9	0.02	500 for reaching a goal, −30 for collision

4组不同的参数

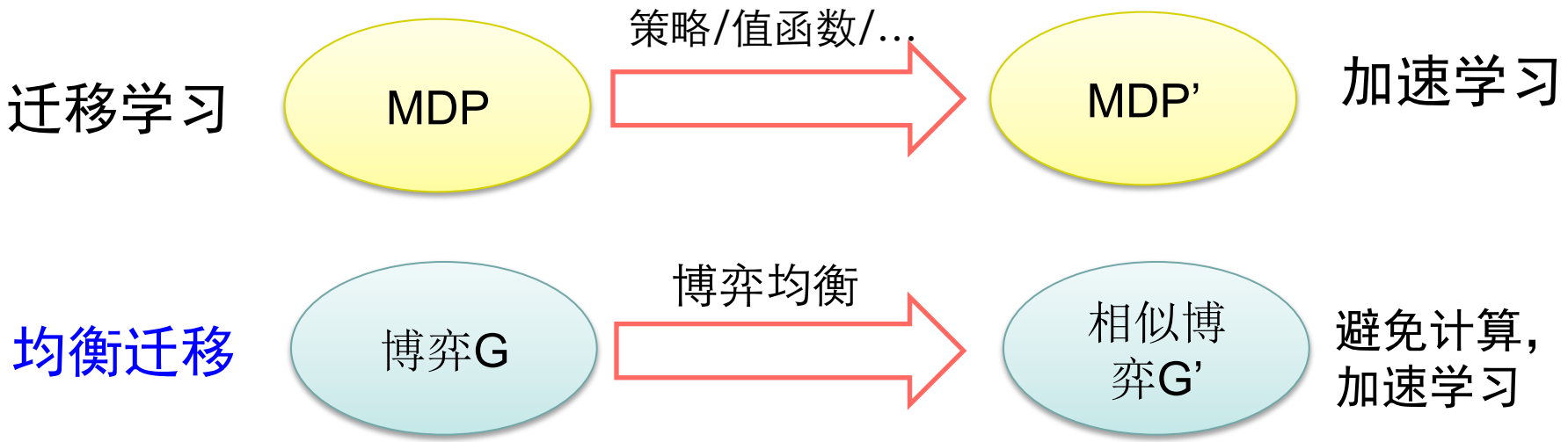


百分比最高达90%，平均为50%

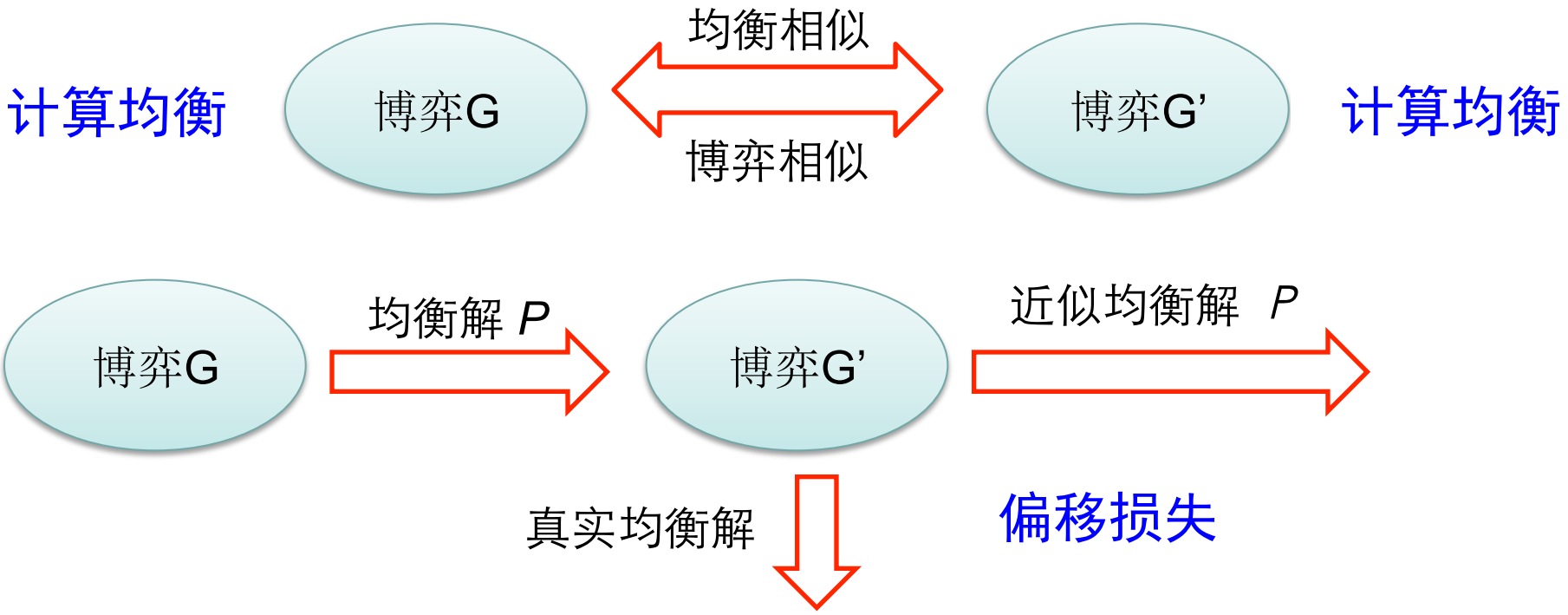
知识复用/均衡迁移

经验观察结论

- ✓ 基于均衡的多智能体学习存在大量的相似博弈
- ✓ 其原因在于值函数更新缓慢
- ✓ 通过复用相似博弈的均衡解，可以简化计算



博弈相似度度量



近似纳什均衡：

博弈的策略组*P*被称作 ε −近似纳什均衡（$\varepsilon>0$）当且仅当对任意智能体*i*，均有

$$U_i(p)+\varepsilon \geq \max_{a_i \in A_i} U_i(a_i, p_{-i})$$

博弈相似度度量 - 举例

(1,2)	R	S	T
O	(0,0)	(9,9)	(0,0)
P	(1,3)	(8,8)	(0,0)
Q	(2,1)	(0,0)	(0,0)

