天津大学

本科生毕业论文

学　　院　　软件学院

专　　业　　软件工程

年　　级　　2013 级

姓　　名　　郝晓田

指导教师　　郝建业

2017 年　6 月　20 日

天津大学

本科生毕业论文任务书

题目：有限通讯下基于强化学习技术的社会规范涌现方法研究

学生姓名 郝晓田

学院名称 软件学院

专　　业 软件工程

学　　号 **3013218138**

指导教师 郝建业

职　　称 副教授

社会规范是实现多智能体系统有机协作的重要技术手段之一，而如何自动生成高效的社会规范是多智能体系统领域重要的研究课题之一。本课题拟研究在通讯受限情况下，如何设计高效的基于强化学习技术的协作方法，实现多 agent 群体内社会规范的快速涌现。

指导教师（签字）

年　月　日

审题小组组长（签字）

年　月　日

天津大学本科生毕业设计开题报告

课题名称	有限通讯下基于强化学习技术的社会规范涌现方法研究		
学院名称	软件学院	专业名称	软件工程
学生姓名	郝晓田	指导教师	郝建业

一、课题的来源及意义

多智能体系统(multi-agent system(M.A.S.))是一个由在某种环境中交互的多个智能体组成的计算系统。其中合作式的多智能体系统（Cooperative multi-agent systems）在现实世界中应用广泛，比如机器人学，传感器网络，分布式的控制系统。

在以往的研究中，多智能体系统规模往往比较小，而且大多假设各个 agent 对外部环境可观察并且能够全局通信，即每个 agent 知道外部环境即其他 agent 的策略和行动，这样实际上组成了一个集中式的决策系统。

但是在现实中，往往由于系统规模比较大，实际可利用的计算资源十分有限，通信的距离及通信并发数量也是非常有限，部分系统对实时性要求很高。所以应对这种情况，如何使各 agent 基于局部信息独立决策，并通过有限的有效通信，实现高效协作，在实现分布式系统有机控制中具有重要意义。

社会规范是规范多 agent 之间行为并且促进多 agent 合作的重要机制，是实现多智能体系统有机协作的重要技术手段之一，而如何自动生成高效的社会规范是多智能体系统领域重要的研究课题之一。

本课题拟研究基于有限通讯条件下，如何设计高效的基于强化学习技术的协作方法，实现多 agent 群体内社会规范的快速涌现，以最大化整体的效益。

二、国内外发展状况

上世纪 90 年代以来，关于多 agent 的研究逐步引起重视并成为人工智能研究的热点。在多 Agent 系统中，各 Agent 必须对其目标和资源使用进行协作，协作是多 Agent 系统研究的核心问题。

强化学习一直被认为在现代及未来的人工智能领域存在无限可能，利用强化学习的方法，各 Agent 通过与环境的不断交互获得新知，并通过相互通信，改进行为策略。

Multi-agent reinforcement learning(MARL)，已经相当成熟。但是以往的研究，及算法大都针对较小的系统及较小的状态空间，而且大都需要很强的通信。但是

在 Agents 规模很大、通信受限的现实系统中，难以集中式的决策，各 Agent 如何协作，加快协作规范涌现成为难点。

针对这个问题，不少学者已经做了相关的研究。例如：Chongjie Zhang 和

Victor Lesser 提出了在有限通信带宽的情况下，运用 DCOP 算法和多智能体系统的强化学习该环境下的独立学习和协调通信问题【1】。Tianpei Yang 和 Jianye Hao 提出了在基于分层的启发式学习方法来加速群体内社会规范涌现的思路。【2】

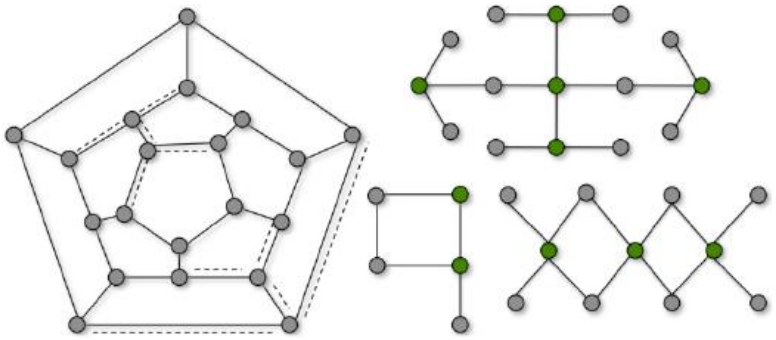


图 1 传感器网络

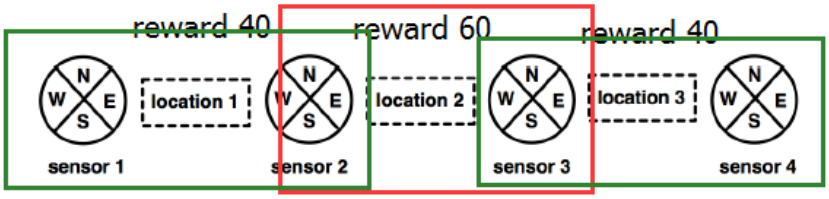


图 2 通信对象的选择

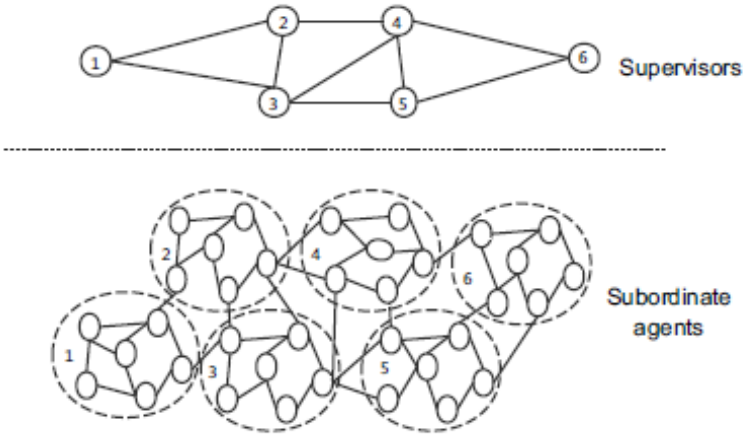


图 3 层次学习结构

三、课题的研究目标

社会规范是实现多智能体系统有机协作的重要技术手段之一，而如何自动生成高效的社会规范是多智能体系统领域重要的研究课题之一。本课题拟研究在通讯受限情况下，如何设计高效的基于强化学习技术的协作方法，实现多 agent 群体内社会规范的快速涌现。

四、研究内容

有限通信，分布式的条件下，基于强化学习的方式，在保证信息损失在可接受范围条件下，将集中式，大规模计算拆分为分布式，合作式计算。

在通讯受限情况下，如何设计高效的基于强化学习技术的协作方法，尽量减少各个 agent 之间的通信，或者如何设计通信规则让各个 agent 能更加有效利用有限的信息交换，通过有效的交流，让各个 agent 更快的趋向于同一个 action，实现多 agent 群体内社会规范的快速涌现。

Markov Game

$$Q(\vec{h}^t, a^t) = (1-\alpha)Q(\vec{h}^t, a^t) + \alpha[r^t + \gamma \max_a Q(\vec{h}^{t+1}, a)]$$

$$\hat{Q}(\vec{h}, a) = \sum_{i \in I} \sum_{g_i \in G_i} Q_{g_i}(\vec{h_i}, a_i, a_{g_i}).$$

$$Q_{g_i}(\vec{h_i}^t, a_i^t, a_{g_i}^t) = (1-\alpha)Q_{g_i}(\vec{h_i}^t, a_i^t, a_{g_i}^t) + \alpha[r_{g_i}^t + \gamma Q_{g_i}(\vec{h_i}^{t+1}, a_i^*, a_{g_i}^*)]$$

五、课题的研究方法和研究手段

针对问题，提出设想，建立模型，并通过模拟实验验证模型的正确性和准确性，其次假设与理论结合，设计适用于模型的合理的算法和解题思路，然后结合实验观察结果，总结归纳，总结发现的问题和解决问题的。

六、进度安排

在老师的指导下，尽快确立解决方案及思路，尽快将目前思路转化为代码实现，在此基础上，不断调整，尽快达到一个好的结果。

1. 时间：2016.12 月初 – 2016.12 月中旬
   确立题目，及研究方向。
2. 时间：2016.12 月中旬 – 2016.12 月末
   明确课题方向、浓缩课题范围、提示焦点，撰写开题报告。
3. 时间：2017.01 月初 – 2017.02 月末
   阅读相关的文献资料，整理思路，初步确立模型和解题思路，并用代码实现初步思路。
4. 时间：2017.03 月初 – 2017.03 中旬
   在不断阅读文献的同时，结合前期实验结果，调整思路，优化算法设计。
5. 时间：2017.03 月中旬 – 2017.04 月末

优化代码，实验调参。在与之前的方法进行比较基础上改进。

6. 时间：2017.05 月初 – 2017.05 月中旬

进一步完善、改进算法中的各个模块，撰写论文。

七、实验方案可行性分析

目前，强化学习在解决多智能体系统全局最优的问题上已经存在一定的研究基础，在有限通讯条件下，区域内社会规范涌现也做出了很多的探索。

基于这些科研基础，在有限通信，分布式的条件下，基于强化学习的方式，在保证信息损失在可接受范围条件下，将集中式，大规模计算拆分为分布式，合作式计算，结合博弈论模型，解决问题是可行的。

八、已具备的实验条件

文献资料可通过 Google 学术，及天大图书馆检索，实验室机器等设备完善。

九、主要参考文献

[1] Zhang C, Lesser V. Coordinating multi-agent reinforcement learning with limited communication[C]//Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems. International Foundation for Autonomous Agents and Multiagent Systems, 2013: 1101-1108.

[2] Tianpei Yang ,and Zhaopeng Meng , Jianye Hao Accelerating Norm Emergence Through Hierarchical Heuristic Learning ECAI 2016 G.A. Kaminka et al. (Eds.)

[3] R.Becker,S. Zilberstein, V. Lesser, and C. V. Goldman.Transition-Independent Decentralized Markov Decision Processes. In Proceedings of the Second International Joint Conference on Autonomous Agents and Multi Agent Systems,pages 41–48, Melbourne, Australia, 2003. ACM Press.

[4] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein.The complexity of decentralized control of markov decision processes. Mathematics of Operations Research,27(4):819–840, 2002.

[5] S. Cheng. Coordinating Decentralized Learning and Conflict Resolution Across agent Boundaries. PhD thesis, University of North Carolina at Charlotte, 2012.

[6] C. Claus and C. Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. In AAAI’98,pages 746–752. AAAI Press, 1998.

[7] C. Guestrin, D. Koller, and R. Parr. Multiagent planning with factored mdps. In NIPS-14, pages 1523–1530, 2001.

[8] C. Guestrin, M. G. Lagoudakis, and R. Parr. Coordinated reinforcement learning. In ICML ’02: Proceedings of the Nineteenth International Conference on Machine Learning,pages 227–234, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[9] C. E. Guestrin. Planning under uncertainty in complex structured environments. PhD thesis,

Stanford University,Stanford, CA, USA, 2003.

[10] J. R. Kok and N. Vlassis. Collaborative multiagent reinforcement learning by payoff propagation. Journal of Machine Learning Research, 7:1789–1828, 2006.

[11] R. Stranders, A. Farinelli, A. Rogers, and N. R. Jennings. Decentralised coordination of mobile sensors using the max-sum algorithm. In IJCAI, pages 299–304, 2009.

[12] P. Varakantham, M. Tambe, and M. Yokoo. Networked distributed pomdps: A synthesis of distributed constraint optimization and pomdps. In AAAI, pages 133–139, 2005.

[13] S. J. Witwicki and E. H. Durfee. Influence-based policy abstraction for weakly-coupled dec-pomdps. In R. I.Brafman, H. Geffner, J. Hoffmann, and H. A. Kautz, editors, ICAPS, pages 185–192. AAAI, 2010.

[14] C. Zhang, S. Abdallah, and V. Lesser. Integrating organizational control into multi-agent learning. In AAMAS'09, 2009.

[15] C. Zhang, V. Lesser, and S. Abdallah. Self-organization for coordinating decentralized reinforcement learning. In AAMAS'10, 2010.

[16] C. Zhang and V. R. Lesser. Coordinated multi-agent reinforcement learning in networked distributed pomdps. In W. Burgard and D. Roth, editors, AAAI. AAAI Press, 2011.

选题是否合适： 是□ 否□

课题能否实现： 能□ 不能□

指导教师（签字）

年 月 日

选题是否合适： 是□ 否□

课题能否实现： 能□ 不能□

审题小组组长（签字）

年 月 日

摘　要

社会规范，对规范多智能体系统中各个智能体的行为，并加快群体协作的达成具有重要的意义。其中，一个重要的问题是，在现实的应用系统中，智能体之间通信资源有限的情况下，如何高效利用仅有的通信资源而加速社会规范的生成。本文中，我们设计了一种算法，使各个智能体在学习的过程中，能够根据自己当前时刻的状态，动态调整需要与自己协调的智能体集合，选择出最为关键的智能体进行通信协调，以最大限度的利用仅有的通信资源。方法大大减小了智能体之间在决策时的相互依赖，降低了通信资源的消耗。实验表明，我们的方法能够根据不同系统中不同通信资源的限制，快速地生成社会规范，加速群体协作的达成。同时我们的方法适用于多种不同的网络结构及规模足够大的多智能体系统。

关键词：合作式多智能体系统；强化学习；有限通信；社会规范；网络拓扑

ABSTRACT

Social norms is an important mechanism to regulate the behavior of agents and facilitate the coordination among them in cooperated multi-agent systems. One important problem is how a norm can rapidly emerge for most applications in realistic with limited communication resources. In this paper, we propose a learning approach that the agents can dynamically adjust their coordination set according to their own observations and pick out the most crucial agents to coordinate. Through this, our method reduces the dependence among agents. Hence our method can trade off the agents performance and the communication cost. The experiment results say that our method can efficiently facilitate the social norm emergence among MAS, and scale well in large systems.

