交互式动态影响图研究及其最优 K 模型解法

潘颖慧[1)] 曾一锋[2)]

[1)](江西财经大学信息管理学院 南昌 330013)

[2)](厦门大学自动化系 福建 厦门 361005)

摘 要 不确定性多智能体序贯决策是人工智能研究领域一个重要的研究问题,主要求解智能体如何在与其他智能体的交互中优化本身的决策.特别在部分可观测的随机博弈设置下,智能体不能探测到真实的外部环境状态,必须依靠所接收的观察来推断可能的状态;同时,智能体的动作也具有相当的随机性,直接影响到其他智能体的决策.智能体的交互主要通过对共同环境状态的影响决定它们各自决策的报酬.因此,如何对多智能体之间的交互进行建模是求解该问题的核心任务.目前大部分的研究主要通过对整个智能体系统进行建模,采取集中规划、分散控制的求解机制:首先,统一计算所有智能体的联合决策;然后,各个智能体执行分配得到的局部决策.该求解技术往往要求所有的智能体必须对全局环境有一个共同的知识假设,因此该研究工作一般只适用于合作型的多智能体系统.相比之下,交互式动态影响图是从个体决策者的角度研究不确定性多智能体序贯决策问题的一种普遍适用的建模方法,克服了传统的博弈论方法求解多智能体决策问题的局限性.求解交互式动态影响图模型的主要困难在于复杂的智能体相互建模过程.特别是在竞争的环境下,由于智能体缺少相互交流的机会,也不能预知其他智能体的真实模型,必须通过预测和推理其他智能体的行为来决定本身的动作.主要求解思路是首先假设其他智能体的可能模型,然后通过求解这些可能的模型来预测智能体的行为.由于其他智能体的备选模型往往有很多,而且随着决策时间的推移,模型的不确定性增强,导致可能的模型呈指数增长,这给求解交互式动态影响图带来了极大的困难.基于目前大量的交互式动态影响图研究工作,文中旨在总结归纳模型的具体表达方式和求解方法,并在此基础上提出一种新的模型求解方法.针对巨大的其他智能体备选模型空间,新方法侧重于研究模型的选取技术,把模型选取问题转化为一个构造最优 K 模型的函数优化问题.优化的目标是尽量使得选取的 K 个模型能在最大程度上覆盖整个其他智能体的模型空间.从本质上说,新的函数优化问题具有 NP 难度.文中通过挖掘目标函数的单调子模特性提出一种贪婪算法以迅速求解该优化问题,并在理论上保证了解的质量.此外,新的求解方法克服了目前近似方法的随机性和参数设置的复杂性.该方法在一个经典计算机游戏领域得到了大量的实验验证,展示了较强的实际应用能力.

关键词 多智能体系统;影响图;序贯决策问题;行为等价

中图法分类号 TP18 **DOI 号** 10.11897/SP.J.1016.2018.00028

Interactive Dynamic Influence Diagram Research Summary and New Solutions on Top-K Model Selection

PAN Ying-Hui[1)] ZENG Yi-Feng[2)]

[1)](*School of Information Management, Jiangxi University of Finance and Economics, Nanchang* 330013)

[2)](*Department of Automation, Xiamen University, Xiamen, Fujian* 361005)

Abstract Multiagent sequential decision-making problem under uncertainty is an important research issue in the area of artificial intelligence, and mainly focuses on solutions to the problem of how agents shall optimize their decisions in the interactions. Particularly in a setting of partially observable, stochastic games agents can't perceive the precise states of external environments

收稿日期:2016-04-21;在线出版日期:2017-07-28.本课题得到国家自然科学基金(61375070,61562033,61772442,71361011)、江西省社会科学规划基金(16GJ20)、江西省自然科学基金(20171BAB202022)资助.**潘颖慧**,女,1981 年生,博士,副教授,主要研究方向为多智能体推理与决策、人工智能. E-mail: panyinghui@jxufe.edu.cn. **曾一锋**(通信作者),男,1976 年生,博士,教授,主要研究领域为多智能体决策、人工智能、社交网络、计算机游戏等. E-mail: yfzeng@xmu.edu.cn.

and rely on received observations to infer the hidden states. Meanwhile, the stochastic actions of agents have a direct influence on decisions of other agents. Their interactions impact the state changes of the common environment, which decides rewards in executing their actions. Hence the core task is to model agents' interactions and subsequently to solve the model. Most of the existing research models the entire multiagent systems and follows the mechanism of centralized plan and decentralized control to solve the problem. It first computes a joint policy for all the agents and then assigns the local policies to the agents for a final execution. This approach often demands that all the agents hold common knowledge of the global environment, which can only be applied in cooperative multiagent systems. In contrast, interactive dynamic influence diagram (I-DID), which takes the individual decision-making perspective, provides a general framework for solving-multiagent sequential decision problems under uncertainty. The solutions remove limitations of traditional multiagent decision approaches based on game theory. The main difficulty arises from the complicated process of mutually modeling of multiple agents in I-DID. In particular, agents can't communicate in a competitive setting so that they can't perceive the true model other agents, which requires the agents to predict and reason with other agents' behavior in order to optimize their own decisions. The solution is to first hypothesize a number of candidate models assigned to other agents and then solve the models to predict their behavior. Since the number of candidate models could be rather large and the uncertainty of the models increases with the receiving of new information over time, the number of models grows exponentially with time. This significantly increases the difficulty in solving I-DID. This paper summarizes the recent development of I-DID research and proposes a new I-DID solution. Considering the large space of candidate models of other agents, the proposed approach focuses on the selection of a proper subset of models in order to efficiently solve the I-DID. It converts the model selection problem into one top-K model optimization through function optimization techniques. The optimization problem aims to maximize the coverage of the selected K models over the entire space of other agents' candidate models. Essentially the top-K model optimization is NP hard. By exploiting the monotonic, submodular property of the objective function, this article develops a greedy selection algorithm to efficiently solve the problem while ensuring the solution quality in a theoretical way. In addition, the new solution avoids randomness of the model selection and reduces the complexity of parameter settings in approximate I-DID solutions. Performance of the new method is experimentally verified in a new problem domain of computer games and shows strong capability in practical applications.

Keywords　multiagent systems; influence diagrams; sequential decision-making problem; behavioral equivalence

1　引　言

多智能体系统可以完成单个智能体难以胜任的复杂任务，已经在很多领域有着相当广泛的应用，譬如航天、军事、机器人、灾难援救、供应链管理等等. 在上述众多的应用中，特别是不确定环境下，如何优化多智能体的决策一直是科学研究中的一个难点. 尤其随着多智能体系统规模的日益膨胀，其求解方法也面临着严峻的考验.

求解不确定性多智能体决策问题的传统方法往往是从整个多智能体系统的角度出发，对系统中所有的智能体进行统一的建模并求解其联合决策，最后把所求的局部决策分配给各个智能体执行. 该方法属于典型的集中规划、分散控制的求解机制. 显而易见，当多智能体系统随着智能体数目的增加迅速

膨胀,其方法将遭遇到无法解决的维数灾难问题.这个问题已经得到多智能体研究者的高度重视,从而引起了对传统集中式规划方法的重新审视.针对规模庞大的智能电网决策问题,著名的智能体研究科学家 Jennings 研究团队直接指出传统方法的维数灾难问题,并提出采用基于个体控制的多智能体决策求解方法[1].同时,Durfee 研究团队集中研究基于系统维数的多智能体决策的复杂度衡量问题,从理论上证实了集中规划等求解方法的不可扩展性[2].

多智能体数目增加的直接后果是系统的异质性更为突出:各种各样的智能体存在于一个大规模的系统中,智能体之间的合作和竞争关系并存.譬如,在 Web 服务问题上,网络中每个服务智能体力争自己的收益最大,它们之间有竞争关系,但在此基础上,智能体还需考虑资源共享以便提高自己的服务质量,因此他们之间还有合作关系[3].在一个拥有数目众多能源消费者和供应商的智能电网中,供应商彼此相互竞争以争取各自最大的商业利润,与此同时,消费者必须相互协调以优化(从供应商)获取的能源,系统中智能体之间也是合作竞争关系.如果智能体之间存在着竞争关系,它们将不会共享所有的信息,那么基于集中规划的多智能体决策传统求解方法将直接失效.因此,针对系统维数膨胀而带来的多智能体决策问题的求解方法,将不是传统求解方法的简单扩展,需要进行全面而细致的研究.

从单个智能体的角度出发研究不确定性多智能体决策问题是目前出现的一种新型建模理论.最为典型的方法是交互式部分可观测马尔可夫决策过程 I-POMDP(Interactive Partially Observable Markov Decision Process)[4] 和交互式动态影响图 I-DID (Interactive Dynamic Influence Diagram)[5],其核心思想是采用智能体相互建模技术,把多智能体的决策问题转化为个体决策问题.通过建立交互状态空间,个体智能体可以清晰地表示其他智能体决策过程.建模过程并不需要对多智能体决策过程做出共同知识的假设[6],从而突破了纳什平衡点的解约束.因此,该方法不仅能够求解合作型的多智能体决策问题,也可以求解竞争型的多智能体决策问题.由于难以预知其他智能体的真实模型,模型求解的主要难点在于计算其他智能体的数目众多的候选模型.与 I-POMDP 相比,I-DID 具有更好的问题表征能力,能够有效地利用潜在的问题结构,更为高效地求解模型.

为了压缩巨大的其他智能体候选模型空间,目前大量的 I-DID 研究工作采用行为等价原理[7],提出多种精确、近似算法以求解大规模 I-DID 模型.这些研究已经陆续发表在近 10 年的智能体/人工智能的主要国际顶级会议和期刊上,形成了一个可持续发展的研究主题.鉴于此,本文主要做出如下两个贡献:

(1)对 I-DID 模型的构造进行详尽地阐述,全面总结目前典型的 I-DID 模型求解方法,为多智能体的研究者提供系统的 I-DID 研究介绍;

(2)提出一个崭新的 I-DID 模型求解技术.目前的 I-DID 研究主要通过近似的模型选择方法来压缩模型空间,但方法往往具有随机性,选择不严谨.本文把模型选择转化成一个函数优化问题,提出了最优 K 模型选择技术,最大化所选 K 模型对整个模型空间的覆盖程度.

本文第 2 节概括总结与 I-DID 相关的多智能体决策模型研究工作,以明确 I-DID 研究的必要性和主要特点;第 3、4 节分别对 I-DID 模型的建立和求解进行详细地介绍和归纳;在此基础上,第 5 节详细阐述最优 K 模型选择技术,并在计算机游戏决策问题中验证该技术的性能;最后,本文讨论 I-DID 应用前景及其后续主要研究工作.

2 不确定性多智能体决策的相关研究

I-DID 的研究隶属于不确定性多智能体决策问题,本节将从多智能体决策问题的算法及应用、I-DID 模型及算法这两个方面论述多智能体决策的相关工作.

2.1 多智能体决策问题的算法及应用

多智能体决策问题一直是多智能体系统研究领域的重点和热点,其研究成果能直接提高自主智能体的研发水平,促进智能体技术的实际应用.