博弈G，纳什均衡p=(O,S)

效用矩阵改变

(1,2)	R	S	T
O	(0,0)	(9,9)	(0,0)
P	(1,3)	(10,10)	(0,0)
Q	(2,1)	(0,0)	(0,0)

博弈G’

p的偏移损失

智能体1：$\varepsilon_1 = \max_{a_1 \in A_1} U_1(a_1, S) - U_1(p) = U_1(P, S) - U_1(O, S) = 1$

智能体2：$\varepsilon_2 = \max_{a_2 \in A_2} U_2(O, a_2) - U_2(p) = U_2(O, S) - U_2(O, S) = 0$

所以，博弈G的纳什均衡p在博弈G’的偏移损失为1

迁移损失

定义 迁移损失**[Transfer Loss]**:对于一般式博弈G和它的一个均衡策略p，将p迁移到博弈G'的迁移损失定义为在G'中从策略p的偏移到其他策略时的最大效用损失ε，也即是说，p是G'的 ε -近似均衡

纳什均衡下的迁移损失定义：
一般式博弈G和 G'， p^*为博弈G的纳什均衡策略，将p^*迁移到G'的迁移损失为：

$$\epsilon^{NE} = \max_{i \in N} \max_{a_i \in A_i} (U_i^{G'}(a_i, p_{-i}^*) - U_i^{G'}(p^*)).$$

相关均衡下的迁移损失定义：
一般式博弈G和 G'， q^*为博弈G的相关均衡策略，将q^*迁移到G'的迁移损失为：

$$\epsilon^{CE} = \max_{i \in N} \max_{a_i \in A_i} \max_{a_i' \in A_i} \sum_{\vec{a}_{-i}} q^*(a_i, \vec{a}_{-i})(U_i^{G'}(a_i, \vec{a}_{-i}) - U_i^{G'}(a_i', \vec{a}_{-i})) .$$

迁移条件

设定迁移损失不超过 τ

- ✓ $\varepsilon_\theta = 0$，p^*为精确的均衡解
- ✓ $\varepsilon_\theta \le \tau$，$p^*$为τ-近似均衡
- ✓ $\varepsilon_\theta > \tau$，$p^*$不被迁移，均衡将被重新计算

定义 迁移条件**[Transfer Condition]**:对于一般式博弈*G*、它的一个均衡策略*p*、以及给定正数τ，将*p*迁移到博弈*G'*的迁移条件为关于*p*在*G'*中迁移损失ε小于τ，即 $\varepsilon(G, p, G') \le \tau$ 。

如果条件成立，*p*至少为博弈*G'*的τ-近似均衡

Algorithm 2: Equilibrium transfer-based MARL

Input: Learning rate α, discount rate γ, exploration factor ε, and threshold of transfer loss τ

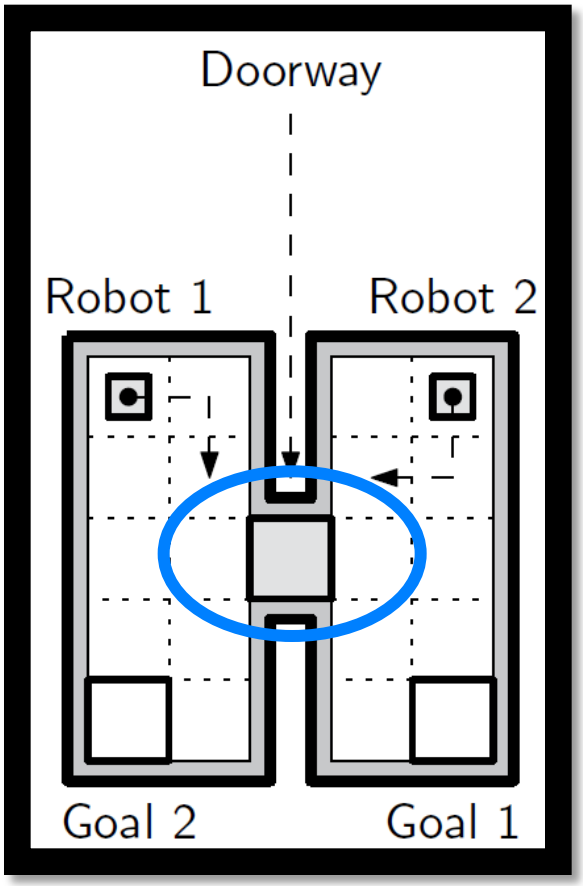
Initialization. $\forall s \in S, \forall i \in N, \forall \vec{a}, Q_i(s, \vec{a}) \leftarrow 0$;
foreach *episode* **do**
  Initialize state s;
  repeat
    $G_c \leftarrow$ the current one-shot game occurring in s;
    $p^* \leftarrow$ the equilibrium previously computed in s;
    if *s has been visited* **then**
      $\varepsilon^{\Theta} \leftarrow$ the agents' maximum utility loss for transferring p^* to G_c;
    else
      $\varepsilon^{\Theta} \leftarrow +\infty$;
    if $\varepsilon^{\Theta} > \tau$ **then**
      Compute the equilibrium p^* for G_c;
    else
      p^* is directly used in G_c;
    $\vec{a} \leftarrow$ the joint action sampled from p^* (using ε-greedy);
    Receive experience $(s, \vec{a}, r_i, s')$ for each $i \in N$;
    $p' \leftarrow$ the equilibrium stored for the next state s';
    foreach *agent* $i \in N$ **do**
      $V_i(s') \leftarrow$ the expected value of p' in s';
      $Q_i(s, \vec{a}) \leftarrow (1-\alpha) Q_i(s, \vec{a}) + \alpha(r + \gamma V_i(s'))$;
    $s \leftarrow s'$;
  until *s is a terminal state*;