Keywords：cooperative multi-agent system； reinforcement learning；
limited communication；social norms；network topology

目 录

第一章　绪论

1.1　课题背景及相关工作

多智能体系统（multi-agent system）由处于同一个环境中的多个智能体（agent）组成，其中每个 agent 都能够与其他某些个 agent 进行沟通，以达到一定的目的(Sycara, 1998[1]; Weiss, 1999[2]; Durfee, 2001[3]; Vlassis, 2003[4])。多智能体系统的出现，为解决复杂的分布式问题提供了新的思路。在现实应用中，合作式的多智能体系统（Cooperative multi-agent system, MAS）十分普遍，比如：机器人系统、传感器网络、分布式的协调控制、合作式的决策系统等等。一个合作式的多智能体系统（由许多个能够独自决策的智能体（agent）组成。每个智能体（agent）通过在一个公共的环境中，与可交互范围内的其他个体不断通信协调，各自选择最合适的动作，以达到群体既定的目标，或者提高群体的整体收益（payoff）。在分布式多智能体系统（MAS）的环境下，一个最主要的问题是如何设计每个智能体的决策策略，以协调彼此之间的动作选择，从而提高系统的整体收益。例如，在由四个传感器（sensor）组成的传感器网络，如图 1-1 所示，每个传感器可以监测上下左右四个位置上的环境变化，为了保证监测的精准度，规定只有当两个传感器同时监控同一个位置时，才会得到监测环境的具体数值并且获得一定的收益。当 sensor1 与 sensor2 同时监测 location 1 时获得+30 收益；当 sensor2 与 sensor3 同时监测 location2 时；获得+50 收益，当 sensor3 与 sensor4 同时监测 location 3 时，获得+40 收益。如果各个 sensor 只考虑自己的收益，那么 sensor2 与 sensor3 会选择同时监测 location2 以获得各自最大收益+50。但是如果考虑系统整体的收益，则 sensor1 与 sensor2 应该监测 location 1，sensor3 与 sensor4 应该监测 location3，此时系统可以收到最大收益+70。

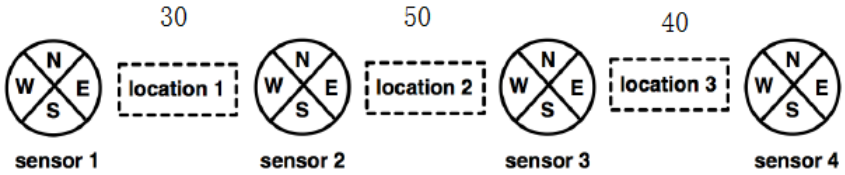


图 1-1　传感器网络举例

在合作式多智能体系统研究工作中，社会规范（social norms）在规范每个智能体的行为，加速群体合作行为的达成具有极其重要的意义。比如，我们人类社会中的交通规则，在马路上是靠左行驶，还是靠右行驶。对社会规范（Norm）的一个比较普遍的理论描述是，当 agent 的动作空间（例如：靠左行驶，靠右行驶）

中存在多个纳什均衡点时，规范（Norm）是群体通过协商选择出的一个确定的纳什均衡点。其中，在存在多个纳什均衡点的多智能体系统中，如何快速使整个系统统一于同一个纳什均衡点的选择，从而加速规范（Norm）的涌现，具有重要的意义。在分布式的环境下，由于外部环境的变化频繁，往往无法针对每种可能出现的情况，提前设计一个规范来约束每个智能体的行为，并且能够实现群体的最大收益。因此在分布式的多变环境中，通过各个智能体不断与外界环境（包括通信范围内的其他智能体）进行交互，根据环境的反馈收益，学习到自己的策略，并不断学习更新自己的策略，从而最终自动生成一个良好的社会规范。

当前，社会规范涌现问题的研究已经取得了很快的发展。Sen, Airiau[13][5] 通过随机生成的网络，来模拟群体间的规范涌现问题。网络中，每个节点代表具备学习能力，并能够独立决策的 agent，每两个节点之间的交互过程可以抽象成由两个 player 组成的常规博弈游戏，如图 1-2 所示，一个由两个 player、每个 player 有两个可选动作的常规博弈游戏。并且规范代表博弈中的一个具体的确定的纳什均衡点。随后，在其基础上，很多研究[6,7,8,9]通过生成更加复杂的或者更加符合某种问题模型的（比如 small-world 网络结构模拟人的交际圈）的网络结构，来模拟群体之间的交互关系。

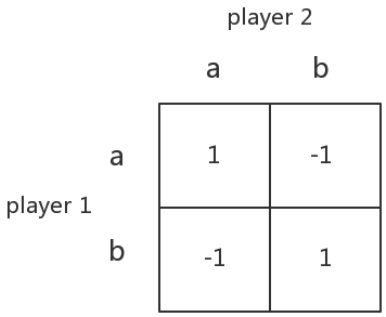


图 1-2　two player-two action game

但是，大多数当前的研究工作，所针对的博弈游戏的规模往往比较小，并不能良好的反应现实中，agent 及每个 agent 可选动作的数量比较多的情况。当每个 agent 的动作空间比较大时，很多研究设计的算法便不能很快的生成规范，甚至不能生成规范，因此这类算法不适用于大规模的合作式多智能体系统。随后，针对此问题，Yu et al.[10]设计了一系列基于层次学习的算法，来加快大网络结构，大动作空间下，社会规范的涌现问题。协调的多智能体学习算法（Coordinated multi-agent learning approaches[11,12,13]），利用分布式约束满足算法（DCOP：Distributed constraint optimization），来协调学习过程中每个 agent 的动作选择。但

是在算法中，假设每个 agent 都可以跟其他 agent 进行很强的通信，即每个 agent 可以与通信范围内的 agents 进行无限次通信。但是，现实中，情况往往不够理想，每个 agent 通信的距离及通信的带宽往往是有限的，本文拟在大规模网络结构及动作空间下，通过使用有限的通信资源，最快的达到社会规范的快速涌现。

1.2 本文创新工作

在现实的 agent 网络中，虽然当前 agent 的决策依赖于由周围很多 agent 所组成的集合（CS, Coordination Set），但是在很多情况下（比如：当前局部网络结构下，各个 agent 已经达成了最优的协调，所以他们不再需要另外的通信协调），每个 agent 只需要与周围最影响其表现（收益）的几个特定的 agents 进行通信协调。于是，我们针对这个特点，对网络中任意的一个 agent i 设计了一种方式能够衡量其各个邻居 agent 对其收益的影响。通过约束系统允许的最大损失，进一步可以选择出那些对其影响最大的 agent 子集，作为当前时刻需要交互的协调集合（Coordination Set），于是减少了初始网络结构中的很多条边，并且往往初始的复杂网络会被分割成多个小网络，从而大大减小了 agent 之间通信的数量，节约了通信资源。实验结果证明了，我们的方法可以有效的在系统的表现及系统的通信资源耗费之间取得折中。并且，我们通过比较了在不同网络结构的环境中，方法的收敛速度。在规模足够大的网络结构，及 agent 的动作空间条件下，与传统方法相比，我们的方法能更快地达到收敛，即更快地生成规范。最后，通过随机调整系统在两个 agent 之间设定的收益函数，发现，我们的方法都能够较快的达到系统最优的纳什均衡点。

1.3 论文结构

后续文章的结构如下：第二章，介绍了后续文章中可能用到的理论基础。第三章是对文章所针对的单状态 Coordination Game 的理论及符号描述。第四章，是文章核心算法部分的详细描述。第五章，是针对几个特定 game 所做的对比实验及结果分析。第六章，对文章做出总结，并介绍后续一些研究工作的方向。

第二章　理论基础

2.1 博弈论&纳什均衡

2.1.1 博弈论（Game theory）

博弈论也称为对策论，或者赛局理论，是研究具有斗争或竞争性质现象的数学理论和方法。简单来说**博弈论**是对包含**相互依存**情况的环境中理性行为的研究，主要对参与者之间策略交互的行为进行建模。

- 相互依存：通常是指博弈中的任何一个博弈方的行为受到其他博弈方行为的影响，反过来，他的行为也影响到其他博弈方。
- 理性行为：博弈论中的理性，一般指的不是道德标准，一般是与博弈方自身利益或整体利益相关。
- 博弈方：参与博弈但利益不完全一致者，有二人博弈与多人博弈之分。每个个体，都希望在博弈中，能够尽可能的提高自己的收益。
- 策略：每个博弈方都会有一系列的策略可选，称为对应于每个博弈方的策略集。
- 收益：博弈方选定一组策略后，按照此策略执行动作后的得益情况。

2.1.2普通的表格游戏（Normal Form Games）

Normal Form game 由元组（$n, A_{1,\dots n}, R_{1,\dots,n}$）组成，其中：

- $1, \dots, n$ 是游戏中的博弈方组成的集合，一般叫做博弈游戏的 player。
- A_k 是每个 player k 可选的动作集合（$A_k = \langle a_1, a_2, \dots, a_m \rangle$集合中包含 m 个可选动作）。
- R_k: $A_1 \times \dots \times A_n \to R$，是每一次博弈中，各个博弈方选择动作$\mathbf{a} \in A_1 \times \dots \times A_n$时，博弈方 k 收到的收益。
- 策略π_k: $A_k \to [0,1]$，是博弈方 k 在其动作集合A_k 中选择各个动作a_m的概率。
- 纯策略（pure strategy）：$\pi_k(a_k) = 1$ 对与当前选择的动作，并且对于其他的动作，$\pi(a_{j,j\neq k}) = 0$。混合策略（mixed strategy），选择各个动作的概率满足特定的概率分布。可以把纯策略看作是特殊情况的混合策略。

2.1.2 纳什均衡（Nash Equilibrium）

我们用包括两个博弈方 player1，player2 的 Norm Form Game 来解释这个问题

- **Best Response**

当 player1 选择动作a_1 的条件下，player2 在针对 player1 的动作选择，在自己的动作集合中选择能最大化自己收益的动作，即 $a_2 = argmax_{a_2} R_2$

- 纳什均衡

当博弈方都不改变自己策略的前提下，每一个博弈方，都是对其他博弈方动作惦的最佳响应，即使纳什均衡。player1 选择 a_1与 player2 选择 a_2，当双方都不改变自己动作选择的条件下，$a_1 = \text{Best Response of player2}(a_2), \text{and } a_2 = \text{Best Response of player1}(a_1)$。

2.2 Cooperative multi-agent system

合作式的多智能体系统（Cooperative multi-agent system），由许多个能够独自决策的智能体（agent）组成。每个智能体（agent）通过在所处的公共环境中，与交互范围内的其他 agent 不断交互，各自选择最合适的动作，以达到群体既定的目标，或者提高群体的整体收益（payoff）。

2.3 社会规范

在合作式多智能体系统研究工作中，社会规范（social norms）在规范每个智能体的行为，加速群体合作行为的达成具有极其重要的意义。比如，我们人类社会中的交通规则，在马路上是靠左行驶，还是靠右行驶。对社会规范（Norm）的一个比较普遍的理论描述是，当 agent 的动作空间（例如：靠左行驶，靠右行驶）中存在多个纳什均衡点时，规范（Norm）是群体通过协商选择出的一个确定的纳什均衡点。

2.4 强化学习

2.4.1 MDP（Markov Decision Process）马尔科夫决策过程

一个基本的 MDP 可以用（S,A,P）来表示，S 表示状态集合，A 表示动作集合，P 表示状态转移概率，也就是根据当前的状态s_t和a_t转移到s_{t+1}的概率。下一个状态s_{t+1}只取决于当前的状态s_t和当前的动作a_t，而与以前更早的状态和动作无关。当我们知道了状态转移概率 P，也就是我们获得了**模型 Model**。有了模型，未来就可以求解，那么获取最优的动作也就有可能，这种通过模型来获取最优动作的方法也就称为 Model-based 的方法。但是现实情况下，很多问题是很难得到准确的模型的，因此就有 Model-free 的方法来寻找最优的动作。

2.4.2 强化学习简介

强化学习简单来说是，环境中的 agent 通过不断与环境交互，并根据环境的

收益反馈，来更新自己选择动作的策略，以最大化自己的长期收益的过程，它是一种试错式的学习方式，试错和延迟回报（收益）是强化学习的两个重要特征。由以下几个要素组成：

- agent：学习的主体，与环境交互的对象。
- 环境：agent 所处的空间，分为静态环境与动态环境。
- 动作（action space）：环境下，每个 agent 可行的动作集合，分为离散和连续。
- 反馈（收益、回报）：当前环境下，衡量 agent 当前动作好坏的方式。

2.4.3 Q-Learning

Q-learning 是强化学习中的一个重要里程碑，它是一种模型无关（Model Free）的算法，是 TD(0)算法的典型应用。Q-learning 中一种最简单的形式如公式（2-1）所示：

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha[r_{t+1} + \gamma max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (2\text{-}1)$$

上式中，α称为学习率（表示学习新知识的快慢），γ为折扣率，表示当前收益与未来可能的收益的权衡。$Q(s_t, a_t)$是状态动作值函数，表示在状态s_t下，执行动作a_t，所得到的累积收益。一个典型的 Q-learning 过程描述如下：

Algorithm 1 Single Q-learning procedure

initialize $Q(s, a) = 0$
repeat for each episode:
    initialize s_0
    repeat for each step:
        choose a from s using policy derived from $Q(e.g., \varepsilon - greedy)$
        take action a, and observe reward r and the next state s'
        $Q(s, a) = Q(s, a) + \alpha[r + \gamma max_a Q(s', a) - Q(s, a)]$
        $s = s'$
    until s is terminal

2.5 网络拓扑结构

2.5.1 Grid Network

网格网络结构，即网络中的各个结点，是方格状布局。

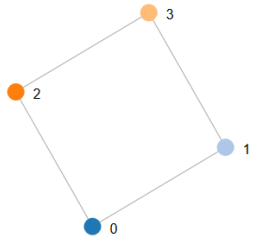


图 2-1　grid Network

2.5.2　Regular Network

规则网络结构，在环状网络的基础上，网络中的 n 个节点分别与自己最近的 m 个节点连接。当 m=n-1 时，则为全相连网络（fully-connected network）。

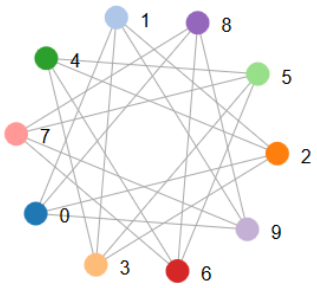


图 2-2　regular network

2.5.3　Random Network

Random 网络是一种随机连接组成的网络结构。经典的模型是埃尔德什和雷尼共同研究提出的 ER 模型。ER 模型是指在给定图中的 n 个顶点后，规定每两个顶点之间都有 p 的概率相互连接（$0 \leqslant p \leqslant 1$）。

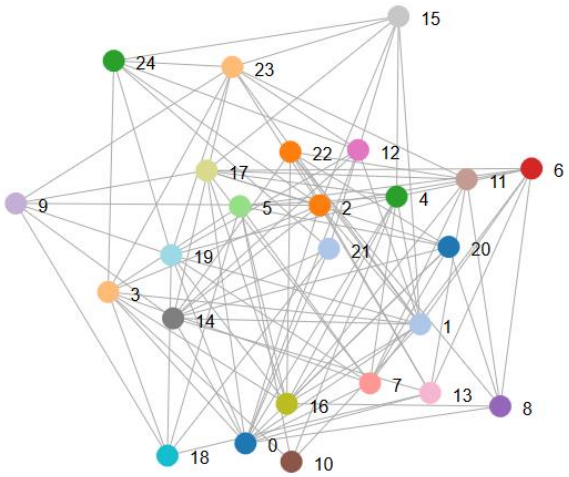


图 2-3　random network

2.5.4 Small World Network

Small world 最早是在社会人际关系网络中被提出的，将每个人作为结点，人与人之间的人际关系（认识与否，熟悉与否等）作为网络中的边。瓦茨-斯特罗加茨模型，WS 模型是基于一个假设：小世界模型是介于规则网络和随机网络之间的网络。因此模型是从一个完全的规则网络出发，以一定的概率将网络中的连接打乱重连。