2.1.1 多智能体决策问题

求解多智能体单步决策问题是通过建立收益表(Payoff Matrix)或对策树寻找纳什平衡点,主要的研究困难在于庞大的搜索空间[8].与传统的求解方法不同,斯坦福大学 Koller 等人利用影响图 ID (Influence Diagram)的结构化特点[8],创立多智能体影响图 MAID(MultiAgent Influence Diagram)模型用以有效地表示多智能体之间的静态结构关系,分解可行解的搜索空间,能够处理复杂的多智能体对策问题[9].姜鑫等人在博弈论框架下,利用 MAID 对军事决策进行分析和建模[10].MAID 是从多智能体

系统局外者的角度对决策进行全局分析，往往假设每个智能体的信息和知识是共享的.因此这种建模方法仍然属于集中规划、分散控制的范畴.在 MAID 发展的基础上，网络影响图 NID(Network of Influence Diagram)对其他智能体模型的不确定性进行了建模，采用分层求解方法解决多智能体的决策问题[11].不管 MAID 还是 NID 模型，它们的求解结果都是纳什平衡点.由于纳什平衡点的不完全性和多个解并存的特性，求解方法并不能适用于一般的决策控制问题.

与多智能体单步决策问题相比，多智能体序贯决策问题不仅需要考虑决策行为的即时收益(Immediate Reward)，而且必须考虑其产生的未来收益.目前，主要的研究方法集中在分散式部分可观测马尔可夫决策过程 DEC-POMDP(Decentralized Partially Observable Markov Decision Process)[12-13]，即把单个智能体的部分可观测马尔可夫决策过程 POMDP(Partially Observable Markov Decision Process)[14-16]推广到多智能体的环境设置当中.由于采取全局建模的方式，DEC-POMDP 的模型求解是相当困难的，属于 NEXP 完全问题.最新的 DEC-POMDP 技术是基于取样的期望-最大化算法，可以求解较大规模的多智能体决策问题[17].

在国内，张迎晓等人提出基于 DEC-POMDP 的多用户频谱接入算法，提高了无线频谱的利用率[18].刘海涛等人提出基于有向无环图的分散式通信决策算法，应用于经典的 DEC-POMDP 问题——老虎问题[19].在国外，Marecki 等人充分利用智能体局部合作关系以提高 DEC-POMDP 的求解能力[20].Velagapudi 等人允许智能体进行部分的通信交流以进行决策协商，可以求解超过 100 个智能体的决策系统[21].但基于 DEC-POMDP 的求解方法始终需要假设所有智能体必须对环境状态有共同的信度(Belief)，因此，该方法一般只适用于具有合作关系的多智能体系统.

2.1.2　多智能体决策系统研究发展及应用

多智能体系统的研究和应用在国内外一直相当活跃，从传统的足球机器人到目前倍受关注的智能电网、国家安全和电子市场等涉及到国计民生的各个领域.譬如，Varakantham 提出在 D-TREMOR (Distributed-Team's Reshaping of Models for Rapid Execution)模型中智能体通过交互个人规划重组自己的值函数和状态转移函数，规划的速度和质量取决于智能体的优先级顺序[22]，并通过灾难救援问题验证了方法的有效性[23].Okamoto 研究在竞争环境中考虑攻击成本的情况下，如何对网络流量发送者、竞争对手两个智能体进行建模，并提出一种非零和多智能体网络流量安全博弈方法[24].卡耐基梅隆大学 Sycara 长期致力于多智能体的分散控制决策，并将其成果应用于足球机器人、城市搜索和救援等领域[25].德州大学奥斯汀分校的 Stone 侧重于研究多智能体系统中的团队合作学习问题，应用到足球机器人比赛、机器人导航[26]、智能电网等领域.Jennings 目前侧重于多智能体在提高能源效率方面的应用，主要通过预测用户及其能源供应商(多智能体)的决策行为优化能源的分配[27].南加州大学的 TEAMCORE 研究团队主要研究多智能体的博弈问题，并将其成果应用于机场(码头)、公交系统的安全保卫[28]、野生动物保护[29]等.牛津大学的 Woodridge 主要研究多智能体复杂计算和多智能体博弈，应用于电子市场等领域[30].

在国内，目前大多多智能体的研究工作集中在求解多智能体的协作决策问题.如 Jiang 等人提出一种基于声誉机制的分配模型，将其应用于社会网络中不可靠的多智能体系统的任务分配[31].Jiao 提出智能体根据对其他智能体行为的预测进行更加有效地合作[32].刘春阳等人提出基于投票的多智能体强化学习方法，将其应用于足球机器人比赛的团队协调决策[33].马广富等人研究有向网络下基于一致性理论的非线性多智能体系统的协调跟踪[34].Chen 等人解决了无人驾驶机协同攻击的路径规划问题[35].蒋伟进等人将多智能体协作技术应用于知识管理和知识动态复用领域[36].

关于智能体之间竞争关系的研究近年来也逐渐成为热点.例如，钟伟才等提出一种将多智能体系统与遗传算法相结合的组合优化进化算法，仿真模拟了多智能体的竞争行为和自学习行为，将其应用于上千维的欺骗问题和等级问题[37].范波等人将多智能体系统分为团队内部的合作关系和团队之间的竞争关系，提出分层的马尔可夫对策协调方法，利用足球机器人仿真验证了方法的有效性[38].朱曼玲等人研究在多智能体竞争环境中 Web 服务选择的可信问题，即服务智能体发出服务请求后，如何在响应请求的服务智能体中选择可信任的一个[3].张庆杰等人提出复杂网络条件下多无人驾驶机系统任务区集结问题的非合作求解方法[39].

随着多智能体系统规模的扩大，网络特性成为必须考虑的重要因素.Delgado 提出复杂网络拓扑

结构与多智能体系统中社会公约传播效率有密切关系[40]. Gaston 等人提出复杂网络的拓扑结构对多智能体系统动态行为有显著影响[41]. 徐杨等人研究了复杂网络特性对大规模多智能体协同控制的影响[42]. 李晓等人阐述了复杂网络的特性,并分析了与多智能体的一致性关系[43].

由此可见,目前多智能体决策系统的研究大多集中在多智能体合作完成任务,主要应用于无人机驾驶、足球机器人、路径规划、任务分配、知识管理、智能电网等领域. 大量文献都提到大规模多智能体系统研究的必要性,也利用复杂网络特性、启发式算法、进化算法等或多或少的解决了一些问题;但是维度灾难仍然是研究的最大难点,限制其在实际生活当中的应用. 纵观多智能体决策系统求解方法及其应用的发展历程,可以发现主要的研究方向趋向于模型所依赖的假设越来越少,从只适用于合作关系到适用于各种关系,从解决静态决策问题到序贯(动态)决策问题. 对于无假设的序贯决策问题,目前最具适应性的方法之一是 I-DID.

2.2 I-DID 模型及算法

相对于 I-POMDP,I-DID 的优势在于能够利用决策问题中各种变量之间的条件独立性更加清楚直观地描述待解决的决策问题. 相对于 DEC-POMDP,I-DID 的优势在于取消了"多智能体必须对环境状态保持共同信度"的假设,其应用并不局限于合作性的多智能体系统,从而更加适用于具有普遍意义的多智能体系统. 目前,I-DID 的应用领域不仅涵盖有敌对关系的反洗钱智能体系统,也包括具有合作关系的仓库自动小车系统[44-45].

I-DID 算法主要采用行为等价方法压缩其他智能体的模型空间,达到求解多个时间片规划的目的. 譬如,Doshi 等人采用区分性模型更新方法 DMU(Discriminate Model Update)首次突破求解 5 个规划时间片老虎问题的瓶颈[46]. Zeng 等人提出(ε-d)近似行为等价原理合理修正了行为等价原理的严格条件[47],并利用增量式策略识别近似行为等价模型[47],进一步限制了候选模型的数量,取得了较大成效,拓展了 I-DID 的应用范围[48]. 此外,Zeng 等人通过动作等价方法合并行为等价模型,进一步提高了 I-DID 的求解能力. 潘颖慧等人以 3 个智能体为例初步探讨了多智能体 I-DID 的建模方法[49]. 最近,Chen 等人提出了 I-DID 模型的在线求解方法,利用多智能体的实时交互信息,逐渐判断其他智能体的真实模型[50]. Conroy 等人主要研究基于数据的 I-DID 模型建立及其求解方法,克服了有限的领域知识对模型建立产生的不准确性[51]的缺点.

由于 I-DID 没有对智能体的行为做出任何的假设,主体智能体与其他智能体也没有进行通信,没有事先协议,每个智能体都是独立的而且对环境和其他智能体行为是部分可观测的. 因此,I-DID 模型具有求解一般多智能体序贯决策问题的天然优势. 目前大量的高效 I-DID 算法也为拓展 I-DID 应用提供了重要保证. 此外,类似于 I-DID 的相互建模技术是研究 ad hoc 智能体决策系统的一个有前景的研究方向,在 ad hoc 智能体决策领域倍受青睐[52]. 同时,在混合型多智能体决策的研究中,主要挑战在于如何对不同类型的智能体进行建模从而预测其对应的行为状态. 由于 I-DID 的研究技术并没有对智能体的类型进行预先假设,因此 I-DID 自然成为一个求解混合型多智能体决策问题的主要研究方法[53].

3 I-DID 的建模

I-DID 是概率图模型求解多智能体决策问题的一个具体方法. 本节从单个智能体的 ID、动态影响图 DID(Dynamic Influence Diagram)到多个智能体的 I-DID 详细介绍这些模型的建立方法. 在基本的概率图模型 ID 基础上,DID 表示了单个智能体多步决策过程. 在此基础上,I-DID 拓展 DID 以求解多智能体序贯决策问题.

3.1 ID 模型

ID 模型是在马尔可夫决策过程和图论的基础上提出的一种求解单个智能体决策问题的有效方法[8]. 从数学上解释,ID=(A_i,S,O_i,Ω,R_i),如图 1 所示. 其中矩形表示决策节点,A_i即智能体 i 的决策行为. 椭圆表示随机节点,S 是客观存在的环境状态,O_i是智能体 i 的观察函数,观察值用 Ω 表示. 菱形表示值节点,R_i即智能体 i 的值函数.

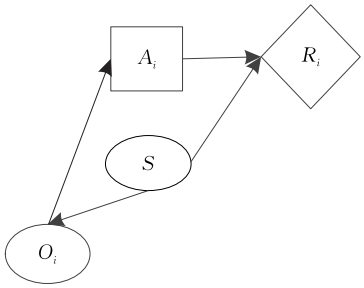

图 1 ID 的一般结构

本文拟用老虎问题[8]来阐述模型的建立过程. 老虎问题是探讨 ID 模型及其求解方法的经典案例.

[**单个智能体的老虎问题**]：存在一个智能体 i，它有三种决策动作：开左门、开右门和听，分别用 OL、OR 和 L 表示. 环境状态有两种：老虎在左门后或在右门后，分别用 TL 和 TR 表示，另一个门后是金子. 如果老虎在左门，而 i 打开左门则会受到惩罚，惩罚值为 -100；开右门则获得金子，奖励值 10；也可以选择听，值为 -1. 如果选择听，观察值包括老虎吼叫在左边、老虎吼叫在右边，分别用 GL 和 GR 表示，正确率为 0.85. 老虎问题的 ID 模型如图 2 所示.