基于均衡迁移的多智能体强化学习框架

多智能体学习中的博弈和近似均衡

稀疏交互下的知识迁移和博弈约简

稀疏交互的多智能体系统



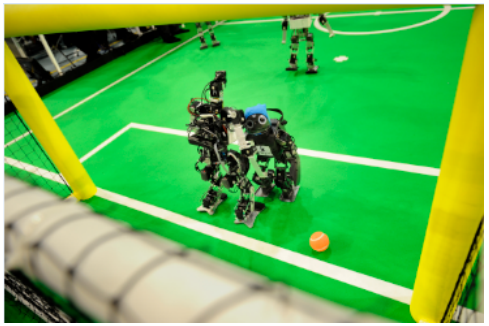
两个机器人可以在房间中自由行走

两个机器人不可以同时通过门

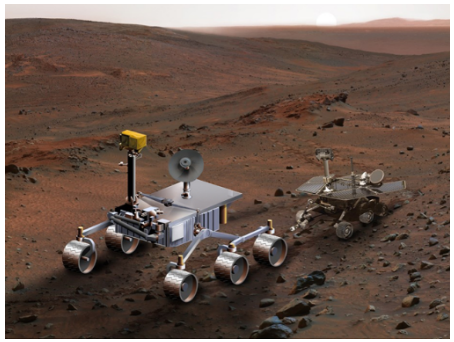
Sparse-interaction MAS

交互仅在局部出现的多智能体系统

实际应用例子

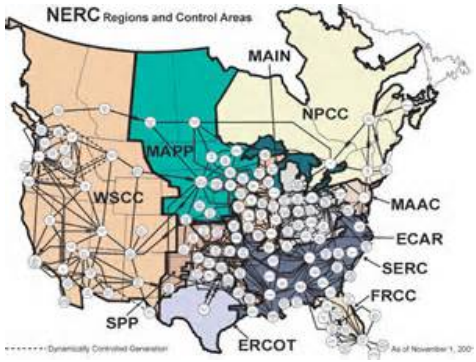


机器人足球

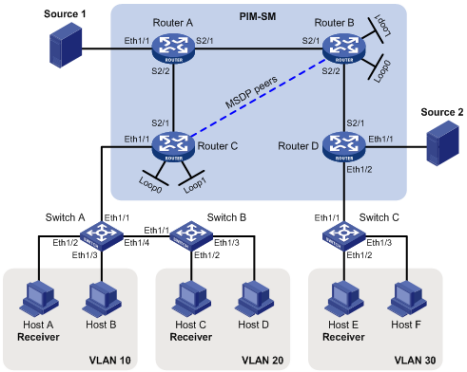


火星探测器

There is no need to interact all the time.



电力网格

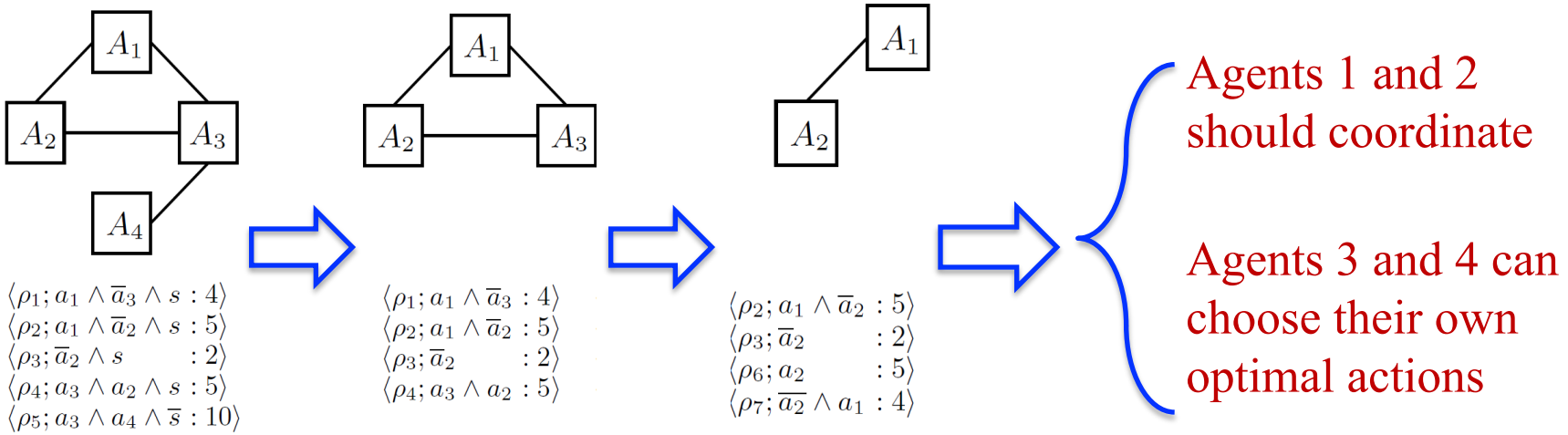


网络路由

There is no need to interact with all other agents.