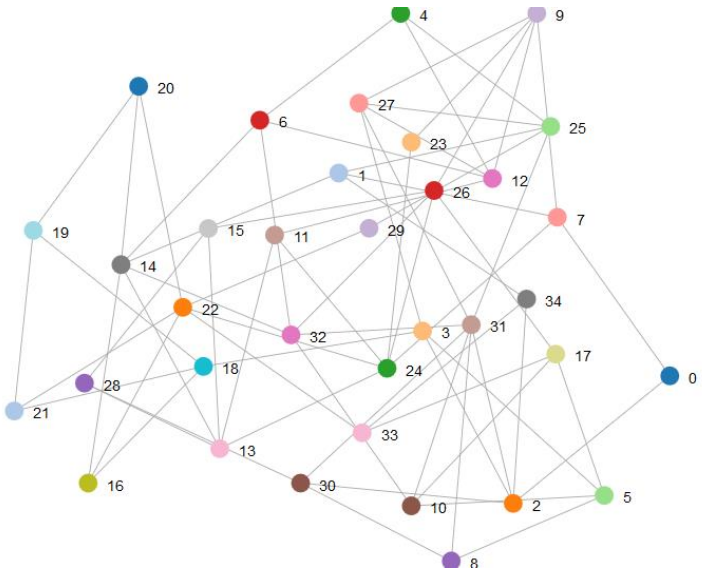


图 2-4 small-world network

2.5.5 Scale Free Network

无尺度网络中的节点之间并不是随机相连，网络中只有一少部分节点作为网络的中心节点，负责连接很多个节点，而其它大多数节点仅与以少部分节点相连接。

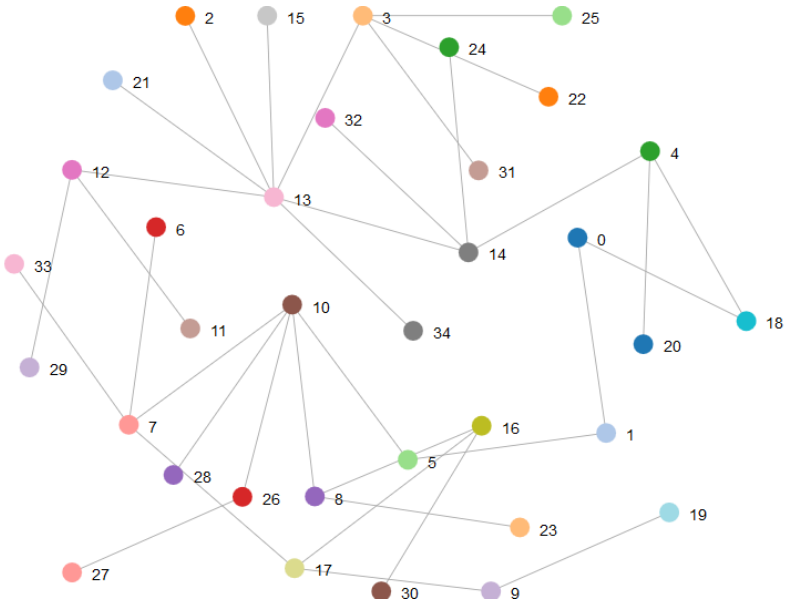


图 2-5　scale-free network

第三章　问题描述

3.1 符号定义

- n 是系统中 agents 的数量。
- 每个 agent 只有一个状态。
- $\boldsymbol{\Gamma(i)}$ 代表 agent i 的所有邻居。
- ***CS(i)*** agent i 的 Coordination Set，***NC(i)*** agent i 的邻居中，不在当前 ***CS(i)*** 中的 agents 组成的集合 $\boldsymbol{NC(i)=\Gamma(i)\backslash CS(i)}$。
- $\boldsymbol{CG}$ Coordination Graph。
- $\boldsymbol{A_i} = \langle a_1, a_2, \dots a_k \rangle$ 是 agent i 的动作空间，即 agent i 有 k 个可选动作 $a_1, a_2, \dots a_k$。 $\boldsymbol{A} = \boldsymbol{A_1} \times \dots \times \boldsymbol{A_n}$ 是系统 agents 的联合状态空间。其中 $\boldsymbol{a} = \langle a_1, a_2, \dots a_n, \rangle$， $\boldsymbol{a} \in \boldsymbol{A}$，表示当前所有 agents 的动作选择。
- $r(a_i, a_j)$ 是系统预设的 reward table，a_i, a_j 是环境中相邻 agents i,j 选择的 action，当 $a_i = a_j$，$r(a_i, a_j) = 1$ 否则，$r(a_i, a_j) = -1$。系统整体收益 $R(\boldsymbol{a}) = \sum_{i=1}^{n} r(a_i)$。
- 假定每个 agent i 可以观察到与其交互的 agent j 的 action 选择，并且可以统计最近时间段内，对手选择各个 action 的频率。
- $Q(i,j)$ 用来记录相邻 agent i,j 之间的学习经验，以对 agent 每个 action 的优劣进行评估。$\boldsymbol{Q(a)}$ 代表系统，对联合 action $\boldsymbol{a}$ 的评估。
- $\boldsymbol{\pi_i}$ 是 agent i 选择 action 的策略，$\boldsymbol{\pi_i \rightarrow a_i}$。$\boldsymbol{\pi} = \boldsymbol{argmax_{a \in A} Q(a)}$，是系统的整体策略（global policy）。

3.2 基于单状态的协调问题

定义，系统环境是由 n 个 agent 组成的合作式的多智能体系统，每个 agent 独立决策，并且通过对环境的探测与学习，选择对整体最优的动作，来最大化系统整体的收益。系统中每个 agent i 根据自己的策略，选择出动作 action a_i，随机地与邻居进行交互。随即，当动作执行后，一轮游戏结束，并且每个 agent i 各自收到一个回报 r_i。每个 agent i 的目标是选择出各自最优的动作 a_i^* 以最大化系统整体收益 $R(\boldsymbol{a}^*) = \sum_{i=1}^{n} r(a_i^*)$。

每个 agent i 在每一轮收到的回报 r_i 取决于与其交互的邻居 agent j。依赖关系可以通过无向图 $G = (V, E)$ 进行表示，其中每一条边 $(i,j) \in E$ 对应于相邻节点 agent i,j 选择各自动作 a_i, a_j 后的收益 $r(a_i, a_j)$，如图 3-1 所示，收益函数 $r(a_i, a_j)$ 由系统提前设定。例如，对每个 agent i，$a_i \in \boldsymbol{A_i}$，$\boldsymbol{A_i} = \langle a_1, a_2, \dots a_k, \rangle$，回报函数 $r(a_i, a_j)$ 定义如下：

$$r(a_i, a_j) = \begin{matrix} \\ a_i \end{matrix} \begin{matrix} \begin{matrix} a_j \end{matrix} \\ \begin{bmatrix} 1 & \cdots & -1 \\ \vdots & \ddots & \vdots \\ -1 & \cdots & 1 \end{bmatrix} \end{matrix}$$

如果 agent i, j 同时选择在对角线上的动作组合 a_i, a_j，其中 $i = j$,则双方各自收到 reward +1，否则协调失败，收到 reward -1，如图 3-2 所示。

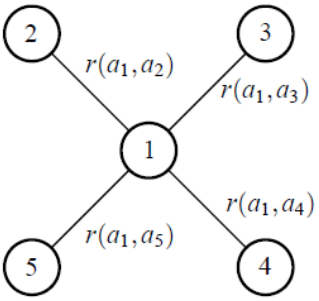


图 3-1 收益依赖关系

player 1 \ player 2	a	b	c	d
a	1	-1	-1	-1
b	-1	1	-1	-1
c	-1	-1	1	-1
d	-1	-1	-1	1

图 3-2 coordination game

第四章　算法

4.1 Coordination Graph

在协作式的多智能体系统中，每个 agent 的动作选择会对其他 agent 产生潜在的影响，即系统中各个 agent 之间存在依赖关系，一个 agent 动作的选择会取决于其他 agent 的决定，比如：图 1 所描述的传感器的例子，每个传感器的动作选择，依赖于相邻传感器的动作选择，只有相邻传感器同时选择监测同一个地方时，才会收到正的收益。所以保证各个 agent 每个时刻选择的动作都是针对整个系统的最优决策，对提高系统的整体收益具有重要的意义。通常这种问题被定义为协调问题 (Coordination Problem)。本章节，我们首先回顾由 Guestrin et al. (2002a)[14]提出的问题，计算对由n个 agents 组成的协作式多智能体系统整体最优的动作组合。系统中每个 agent i 从各自的动作集合 $\boldsymbol{A_i}$ 中选择一个 action a_i 整体组成一个动作向量（联合动作） $\boldsymbol{a} = (a_1, a_2 \dots a_n)$，进一步系统得到环境提供的一个收益 $u(\boldsymbol{a})$。协调问题的目标是选择一个动作向量 $\boldsymbol{a}^*$ 以最大化系统的整体收益 $R(\boldsymbol{a})$，即 $\boldsymbol{a}^* = \ argmax_a u(\boldsymbol{a})$。

针对这个问题，可以遍历所有可能的动作向量，并且选择可以最大化 $u(\boldsymbol{a})$ 的动作向量。但是，很快发现这个思路是不现实的，因为问题的解空间 $|\boldsymbol{A_1} \times \boldsymbol{A_2} \times \dots \times \boldsymbol{A_n}|$ 的规模，随着系统中 agent 的数量 n 成指数增长。幸运的是，现实的很多问题中，每个 agent 的决策只依赖于与其非常相关的一少部分。

由Guestrin et al.,2002a[14]提出的协调图(coordination graphs ,CGs)架构是解决此类策略相互依赖问题的一种方式。此架构假设对一个 agent i，其动作的选择只依赖与与其相关的 agent $j \in \boldsymbol{\Gamma(i)}$ 集合，其中$\boldsymbol{\Gamma(i)}$代表 agent i 的所有邻居。系统整体的收益 $R(\boldsymbol{a})$ 由系统中每个 agent i 的收益 $r(i)$ 之和组成，如公式（4-1）所示：

$$R(\boldsymbol{a}) = \sum_{i=1}^{n} r(\boldsymbol{a_i}) \qquad (4-1)$$

每个 agent i 的收益 $r(i)$ 取决于与其密切相关（有依赖关系）的所有 agent 的动作选择，$\boldsymbol{a_i} \subseteq \boldsymbol{a}$，　$\boldsymbol{a_i} = \boldsymbol{A_i} \times (\times_{j \in \boldsymbol{\Gamma(i)}} \boldsymbol{A_j})$，这种相互依赖关系可以通过无向图 $G = (V, E)$表示，其中每个节点 $i \in V$ 表示 agent，每条边 $(i,j) \in E$ 表示相关的 agents i,j 需要协调各自动作的选择，$j \in \boldsymbol{\Gamma(i)}$ 并且 $i \in \boldsymbol{\Gamma(j)}$，每条边上标记的值，代表相关联的两个 agent i,j 各自选择 action a_i, a_j 所得到的收益 $r(a_i, a_j)$。于是整个系统的协调问题，被拆分为一定数量的局部协调问题，并且减小了问题的规模。协调图（CG）的示例如图 4-1 所示。

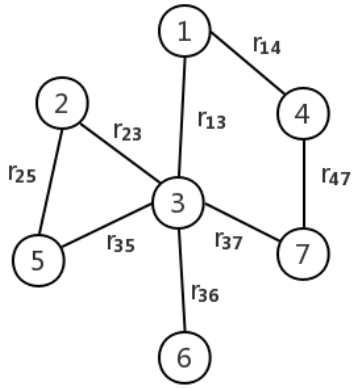


图 4-1　协调图(CG)

4.2　Learning Processes with Emergent Coordination

Algorithm 2 描述了网络中, agents 合作式学习的过程。学习过程中，将最大化所有 agents 整体的$\boldsymbol{Q}(\boldsymbol{a})$ 分散到了每对 agent *i,j* 组合,每个 agent i 记录所有与其 Coordination Set 中的 agent j 的动作值函数$\mathrm{Q}_i(a_i, a_j)$。loss rate $\delta \in [0,1]$ 用来计算在当前时刻，所需要的最优的 Coordination Set（Coordination Set 中包含了需要与当前 agent 进行协调选择 action 的 agent 集合）。δ 是用来在保证系统整体收益满足要求的条件下，减少各个 agent 之间通信次数。网络中，每个 agent i 都需要不断地统计每个交互的 agent j, 在自己选择 action a_i的条件下，对手选择 action a_j的概率 $\mathrm{P}_j(a_j|a_i)$。概率$\mathrm{P}_j(a_j|a_i)$ 将用来计算当 agent $j \notin \boldsymbol{CS}(i)$ 时，agent i 选择 a_i 后可能收到的来自 agent j的收益$Q_i(a_i)$。比如，当$\boldsymbol{CS}(i) = \emptyset$，agent i 需要根据其所有邻居 action 选择的统计概率计算自己当前 action a_i的预期回报。计算方式如如公式（4-2）所示：

$$Q_i(a_i) = \sum_{j \in \Gamma(i),\ j \notin C} \sum_{a_j \in \boldsymbol{A_i}} P_j(a_j|a_i) Q_{ij}(a_i, a_j) \qquad (4\text{-}2)$$

在后面，$P_j(a_j|a_i)$将用来计算 agent i 的最优 Coordination Set。

注意，Coordination Set 可能会随着 Coordination Action Selection 的过程中发生变化。比如当前 agent i 的 Coordination Set 不包含 agent j，但是 agent j 的 Coordination Set 中包含 agent i，在执行 DCOP 时，agent i 会把 agent j 加入到其 Coordination Set 中。

Algorithm 2 The coordinated learning process

1：　initialize learning rate $\alpha = 1$,explore rate $\varepsilon = 1$, loss rate $\delta = 0.001$

2：　**while** not converge **do**

3：　　runDCOP() to select the best action a_i^* for each agent i

4： **for** every agent *i* **do**

5： random select a neighbor *j* to interact

6： each agent *i,j* select the its' action a_i, a_j(each select the best action a_i^*, a_j^*

7： with some explore rate ε)

8： each agent observed the reward $r(a_i, a_j)$,and observed each other's action (for

9： record $P_j(a_j|a_i)$ *and* $P_i(a_i|a_j)$)

10： each agent update its' Q table

11： agent *i* update its' learning rate α and explore rate α with some decay

12： computeCoordinationSet(*i*)

13： **end for**

4.3 Coordination Action Selection

4.3.1 Cooperative Q-learning

由于实验环境中，每个 agent 不能直接获取系统预设的回报函数（或 reward table），因此需要通过学习不断与环境进行交互、探测，进而对自己动作集合 A 中的每个 action 的优劣进行评估。这里使用 Q-learning 来对相邻 agent 的学习行为进行建模。

于是，对于此问题，各 agent 之间的依赖关系，可以通过协调图 $G = (V, E)$ 表示，其中每个节点 $i \in V$ 表示每个 agent，每条边 $(i, j) \in E$ 表示相关的 agents i, j 的局部 Q 函数 $Q(i, j)$，如图 4-2 所示。

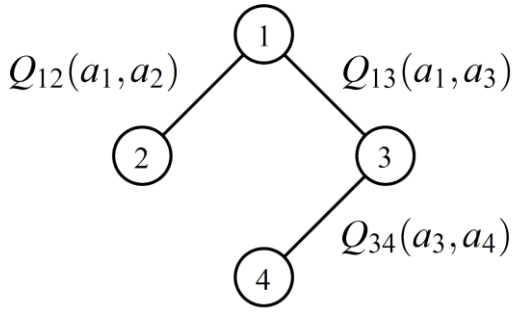


图 4-2　agent 依赖关系图

我们的目标是，找到一个策略$\boldsymbol{\pi} = argmax_{\boldsymbol{a} \in \boldsymbol{A}} \boldsymbol{Q}(\boldsymbol{a})$，以最大化系统的整体收益。对于一个包含多状态的 MDP 问题，可以简单的对整体使用 single Q-learning，如公式（4-3）所示：

$$\mathbf{Q}(\boldsymbol{s_t}, \boldsymbol{a^t}) = (1 - \alpha)\boldsymbol{Q}(\boldsymbol{s_t}, \boldsymbol{a^t}) + \alpha\left[r^t + \ \gamma \boldsymbol{max}_{\boldsymbol{a^{t+1}}} \boldsymbol{Q}(\mathbf{s_{t+1}}, \boldsymbol{a^{t+1}})\right] \quad (4\text{-}3)$$