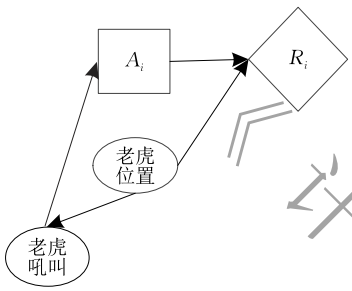

图 2　老虎问题的 ID 模型

观察函数 O_i 和状态转移函数 S 分别如附表 1 和附表 2 所示. 其中附表 2 中概率为先验概率，即假设初始状态为老虎在左边门中的概率与右边相等. 值函数 R_i 如附表 3 所示. 求解 ID 的主要技术是基于传统的消元法[54]. 譬如，在图 1 中，可以先消去 S 节点，链接节点 O_i 和 R_i，然后就可以根据不同的观察值（在观察节点 O_i）在决策节点 A_i 中通过期望值最大化原则选择不同的最优决策.

3.2　DID 模型

为了求解单个智能体多步决策问题，Tatman 和 Shachter 提出了 DID[55]，也就是多个时间片的 ID 模型. 图 3 是两个时间片的 DID，决策节点 A_i 的取值为智能体的有限动作集；随机节点 S 和 O_i 的变量取值分别为有限状态集和有限观察集合；另外随机节点 S^{t+1} 和 O_i^{t+1} 上的条件概率分布 $\Pr(S^{t+1}\mid S^t, A_i^t)$ 和 $\Pr(O_i^{t+1}\mid S^{t+1}, A_i^t)$ 分别是状态转移函数、观察函数. R_i 上的条件概率表是值函数.

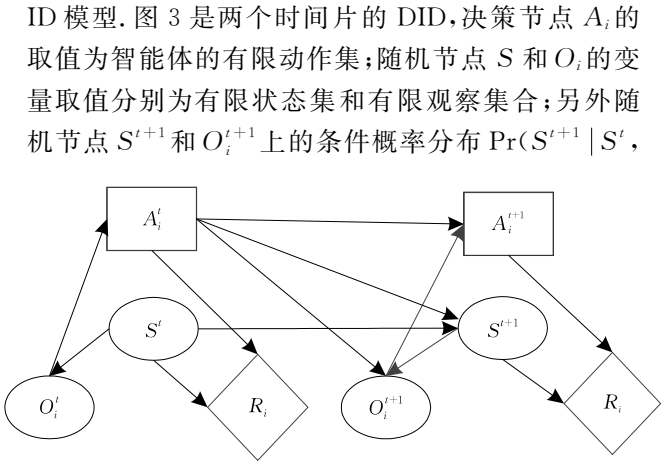

图 3　两个时间片的 DID

将 ID 模型推广到 DID，两个时间片的老虎问题模型如图 4 所示. 第 0 个时间片模型的条件概率表见附表 1、附表 2、附表 3. 假设如果在第 t 时间片中智能体 i 打开了任何一扇门，全部数据重置. 观察函数和状态转移函数如附表 4 和附表 5 所示，值函数不变.

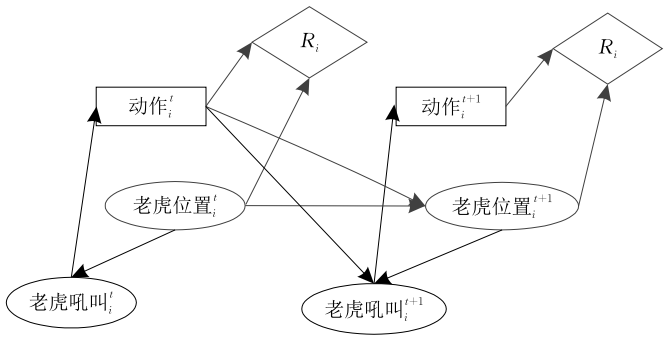

图 4　老虎问题的 DID 模型

3.3　I-DID 模型

交互式影响图 I-ID（Interactive ID）把 ID 扩展到了多智能体环境. 除了已有的决策节点、随机节点、值节点外，I-ID 还包含一种新的节点：模型节点（Model Node）. 如图 5(a)所示，在 l 层的 I-ID 中，用六边形表示模型节点 $M_{j,l-1}$. 除了模型节点，I-ID 中还增加了策略链，图中用随机节点 A_j 与模型节点之间的虚线表示. 随机节点 A_j 表示智能体 j 动作的概率分布. 图中 S 节点表示智能体共处的环境变量、O_i 节点表示智能体 i 的观测值、A_i 节点表示智能体 i 的决策、R_i 节点表示智能体 i 的值函数[55]. 如果不存在其他智能体，则不存在模型节点和随机节点 A_j，此时 I-ID 变成普通的 ID.

$l-1$ 层智能体 j 的模型节点 $M_{j,l-1}$ 的具体构造如图 5(b)所示. 智能体 j 所有可能存在的模型都包含在模型节点里，假设模型有两个，用 $m_{j,l-1}^1$ 和 $m_{j,l-1}^2$ 表示，这些模型可能是 I-ID，也可能是 ID. 求解智能体 j 的模型，可以得到每个模型的最优动作集及其动作的概率分布，譬如图 5(b)中的 A_j^1 和 A_j^2. 用 OPT 表示求解 $m_{j,l-1}^1$ 得到的最优动作集，则模型 $m_{j,l-1}^1$ 中当 $a_j\in OPT$ 时，动作的概率分布为 $\Pr(a_j\in A_j^1)=1/|OPT|$，否则为 0. 因此每一个 $l-1$ 层 I-ID 模型的动作节点转换成一个随机节点（在图中为 A_j^1, A_j^2），并且产生最优策略相对应的动作 a_j 的概率分布均为 $1/|OPT|$ $(a_j\in OPT)$. 图 5(b)中，不同的模型对应的动作节点（A_j^1, A_j^2）与节点

$Mod[M_j]$作为节点 A_j 的父节点. 由于每一个动作节点对应于一个模型,则在模型节点 $M_{j,l-1}$ 中的动作节点个数与模型节点中的模型数目相同. 随机节点 A_j 的条件概率分布采用动作节点 A_j^1 或者 A_j^2 的概率分布,至于选择哪个动作节点的分布则依赖 $Mod[M_j]$ 的值,用 $Mod[M_j]$ 的值来区分 j 的不同模型. 举例来说,当 $Mod[M_j]$ 的值为 $m_{j,l-1}^1$ 时,随机节点 A_j 采用节点 A_j^1 的分布;当 $Mod[M_j]$ 的值为 $m_{j,l-1}^2$ 时,随机节点 A_j 采用节点 A_j^2 的分布. 节点 $Mod[M_j]$ 的概率分布就是智能体 i 对 j 模型的信度. 图 5(b)使策略链的概念更加清晰,可以看出策略链可以用传统 ID 的弧(或者边)表示出来.

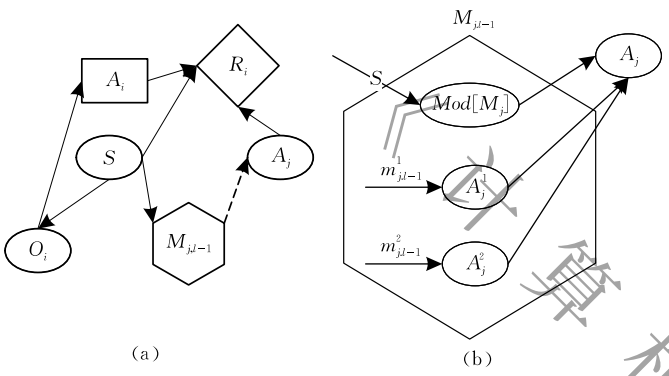

图 5 l 层上的 I-ID 和模型节点内部结构

在图 6 中,我们用一个 ID 表示 I-ID,把图 5 中的模型节点用图 6 中的随机节点和之间的边来替代. 这样就没有特殊的策略链,取而代之的是传统 ID 中节点类型和节点之间的依赖关系. 这样便可以用求解传统 ID 的方法来求解 I-ID.

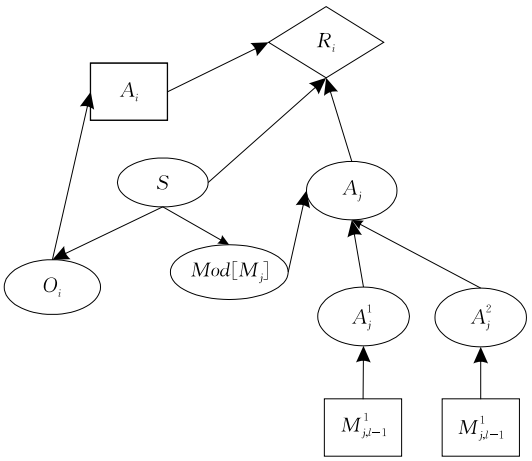

图 6 用 ID 表示 I-ID

如同 DID 扩展 ID 一样,I-ID 在时间上扩展便得到 I-DID 模型,用来处理动态决策问题. 图 7 给出第 l 层两个时间片的 I-DID,其中模型节点间的圆点虚线箭头叫做模型更新链[56].

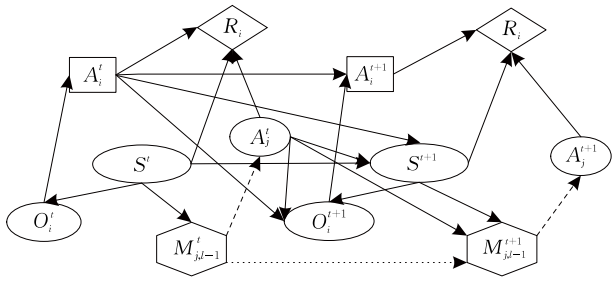

图 7 两个时间片的 I-DID

与 DID 类似,随机节点 S^{t+1} 上的条件概率分布 $\Pr(S^{t+1}|S^t,A_i^t,A_j^t)$ 和 O_i^{t+1} 上的条件概率分布 $\Pr(O^{t+1}|S^{t+1},A_i^t,A_j^t)$ 分别对应决策问题中的状态转移函数 T_i 和观察函数 O_i. I-DID 与 DID 的主要不同之处是模型更新链,下面详细说明模型如何更新.

模型节点更新需两步:(1)给定 t 时刻的模型,确定 $t+1$ 时刻模型节点中的模型更新集合. 一个智能体的意图模型包含它的信度,当智能体执行动作进入当前状态并且获得新的观察时,它们的模型和信度就得到更新. 在某些情况下,这种更新可能会导致模型的结构与以前有所不同. 由于模型的最优动作集不确定,可能是任意一个动作,且智能体 j 可能的观察值不确定,可能是 $|\Omega_j|$ 个观察中的任意一个,因此在 $t+1$ 时刻的更新集合最多包含 $|M_{j,l-1}^t||A_j||\Omega_j|$ 个模型. 其中,$|M_{j,l-1}^t|$ 是 t 时刻的模型个数,$|A_j|$ 和 $|\Omega_j|$ 分别表示智能体 j 可能的动作个数和可能的观察值个数;(2)根据 t 时间片智能体 j 模型的概率分布、可能得到的观察、做出的行为决策,求解 $t+1$ 时间片智能体 j 模型的概率分布.