值函数分解和协调图

- 局部交互
- 合作任务 (Multi-agent MDP)
- 全局值函数分解为局部值函数
- 构造协调图来选择动作



联合学习和独立学习交互

- 相对于Multi-agent MDP，模型更通用
- Markov Game, Decentralized MDP with Sparse Interaction (Dec-SIMDP)
- 没有交互时，就独立学习；反之，联合学习

典型算法－1

Algorithm 1 Learning algorithm for agent k

1: Initialize Q_k^* and Q_k^C;
2: Set $t = 0$;
3: **while** (FOREVER) **do**
4: Choose $A_k(t)$ using π_e;
5: **if** $A_k(t) =$ COORDINATE **then**
6: **if** $ActivePercept =$ TRUE **then**
7: $\hat{A}_k(t) = \pi_g(Q_k^C, X(t))$;
8: **else**
9: $\hat{A}_k(t) = \pi_g(Q_k^*, X_k(t))$;
10: **end if**
11: Sample $R_k(t)$ and $X_k(t+1)$;
12: **if** $ActivePercept =$ TRUE **then**
13: QLUpdate$\left(Q_k^C; X(t), \hat{A}_k(t), R_k(t), X_k(t+1), Q_k^*\right)$;
14: **end if**
15: **else**
16: Sample $R_k(t)$ and $X_k(t+1)$;
17: **end if**
18: QLUpdate$\left(Q_k^*; X_k(t), A_k(t), R_k(t), X_k(t+1), Q_k^*\right)$;
19: $t = t + 1$;
20: **end while**

Coordinate if the chosen local action suggests coordination

Act independently if it is better to do so

典型算法－2

Algorithm 1 CQ-Learning algorithm for agent k

1: Initialise Q_k and Q_k^j;
2: **while true do**
3: **if** $\forall$ Agents k, state s_k of Agent k is a safe state **then**
4: Select a_k for Agent k from Q_k
5: **else**
6: Select a_k for Agent k from Q_k^j
7: **end if**
8: $\forall$ Agents A_k, sample $\langle s_k, a_k, r_k \rangle$
9: **if** t-test detects difference in observed rewards vs expected rewards **then**
10: **for** $\forall$ seen states s_k of agent A_k **do**
11: **if** t-test detects difference between independent state s and joint state js **then**
12: add js to Q_k^j
13: mark js as dangerous
14: **else**
15: mark js as safe
16: **end if**
17: **end for**
18: **end if**
19: **if** s_k is safe for Agent k **then**
20: No need to update $Q_k(s)$.
21: **else**
22: Update $Q_k^j(js) \leftarrow (1 - \alpha_t)Q_k^j(js) + \alpha_t[r(js, a_k) + \gamma \, max_a Q(s'_k), a]$
23: **end if**
24: **end while**

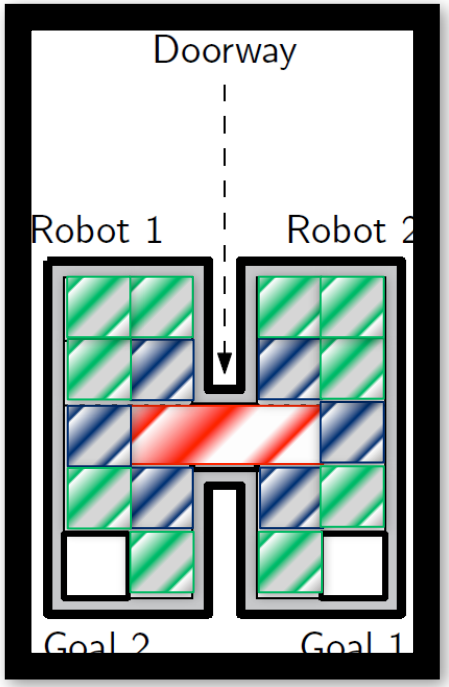
Mark the joint state when coordination is needed

Only update the joint-state Q-function

现有方法不足

The range of coordination: coordination should be executed at a more global level.

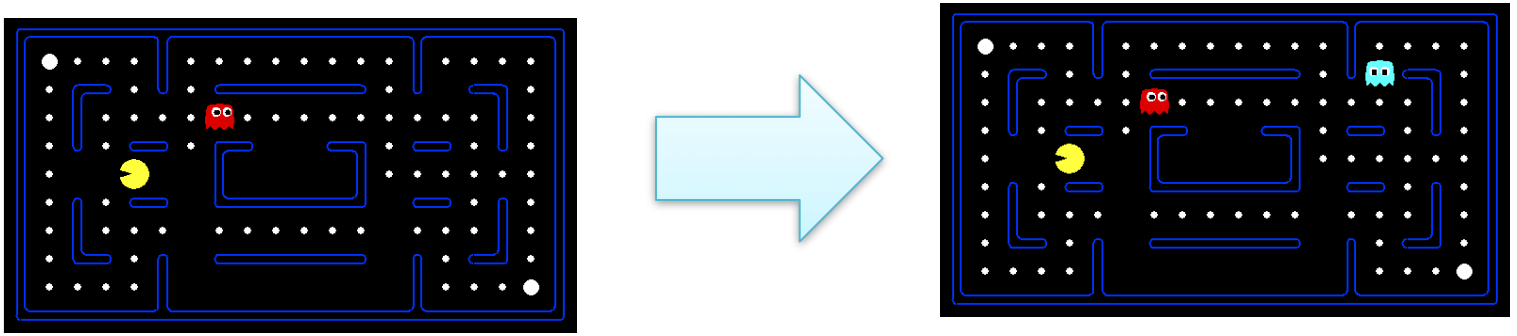
- The decision made in the current state determines which states can be reached next
- Agents may have to coordinate in states around the interaction areas
- Whether agents should coordinate may be a difficult problem



The magnitude of coordination: the way of coordination should be improved.

- It is not sufficient to just take joint actions or joint states into consideration
- We need more sophisticated coordination mechanism
- Equilibrium in each state?