但是，由于系统整体的策略空间随 agents 的数量 n，并且往往无法观察到其他 agent 的所有信息，因此进一步把整体的 **Q** 函数拆分成各个 agent Q 函数的线性组合，如公式（4-4）所示：

$$\mathbf{Q}(\mathbf{s_t}, \boldsymbol{a^t}) = \sum_{(i,j)\in E} Q_{ij}(s_{i,j}^t, a_i, a_j) \quad (4\text{-}4)$$

于是，等式(1) 可以被重新表示为：

$$\sum_{(i,j)\in E} Q_{ij}(s_{i,j}^t, a_i, a_j) = (1-\alpha)\sum_{(i,j)\in E} Q_{ij}(s_{i,j}^t, a_i, a_j) + \alpha\big[r_{ij}{}^t + \gamma \boldsymbol{max}_{\boldsymbol{a^{t+1}}} \boldsymbol{Q}(\mathbf{s_{t+1}}, \boldsymbol{a^{t+1}})\big] \quad (4\text{-}5)$$

上式（4-5）中，因为$\boldsymbol{max}_{\boldsymbol{a^{t+1}}} \boldsymbol{Q}(\mathbf{s_{t+1}}, \boldsymbol{a^{t+1}})$ 取决于对整体最优的联合 action $\mathbf{a}^*$，因此不能直接拆分为各个 agent 局部最优 Q 值之和。但是我们可以通过 VE(Variable Elimination Guestrin et al. (2002a)[14])或 Max-Plus(J. R. Kok and N. Vlassis.(2006)[12])等方式，通过使每个 agent i 选择出对整体最优的 action a_i^* 来计算出对整体最优的联合 action $\mathbf{a}^*$。其中$\boldsymbol{max}_{\boldsymbol{a^{t+1}}} \boldsymbol{Q}(\mathbf{s_{t+1}}, \boldsymbol{a^{t+1}}) = \boldsymbol{Q}(\mathbf{s_{t+1}}, \boldsymbol{a}^*) = \sum_{(i,j)\in E} Q_i(s_{i,j}^{t+1}, a_i^*, a_j^*)$。于是对于每一个 agent 对，有：

$$Q_{ij}\big(s_{i,j}^t, a_i, a_j\big) = (1-\alpha)Q_{ij}\big(s_{i,j}^t, a_i, a_j\big) + \alpha r\big(a_i, a_j\big) + \gamma Q_{ij}\big(s_{i,j}^{t+1}, a_i^*, a_j^*\big) \quad (4\text{-}6)$$

对于单状态的协调问题，下一个状态的 Q 函数没有定义，因此在本实验中，每个 agent i 在每一轮中，选择自己的 action 时，直接考虑选择对当前系统整体最优的action a_i^*，并且以一定的探索率 ε 随机对动作空间中的action进行探索。

4.3.2 Payoff Propagation and Max-Plus Algorithm

如上节所示，实验中各 agents 的协调图 CG（Coordination Graph）如图 11 所示。为了计算对整体最优的 action a^* (最大化$\mathbf{Q}(\mathbf{s}, \boldsymbol{a})$)，于是，每个 agent i (CG 中的节点)，向它的邻居 agent $j \in \boldsymbol{\Gamma(i)}$ 不断的发送消息μ_{ij}，发送的消息μ_{ij}定义为：

$$\mu_{ij}\big(a_j\big) = max_{a_i}\big\{Q_{ij}\big(a_i, a_j\big) + \sum_{k\in\Gamma(i)\backslash j}\mu_{ki}(a_i)\big\} + c_{ij} \quad (4\text{-}7)$$

其中 $\Gamma(i)\backslash j$ 表示 agent i 除了 j 以外的所有邻居。参数 c_{ij} 是为了标准化消息数值的取值范围。这个消息 μ_{ij} 是对给定一个目标 agent j 的动作 a_j，agent i 所能实现的最大收益值的近似（即 best response to action a_j）。通过最大化与目标 agent j 之间的平均回报 $Q_{ij}\big(a_i, a_j\big)$ 以及 agent i 的所有邻居（j 以外的）向其发送的消息数值总和来计算当前消息 μ_{ij}。注意，这个消息只取决于 agent i与 agent j之

间的收益和所有发送到 agent i的消息。每个 agent 不断向邻居发送消息直到消息的值不再变化（收敛到一个稳定值），或者到达指定的最大发送轮数（或者收到某些终止信号）。当网络中所有消息值都达到稳定时，每个消息中都包含了网络中所有边（i,j）上的收益，所以最大化当前消息值即最大化了系统的整体收益 **Q**。如图 4-3 所示，展示了一个由 4 个 agent 组成的 Coordination Graph 中，消息传递的过程。

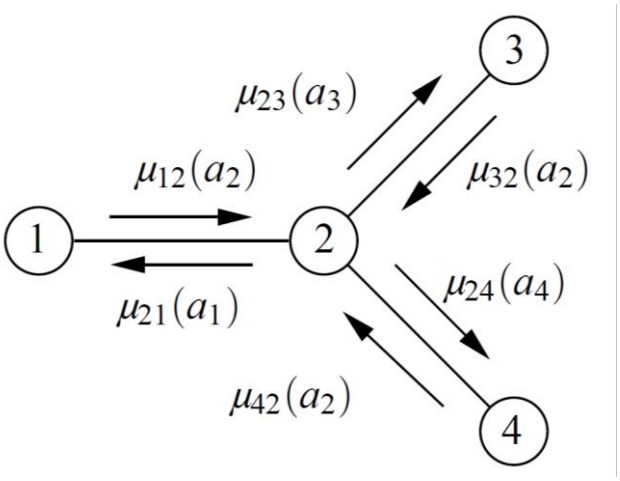


图 4-3　消息传递过程

当网络结构是一棵树时，很明显，经过有限次消息发送后，所有消息的值收敛到一个固定值(Pearl, 1988[15];Wainwright et al., 2004[16]).因为每个消息等于其所有子树产生的收益总和，所以在每一步，对每个agent i，即找到了能最大化整体收益的 action a_i^*：

$$a_i^* = \ argmax_{a_i} \sum_{k\in\Gamma(i)} \mu_{ki}(a_i) \qquad (4\text{-}8)$$

但是，当网络中存在环状结构时，并不能保证 max-plus 可以收敛到一个固定值，因此并不能保证当前根据式（4-8）选择出的最优 action a_i^*的质量，但是，大量的实验表明，max-plus 已经被成功的应用在又环图的网络结构中，并且取得不错的效果(Murphy et al., 1999[17]; Crick and Pfeffer, 2003[18];Yedidia et al., 2003[19])。对于有环图，最大的问题是，当前由 i 发送出去的消息，一定时间后，又发送到 i，进而导致消息值的无限增大。对此依据(Wainwright et al., 2004)，我们使用当前 agent i 发送出去消息的平均值 $c_{ij} = \frac{1}{|\Gamma(i)|}\sum_{k\in\Gamma(i)} \mu_{ik}(a_k)$来约束消息的无限增大。仍然，在很多情况下，随着消息值的抖动，各个 agent 的最优 action a_i^* 也在不断变化，针对此问题进一步拓展，只有当 agent 收到的收益 $g_i(a_i')$ 提高时，才对其最优 action a_i^* 进行更新（anytime max-plus algorithm）。

4.3.3 Coordination Action Selection

总上所述，[anytime] max-plus algorithm 算法计算过程如 Algorithm 3 所示：

Algorithm 3 runDCOP(centralized max-plus algorithm for CG(V,E))

14： initialize $\mu_{ij} = \mu_{ji} = 0$ for $(i,j) \in E$,m=$-\infty$,fixed_point=false

15： **while** fixed_point = false and deadline to send action has not yet arrived **do**

16： // run one iteration

17： fixed_point = true

18： **for** every agent *i* **do**

19： **for** all neighbors $j = \Gamma(i)$ **do**

20： send j messages $\mu_{ij}(a_j) = max_{a_i}\{Q_{ij}(a_i, a_j) + \sum_{k\in\Gamma(i)\setminus j}\mu_{ki}(a_i)\} + c_{ij}$

21： **if** $\mu_{ij}(a_j)$ differs from previous message by a small threshold **then**

22： fixed_point = false

23： determine $g_i(a_i) = \sum_{j\in\Gamma(i)}\mu_{ji}(a_i)$ and $a_i' = argmax_{a_i}g_i(a_i)$

24： **if** use anytime extension **then**

25： **if** $g_i(a_i') > m$ **then**

26： $a_i^* = a_i'$ and $m = g_i(a_i')$

27： **else**

28： $a_i^* = a_i'$

29： set best action for agent *i* = a_i^*

30： **end for**

31： **end for**

4.4 Coordination Set Selection

Algorithm 3 中，消息的数量与系统的 CG（Coordination Graph）中，边的条数成正比。对于一个足够大网络结构来说，各个 agent 的相互依赖关系比较复杂，图中每个节点的度数可能比较大，因而消息发送的次数频繁。但是在现实环境中，每个 agent 通信的资源数量往往是有限的，并且通信的代价往往比较昂贵，因此我们设计了一种动态调整，选择出当前时刻，对各个 agent 最有益的最小协调子集（Coordination Set，$CS \subset \Gamma(i)$），以减少在 CG 中相互依赖的边的数量，进而减少每个 agent 发送 message 的数量，进而降低通信的代价。为了给每个 agent i 找到当前时刻最优的 Coordination 子集，我们定义了一种定量的 agent 之间交互的衡量方式 Potential loss in lack of coordination（PLILOC），来衡量不与邻居中某

几个 agent 进行协调而可能带来的损失。这种衡量方式基于我们定义的另一种方式：Potential expected utility(PV)来衡量 agent i 紧与选定的 CS(i)交互，而预期可得到的最大收益。

定义一：在稳定状态的 Coordination Set（CS）中，对任意 agent i, 其 CS(i)中的邻居 agent j，将无条件的配合 agent i 的行为选择 action，以最大化其局部的整体最大收益。对于初始网络中 agent i 的邻居 $k \in \boldsymbol{\Gamma(i)}\ and\ k \notin CS$（$\boldsymbol{\Gamma(i)}$ 是网络初始化时，agent i 的所有邻居组成的集合），agent i 能够根据对 agent k 行为的观察，统计出当前其选择各个 action 的概率，进一步可计算其对 agent i 选择 action a_i 收益的平均影响。这里，计算方式如下：

$$Q_i(a_i) = \sum_{k\in\Gamma(i),\ k\notin C} \sum_{a_k\in \boldsymbol{A_k}} P_k(a_k|a_i) Q_{ik}(a_i, a_k) \qquad (4\text{-}9)$$

定义二：当选定 Coordination Set = CS(i)，并且仅与 CS(i)中的 agent 进行协调时，agent i 的预期最大收益（the potential expected utility） $\mathrm{PV}(\mathrm{a}_i, \mathrm{CS}(i))$如式（4-10）所示：

$$\mathrm{PV}(\mathrm{a}_i, \mathrm{CS}(i)) = \sum_{j\in \mathrm{CS}(i)} max_{a_j} Q_{ij}(a_i, a_j) \ + \sum_{k\in\Gamma(i),\ k\notin C} \sum_{a_k} P_k(a_k|a_i) Q_{ik}(a_i, a_k) \qquad (4\text{-}10)$$

其中，$P_k(a_k|a_i)$，$k \in \Gamma(i)\ and\ k \notin \mathrm{CS}(i)$，是 agent i 在自己选择 action a_i的条件下，对 agent k 最近一段时间选择各个 action 可能性的概率统计。对 agent i，PV(i)由两部分构成：（1）由其 CS(i)中的 agent 组成，其中没一个 agent 将无条件的配合 agent i 的行为选择 action；（2）由属于初始化网络中 agent i的邻居但不属于当前 i 的 CS(i)的 agents 组成，agent i 根据对其的统计信息，估计出对自己的影响。很明显如果 $\mathrm{CS}_1(\mathrm{i}) \subseteq \mathrm{CS}_2(\mathrm{i}) \subseteq \boldsymbol{\Gamma(i)}$，则对任意 action a_i，有 $\mathrm{PV}(\mathrm{a}_i, \mathrm{CS}_1(i)) \leq \mathrm{PV}(\mathrm{a}_i, \mathrm{CS}_2(i))$。

定义三：不与 NC(i)协调而造成的预期损失（the potential loss in lack of coordination）。是当前 agent i 与所有邻居$\boldsymbol{\Gamma(i)}$ 通信协调期望所得的最大收益与紧与$\mathrm{CS(i)} = \boldsymbol{\Gamma(i)}\backslash\mathrm{NC(i)}$中的 agents 通信协调所期望最大收益之差。

$$\mathrm{PLILOC}_i(\mathrm{NC(i)}) = max_{a_i, i\in\Gamma(i)} \mathrm{PV}(\mathrm{a}_i, \Gamma(i)) - max_{a_k, k\in\Gamma(i)\backslash NC(i)} \mathrm{PV}(\mathrm{a}_k, \Gamma(i)\backslash\mathrm{NC(i)}) \qquad (4\text{-}11)$$

其中，(1)如果$\mathrm{NC}_1(\mathrm{i}) \subseteq \mathrm{NC}_2(\mathrm{i}) \subseteq \boldsymbol{\Gamma(i)}$，则对任意 action a_i,有 $\mathrm{PLILOC}_i(\mathrm{NC}_1(\mathrm{i})) \leq \mathrm{PLILOC}_i(\mathrm{NC}_2(\mathrm{i}))$；(2) $\mathrm{PLILOC}_i(\emptyset) = 0$；(3)对于所有$\mathrm{NC(i)} \subseteq \boldsymbol{\Gamma(i)}$，有 $0 \leq$

$\mathrm{PLILOC}_i(\mathrm{NC(i)}) \leq \mathrm{PLILOC}_i(\boldsymbol{\Gamma(i)})$。

任意 agent i 将通过计算$\mathrm{PLILOC}_i(\mathrm{NC(i)})$来选择出损失范围内的最优 Coordination Set 子集，进而减少 Coordination 网络中边的个数，减少 DCOP 运行时的通信次数。

Algorithm 4 通过在可能的 Coordination Set 中搜寻（子集组合问题），在系统收益与通信资源的耗费直接取个折中。整个算法过程描述如下：其中δ 代表系统允许的最大损失率，此处设置为 0.001，当$\delta = 0$，每个 agent i 将与其所有 neighbor $\boldsymbol{\Gamma(i)}$通信进行协调；当$\delta = 1$时，每个 agent i 将不与其任何 agent 进行通信协调。

Algorithm 4 computeCoordinationSet(i)

32：initialize $maxLoss = \delta * \max\{|max_{a_i}\mathrm{PV}(a_i, \boldsymbol{\Gamma(i)})|, \mathbf{PLILOC}_i(\boldsymbol{\Gamma(i)})\}$

33： find $\mathrm{C} \subset \boldsymbol{\Gamma(i)}$, such that

34： (1) $\mathbf{PLILOC}_i(\boldsymbol{\Gamma(i)} \backslash C) \leq maxLoss$

35： (2) $\mathbf{PLILOC}_i(\boldsymbol{\Gamma(i)} \backslash D) > maxLoss$, for all $\mathrm{D} \subset \Gamma(i)$ and $|\mathrm{D}| < |C|$