图 8 表示在 I-DID 中模型更新链的实现. 假设在第 t 时间片,智能体 j 在 $l-1$ 层存在 2 个模型,分别为 $m_{j,l-1}^1$ 和 $m_{j,l-1}^2$,2 个模型都只有一个最优动作,而且智能体 j 可能得到的观察值也有 2 个,分别

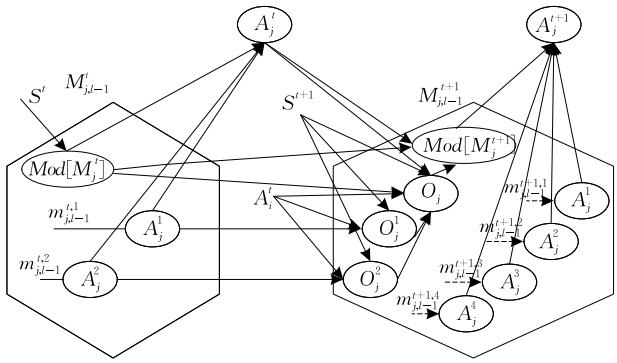

图 8 I-DID 中模型更新链的实现

为 o_j^1 和 o_j^2，则在 $t+1$ 时间片包含 4 个新模型，在图 8 中用 $m_{j,l-1}^{t+1,1}$，$m_{j,l-1}^{t+1,2}$，$m_{j,l-1}^{t+1,3}$ 和 $m_{j,l-1}^{t+1,4}$ 表示. 由于 j 通过其动作和观察来更新它的信度，因此这些模型的信度不同. 这 4 个模型代表 $l-1$ 层上的 I-DID 或者 DID，其动作节点分别与随机节点 A_j^1，A_j^2，A_j^3 和 A_j^4 相对应[56].

接下来讨论怎样计算模型更新后的概率分布，即随机节点 $Mod[M_j^{t+1}]$ 的分布. 智能体 j 更新后的模型(例如 $m_{j,l-1}^{t+1,1}$)的概率分布依赖于导致此模型的动作和观察值，并且取决于 t 时刻模型的概率分布. 图中 A_j^t 动作节点的分布由 $Mod[M_j^t]$ 确定，智能体 j 决策行为的概率由 A_j^t 确定. 与 A_j^t 类似，O_j 观察节点的分布也由 $Mod[M_j^t]$ 确定，观察到 o_j^1 或 o_j^2. 因为智能体 j 观察节点的概率分布条件依赖于环境状态和智能体 i 和 j 的联合动作，因此 o_j^1 和 o_j^2 分别与 S^{t+1}，A_j^t 和 A_i^t 相连. 最后，t 时刻模型的分布从随机节点 $Mod[M_j^t]$ 中得到. 所以，随机节点 $Mod[M_j^t]$，A_j^t 和 O_j 作为 $Mod[M_j^{t+1}]$ 的父节点.

由此可见，模型更新链可以用两个模型节点中随机节点间的依赖关系来替换. 图 9 表示了两个时间片的 I-DID，其模型节点用随机节点来替代，这样，I-DID 就可以转换成 DID，便可以用求解传统 DID 的方法来求解 I-DID. 从图 9 还发现，除去图中用加粗表示的节点和边，I-DID 就变成单个智能体的 DID 了[56].

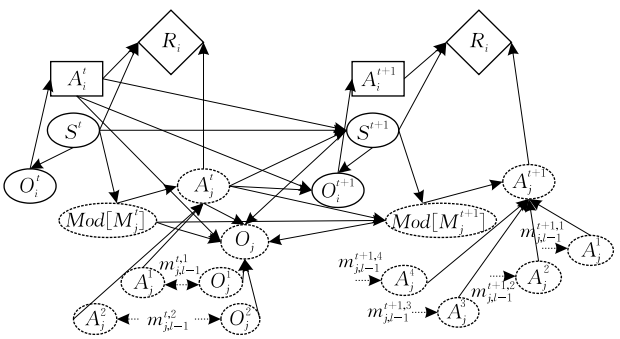

图 9　用 DID 表示 I-DID

重复模型更新的步骤，可以得到多时间片的 I-DID. 假设时间片长度为 T，则智能体 j 模型节点中的模型个数最多为 $M_{j,l-1}^{t=1}(|A_j||\Omega_j|)^{T-1}$ 个. 可以看出，随着时间片的增长，模型个数呈指数增加.

[**两个智能体的老虎问题**[56]]：假设有两个智能体 i 和 j，每个智能体能做的决策动作与单个智能体老虎问题相同，即 OL、OR 和 L. 观察函数与单个智能体不同，除了 GL 和 GR，智能体听到的声音还包括另一个智能体的开门声，即开门声在左边(CL)、开门声在右边(CR)或没有声音(S). 假设听老虎吼叫的正确率仍然为 0.85，听开门声的准确率为 0.9，则智能体更加倾向于听开门声. 如果有任一智能体开门，环境状态——老虎的位置重置.

现对智能体 i 建模，首先建立 i 的第 1 层 I-ID 模型. 由于 i 包含对 j 模型的信度，也就是说第 1 层 I-ID 包含 j 第 0 层的模型. 假设 i 考虑两个不同的 0 层模型，图 10 表示一个两层的 I-ID. 图 10(a)表示 i 的第 1 层 I-ID，(b)表示 j 的第 0 层 ID，这里两个 0 层 ID 的动作节点与(a)中的随机结点 A_j^1 和 A_j^2 相连. 求解 j 的 0 层模型，为 i 的第 1 层模型提供 j 动作的概率分布，实现 i 对 j 动作的预测(在图中用 A_j^t 表示)，i 的第 1 层 I-ID 变成 ID，就可用解 ID 的方法进行求解.

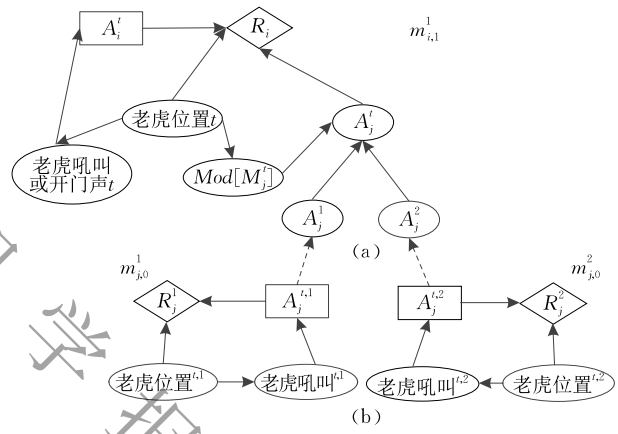

图 10　老虎问题的 I-ID 模型

将 I-ID 扩展到多个时间片就得到 I-DID，以两个时间片为例，图 11 表示智能体 i 第 1 层两个时间片的 I-DID. 模型结点 $M_{i,0}^t$ 中包含第 0 层的 DID. 且在当前时刻，这些 DID 可能对老虎的位置有不同的信度. 求解智能体 j 在 0 层的这些 DID 得到 j 动作的概率分布，并表示于随机结点 A_j^t 中. 当 j 执行在 t 时刻的最优动作后，听到老虎吼叫声或者开门声，然后得到 $t+1$ 时刻 j 对老虎位置的信度.

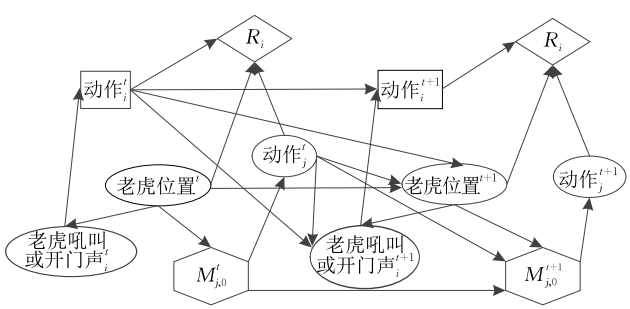

图 11　两个时间片的老虎问题的 I-DID 模型

将模型节点和模型更新链用随机节点和依赖关系取代，可得到如图 12 所示模型. 图中状态转移函数 S^{t+1}、观察函数 O_i^{t+1} 和值函数 R_i 如附表 6、附表 7、附表 8 所示.

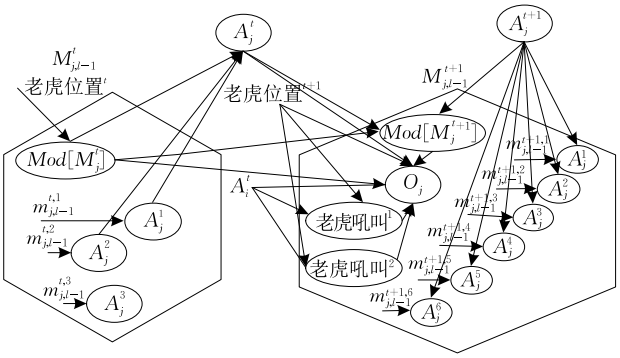

图 12 老虎问题中模型更新链的实现

4 I-DID 的求解方法

I-DID 的求解非常复杂，主要原因在于：其他智能体的候选模型数量随时间变化呈指数增长(从图 8 可以看出)；S 的状态空间是智能体之间的交互式状态，状态空间很大；且多智能体之间相互建模使得模型状态空间更是随着建模嵌套层数的提高而急剧增加. 现有的解决方法中常用的是基于行为等价原理 BE(Behavioral Equivalence)的求解方法，基本思想是：如果智能体 i 对 j 的两个候选模型的行为预测完全一致，那么这两个模型对智能体 i 决策的影响是没有任何区别的，属于 BE 模型，可以将其中一个删除，把该模型的信度赋予另一个模型. 本节将全面总结基于 BE 的 I-DID 求解方法，这些方法构成了 I-DID 目前的主要研究内容和发展基础.

4.1 I-DID 基本算法框架

从第 3 节的阐述可以知道，求解 I-DID 主要包括两个步骤：(1) 通过求解其他智能体 j 的候选模型逐步拓展模型节点，建立 I-DID 模型；(2) 求解 I-DID 模型. 算法 1 描述了求解 I-DID 的主要步骤.

算法 1. I-DID 基本求解算法.

输入：I-DID 模型

输出：智能体 i 的决策

1. FOR t from 0 to $T-1$ DO
2. IF $l\geq 1$ THEN

最小化 $M_{j,l-1}^t$

3. FOR each m_j^t in $M_{j,l-1}^t$ DO
4. 递归调用 $l-1$ 层 I-DID(或 DID)算法表示 m_j^t 和时间片 $T-t$
5. 将解 I-DID(或 DID)的决策节点 $OPT(m_j^t)$ 映射到相应的节点 A_j
6. $PruneBehavioralEq(M_{j,l-1})\rightarrow M_{j,l-1}^t$

推广到 $M_{j,l-1}^{t+1}$

7. FOR each m_j^t in $M_{j,l-1}^t$ DO
8. FOR each a_j in $OPT(m_j^t)$ DO
9. FOR each o_j in O_j(m_j^t 中的组成部分)DO
10. 更新智能体 j 的信度，$SE(b_j^t,a_j,o_j)\rightarrow b_j^{t+1}$
11. 新的 I-DID(或 DID) b_j^{t+1} 作为初始信度 $\rightarrow m_j^{t+1}$
12. $M_{j,l-1}^{t+1}\bigcup\limits_{\leftarrow}\{m_j^{t+1}\}$
13. 增加模型节点 $M_{j,l-1}^{t+1}$ 以及 $M_{j,l-1}^t$ 和 $M_{j,l-1}^{t+1}$ 之间的模型更新链
14. 为 $t+1$ 时间片模型增加随机节点、决策节点和效用节点和它们之间的弧
15. 建立每个随机节点和效用节点的条件概率表
16. 将 $l>=1$ 的 I-DID 转换成普通的 DID，然后应用标准的 DID 求解方法进行求解

算法 1 采用自下而上的方法求解 l 层的 I-DID 模型(或者 I-ID 模型，如果只考虑单步决策)以得到决策树. 算法 3～5 步主要求解 $l-1$ 层的模型以获取其他智能体 j 的决策. 根据这些决策，智能体 i 可以对 j 每一步的动作做出预测. 算法第 5 步约减迅速膨胀的 j 的候选模型，降低预测动作嵌入到 I-DID 的复杂度. 算法 7～15 步遵循图 8 的方法以拓展 I-DID 在下一个时间片的模型表示，其中 $SE(b_j^t,a_j,o_j)$ 是信度更新的缩写. 算法第 16 步根据图 9 把 I-DID 转化成标准的 DID 模型，可以采用标准的 look-ahead 算法求解. 上述算法可以通过一些 ID 工具(如 Hugin API)实现.