动　机



Agent学习自身任务，然后学习协调

单Agent的知识或者经验，向其他Agent迁移

直接迁移局部值函数

Input: Learning rate α, discount rate γ, exploration factor ϵ, local value function table q_i for each agent i
Initialize the Q-table Q_i for each agent i:
For each agent $i \in N$
  For each state $s \in S$
    For each joint action $\vec{a} \in A$
      $Q_i(s, \vec{a}) \leftarrow q_i(s_i, a_i)$;
    End For
  End For
End For
The following is the same as the game theory-based MARL algorithm…

Algorithm 3. Value function transfer in game theory-based MARL with

直接迁移单Agent策略

Input: Learning rate α, discount rate γ, exploration factor ϵ, local policy π_i for each agent i, probability η for executing π_i, decreasing factor λ of η

Initialization. $\forall s \in S, \forall i \in N, \forall \vec{a}, Q_i(s, \vec{a}) \leftarrow 0$;

For each episode **do**

  Initialize state s;

  For each step **do**

    Sample a random probability p;

    If $p < \eta$

      $\vec{a} \leftarrow (\pi_1(s_1), \ldots, \pi_n(s_n))$;

      $\eta \leftarrow \eta\lambda$;

    Else

      $\vec{a} \leftarrow \Omega(\mathrm{Q}_1(\mathrm{s}), \ldots, \mathrm{Q}_\mathrm{n}(\mathrm{s}))$;

    Receive the experience $(s, \vec{a}, r_i, s')$ for each agent i;

    For each agent $i \in N$

      $Q_i(s, \vec{a}) \leftarrow (1 - \alpha)Q_i(s, \vec{a}) + \alpha(r_i + \gamma\Phi_i(s'))$;

    End For

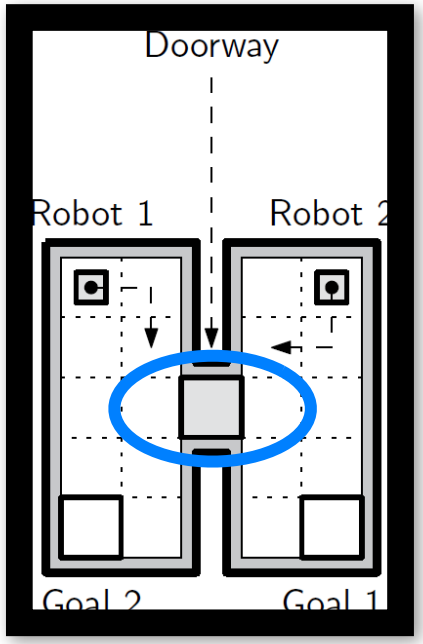
    $s \leftarrow s'$;

  End For

End For

Algorithm 4. Policy transfer in Game theory-based MARL

选择性迁移



The above two approaches transfer the single-agent knowledge to all states.

However, in some states (e.g., interaction area), agents should not act as they do in the single-agent tasks.

It will be better to identify to which states knowledge can be transferred and to which it cannot.

Markov博弈与多MDP

Assume that a Markov game $< N, S, \{A_i\}_{i=1\ldots n}, \{R_i\}_{i=1\ldots n}, T >$ can be equivalently represented by Multiple MDPs $\{\widehat{M}_i =< S_i, A_i, \hat{r}_i, \hat{T}_i >\}$ (the transformation is unknown).

We want to find for each agent i that in which state of the potential MDP $\widehat{M}_i =< S_i, A_i, \hat{r}_i, \hat{T}_i >$, the agent can act as it do in the MDP $M_i =< S_i, A_i, r_i, T_i >$.

$\widehat{M}_i =< S_i, A_i, \hat{r}_i, \hat{T}_i >$ can be constructed by sampling in the multi-agent task.
$M_i =< S_i, A_i, r_i, T_i >$ is stored in some model-based RL algorithms (such as Rmax).

Transferring both model and value function

状态和模型距离的度量

If $d_M(s_1, s_2) \approx d_{\widehat{M}}(s_1, s_2)$ and $d_M(s_1, s_3) \approx d_{\widehat{M}}(s_1, s_3)$, then we can determine that the s_1 in M is similar to the s_1 in $\widehat{M}$.

Definition 2 (State Similarity Between Two MDPs)
For the same state s_i in agent i's two MDPs $M_i =< S_i, A_i, r_i, T_i >$ and $\widehat{M}_i =< S_i, A_i, \hat{r}_i, \hat{T}_i >$, the similarity of state s_i between M_i and $\widehat{M}_i$ is defined as

$$D_{M_i,\widehat{M}_i}(s_i) = \sqrt{\sum_{s_i' \in S_i} \left(d_{M_i}(s_i, s_i') - d_{\widehat{M}_i}(s_i, s_i')\right)^2}.$$

A very small $D_{M_i,\widehat{M}_i}(s_i)$ indicates that the environment dynamics related to s_i in the Markov game is similar to that in M_i. Therefore, knowledge can be transferred in this state.

结合模型相似性的选择性迁移

Input: Learning rate α, discount rate γ, exploration factor ϵ, local value function q_i, local MDP model M_i (reward function and transition function) for each agent i, a threshold value τ for transferring knowledge

Experience several episodes with a random policy;

Construct an empirical MDP model $\widehat{M_i}$ for each agent i;

//Begin to transfer the value function

For each agent $i \in N$

  For each state $s \in S$

    For each joint action $\vec{a} \in A$

      $s_i \leftarrow$ the corresponding local state of agent i in s;

      $D_i \leftarrow$ state similarity of s_i between M_i and $\widehat{M}_i$;

      if $D_i \leq \tau$

        $Q_i(s, \vec{a}) \leftarrow q_i(s_i, a_i)$;

    End For

  End For

End For

The following is the same as the game theory-based MARL algorithm…

基于模型迁移的博弈约简

Definition 4 (Abstracted Game)
Let $G(s) =< N, \{A_i\}_{i=1,\dots,n}, \{Q_i(s)\}_{i=1,\dots,n} >$ be the one-shot game in state s of a Markov game. Let $K = \{i \in N | D_i(s_i) \leq \tau\}$, where s_i is the local state of agent i in state s, $D_i(s_i)$ is the state similarity defined in **Definition 2**, and τ is a threshold value. The abstracted game of $G(s)$ derived from K is defined as

$$G_K(s) =< N \setminus K, \{A_i\}_{i \notin K}, \{Q_i(s)\}_{i \notin K} >.$$

The value of τ can be decided by the algorithm designer. Smaller values means more coordination between agents.