36： (3) $\mathbf{PLILOC}_i(\boldsymbol{\Gamma(i)} \backslash C) \leq \mathbf{PLILOC}_i(\boldsymbol{\Gamma(i)} \backslash D)$ for all $\mathrm{D} \subset \Gamma(i)$ and $|\mathrm{D}| = |C|$

37： return C

4.5 FMQ

考虑到不同 game 的 reward matrix，为保证我们的方法总能收敛到最优的纳什均衡，引入 FMQ heuristic[21]，在每一轮更新 Q-function 时，采用式（4-12）更新：

$$\mathrm{FMQ}_i(s,a) = Q_i(s,a) + freq(s,a) * r_{max}(s,a) * C \qquad (4\text{-}12)$$

其中，$r_{max}(s,a)$ 是对 agent 的每个 action，到目前出现过的最大 reward，$\mathrm{freq(s,a)}$是 agent 在 s 状态，选择 action a 而出现最大 reward $\mathrm{r}_{max}(s,a)$的频率。C 是权重参数，调整最大值对更新的影响程度。其中$\mathrm{freq(s,a)}$计算方式如（4-13）所示：

$$\mathrm{freq(s,a)} = \frac{|\{\langle s_k,a_k,r_k\rangle | s_k=s,a_k=a,r_k=r_{max}(s,a)\}|}{|\{\langle s_k,a_k,r_k\rangle | s_k=s,a_k=a\}|} \qquad (4\text{-}13)$$

第五章 实验及结果分析

在本章，通过不同的网络结构及不同的 game，我们对算法生成社会规范（Norm）的表现进行评估。随后，我们实验中涉及到的不同参数对社会规范生成的影响程度进行讨论。

5.1 算法评估

在相同网络结构及相同 game 模型下，我们对我们算法生成社会规范时的收敛轮数进行评估，以比较我们算法在生成规范速度上的优势。

5.1.1 网络结构

在这一小节，我们在不同网络结构下，比较各种算法的收敛速率

- **Grid Network**：网格结构
- **Regular Network**：规则网络
- **Random Network**：随机网络
- **Small World Network**：小世界网络
- **Scale Free Network**：无尺度网络

5.1.2 Game 参数设置

- 状态数：1（单状态 Normal Form Game）
- agent 个数：50
- action 个数：2
- payoff matrix：如图 5-1 所示

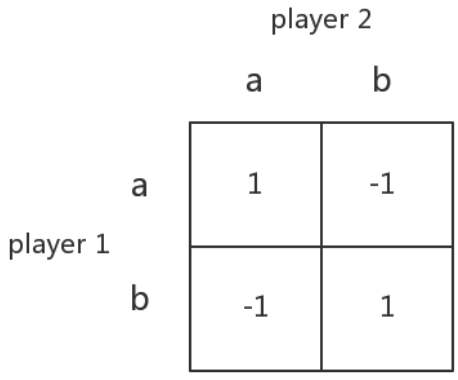


图 5-1 收益矩阵

5.1.3 算法比较

- Independent Learners (IL)

网络中每个 agent i 采用 Q-learning，只根据自己的动作a_i、收益r_i记录自己的 Q-function $Q_i(a_i)$，每次只根据自己的收益采用式 5-1 进行更新。各个 agent 根据自己的 Q-function Q_i选择自己当前的动作，以最大化自己的收益。

$$Q_i(s,a_i) = Q\ (s,a_i) + \alpha[R_i(\boldsymbol{s},\boldsymbol{a}) + \gamma max_{a_i'} Q_i(s',a_i') - Q_i(s,a_i)] \quad (5\text{-}1)$$

- Distributed Value Functions(DVF)

网络中每个 agent i 根据自己的动作a_i存储自己的 Q-function $Q_i(a_i)$。但是每一次更新都结合自己所有邻居 agents 的 Q-function 进行更新，更新方式如式 5-2 所示。其中f(i, j)表示邻居 agent j 的贡献率（这里取$1/|\Gamma(i)|$）。在单状态的问题中，做出以下调整，在每个 agent i 每一轮学习过程中，选择自己当前最优 action 时，同时考虑所有邻居 agents 的 Q-function，即 $a_i^* = argmax_{a_i \in A_i} \sum_{j\in\{i\cup\Gamma(i)\}} f(i,j)\, Q_j(s',a_i')$。

$$Q_i(s,a_i) = Q_i(s,a_i) + \alpha[R_i(\boldsymbol{s},\boldsymbol{a}) + \gamma \sum_{j\in\{i\cup\Gamma(i)\}} f(i,j)\, max_{a_j'} Q_j(s',a_j') - Q_i(s,a_i)] \quad (5\text{-}2)$$

- Our Method （loss rate $\delta = 0.1$ and 0.8）

5.1.4 实验结果

在不同网络结构下（各种网络结构平均各个节点的平均度数为 5），我们比较了上小节所提到的不同算法的收敛速率。其中，在随机网络、small world 网络结构下，各种算法都可以很快达到收敛（如图 5-3 和图 5-5 所示）。其中，我们的算法（with loss rate 0.1[color purple] 和 loss rate 0.8 [color blue]）收敛速度最快，DVF[color red]次之，IL[color green]收敛速度最慢。在 Grid 网络、Regular 网络、Scale Free 网络下，只有我们的算法 with loss rate 0.1[color purple] 和 loss rate 0.8 [color blue]）可以在很快的时间内达到收敛。不同网络结构下算法的具体收敛情况，如图 5-2~5-6 所示，因此我们的算法在生成社会规范的问题上更有效率，能够加速社会规范的生成。

图 5-2　Grid network

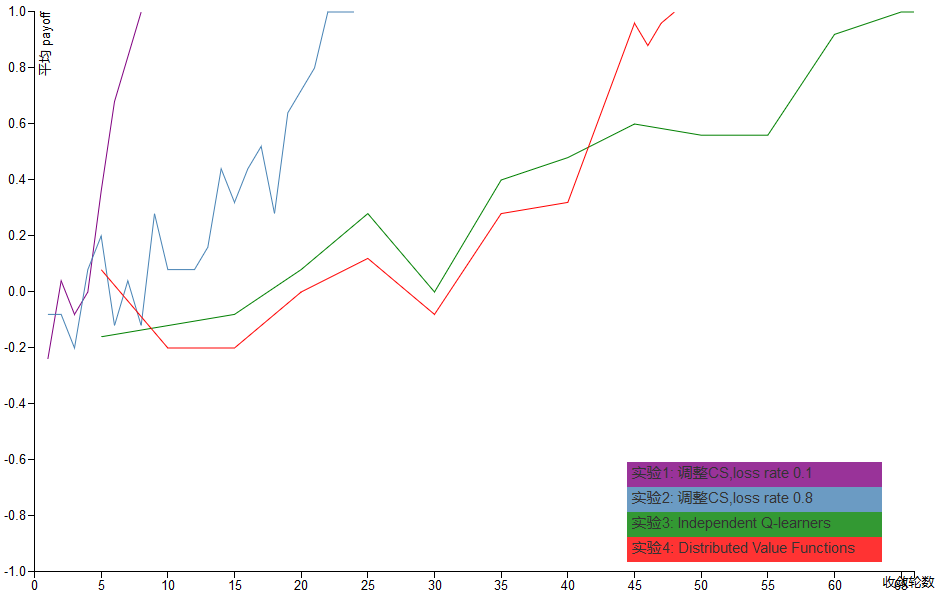


图 5-3　Radom network

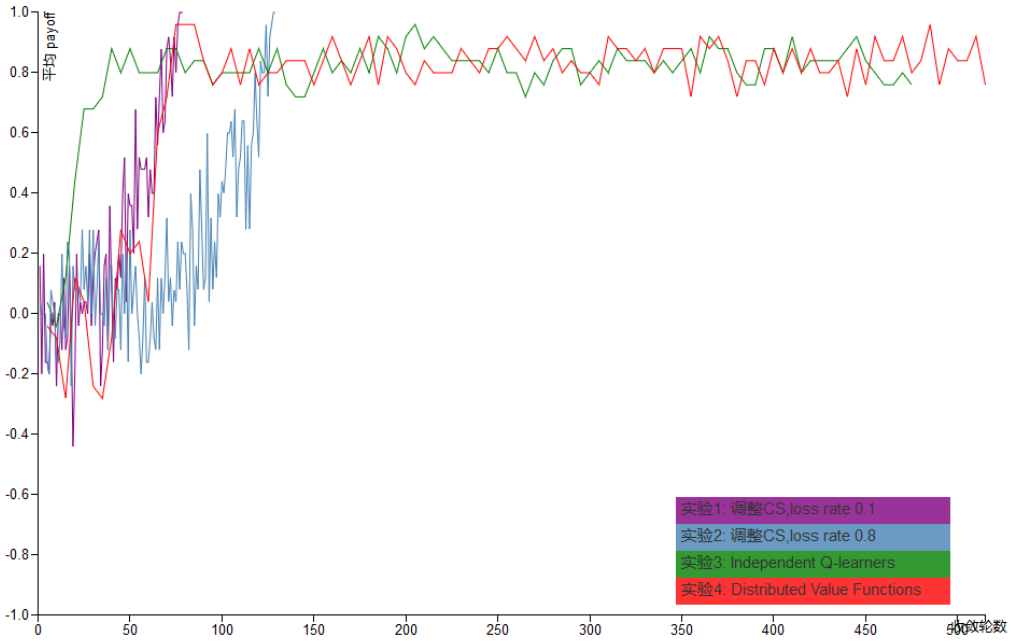


图 5-4　regular network

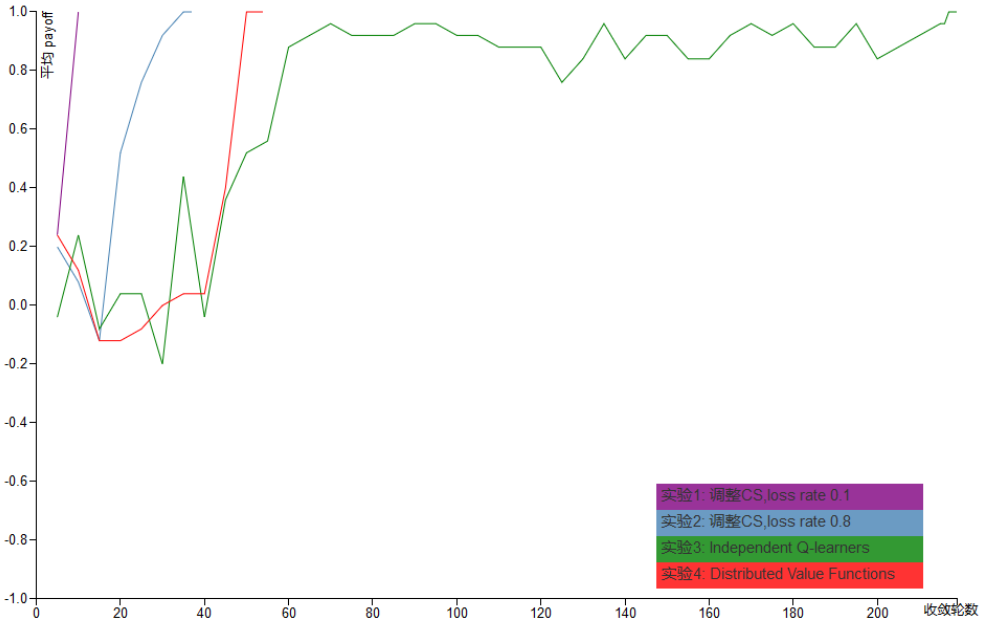


图 5-5　small-world network

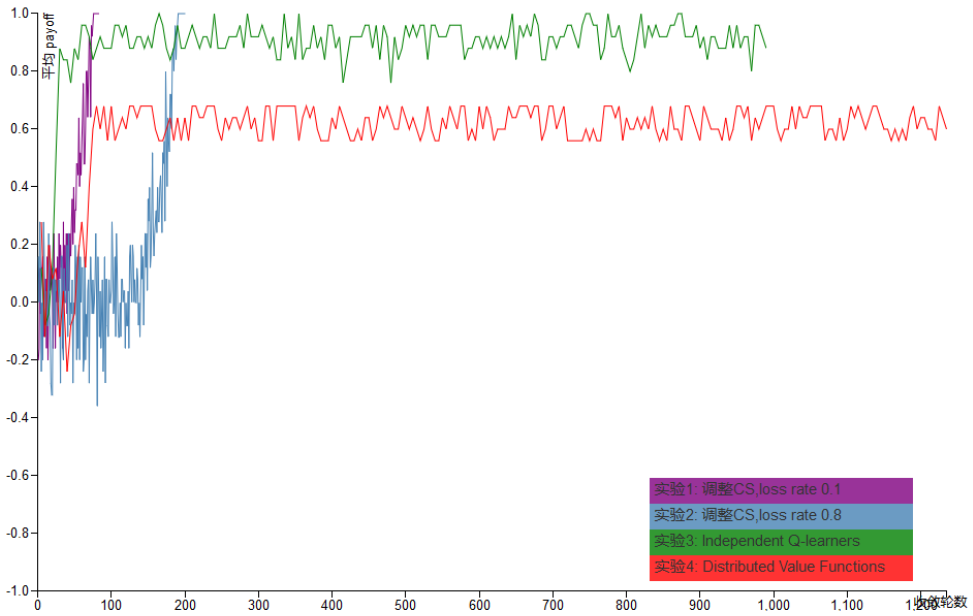


图 5-6　scale-free network

5.2　评估参数影响

在这一节，我们评估算法中的涉及到的参数对收敛速度及 DCOP 算法中各个 agent 之间通信数量的影响。

5.2.1　loss rate δ 对收敛速度及通信次数的影响

loss rate δ 的大小，反映了算法在系统收益与通信资源消耗之间的折中。当$\delta = 0$时，表示算法可容忍的系统收益损失百分比为0，则系统各个 agent 之间的通信量较大，agent 之间需要不断通信来协调自己的 action 选择，以来提高系统的收益。当$\delta = 1$时，表示系统可容忍的损失百分比为 100%，即系统各个 agent 之间无需进行通信。当$0 < \delta < 1$时，即在收益与通信之间取折中。实验中，参数设置如下：

- 状态数：1（单状态 Normal Form Game）
- agent 个数：100
- action 个数：10
- payoff matrix：

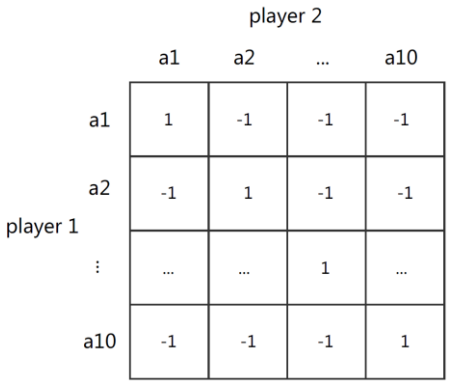


图 5-7 收益矩阵

- 网络结构：small world
- 网络中各节点平均度数（agent 平均邻居个数）：5

实验结果，如下图所示。图 5-8 展示了不同 loss rate δ对收敛速度的影响，比较发现，agents 的收敛速度会随着 loss rate δ的增大而减慢，其中 loss rate $\delta < 0.7$ 时，对 agents 收敛速度影响不明显。当 loss rate $\delta > 0.7$ 时，则算法不能再很快时间范围内达到收敛状态。图 5-9 展示了不同 loss rate δ 对运行 DCOP 时 agent 间通信次数的影响。实验结果表明，随着 loss rate δ 的增大且 loss rate $\delta < 0.7$ 时，满足在保证 agents 很快达到收敛状态的情况下，减少 agent 之间的通信次数，进而节约了系统的通信资源，满足了在通信资源受限的条件下，实现社会规范的快速涌现。但当 loss rate $\delta > 0.7$时，由于系统不能再很快的时间段内达到收敛状态，因此 agents 之间会不断进行通信，直到达到收敛状态，因此通信次数又会有所增大。因此，针对不同系统的通信资源数量，可以通过条件 loss rate δ 的大小，来最大限度的满足系统的要求，以提高系统的效益。