可以看出，求解 I-DID 的复杂度在于大量的智能体 j 模型出现在各个阶段的模型节点. 因此，如何减少智能体 j 模型空间成为求解 I-DID 的关键. 接下去的第 4.2～4.4 小节将阐述目前存在的主要模型约减技术并分析求解的复杂度和准确性. 这些技术将通过取代基本算法中的 $PruneBehavioralEq(M_{j,l-1})$ 过程，达到迅速求解 I-DID 的目的.

4.2 BE 精确算法

决策模型的解通常可以用策略树(Policy Tree)来表示，参考图 4 和图 13 进行说明. 图 4 为 2 个时间片单个智能体老虎问题的 DID 模型，图 13 为相应的策略树. T 个时间片的策略树用 $\pi_{m_{j,l-1}}^T$ 表示，则 $OPT(m_{j,l-1})\triangleq\pi_{m_{j,l-1}}^T$ 为模型的最优解. 在策略树中，每一个从根节点到叶子节点的分支组成一个动作及观察序列，用 $h_j^{T-1}=\{a_j^t,o_j^{t+1}\}_{t=0}^{T-1}$ 表示. 譬如，图 13 中有两个分支，分别为 $L\rightarrow GL\rightarrow OR$ 和 $L\rightarrow GR\rightarrow L$.

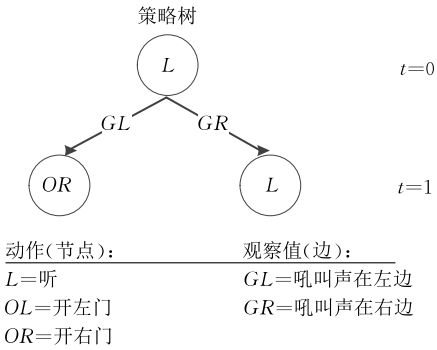

图 13　策略树和分支

定义 1.　行为等价原理：当且仅当 $\pi^T_{m_{j,l-1}} = \pi^T_{\hat{m}_{j,l-1}}$. 模型 $m_{j,l-1}, \hat{m}_{j,l-1} \in M_{j,l-1}$ 是行为等价的.

简单来说，当两个模型被认为是 BE，则它们对其他智能体的行为预测是完全一致的. 图 14 表示两个 3 步模型的策略树. 这两个模型 $m^1_{j,l-1}$ 和 $m^2_{j,l-1}$ 在 $t=3$ 时间片中深灰色和浅灰色的行为不相同，因而两个模型不是 BE 的. 对于个体决策来说，其他智能体 j 只能通过行为对主体智能体 i 产生影响，因而不需要区别行为完全一致的 BE 模型.

在基本算法 1 中，$PruneBehavioralEq(M_{j,l-1})$ 是从 $M_{j,l-1}$ 中删掉 BE 模型，返回所有代表模型的过程. 首先求解任何一个其他智能体 j 的模型，把模型的解表示成策略树；然后对所有策略树进行两两对比，删除具有完全相同分支的策略树，保留行为相异的其他智能体模型.

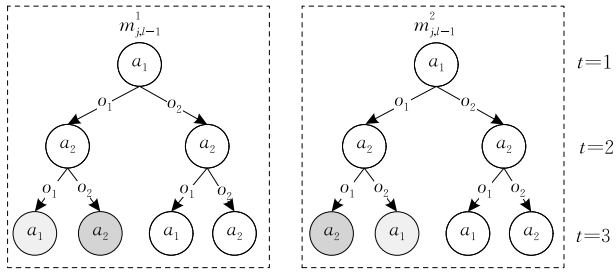

图 14　两个策略树是否行为等价

4.3　DMU 算法

基于 BE 原理的方法可以有效地剔除大量的行为等价模型，达到压缩模型空间的目的. 但其简单的策略树比较方法导致在不同时间段的模型节点仍然保留了很多 BE 模型. Doshi 和 Zeng 通过自底向上的策略树合并方法(DMU)，得到了最小模型集[57].

DMU 方法分别对策略树进行自底向上的合并，得到策略图(Policy Graph)[57]，具体过程如图 15 所示. 图中假设问题模型域中有 2 个观察值，3 个决策选择，最上面的图是求解 3 个智能体 j 的模型得到的策略树. 首先，可以分别合并叶子节点中所有的 8 个 a_3(黑色线条表示合并)和 3 个 a_1(灰色线条表示合并)，得到左下角的图. 在左下角图中，可以看出 $t=2$ 时刻决策是 a_3 的节点与 $t=1$ 时刻决策是 a_3 的节点有很多链接，其中灰色的决策节点都是在观察是 o_1 的情况下得到的 $t=2$ 时刻的 a_3，在观察是 o_2 的情况下得到的 $t=2$ 时刻的 a_1，因此可以把这些节点合并，得到右下角的图.

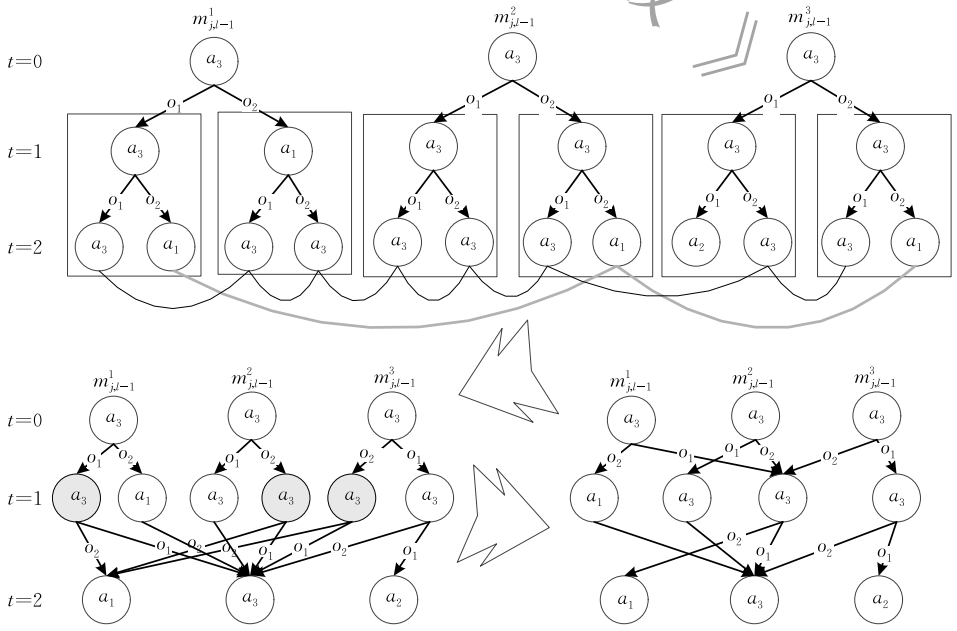

图 15　策略树的合并过程

相比基本的 BE 比较方法，DMU 方法合并在不同模型节点中具有相同将来行为的模型. 譬如在图 15 中，如果采用 BE 方法，在 $t=2$ 的模型节点中，将有 12 个模型，而不是 DMU 得到的 3 个模型.

在实现 DMU 方法的同时，可以首先从所有的 j 的模型中任意选择 K 个，$K \ll |M^0_{j,l-1}|$；然后对于剩余的 $|M^0_{j,l-1}|-K$ 个模型，如果其信度与 K 个模型的信度都相差不大(譬如 $\|b^k_j - b^{\bar{k}}_j\|_1 \leq \mu$)，就只求解这已选的 K 个模型. 最后，DMU 只需合并 K 个策略树，在各个模型节点保留行为相异模型.

4.4 (ε-d)-BE 算法

DMU 方法虽然可以大大压缩模型空间，但仍然需要根据整个策略树的异同来决定模型的约减. 随着时间片的增加，策略树的大小呈指数增长，一般很难找到两个策略树完全匹配，因此模型删除的机会并不多，同时，对比操作的复杂度还大大增加. 因而 Zeng 等人提出了近似 BE 原理，即(ε-d)-BE 算法[47].

近似 BE 原理的基本思想是：通过比较部分策略树，达到判断 BE 模型的目的. 显然，如果两个模型是 BE 的，那么它们所对应的每一个深度的部分策略树均相同；否则，它们对应某一深度(小于完全深度)的策略树可能相同，但高深度的策略树将相异. 那么一个核心的问题是如何确定策略树的比较深度，以判断近似 BE 模型.

判断的主要原理是基于随机转移矩阵中的最小混合率[58]：在马尔可夫随机过程中，相同状态空间下的信度概率的 KL(Kullback-Leibler)距离以一定的混合率逐渐缩小. 具体地说，最小混合率如下定义.

$$\gamma_{F_{a,o}} = \min_{m_{j,l-1},\hat{m}_{j,l-1}} \sum_{s' \in S} \min\{F_{a,o}(s' \mid s_{m_{j,l-1}}), F_{a,o}(s' \mid s_{\hat{m}_{j,l-1}})\} \quad (1)$$

这里 $F_{a,o}(s' \mid s)$ 是在智能体 j 的可能的观测值 o 和决策 a 的共同作用下，状态从 s 转化到 s' 的一个随机过程；$\gamma_{F_{a,o}}$ 是这一随机过程中的压缩率.

由于分支可能包括不同的动作-观察序列，要求的最小混合率 γ_F 为

$$\gamma_F = \min\{\gamma_{a,o} \mid a \in A_j, o \in \Omega_j\} \quad (2)$$

也就是说，当信度概率转移到某一阶段(对应策略树的深度)，其相互的 KL 距离将小于一个界定值 ε；而这个界定值决定着剩余部分策略树的相异程度. 如果这个界定值足够小，那么部分策略树将不会有差别. 鉴于此，可以定义近似行为等价模型(ε-d)-BE 如下.

定义 2. (ε-d)-BE 模型. 如果满足下列两个条件，智能体 j 的两个候选模型 $m_{j,l-1}$ 和 $\hat{m}_{j,l-1}$ 是(ε-d)-BE 的：(1) 长度为 d 的策略树是完全一致的，$d \leq T-1$，即 $\pi^T_{m_{j,l-1}} = \pi^T_{\hat{m}_{j,l-1}}$；(2) 当 $d < T-1$ 时，两个有序策略树叶子节点的信度 KL 距离不大于 ε，即 $\max_{k=1\cdots|\Omega_j|^d} D_{KL}[b^{d,k}_{m_{j,l-1}} \| b^{d,k}_{\hat{m}_{j,l-1}}] \leq \varepsilon$.