基于模型迁移的博弈约简

Input: Learning rate α, discount rate γ, exploration factor ϵ, local MDP model M_i (reward function and transition function) for each agent i, a threshold value τ for game abstraction
Initialize the joint-action value function, $\forall s \in S, \forall i \in N, \forall \vec{a}$, the joint-action Q-value $Q_i(s, \vec{a}) \leftarrow 0$;
Initialize the local value function, $\forall i \in N, \forall s_i \in S_i, \forall a_i \in A_i$, the local Q-value $q_i(s_i, a_i) \leftarrow 0$;
Experience several episodes with a random policy construct an empirical MDP model $\widehat{M}_i$ for each agent i;
For each episode **do**
  Initialize state s;
  For each step **do**
    $G(s) \leftarrow < N, \{A_i\}_{i=1,\dots,n}, \{Q_i(s)\}_{i=1,\dots,n} >$;
    $K \leftarrow \{i \in N | D_i \leq \tau\}$ (as defined in **Definition 4**);
    $G_K(s) \leftarrow$ the abstracted game of $G(s)$ derived from K;
    For each agent $i \in K$
      $a_i \leftarrow \text{argmax}_{a_i'} q_i(s_i, a_i')$;
    End For
    For each agent $i \notin K$
      $a_i \leftarrow$ the action sampled by the equilibrium of $G_K(s)$ for agent i;
    End For
    $\vec{a} \leftarrow (a_1, \dots, a_n)$;
    Receive the experience $(s, \vec{a}, r_i, s')$ for each agent i;
    For each agent $i \in K$
      $q_i(s_i, a_i) \leftarrow (1-\alpha) q_i(s_i, a_i) + \alpha(r_i + \gamma \Phi_i(s'))$;
    End For
    For each agent $i \notin K$
      $Q_i(s, \vec{a}) \leftarrow (1-\alpha) Q_i(s, \vec{a}) + \alpha(r_i + \gamma \Phi_i(s'))$;
    End For
    $s \leftarrow s'$;
  End For
End For

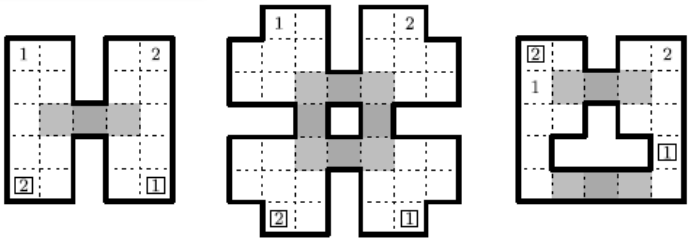
Game Abstraction

$\Phi_i(s')$ is $\max_{a_i} q_i(s_i', a_i)$ or the equilibrium value of agent i in the (maybe abstracted) game in s'

Algorithm 6. Game theory-based MARL with game abstraction

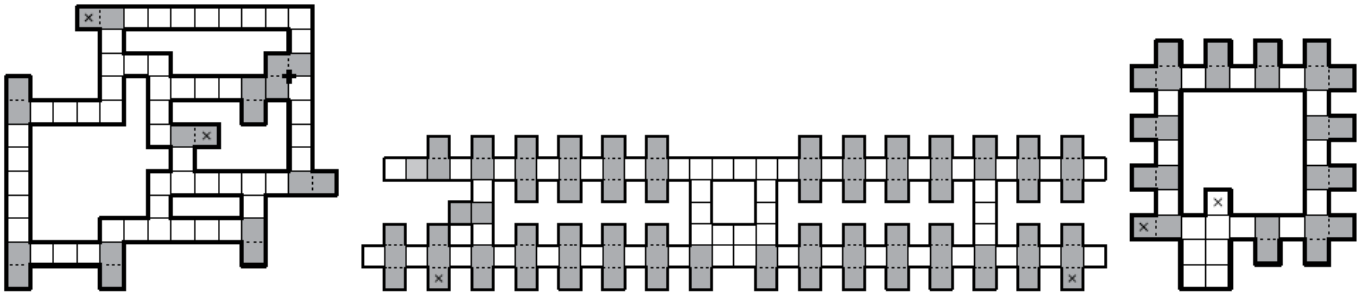
实　验

Benchmark

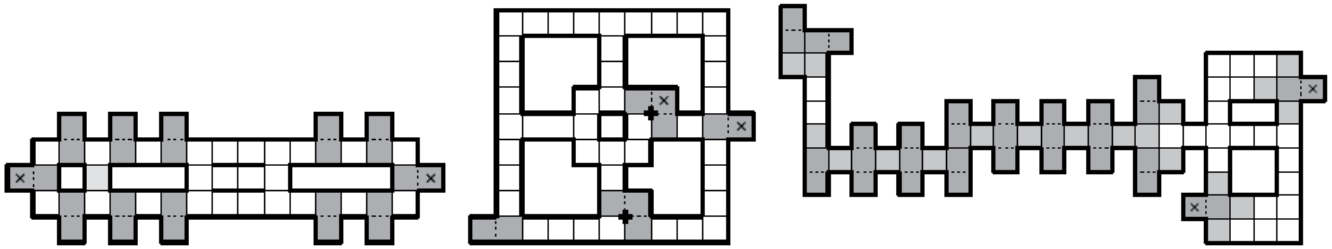


(a) Map 1.　(b) Map 2.　(c) Map 3.