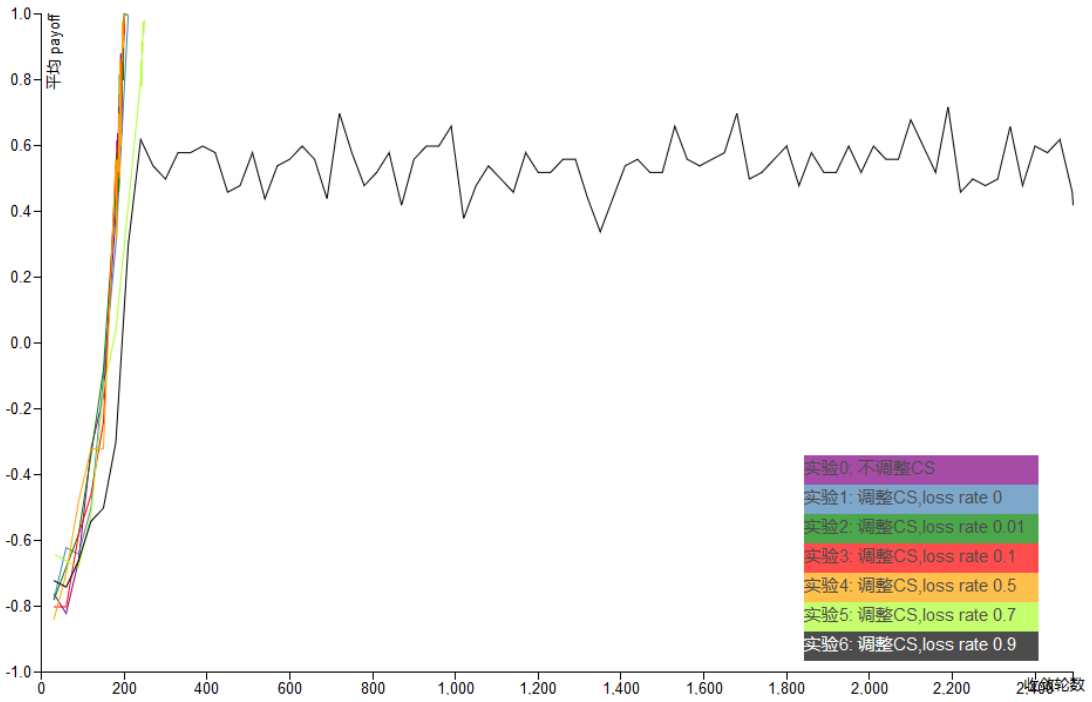


图 5-8　loss rate δ 对收敛速度的影响

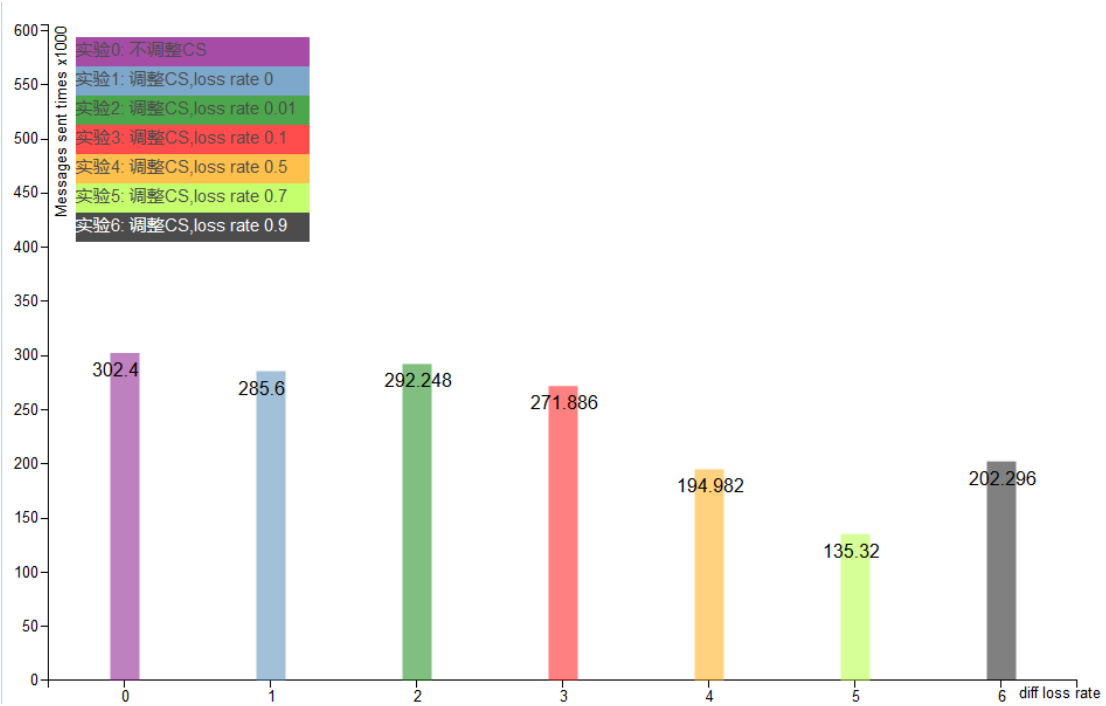


图 5-9　loss rate δ 对通信次数的影响

5.2.2　网络结构中，agent 数量的影响

我们在 small world 网络结构下，在 100~1000 之间，对 agent 的数量做了多次采样，如图 5-10 所示，是网络中有 1000 个 agents 的情况。实验结果表明（如

图 5-11 所示），在即便有 1000 个 agents 的情况下（small-world network，1000-agents，10-actions，loss rate 0.1），我们的方法在 200 轮左右的情况下，已经达到了收敛状态。所以我们的方法能够应对 agents 规模足够大的情况。图 5-12 展示了，在 small-world 网络、每个 agent 10 个 action 的条件下，agents 规模分别在 100、200、500、1000 时的收敛情况。

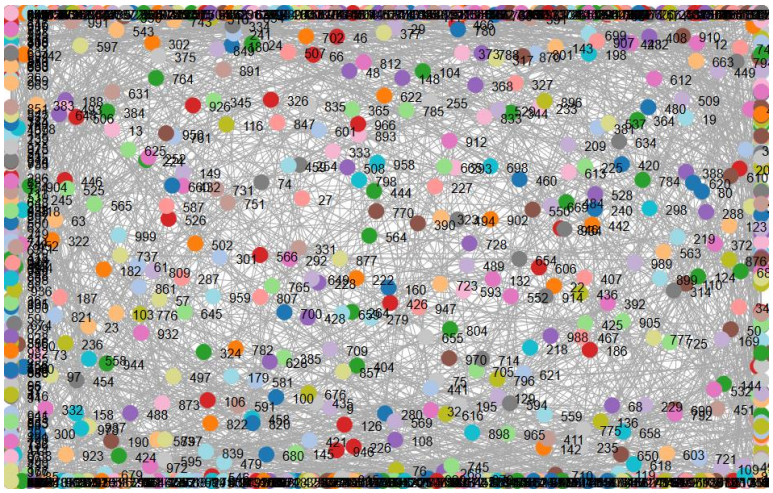


图 5-10　1000 个 agent 组成的小世界网络

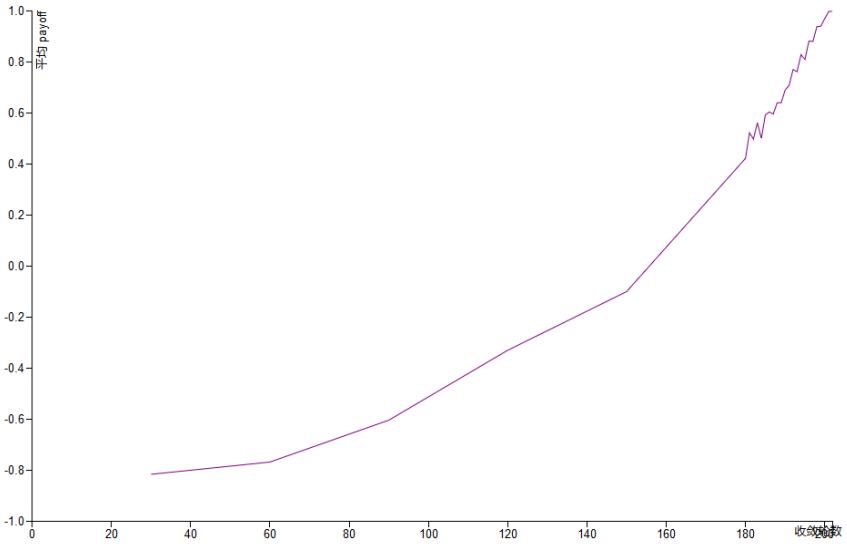


图 5-11　1000 个 agents 收敛轮数（small world）

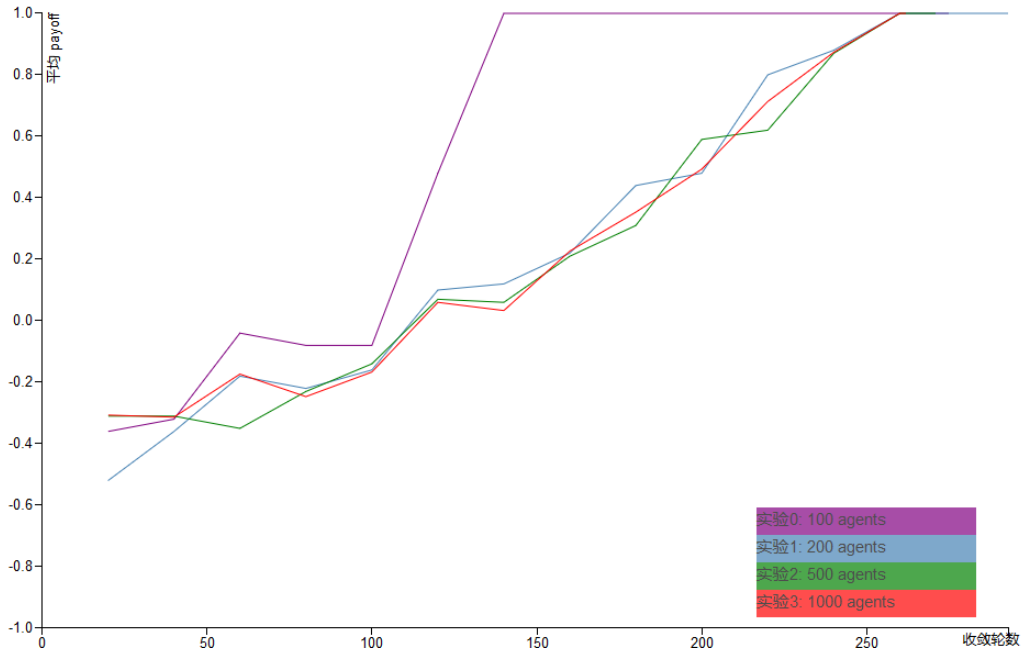


图 5-12　agents size 的影响

5.2.3　Reward-matrix 的影响

在不同的 reward matrix 中，如图 5-13，针对不同的收益矩阵，我们的方法都能收敛到最优的纳什均衡。

player 2

player 1	a1	a2	a3	a4
a1	1	-1	-1	-1
a2	-1	2	-1	-1
a3	-1	-1	1	-1
a4	-1	-1	-1	1

player 2

player 1	a1	a2	a3	a4
a1	6	-1	-1	-1
a2	-1	8	-1	-1
a3	-1	-1	9	-1
a4	-1	-1	-1	8

图 5-13 收益矩阵的影响

5.2.4　随机因素的影响

随机 AI 帮助人类提高决策效率[22]，我们在 DCOP 算法中，引入随机通信的因素，产生 0~1 直间的随机数ε，当$\varepsilon < 0.1$时，发送消息，否则不发送。实验环境：100 个 agent，10 actions，payoff matrix 如图 5-7 所示。实验结果如下图 5-14 和 5-15 所示，实验表明，在引入随机通信的条件下，对收敛速率没有太大的影响，但是大大减少了通信次数，从而极大节约了通信资源。