根据式(2)，可以推导出 d 为

$$d = \min\{T-1, \max\{0, \lfloor In(\varepsilon / D_{KL}[b^{0,k}_{m_{j,l-1}} \| b^{0,k}_{\hat{m}_{j,l-1}}]) / In(1-\gamma_F) \rfloor\}\} \quad (3)$$

因此，在确定(ε-d)-BE 模型过程中，必须根据一个信度距离界定值计算策略树的比较深度 d. 如果两个模型对应的 d 层策略树相同，那么这两个模型是(ε-d)-BE 模型；否则不是.

注意到最小混合率可能出现两种极端的情况：$\gamma_F = 1$ 和 $\gamma_F = 0$. 前者意味着每组状态更新后的信度是完全一样的，KL 距离为 0，可以令 $d=1$；后者则表示至少有一组状态更新后完全不同，这种情况下 KL 距离完全没有压缩，d 无法通过公式求得，可以假设任意选择 d，即 $d \leq T$-1.

4.5 增量式 ε-BE-I 算法

(ε-d)-BE 算法需要求解出完整的策略树，然后比较深度为 d 的部分策略树，且该方法不能在最小混合率为 0 的问题领域中确定需要比较的部分策略树的深度. 鉴于此，Zeng 等人提出增量式 ε-BE-I 算法，允许部分策略树的分支比较深度不同，设定最大比较深度 d[59]. 即只有当叶子节点更新后的信度差距不足以判断模型是否等价时，才考虑增大策略树的深度，直到所有分支的比较深度达到最大为止. 通过这个改进，策略树分支的比较深度将有所不同，从而提高了效率.

增量式技术的具体思想是：随着每一步时间片的增加，对比部分策略树和更新后叶子节点的信度差距，当动作选择不同时，立刻终止对比操作. 得出结论：两个模型不是近似(ε-d)-BE 等价；否则，扩大策略树的规模，直到更新后叶子节点的信度差距小于 ε 或者到达最大深度 q_U. 注意到即使没有到达 q_U，只要部分策略树在深度 q_r(小于 q_U)相同且更新后叶子节点的信度差距小于 ε，策略树就不会继续扩张，对比过程结束，取最小长度为 q_L. 这是因为足够小的信度差距已经能够用来判断未来的行为是否等价. 整个过程是一个增量式的策略树比较，简称这种方法为 ε-BE-I.

显然，如果两个模型在(q_L，q_U)层的策略树有相同的解，并且叶子节点更新后的信度差距不大于 ε，它们是(ε，q_L，q_U)-BE 模型. 当 $\varepsilon=0$ 时，(ε，q_L，q_U)-

BE 是精确的 BE 模型；如果部分策略树是对称的，即每个分支深度一致 $q_L=q_U$，(ε,q_L,q_U)-BE 等同于 $(\varepsilon\text{-}d)$-BE 模型. 因此 (ε,q_L,q_U)-BE 模型给出了一个具有普遍意义的近似 BE 概念，既适用于对称结构的策略树，也适用于不对称结构.

可以看出算法 ε-BE-I 不同于 $(\varepsilon\text{-}d)$-BE，当 $D_{KL}[b_{m_{j,l-1}}^{q_r,k}\|b_{\hat{m}_{j,l-1}}^{q_r,k}]$ 小于 ε 时，可以中止分支比较，使之不必进行进一步的对比操作. 但是即使 $D_{KL}[b_{m_{j,l-1}}^{q_r,k}\|b_{\hat{m}_{j,l-1}}^{q_r,k}]$ 是一个非常小的数，$h_{m_{j,l-1}}^{q_r,k}$ 和 $h_{\hat{m}_{j,l-1}}^{q_r,k}$ 之后的策略树（分支深度从 q_r+1 到 T）也不一定是一致的，因而 ε-BE-I 可以比 $(\varepsilon\text{-}d)$-BE 聚类更多的近似 BE 模型.

4.6　算法理论性能比较

从算法复杂度和误差区间两个方面对比 4 种方法.

4.6.1　算法复杂度

对于精确的 BE 方法来说，在 t 时刻，智能体 j 有 $|M_j^0|(|A_j||\Omega_j|)^t$ 个模型，其中 $|M_j^0|$ 是初始模型个数. 对于有 $N+1$ 个智能体的 I-DID 来说，除去主体智能体，其他智能体有 N 个，各个智能体之间互相嵌套建模，假设每层的模型数不超过 $|M|$，则第 l 层求解模型的复杂度为 $o((N|M|)^l)$.

DMU 方法每层最多需要求解 $o((N|\hat{M}^*|)^l)$ 数量级的模型，$\hat{M}^*$ 是所有层里最大的最小模型集. 一般来说 $|\hat{M}^*|\ll|M|$，所以 DMU 方法有效降低了模型求解的复杂度.

对于 $(\varepsilon\text{-}d)$-BE 算法来说，根据式(3)，判定两个候选模型是否是近似 BE 的算法复杂性取决于比较两个 d 层策略树的复杂度，它与遍历策略树产生的需要比较的分支数目成正比. 由于 d 层策略树的叶子节点最多有 $|\Omega_j|^d$ 个，识别模型是否是 $(\varepsilon\text{-}d)$-BE 的算法复杂度为 $o(|M_{j,l-1}|^2|\Omega_j|^d)$，这里 $|M_{j,l-1}|$ 表示模型节点中候选模型的数目. 假设 $|M_{j,l-1}|=2$，则每个时间片每组被比较的模型最多有 $2|\Omega_j|^d$ 个分支. 在算法中，每个分支都需要占用存储空间，$(\varepsilon\text{-}d)$-BE 方法不用存储包括 $|\Omega_j|^{T-1}$ 个分支的全部策略树，因而当 $d\ll T$ 时，大大节省了内存，$(\varepsilon\text{-}d)$-BE 方法比 DMU 方法的复杂度低.

对于 ε-BE-I 算法来说，识别一对模型是否 (ε,q_L,q_U)-BE 主要是执行策略树的分支对比，算法复杂度与对比操作执行次数成正比. 算法在从根节点遍历 q_U 层策略树的过程中进行剪枝操作，这可能产生一个不对称的部分策略树，但叶子节点的数目 $M\ll|\Omega_j|^{q_U-1}$. 同时，需要对比的更新后的信度数量也被 M 所界定，因此算法复杂度为 $o(2|M_{j,l-1}|^2M)$. 此外，注意到 ε-BE-I 在模型求解传播信度的过程中涉及的信度更新计算的代价比 $(\varepsilon\text{-}d)$-BE 算法小. ε-BE-I 方法的最长分支长度是 q_U，每个时间片只需存储 $2M$ 个分支，因此 ε-BE-I 比 $(\varepsilon\text{-}d)$-BE 更加节省存储空间.

4.6.2　误差区间

精确的 BE 方法没有误差.

当 $\mu=0$ 时，DMU 方法也是精确求解，没有误差；当 $\mu>0$ 时，误差产生. 最坏的情况是除了选中的 K 个模型，其他所有的模型都与这 K 个行为不等价. 假设导致最坏情况的模型 $m_{j,l-1}$ 在智能体 j 中，并与已经求解的模型 $m'_{j,l-1}$ 相关. 设 α 和 α'' 分别是模型 $m_{j,l-1}$ 和 $m'_{j,l-1}$ 精确求解得出的策略树，而 α' 是模型 $m'_{j,l-1}$ 近似求解得出的策略树，$b_{j,l-1}$ 表示 $m_{j,l-1}$ 的信度，$b'_{j,l-1}$ 表示 $m'_{j,l-1}$ 的信度，则误差为

$$\begin{aligned}\rho&=|\alpha b_{j,l-1}-\alpha'' b_{j,l-1}|\\&=|\alpha b_{j,l-1}-\alpha'' b'_{j,l-1}+\alpha'' b'_{j,l-1}-\alpha'' b_{j,l-1}|\\&\le|\alpha b_{j,l-1}-\alpha b'_{j,l-1}+\alpha'' b'_{j,l-1}-\alpha'' b_{j,l-1}|\\&\quad(\alpha'' b'_{j,l-1}\ge\alpha b'_{j,l-1})\\&=|(\alpha-\alpha'')(b_{j,l-1}-b'_{j,l-1})|\\&\le|(\alpha-\alpha'')|_\infty\cdot|(b_{j,l-1}-b'_{j,l-1})|_1\\&\le(R_j^{\max}-R_j^{\min})T\cdot\mu\end{aligned}\tag{4}$$

这里 $R_j^{\max}$ 和 $R_j^{\min}$ 分别表示 j 的最大期待效益值和最小期待效益值.

当 $\varepsilon=0$ 时，$(\varepsilon\text{-}d)$-BE 方法是精确的 BE，也没有误差. 随着 ε 的增加，d 的值越来越小，与其他模型行为不等价的 $m_{j,l-1}$ 可能被错误的聚类在模型 $\hat{m}_{j,l-1}$ 类中. 最坏的情况是最终 $m_{j,l-1}$ 被选为代表模型. 设 α^T 和 $\hat{\alpha}^T$ 分别是模型 $m_{j,l-1}$ 和 $\hat{m}_{j,l-1}$ 求解出的完整的策略树，则误差为

$$\rho=|\alpha^{\mathrm{T}}b^0_{m_{j,l-1}}-\alpha^{\mathrm{T}}b^0_{\hat{m}_{j,l-1}}|.$$

如果两个模型 d 层策略树是相同的，则误差变为

$$\begin{aligned}\rho&=|\alpha^{T-d}b^d_{m_{j,l-1}}-\alpha^{T-d}b^d_{\hat{m}_{j,l-1}}|\\&=|\alpha^{T-d}b^d_{m_{j,l-1}}+\hat{\alpha}^{T-d}b^d_{m_{j,l-1}}-\hat{\alpha}^{T-d}b^d_{m_{j,l-1}}-\alpha^{T-d}b^d_{\hat{m}_{j,l-1}}|\\&\le|\alpha^{T-d}b^d_{m_{j,l-1}}+\hat{\alpha}^{T-d}b^d_{\hat{m}_{j,l-1}}-\hat{\alpha}^{T-d}b^d_{m_{j,l-1}}-\alpha^{T-d}b^d_{\hat{m}_{j,l-1}}|\\&\quad(\hat{\alpha}^{T-d}b^d_{\hat{m}_{j,l-1}}\ge\hat{\alpha}^{T-d}b^d_{m_{j,l-1}})\\&=|(\alpha^{T-d}-\hat{\alpha}^{T-d})(b^d_{m_{j,l-1}}-b^d_{\hat{m}_{j,l-1}})|\\&\le|(\alpha^{T-d}-\hat{\alpha}^{T-d})|_\infty\cdot|(b^d_{m_{j,l-1}}-b^d_{\hat{m}_{j,l-1}})|_1\\&\le|(\alpha^{T-d}-\hat{\alpha}^{T-d})|_\infty\cdot2D_{KL}(b^d_{m_{j,l-1}}\|b^d_{\hat{m}_{j,l-1}})\\&\le(R_j^{\max}-R_j^{\min})(T-d)\cdot2\varepsilon\end{aligned}\tag{5}$$

在 ε-BE-I 方法中，由于在长度为 q_L 时修剪策略树分支，误差则变为

$$\begin{aligned}\rho &= \left|\alpha^{T-q_L} b^{q_L}_{m_{j,l-1}} - \alpha^{T-q_L} b^{q_L}_{\hat{m}_{j,l-1}}\right| \\ &\le \left|(\alpha^{T-q_L} - \hat{\alpha}^{T-q_L})\right|_\infty \cdot 2D_{KL}\left(b^{q_L}_{m_{j,l-1}} \middle\| b^{q_L}_{\hat{m}_{j,l-1}}\right) \\ &\quad \text{(根据式(5))} \\ &\le (R_j^{\max} - R_j^{\min})(T-q_L)\cdot 2\varepsilon \end{aligned} \tag{6}$$

以上 4 种方法的算法复杂度和误差区间如表 1 所示，可以看出 ε-BE-I 的算法复杂度最小，但误差区间的上限最大.