The grids with shade are interaction areas, where agent will be punished heavily if collision happens



(a) CIT (2 robots)　(b) CMU (2 robots)　(c) ISR (2 robots)

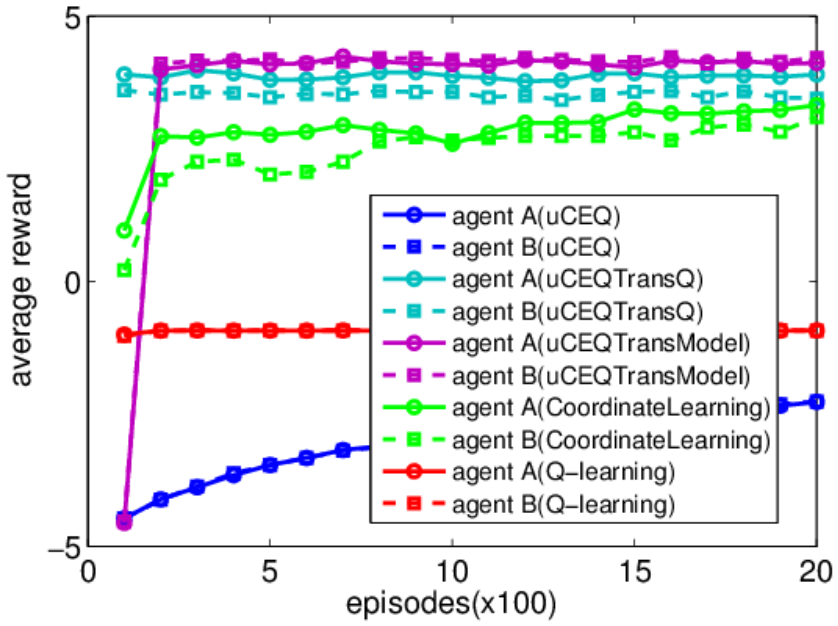


(d) MIT (2 robots)　(e) PENTAGON (2 robots)　(f) SUNY (2 robots)

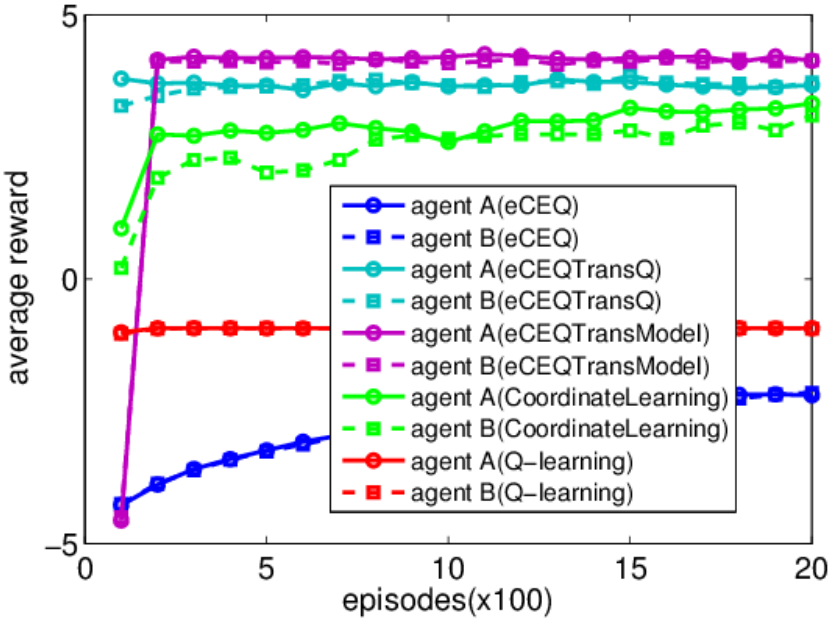
实验结果

The average reward in each learning episode

Reward for reaching the goal: 200
Reward for collision: -20
A negative reward -1 at every step
A probability 0.2 of action failure