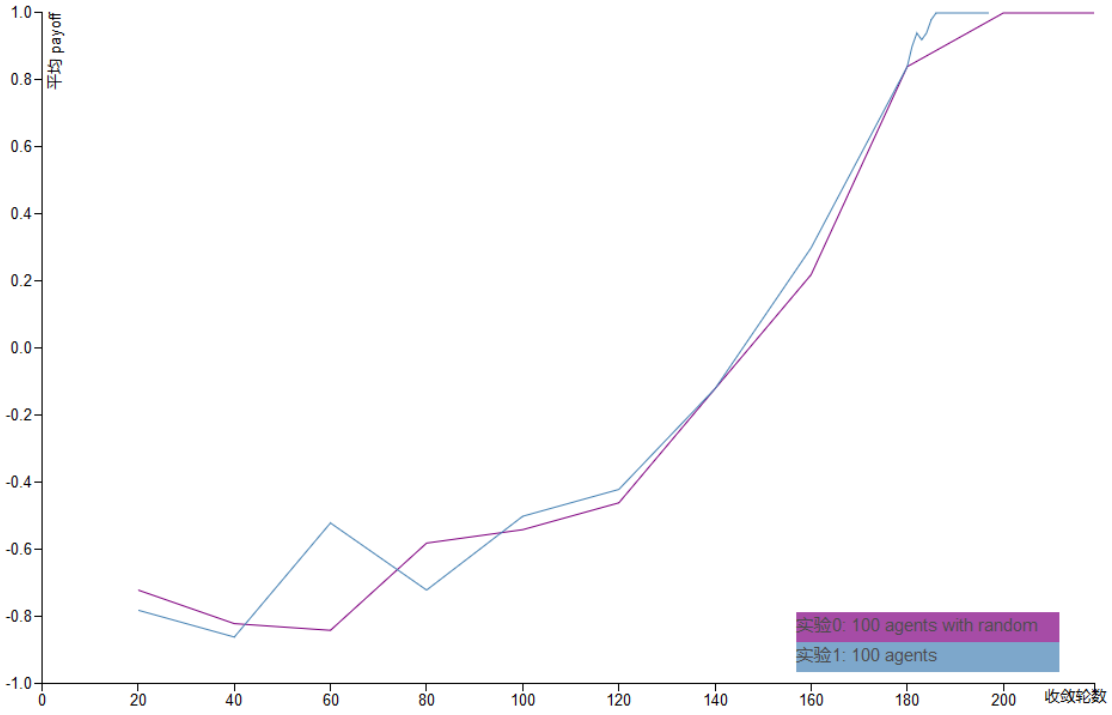


图 5-14　随机通信对收益及收敛速度的影响

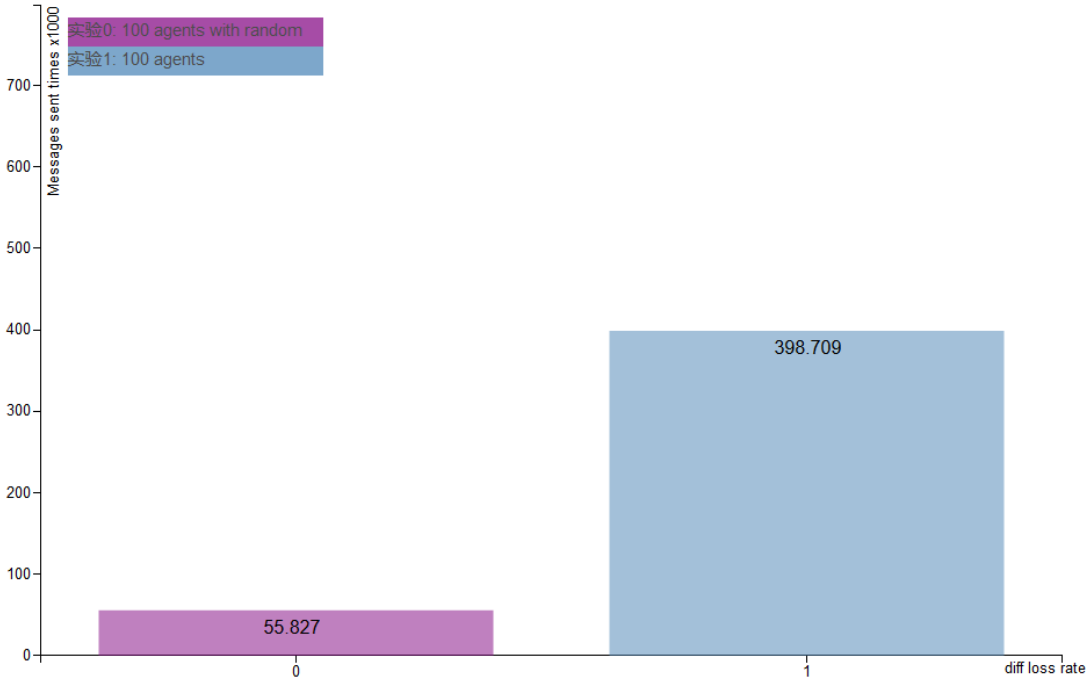


图 5-15　随机通信对通信次数的影响

第六章　总结与展望

6.1 总结

本文针对现实中不同的多智能体系统中对通信资源的不同约束情况，设计了一种能够根据通信资源的数量，动态调整算法中的参数，以满足既定精度的情况下，最大限度的减少系统通信资源的消耗。我们的方法，能够在各个 agent 学习的过程中，根据 agent 当前的状态，动态调整 agent 的 Coordination Set 大小，以减少 agent 之间的动作选择的相互依赖，将整个网络，拆分成多个小网络，从而提高系统的性能，并在很大程度上，减少了各个 agent 之间的通信次数，提高了系统整体的表现能力，进一步，我们的方法可以应用到 agents 数量足够多的大网络环境中（1000 agents）。实验结果证明，我们的方法，在不同的网络结构中，agents 都能够很快的达到收敛状态，即我们的方法能够加快社会规范（Norm）的生成。

6.2 展望

实验中，在网络中，加入一些随机的因素，不仅不会制约 agents 的收敛速率，反而在随机因素波动范围合适的情况下，能够加快社会规范的形成[22]。后续，我们将针对通信资源受限的情况下，探究随机因素对系统收敛情况的影响，以最大限度、最大效率的利用当前数量有限的通信资源，提高合作式多智能体系统的整体收益。

参考文献

[1] K. Sycara. Multiagent systems. AI Magazine, 19(2):79–92, 1998.

[2] G. Weiss, editor. Multiagent systems: A modern approach to distributed artificial intelligence. MIT Press, 1999.

[3] E. H. Durfee. Scaling up agent coordination strategies. IEEE Computer, 34(7):39–46, July 2001.

[4] N. Vlassis. A concise introduction to multiagent systems and distributed AI. Informatics Institute,University of Amsterdam, September 2003.

[5] Sandip Sen and St´ephane Airiau, 'Emergence of norms through social learning.', in International Joint Conference on Artificial Intelligence, volume 1507, p. 1512, (2007).

[6] St´ephane Airiau, Sandip Sen, and Daniel Villatoro, 'Emergence of conventions through social learning', Autonomous Agents and Multi-Agent Systems, 28(5), 779–804, (2014).

[7] Partha Mukherjee, Sandip Sen, and St´ephane Airiau, 'Norm emergence under constrained interactions in diverse societies', in Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2, pp. 779–786. International Foundation for Autonomous Agents and Multiagent Systems, (2008).

[8] Onkur Sen and Sandip Sen, 'Effects of social network topology and options on norm emergence', in Coordination, Organizations, Institutions and Norms in Agent Systems V, 211–222, Springer, (2010).

[9] Daniel Villatoro, Sandip Sen, and Jordi Sabater-Mir, 'Topology and memory effect on convention emergence', in Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 02, pp. 233–240. IEEE Computer Society, (2009).

[10] Chao Yu, Hongtao Lv, Fenghui Ren, Honglin Bao, and Jianye Hao, 'Hierarchical learning for emergence of social norms in networked multiagent systems', in AI 2015: Advances in Artificial Intelligence, 630–643, Springer, (2015).

[11] C. Guestrin, M. G. Lagoudakis, and R. Parr. Coordinated reinforcement learning. In ICML '02: Proceedings of the Nineteenth International Conference on Machine Learning, pages 227–234, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[12] J. R. Kok and N. Vlassis. Collaborative multiagent reinforcement learning by payoff propagation. Journal of Machine Learning Research, 7:1789–1828, 2006.

[13] C. Zhang and V. R. Lesser. Coordinated multi-agent reinforcement learning in networked distributed pomdps. In W. Burgard and D. Roth, editors, AAAI. AAAI Press, 2011.

[14] Guestrin, D. Koller, and R. Parr. Multiagent planning with factored MDPs. In Advances in Neural Information Processing Systems (NIPS) 14. The MIT Press, 2002a.

[15] J. Pearl. Probabilistic reasoning in intelligent systems. Morgan Kaufman, San Mateo, 1988.

[16] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. Tree consistency and bounds on the performance of the max-product algorithm and its generalizations. Statistics and Computing, 14: 143–166, April 2004.

[17] K. Murphy, Y. Weiss, and M. Jordan. Loopy belief propagation for approximate inference: An empirical study. In Proceedings of Uncertainty in Artificial Intelligence (UAI), Stockholm, Sweden, 1999.

[18] C. Crick and A. Pfeffer. Loopy belief propagation as a basis for communication in sensor networks. In Proceedings of Uncertainty in Artificial Intelligence (UAI), 2003.

[19] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. In Exploring Artificial Intelligence in the New Millennium, chapter 8, pages 239–269. Morgan Kaufmann Publishers Inc., January 2003.

[20] Tianpei Yang ,and Zhaopeng Meng , Jianye Hao Accelerating Norm Emergence Through Hierarchical Heuristic Learning ECAI 2016 G.A. Kaminka et al. (Eds.)

[21] Spiros Kapetanakis and Daniel Kudenko, 'Reinforcement learning of coordination in heterogeneous cooperative multi-agent systems', in Adaptive Agents and Multi-Agent Systems II, 119–131, Springer, (2005)

[22] Hirokazu Shirado & Nicholas A. Christakis, 'Locally noisy autonomous agents improve global human coordination in network experiments', Nature 545, pages 370–374, 18 May 2017

外文文献

Hirokazu Shirado & Nicholas A. Christakis, ‘Locally noisy autonomous agents improve global human coordination in network experiments’, Nature 545, pages 370–374, 18 May 2017.

中文译文

随机 AI 帮助人类提高决策效率

群体协调中面临着一个次最优的问题，而理论认为一些随机性有助于实现全局最优化。在这里，我们进行了涉及网络结构的着色实验，其中很多人与自主的软件智能体（机器人）进行交互。4000 名人类参与者与 230 个机器人组成包含 20 个节点的网络，在每个网络中随机添加 3 个机器人。机器人被编程为具有不同级别的行为随机性和分配到不同的地理位置。实验表明，以小量随机噪声行动的机器人分布在中心位置有意义地改善了人类群体的集体表现，将问题的解决时间的中位数提高了 55.6%。特别是当协调问题比较困难的时候。行为随机性的作用不仅仅让通过使机器人连接起来的人类的任务变得更容易，而且还通过影响人与人之间的游戏，从而在这些不均匀的系统中的全局协调中创造更多的级联效益。

集体行动和大规模合作是重大挑战。大多数合作工作侧重于社会困境，即让人们愿意为更大的好处而做出牺牲。 然而，即使可以解决这个困境，仍然存在另一个重大问题：协调[4-6]。 群体中实现最佳集体行动的困难不仅可能来自个人之间，个人与群体之间的利益冲突，也可能是由于个人无法在全局的角度有效地协调其各自的行动。 即使所有个人在局部的互动中都表现得很好，也可能不会导致整个群体出现最佳结果。

针对协调问题，以前的理论工作提出了一个令人惊讶，甚至矛盾的解决方案：增加“噪音”[13-15]。 噪声通常被定义为无意义的信息，并且经常被视为有问题的[16]。 然而，在优化方面，噪声可以帮助系统达到全局最优。 例如，突变在进化中起着至关重要的作用[17]; 错误可以方便搜索信息[18]; 鱼类随机组群可以提高生存率[19]; 合作可能受益于偏差行为[7-9,20]。

在这里，我们评估噪声在解决人类群体协调问题中的益处[21,22]。 由于人类互动融入于社交网络中，我们也考虑到网络位置对噪声潜在有益影响的影响[23]。 我们首先描述了在经典色着色游戏中互动的人网络的集体行动动态。然后，我们使用自主软件代理（机器人）测试噪声对集体性能的影响，调整机器人的噪声和地理位置。通过将机器人添加到实验社会网络中，我们因此探索涉及真实人类和自主代理的异构系统的性能，同时也为全局协调本身的问题展示了一种可能的实际解决方案。

我们在线招募了 4,000 个独立的参与人员，并在 230 个课程中随机分配到 11 个条件中的 1 个（见补充信息）。 通过一个优先添加模型，将每个参与人员在 20 个节点的网络中分配一个位置。 通过将新节点（每个节点有两条边）附加到现有节点，为每个会话重新创建网络结构; 并将测试者随机放入所得网络。 集体

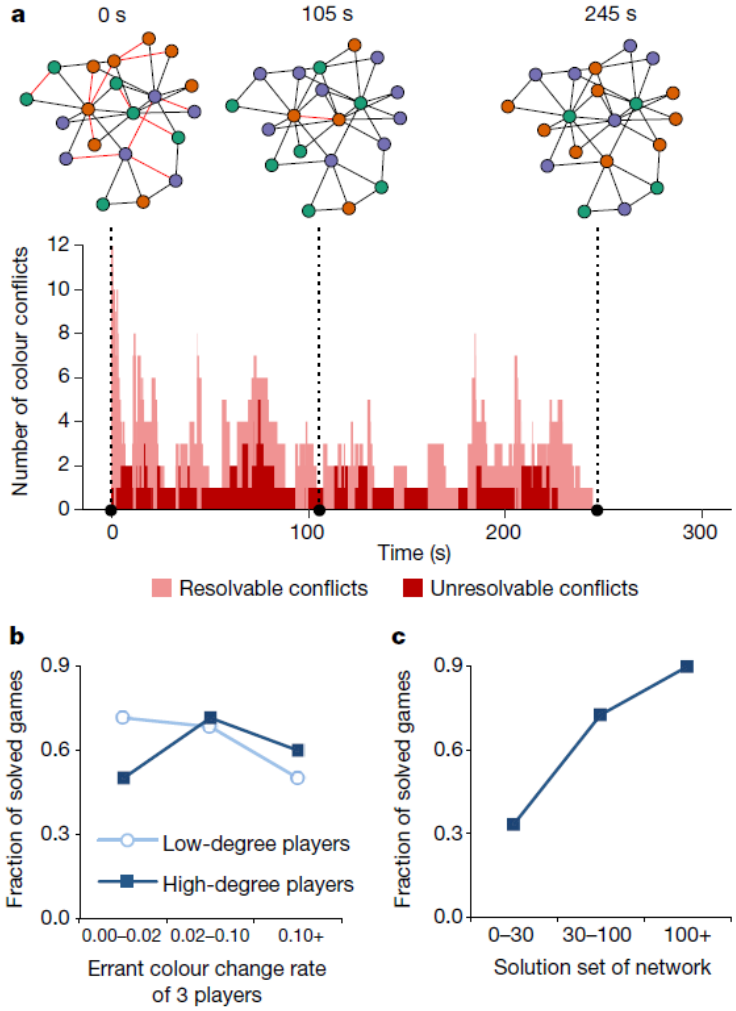
目标是使每个节点具有与其所有相邻节点不同的颜色[10]。 这种配色游戏通过协调的子优化成功地捕获了系统故障的问题; 也就是说，当每个人尝试达到对于该局部（个体）最佳的解决方案时，这可能不是整个组的最佳（图 1a）。

在这些一次一次的会话中，每个参与人都可以随时从三种选择（绿色，橙色和紫色）中选择一种颜色。可用的颜色数量是将整个网络着色为无冲突所必需的最小值，这被称为色数; 我们实验中的所有网络对于着色问题都是全局可解的。然而，虽然所有的网络都允许参与人达到集体目标，网络可能（偶然地）在解决方案数量上有所不同（也就是说，网络范围从 6 到 13,可能存在 824 种的着色方式，称为彩色多项式 ;见补充资料。

除了自己的颜色之外，参与者只能看到与他们直接相连的邻居的颜色。 因此，虽然一个参与人从他或她自己的角度问题可能已经解决，但游戏可能会继续，因为网络在图的其他区域仍然存在冲突。在最优化问题方上，游戏的耗费函数表示为冲突次数的总和。 如以往的工作[10]，参与者根据网络中的所有冲突解决所花费的时间长度而得到回报，并且他们必须在 5 分钟内完成任务（详见附件）。

在这个基本设置中，我们将三个机器人引入网络，以换取相同数量的人（没有机器人被放在控制会话中;见补充表 1）。参与人没有被通知在游戏中有机器人参与。 我们按如下方式控制机器人的噪音干扰：在“零噪声”条件下，机器人的行为表现为一种简单的贪心策略：当机器人有机会最小化与邻居的颜色冲突时，它选择了这种颜色; 否则，它保持当前的颜色。 在其他两个条件下，机器人大多数时候都采用相同的贪婪策略，但是随机选择三种允许选项的颜色，无论其局部情况如何，随机的概率为 10％（“小噪声”）或 30％（“大噪音”）。 在所有条件下，机器人每 1.5 秒作出决定，这是典型的人类反应时间（扩展数据图 1）。

除了机器人的噪声，我们也按如下方式控制他们在网络中的位置：在“中心”，机器人被分配到具有最多邻居数（最高网络度）的三个位置。 同样，在“周边”，机器人被分配到度最低的三个位置。在“随机”条件下，机器人被随机分配到其中的网络位置。 在任何情况下，机器人都可以偶然地相互联系。



如前所述，机器人只使用他们的本地信息。 为了评估这种机器人行为的影响，与对整个网络结构及其全局的解空间要求更高的案例相比，我们还进行了“固定颜色”条件的实验。 在这种额外的条件下，我们评估了每个网络的所有颜色组合，导致没有冲突，然后根据这些组合之一（随机选择）分配三个节点的初始颜色。 也就是说，在游戏过程中，三个节点不受与其邻居协调的机器人的控制，而是这些节点简单地保持在它们的初始颜色，其已按照与全局问题解决方案一致的方式被着色。 我们仅在固定节点处于中心位置的情况下做了这种对比测试。