表 1 方法的算法复杂度和误差区间对比

方法	算法复杂度	误差区间
精确 BE	$o((N\|M\|)^l)$	0
DMU	$o((N\|\hat{M}^*\|)^l)$	$\le (R_j^{\max}-R_j^{\min})T\cdot\mu$
(ε-d)-BE	$o(\|M_{j,l-1}\|^2\|\Omega_j\|^d)$	$\le (R_j^{\max}-R_j^{\min})(T-d)\cdot 2\varepsilon$
ε-BE-I	$o(2\|M_{j,l-1}\|^2 M)$	$\le (R_j^{\max}-R_j^{\min})(T-q_L)\cdot 2\varepsilon$

4.6.3 实验对比

以上阐述的 4 种基于 BE 的 I-DID 方法已在 5 个不同的多智能体决策问题上进行了详尽的测试和对比[48]. 总的来说，采用 BE 方法的确可以大大减少模型空间，近似算法可以求解具有多时间片的大规模决策问题. 表 2 对这些方法进行对比，列出方法在保证相同解的质量的前提下可达到的求解问题的最大规模(计划的时间片). 这也是目前 I-DID 求解方法一个总结性的实验对比. 从结果可以看出，I-DID 精确算法的确不能求解长时间片的多智能体决策问题. 因此，研究近似性的算法是求解 I-DID 的根本需求.

表 2 方法的求解规模对比

决策问题	老虎	机器维修	音乐会	洗钱	无人机
状态数	2	4	2	99	81
观测数	3	4	3	11	4
动作数	3	2	4	9	5
精确 BE	5	4	5	3	3
DMU	8	6	7	5	4
(ε-d)-BE	21	13	15	10	8
ε-BE-I	23	15	18	12	10

5 最优 K 模型选择

基于 BE 的 I-DID 求解方法已经得到了广泛的应用，也是目前 I-DID 的研究重点和主要方向. 但如第 4 节分析，精确 BE 方法的求解能力还是非常有限的，而近似 BE 算法往往需要复杂的参数调节(譬如 ε)，才能提高求解能力和性能.

由于 I-DID 求解算法必须在有限的模型空间中达到最好的求解效果，算法的主要目的就是选择一个合适的 j 的候选模型子集建立 I-DID. 因而本节拟提出一个合理的选择机制，譬如选择 K 个模型，使之有最大的行为覆盖率，从而达到最好的求解效果. 也就是说，在智能体 i 的求解质量不会受到大影响的情况下，尽可能选择那些能代表 j 行为的模型. 代表性行为即为在智能体的交互过程中频繁发生的行为. 鉴于此，本节将把模型选择问题转化为一个优化问题，而不是像近似 BE 算法那样简单选取模型子空间.

在正式建模模型子空间优化问题之前，首先介绍模型选择函数，然后据此提出相应的高效求解算法.

5.1 模型选择优化问题

模型子空间优化问题在于选取一个合适的模型子集(譬如 K 个模型)，使得这 K 个模型的集合能够尽可能地覆盖所有候选模型的代表性行为. 因此，如何衡量 K 个模型的行为代表性是一个首要问题. 这往往需要定义模型之间的行为相似度. 这里首先提出一个测量模型相似度的行为覆盖函数.

$\omega(m_j, m_j')$ 表示模型 m_j 和 m_j' 之间的相似度，也就是求解 m_j 和 m_j' 得到策略树的相似程度.

$$\omega(m_j, m_j') = \sum_{h^T_{m_j}\in\Gamma^T_{m_j}, h^T_{m_j'}\in\Gamma^T_{m_j'}} sim(h^T_{m_j}, h^T_{m_j'}) \tag{7}$$

这里 h 是求解模型所建立的策略树，$sim(h^T_{m_j}, h^T_{m_j'})$ 是计算每个时间片相同的观测值得到相同动作的数量.

紧接着，可以定义每个模型 m_j 被所选择 K 个模型的覆盖程度，即 $\sum_{m_j'\in M^K_{j,l-1}} \omega(m_j, m_j')$. 算法的目标是发现最优 K 个模型 $M^K_{j,l-1}$，能够最大程度地覆盖所有模型 $M_{j,l-1}$ 的行为. 据此，最优 K 模型选择问题可以用下面优化问题来描述.

已知：$M_{j,l-1}, K$

目标函数：

$$\max M^K_{j,l-1}\subseteq M_{j,l-1},\ |M^K_{j,l-1}| = K$$

$$\sigma(M^K_{j,l-1}) = \sum_{m_j\in M_{j,l-1}} \sum_{m_j'\in M^K_{j,l-1}} \omega(m_j, m_j') \tag{8}$$

显然这是一个复杂的单目标组合优化问题，也是一个 NP 问题.

定理 1. 最优 K 模型的选择问题是 NP 问题.

证明. 可以把式(8)转换为单位成本的最大预算覆盖问题[60]，给定实例 φ：集合 $S=\{S_1, S_2, \cdots,$

$S_m\}$. 假设单位成本为 C，领域元素为 $X=\{x_1, x_2, \cdots, x_n\}$，相应的权重为 $\{z_1, z_2, \cdots, z_n\}$，预算为 B. 最优 K 模型的选择实例 ω，设置 $K=\lfloor B/C \rfloor$，$\sigma(S')$ 对应 S' 覆盖元素的总权重. 如果 S' 是 ω 的最优 K 个模型，则 S' 在 φ 中有最大权重. 单位成本的最大预算覆盖问题已经证明是 NP 问题，因而最优模型选择问题也是 NP 问题.　　证毕.

虽然求解模型选择问题是相当困难的，但注意到选择函数 $\sigma(M_{j,l-1}^K)$ 是单调子模函数. 因此可以采用高效的算法来求解该问题.

5.2　求解算法

假设 ν 是一个有限集合，存在函数 $F:\nu\rightarrow\Re$，如果该函数满足边际递减效用，即为子模. 也就是说对所有 $B\subseteq\hat{B}\subseteq\nu$ 且 $s\notin B$，$F(B\cup s)-F(B)\geq F(\hat{B}\cup s)-F(\hat{B})$，$F(B\cup s)-F(B)$ 是 B 中增加元素 s 时 F 的边际增值. 子模函数的特征是增加元素到小集合 B 比到大集合 $\hat{B}$ 所增加的效用要多.

直觉上，$\sigma(M_{j,l-1}^K)$ 是单调的，因为随着候选模型集合的增大，模型集合对行为的覆盖程度必然增加. 假设 $K_1<K_2$，当增加一个新模型到集合 K_1 个模型时的增幅大于增加模型到集合 K_2 个模型. 这是由于新模型的行为可能已经被大集合覆盖，但没有被小集合覆盖. 这说明了 $\sigma(M_{j,l-1}^K)$ 也符合边际递减效益. 因此，模型选择函数的性质如定理 2 所述.

定理 2.　模型选择函数 $\sigma(M_{j,l-1}^K)$ 是单调子模函数.

单调子模函数的性质决定了可以选择贪婪算法来求解模型选择优化问题，如算法 2. 在算法 2 中，贪婪算法设置模型子集的初值为空集，计算每个模型覆盖的行为（见算法 2～4 步），逐步增加可引起边际覆盖增值最大的那个模型，直至 $|M_{j,l-1}^K|=K$（见算法 5～7 步）. 算法 2 可以得到接近 $1-1/e$ 的最优行为覆盖的 K 个模型的近似最优解.

由于贪婪算法检查每轮所有的候选模型（见算法第 6 步），算法的时间复杂度为 $o(K|M_{j,l-1}^K|B(\sigma(\cdot)))$，这里 $B(\sigma(\cdot))$ 是计算模型覆盖率的运行时间.

把模型的选择问题转化为组合优化问题是最优 K 模型的主要特点. 最优 K 模型选择算法从整个模型空间来考虑添加有限模型的适用性，而不是像以前方法那样只通过两两模型之间的简单比较而决定是否添加新的模型. 同时，根据已知的计算资源往往可以确定最大的 I-DID 模型规模，从而确定最大的 K 值.

算法 2.　K 个模型贪婪选择算法.

输入：模型全集，K

输出：选中的 K 个模型集合

1. 函数模型选择（$M_{j,l-1}^K$，K）
2. $M_{j,l-1}^K=\varnothing$
3. FOR $m_j\in M_{j,l-1}^K$ DO
4. 计算 $\sigma(m_j)$
5. FOR $Ite=1$ to K DO
6. $m_j\leftarrow\arg\max_{m_j}[\sigma(M_{j,l-1}^K\cup m_j)-\sigma(M_{j,l-1}^K)]$
7. $M_{j,l-1}^K\leftarrow M_{j,l-1}^K\cup m_j$
8. 返回 $M_{j,l-1}^K$

5.3　实验结果

我们用 5 个经典问题（见表 2）验证了最优 K 模型算法，证实其性能超过目前占主导地位的 BE 方法. 譬如，该算法可以求解超过 12 个时间片的无人驾驶机问题. 本文不再详细阐述这些实验结果，侧重于展示该算法在一个新应用领域（计算机游戏）上的求解性能.

计算机游戏，特别是实时战略游戏，已经成为人工智能技术的一个主要实际应用领域. 游戏的复杂性和真实性有助于验证理论算法的实用性. 目前，把 I-DID 拓展到计算机游戏应用也是验证其实际应用能力的一个主要实验场所. 本文将针对星际争霸的实时战略游戏，测试最优 K 模型算法在攻防决策的应用能力.

在一个复杂的星际争霸游戏中，设计智能的非人类游戏玩家是游戏智能研究和开发的主要目标. 实验应用 I-DID 对非人类玩家进行建模，其他人类玩家作为其他智能体. 目前，实验主要集中在双方的攻防决策当中. 如图 16 所示，在星际争霸中一个复杂的攻防场景，双方均需要通过判断对方的行为，做出准确的攻防动作.

图 16　星际争霸游戏中的攻防场景

在实验中,假设游戏的场景划分为 16 个区域(即状态数),每个玩家可以选择 3 个动作(进攻、防守和等待),同时可以接收到 4 个观测值(不同数目的敌人在附近).在一个特定的状态,玩家动作的报酬由双方的伤亡来决定.据此,该实验为非人类游戏玩家建立了一个 I-DID 模型,帮助其做出攻防的决策;同时,假设人类游戏玩家将不对非人类游戏玩家的决策进行预测,那么其决策模型将是一个简单的 DID 模型,嵌套在非人类游戏玩家的 I-DID 模型之中.

图 17 对比了最优 K 模型算法与其他算法(DMU 和 ε-BE-I)求解 I-DID(时间片=8)模型的性能.由于精确的 BE 方法不能求解这个复杂的决策问题,因此图中没有给出结果.

如图 17 所示,随着模型选择个数的增加,最优 K 模型算法逐渐接近精确的 DMU 求解效果(即最优 I-DID 求解结果).当模型个数达到一定的数量,其性能远优于 ε-BE-I 算法.