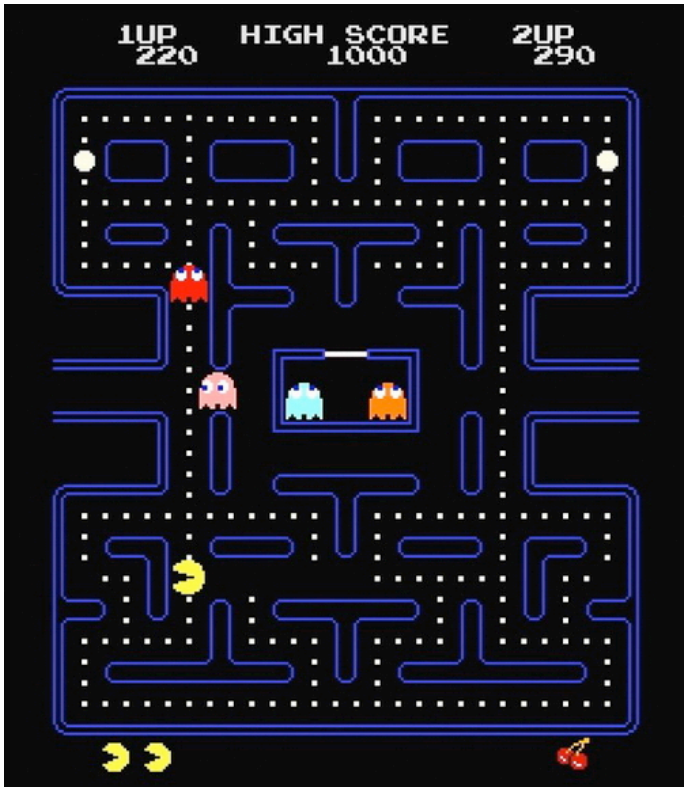


uCEQ



eCEQ

更多实验和分析



- ✓ Yujing Hu, Yang Gao, Bo An. Multi-agent Reinforcement Learning with Unshared Value, IEEE Transactions on Cybernetics, 45 (4):647-662, 2015.
- ✓ Yujing Hu, Yang Gao, Bo An. Accelerating Multi-agent Reinforcement Learning by Equilibrium Transfer, IEEE Transactions on Cybernetics, 45(7):1289-1302, 2015.
- ✓ Yujing Hu, Yang Gao, Bo An. Learning in Multi-agent Systems with Sparse Interactions by Knowledge Transfer and Game Abstraction. AAMAS 2015: 753-761.

总　结

报告要点

- ✓ 大数据决策导致大规模博弈推理
- ✓ 多智能体强化学习是一种复杂的计算范型
- ✓ 非共享支付矩阵情况下的博弈均衡协商机制
- ✓ 相似博弈的博弈均衡迁移
- ✓ 稀疏交互下的知识迁移和博弈约简

谢谢各位专家！

请提宝贵意见！

2015.8.21

参考文献

- Biggio Battista, Fumera Giorgio, Roli Fabio. Design of robust classifiers for adversarial environments. IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2011:977-982
- Michael Brückner, Tobias Scheffer: Stackelberg games for adversarial prediction problems. KDD 2011:547-555
- M. Littman, “Markov games as a framework for multi-agent reinforcement learning,” in Proceedings of the 11th International Conference on Machine Learning, 1994, pp. 157–163.
- M. Littman, “Friend-or-foe Q-learning in general-sum games,” in Proceedings of the 18th International Conference on Machine Learning, 2001, pp. 322–328.
- J. Hu and M. Wellman, “Nash Q-learning for general-sum stochastic games,” The Journal of Machine Learning Research, vol. 4, pp. 1039–1069, 2003.
- A. Greenwald, K. Hall, and R. Serrano, “Correlated Q-learning,” in Proceedings of the 20th International Conference on Machine Learning, 2003, pp. 242–249.
- L. Busoniu, R. Babuska, and B. De Schutter, “A comprehensive survey of multiagent reinforcement learning,” IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 38, no. 2, pp. 156–172, 2008.
- M. E. Taylor and P. Stone, “Transfer learning for reinforcement learning domains: A survey,” The Journal of Machine Learning Research, vol. 10, pp. 1633–1685, 2009.
- M. Osborne and A. Rubinstein, A course in game theory. MIT Press, 1994.

参考文献

- C. Szepesv´ari and M. L. Littman, “A unified analysis of value-functionbased reinforcement-learning algorithms,” Neural computation, vol. 11, no. 8, pp. 2017–2060, 1999.
- R. Sutton and A. Barto, Reinforcement Learning: An Introduction. MIT Press, 1998.
- X. Wang and T. Sandholm, “Reinforcement learning to play an optimal nash equilibrium in team markov games,” in Advances in Neural Information Processing Systems, vol. 15, Vancourver, Canada, December 2002, pp. 1571–1578.
- N. Howard, Paradoxes of Rationality: Games, Metagames, and Political Behavior. Cambridge Massachusetts: MIT Press, 1971.
- T. Sandholm, “Perspectives on multiagent learning,” Artificial Intelligence, vol. 171, no. 7, pp. 382–391, 2007.
- C. Watkins, “Learning from delayed rewards,” Ph.D. dissertation, Department of Computer Science, King’s College, Cambridge, 1989.
- C. Guestrin, D. Koller, and R. Parr. Multiagent planning with factored MDPs. In Proc. Adv. Neural Information Proc. Systems 14, pages 1523–1530, 2001.
- C. Guestrin, S. Venkataraman, and D. Koller. Context specific multiagent coordination and planning with factored MDPs. In Proc. National Conf. Artificial Intelligence, pages 253–259, 2002.
- C. Guestrin, M. Lagoudakis, R. Parr. Coordinated Reinforcement Learning. In Proc. of Int. Conf. Machine Learning (ICML), 2002.

参考文献

- J. Kok, P. Hoen, B. Bakker, and N. Vlassis. Utile coordination: Learning interdependencies among cooperative agents. In Proc. Symp. on Computational Intelligence and Games, pages 29–36, 2005.
- J. Kok and N. Vlassis. Sparse cooperative Q-learning. In Proc. Int. Conf. Machine Learning, pages 61–68, 2004.
- M. Spaan and F. Melo. Local interactions in decentralized multiagent planning under uncertainty. In Proc. Int. Joint Conf. Auton. Agents and Multiagent Systems (AAMAS), pages 525–532, 2008.
- M. Spaan and F. Melo. Interaction-driven Markov games for decentralized multiagent planning under uncertainty. In Proc. Int. Conf. Auton. Agents and Multiagent Systems (AAMAS), pp. 525–532, 2008.
- F. S. Melo, M. Veloso. Learning of Coordination: Exploiting Sparse Interactions in Multi-agent Systems, In Proc. Int. Conf. Auton. Agents and Multiagent Systems (AAMAS), pp. 773-780, 2009.
- Y. D. Hauwere, P. Vrancx, A. Nowe, Learning Multi-agent State Space Representations, In Proc. Int. Conf. Auto. Agents and Multiagent Systems (AAMAS), pp. 715-722, 2010.
- F. S. Melo, M. Veloso. Decentralized MDPs with sparse interaction, Artificial Intelligence, 2011.