总之，我们评估了 11 个条件：1 个控制条件不涉及任何机器人; 9 种机器人的噪声和位置的组合（3 种行为随机性（0%，10%和 30%）与 3 种类型的位置（随机，中心和外围）交叉，1 个固定条件与 3 个固定颜色的节点。 我们为每个条件进行了 30 次控制条件和 20 次会议，总共 230 次和 4000 次。

总之，我们评估了 11 个条件：1 个控制条件不涉及任何机器人; 9 种机器人的噪声和位置的组合（3 种行为随机性（0%，10%和 30%）与 3 种类型的位置（随机，中心和外围）交叉，1 个最终条件与 3 个固定颜色节点。 我们为每个实验条件进行了 30 次实验，为每个控制条件进行了 20 次实验，总共 230 次实验和

4000 参与人员。

对于仅涉及人类受试者的游戏中，30 个中的 20 个实验，在 5 分钟内找到了所在网络的最佳着色方案（中位数= 232.4 秒;四分位数范围（IQR）143.7-300.0）。虽然这些参与者在努力消除一切冲突，但他们往往发现自己无法通过个别地减少局部的冲突来达到集体目标。例如，在图 10 中为 105 秒。 1a（或补充视频 1），每个参与者都选择了自己邻居中最出现最少的颜色之一;也就是说，没有一个人可以改变可以让他们的着色状态变得更好。然而，邻居之间的冲突仍然存在。这种玩家陷入本地不可解决的冲突的状态被认为是游戏成本 cost 函数的局部最小值（与通过本地动作可解决的可解决的冲突相反）。玩家需要从冲突最小化的规范中获得适度的偏差，以克服局部最小化，并达到全局解决方案（例如，图 1a，245s）。

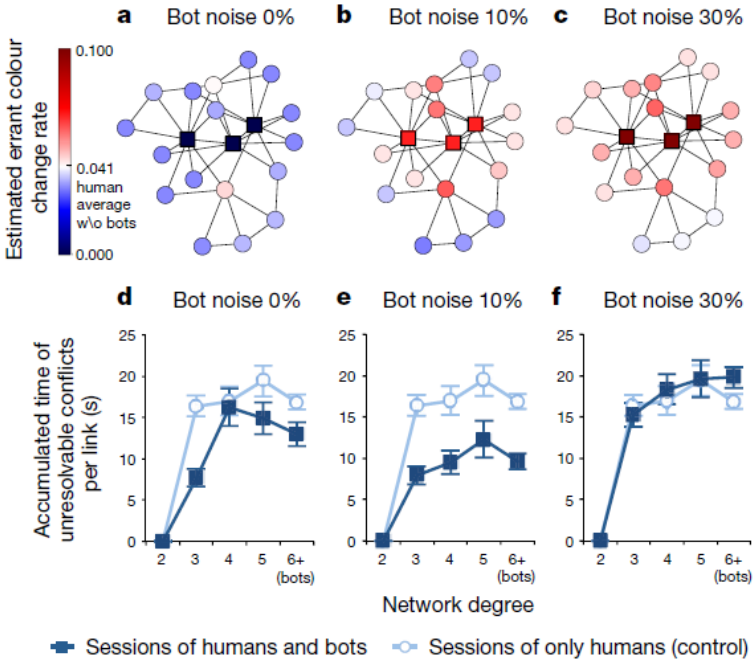
通过分析仅涉及人类参与者的实验，可以看出，有些玩家偶尔会选择不当的颜色，暂时增加冲突，游戏更有可能得到解决。 此外，这种行为偏差的影响与玩家的地理位置的差异而有所不同（图 1b）。 另外，明显地，一些网络可以本身更容易解决（即彩色多项式可能更高）（图 1c）。

为了展示机器人如何改善人类群体的表现， 图 2 显示了涉及到的九种机器人参与的实验过程曲线。在将每个实验组与对照组进行成对比较之前，我们对所有存活曲线相同的无效假设进行了对数秩检验。 该假设被拒绝（P＝0.024），表明存活曲线中至少有两个不同。 在中心地带有 10％噪音的机器人会话最有可能在分配的 5 分钟内解决（20 次会议中有 17 次，或 85％，而 30 次对照会话中的 20 次，或 67％，与人类 单独）; 此外，解决方案比仅涉及人类的中位数（中位数= 103.1 秒（IQR 49.5-170.1））相对于 232.4 秒（IQR 143.7-300.0），快于 129.3 秒（即 55.6％），这明显更好（P＝0.015，对数秩检验）。

然后，我们使用 Cox 比例风险模型检验了各种机器人处理的有效性差异，同时进一步控制了网络的内在可解性。 10％的行为随机性，中心位置和彩色多项式的对数都对完成时间有显着的正面影响（P <0.05; n = 180 个机器人实验过程;见补充信息）。我们还评估了解决方案空间复杂度的另一个度量（即，具有线性概率的平均收敛步骤），并获得了类似的结果（扩展数据图 2 和补充表 4）。具有完全相互作用的统计模型表明，机器人只有当它们具有 10％随机性并且被放置在网络中的中心位置时才影响问题的解决时间（图 3a）;此外，当网络提供许多解决方案时，机器人的有益影响减小，如三维交互所示（图 3b）。总之，当网络在全局范围内难以解决时，机器人显得特别有用。

我们发现，10％的机器人的影响与在已知的配置中具有固定（恒定）颜色的三个节点的影响相当，以便与全局解决方案兼容。 在 10％-noise 机器人和具有

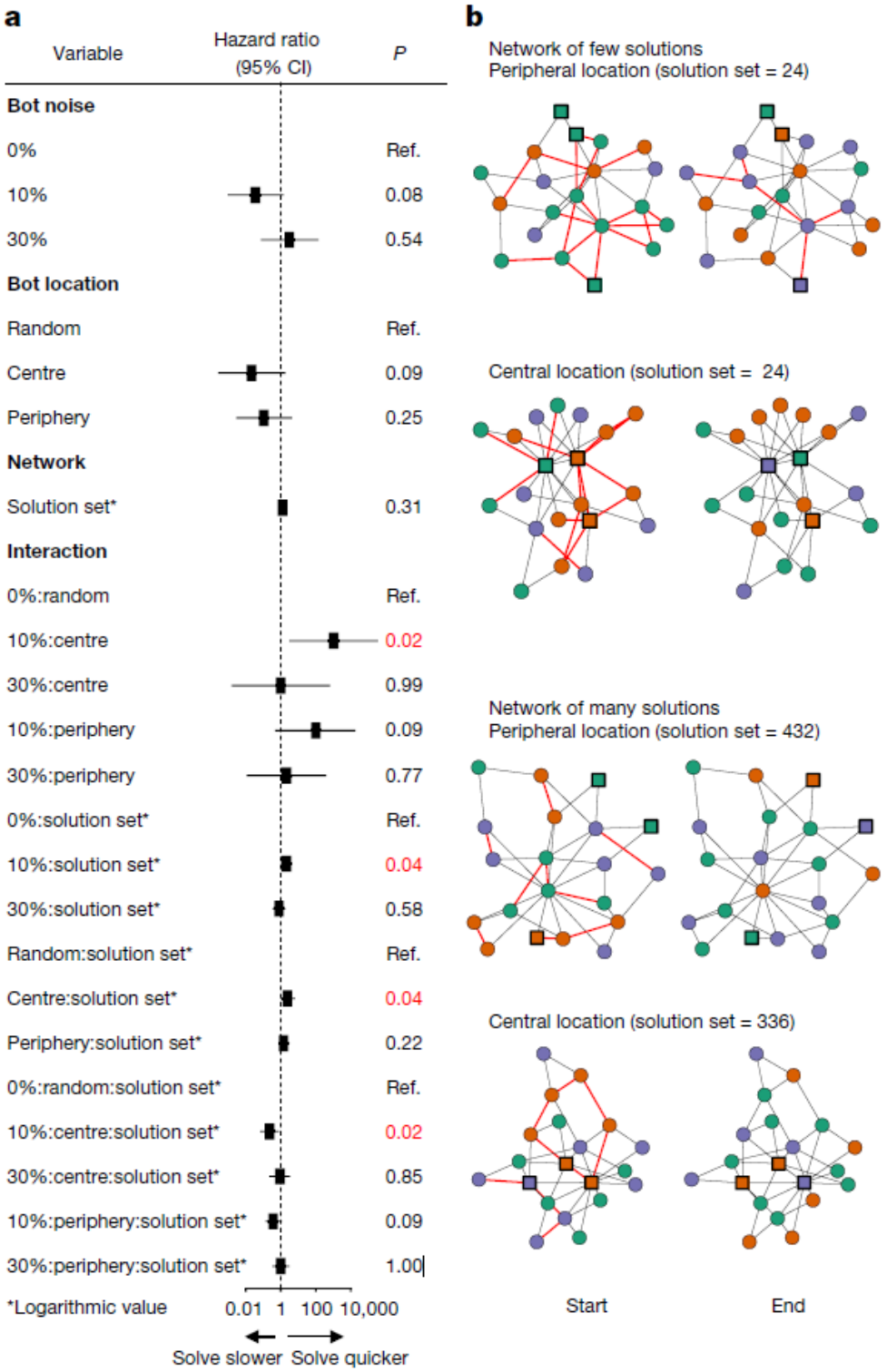
固定颜色的会话之间没有显着差异（P＝0.675，对数秩检验）。 因此，仅基于局部决策的机器人的干预与预先计算的解决方案（需要提前（在典型的情况下）（不切实际地）对整个网络结构及问题的解空间有充分的了解）同样有效。



机器人似乎通过改变整个系统中的人类参与者的色彩冲突行为（扩展数据图 3）部分地改善了集体表现。 当处于高度节点时，具有 0％行为随机性的机器人减少了冲突的数量，但是它们增加了无法解决的冲突的持续时间; 具有 30％随机性的机器人减少了无法解决的冲突的持续时间，但总体冲突增加; 与控制实验相比，只有具有 10％随机性的机器人可以减少冲突的数量和不可解决的冲突的持续时间。 相反，当放置在低度节点时，机器人不太可能影响人类的整个网络，而不管它们的随机噪声的大小。

当机器人被置于度比较高的位置时，他们的行为随机性不仅能够帮助解决自己的冲突，而且还可以推动邻近的人改变行为，从而进一步促进全局的解决方案。具有 0％行为随机性的机器人降低了其他人类玩家的随机性（图 4a），这使得人类玩家，特别是中等度数的玩家被困在不可解决的冲突中（图 4d）。具有 30％行为随机性的机器人使整个网络不稳定，包括在自己的行动中显示更多噪音的低度（图的度）玩家（图 4c）;因此，30％随机性的机器人实验与那些没有机器人的人的实验有一样的效果（图 4f）。具有 10％行为随机性的机器人增加了中央球员的随机性，但是降低了外围球员的随机性（图 4b）;因此，通过其行为随机性的影响，10％的机器人不仅减少了不可解决的冲突，而且还包括整个网络的不可解决的冲突，包括与机器人无关的人类对象之间的联系（图 4e）。事实上，这些结果，在实际上对于对手来说，机器人变得越来越干扰的情况下，依然可以获得（图

4）。



另外，在进一步的实验中，涉及另外的 340 个参与人员和一组 n = 20 个图中，我们发现即使玩家知道他们与机器人进行互动，也可以获得对组合和学习的这些有益效果（见补充信息）。 解决时间在统计学上无法区分（扩展数据图 5），并且对整个系统中的玩家的影响也是类似的（扩展数据图 6）。

将具有简单策略的自主机器人添加到社会系统中可能使人类群体更容易实

现复杂任务全局范围内的最佳化。在这里，设置是一个全局的协调游戏，但其他设置可能包括合作，共享或导航[5,12,25]。 然而，任何这样的机器人，如果它们具有某些属性，包括噪声或特定的测地线位置，可能也会有帮助。实际上，像其他情况[13,14,17,18,20]一样，从组的角度来看，一些噪音可能是好的。 而且，有些只具备本地的信息的噪音机器人，同样可以像提前知道全局信息的机器人一样，提高系统整体的表现能力。

我们发现，这些微小的噪声机器人不仅可以使他们连接起来的人类的任务更容易，而且还可以通过受影响的人与组中的其他人进行交互而影响到网络中的其他人类参与者，从而创造出一系列的效益。即使人们知道他们正在与机器人互动，也会产生这种效果。在这个意义上，即使是简单的人工智能（AI）代理也可以发挥教学功能，改变人类对手的策略，修改人与人的交互，而不仅仅是影响人机交互。 更一般来说，我们的工作说明了组合的异质群体的表现，既不仅仅由人类组成，也不仅仅由机器人来协调他们的行为。未来的工作可以探索更加现实或复杂的交互行为，例如在人类组织内工作的军事或商业机器人，或处在人类驾驶环境中的自动驾驶汽车。

虽然实验室实验提供了强大的因果推理，但它们必须牺牲一些现实性和广度。在先前的理论指导下，我们选择仅关注机器人贡献（噪声和放置位置）的两个方面及其对一个主要结果的影响（标准游戏中的全局协调的成功[10]）。 我们还必须做出其他设计选择，包括使用限于 20 人的无规模网络（如果游戏是易于使用的话）。 但社会交互中的其他特征可能会影响群体协调解决问题的能力，如群体规模，网络拓扑[10]和机器人部分; 网络是动态的还是静态的[26,27]; 或社会机构（例如警务，制裁或规范）是否存在。 这些要素是未来工作的重要方向。

在人类网络中为战略位置添加适度噪声的机器人可能有助于解决各种问题，特别是在特定协调问题困难的时候。例如，狭隘的工作人员可能会努力提高自己的生产力，但实际上这可能会降低整体公司业绩。可以通过在协同工作的组中添加一些机器人或噪音来促进科学中的人群采购（例如解决量子问题[28]或其他类型的从蛋白质折叠[29]到考古或天文图像的评估的“公民科学”）。此外，我们的工作加强了简单和复杂的 AI 都可能有用的想法。例如，简单的机器人可能有助于在线减少种族主义言论[30]。我们在这里探索的简单 AI 中决策的简单性和透明度也可能使人类可以理解，从而引发有效的长期关系[11]。简单的自主性代理人在混合到复杂的社会系统中时，可能会提供实质的优势，并且可以帮助人类团体来帮助自己。

致　谢

首先感谢天津大学这四年来带给我的关怀和成长，感谢天津大学软件学院这四年来的栽培。感谢四年中那些认真负责，严谨教学，给我关心和指导的老师们。其中特别感谢郝建业老师，在学业上的悉心指导与严格要求，生活上的热心关怀与帮助。感谢郝老师，让我在迷茫的时候，给我关怀，让我找到了未来研究生工作的方向和目标。郝老师的认真、勤奋、聪颖与活力无时无刻不让我深深的折服，激励我在学习的道路上，不断前进。

其次感谢父母、感谢家人，感谢他们大学这四年来的支持与无微不至的关怀，感谢亲情与爱情。感谢那些四年来在学习和生活中帮助我的同学们，朋友们，感谢大学四年的友谊！

最后，感谢岁月，感谢自己在不断地成熟，不断地成长。

希望自己在未来的学习和生活中，更加沉稳，更加心平气和，能够不忘初心，坚持自我，保持活力与动力，一步一个脚印。