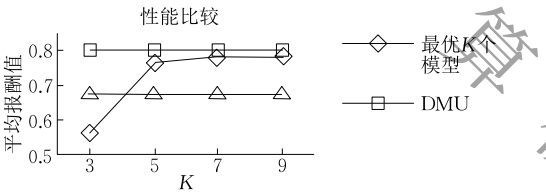

图 17 方法的性能对比

表 3 表明最优 K 模型算法也同时提高了 I-DID 的求解速度.因此,最优 K 模型算法将成为另外一个高效的 I-DID 求解技术.

表 3 方法的求解速度对比

比较的方法	时间/ms
DMU	21000
最优 K 模型 $K=3$	8000
最优 K 模型 $K=5$	15000
最优 K 模型 $K=7$	16180
最优 K 模型 $K=9$	16800
ε-BE-I	17000

6 总结与展望

本文详细阐述了 I-DID 的建模过程,归纳和对比了基于行为等价原理的模型求解方法,并提出了最优 K 模型的求解技术.该求解方法有别于目前主导的行为等价方法,有效地把模型选择问题转化成组合优化问题.这为 I-DID 求解方法的深入研究提供了一个重要的思路.

I-DID 建模多智能体序贯决策问题依赖于两部分的模型.一方面是对主体智能体本身决策过程的建模,建模过程必须考虑其他智能体的决策模型,以便求解 I-DID 模型;另一方面是对其他智能体序贯决策过程的建模.目前 I-DID 的研究工作首先假设主体智能体可以建立描述其他智能体决策过程的模型,然后求解其模型以获得其他智能体的决策行为.

众所周知,依靠问题领域专家建立一个精确的决策模型并不是一件简单的工作,特别是在不确定环境下的决策问题中,很难确定决策模型的部分参数.同时,由于主体智能体并不知道其他智能体的真实模型,往往需要在理论上假设存在数目庞大的其他智能体的候选决策模型.这造成了 I-DID 模型求解的困难及其在实际应用中的障碍.大部分的 I-DID 研究工作集中在如何压缩其他智能体的候选模型空间上,从而降低 I-DID 求解的复杂度.近年来 I-DID 求解技术虽然有了很大的进展,但仍然未能满足求解复杂多智能体决策问题的需求.

鉴于此,今后 I-DID 的可能的发展方向有:

(1)基于数据驱动的建模和求解方法

由于只有其他智能体的动作才能对主体智能体的决策产生影响,因此,考虑从历史数据中学习出其他智能体的决策行为,无需建立描述其决策过程的模型,这样将避免繁杂的 I-DID 建模过程.这个思路对提高 I-DID 的建模和求解技术有很大的推动作用,同时也为其他多智能体决策模型的求解提供一个崭新的技术思路.

(2)扩展 I-DID 到多智能体环境设置

目前 I-DID 只能建立和求解 2～3 个智能体的模型,无法解决大规模多智能体决策问题.智能体个数的增加要求 I-DID 对模型的表示方法和求解算法进行全面的改进.比如如何利用智能体之间的关系简化模型的建立就是一个值得研究的问题.特别是在大规模多智能体协同决策的问题中,可以通过挖掘智能体之间的关系,譬如信任/随从关系,降低建模及其求解的复杂度[49].

(3)I-DID 的应用研究

创立具有实际意义的问题,拓展 I-DID 的应用领域仍然是 I-DID 模型研究的难点.这主要是因为 I-DID 的建模和求解计算量大,实际问题的智能体数目繁多、问题复杂,难以用模型表征.因而 I-DID 的应用研究也是非常重要的工作,充满了挑战.

参 考 文 献

[1] Farinelli A, Rogers A, Jennings N R. Agent-based decentralised coordination for sensor networks using the max-sum

algorithm. Journal of Autonomous Agents and Multi-Agent Systems, 2014, 28(3): 337-380

[2] Witwicki S J, Durfee E H. Towards a unifying characterization for quantifying weak coupling in Dec-POMDP//Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS). Taipei, China, 2011: 29-36

[3] Zhu Man-Ling, Jin Zhi. Approach for evaluating the trustworthiness of service Agent. Journal of Software, 2011, 22(11): 2593-2609(in Chinese)
(朱曼玲，金芝. 一种服务 Agent 的可信性评估方法. 软件学报, 2011, 22(11): 2593-2609)

[4] Gmytrasiewicz P J, Doshi P. A framework for sequential planning in multi-Agent settings. Journal of Artificial Intelligence Research, 2005, 24: 49-79

[5] Zeng Y, Doshi P, Chen Q. Approximate solutions of interactive dynamic influence diagrams using model clustering//Proceedings of the 22nd International Conference on Association for the Advancement of Artificial Intelligence (AAAI). Arlington, Virginia, 2007: 782-787

[6] Brandenburger A M, Nalebuff B. The right game: Use game theory to shape strategy. Harvard Business Review, 1995, 76(7): 57-71

[7] Rathnasabapathy B, Doshi P, Piotr J, Gmytrasiewicz P J. Exact solutions of interactive POMDPs using behavioral equivalence//Proceedings of the 5th Internationl Conference on Autonomous Agents and Multiagents Systems Conference (AAMAS). Hakodate, Japan, 2006: 1025-1032

[8] Howard R A, Matheson J E. Influence diagrams. Principles and Applications of Decision Analysis. USA, Menlo Park: Strategic Decisions Group, 1984

[9] Blum B, Shelton C R, Koller D. A continuation method for Nash equilibria in structured games. Journal of Artificial Intelligence Research, 2006, 25: 457-502

[10] Jiang Xin, Liu Xin-Jian, Chen Chao. Modeling of military decision-making based on multi-Agent influence diagrams and games. Systems Engineering and Electronics, 2011, 33(7): 1312-1319(in Chinese)
(姜鑫，刘新建，陈超. 基于多主体影响图及博弈论的军事决策建模. 系统工程与电子技术, 2011, 33(7): 1312-1319)

[11] Gal K, Pfeffer A. Networks of influence diagrams: A formalism for representing agents' beliefs and decision-making processes. Journal of Artificial Intelligence Research, 2008, 33: 109-147

[12] Nair R, et al. Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings//Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI). Acapulco, Mexico, 2003: 705-711

[13] Akshat K, Hala M, Shlomo Z. Dual formulations for optimizing Dec-POMDP controllers//Proceedings of the 27th International Conference on Automated Planning and Scheduling (ICAPS). London, UK, 2016: 202-210

[14] Kaelbling L P, Littman M L, Cassandra A R. Planning and acting in partially observable stochastic domains. Artificial Intelligence, 1998, 101(1-2): 99-134

[15] Seo J, et al. Training beam sequence design for millimeter-wave MIMO systems: A POMDP framework. IEEE Transactions on Signal Processing, 2015, 64(5): 1228-1242

[16] Grady D K, Moll M, Kavraki L E. Extending the applicability of POMDP solutions to robotic tasks. IEEE Transactions on Robotics, 2015, 31(4): 948-961

[17] Wu F, Zilberstein S, Jennings N R. Monte-Carlo expectation maximization for decentralized POMDPs//Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI). Beijing, China, 2013: 397-403

[18] Zhang Ying-Xiao, et al. Decentralized PODMP-based cognitive radio network spectrum access algorithm. Information and Electronic Engineering, 2010, 8(6): 720-725(in Chinese)
(张迎晓等. 基于 DEC-POMDP 的认知无线电网络频谱接入算法. 信息与电子工程, 2010, 8(6): 720-725)

[19] Liu Hai-Tao, et al. Research on decentralized communication decision in the multi-agent robotic system. Robot, 2007, 29(6): 540-545(in Chinese)
(刘海涛等. 多智能体机器人系统分散式通信决策研究. 机器人, 2007, 29(6): 540-545)

[20] Marecki J, et al. Not all agents are equal: Scaling up distributed POMDPs for agent networks//Proceedings of the 7th International Conference on Autonomous Agents and Multiagents Systems Conference (AAMAS). Estoril, Portugal, 2008: 485-492

[21] Velagapudi P, et al. Distributed model shaping for scaling to decentralized POMDPs with hundreds of agents//Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS). Taipei, China, 2011: 955-962

[22] Varakantham P, et al. Exploiting coordination locales in distributed POMDPs via social model shaping//Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS). Thessaloniki, Greece, 2009: 313-320

[23] Varakantham P, et al. Prioritized shaping of models for solving Dec-POMDPs//Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS). Valencia, Spain, 2012: 1269-1270

[24] Okamoto S, Hazon N, Sycara K. Solving non-zero sum multiagent network flow security games with attack costs//Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS). Valencia, Spain, 2012: 879-888

[25] Lewis M, Sycara K P. Network-centric control for multirobot teams in urban search and rescue//Proceedings of the 44th Hawaii International Conference on Systems Sciences (HICSS). Kauai, USA, 2011: 1-10

[26] Khandelwal P, Stone P. Multi-robot human guidance using topological graphs//Proceedings of the 28th International Conference on Association for the Advancement of Artificial Intelligence (AAAI). Québec City, Canada, 2014: 65-72

[27] Alan A, et al. Mixed-initiative electricity tariff switching for dynamic environments//Proceedings of the 13th International

Conference on Autonomous Agents and Multi-Agent Systems (AAMAS). Paris, France, 2014: 965-972

[28] Fave F M D, et al. Security games in the field: An initial study on a transit system//Proceedings of the 13th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS). Paris, France, 2014: 1363-1364

[29] Ford B, et al. PAWS: Adaptive game-theoretic patrolling for wildlife protection (demonstration)//Proceedings of the 13th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS). Paris, France, 2014: 1641-1642

[30] Sless L, et al. Forming coalitions and facilitating relationships for completing tasks in social networks//Proceedings of the 13th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS). Paris, France, 2014: 261-268

[31] Jiang Y, Zhou Y, Wang W. Task allocation for undependable multiagent systems in social networks. IEEE Transactions on Parallel and Distributed Systems, 2013, 24(8): 1671-1681

[32] Jiao W. Multi-agent cooperation via reasoning about the behavior of others. Computational Intelligence, 2010, 26(1): 57-83

[33] Liu Chun-Yang, et al. Application of multi-agent reinforcement learning in robot soccer. Acta Electronica Sinica, 2010, 38(8): 1958-1962(in Chinese)
(刘春阳等. 多智能体强化学习在足球机器人中的研究与应用. 电子学报, 2010, 38(8): 1958-1962)

[34] Ma Guang-Fu, Mei Jie. Coordinated tracking for nonlinear multi-agent systems under directed networks. Control and Decision, 2011, 26(12): 1861-1871(in Chinese)
(马广富, 梅杰. 有向网络下非线性多智能体系统的协调跟踪. 控制与决策, 2011, 26(12): 1861-1871)

[35] Chen Y M, Wu W Y. Cooperative electronic attack for groups of unmanned air vehicles based on multi-agent simulation and evaluation. International Journal of Computer Science Issues, 2012, 9(2): 1-6

[36] Jiang Wei-Jin, Xu Yu-Hui, Zhang Lian-Mei. Research on knowledge reuse dynamic evolvement model based on multi-agent system component. Systems Engineering-Theory & Practice, 2013, 33(10): 2663-2673(in Chinese)
(蒋伟进, 许宇晖, 张莲梅. 基于 MAS 构件技术的复杂知识复用动态演化模型研究. 系统工程理论与实践, 2013, 33(10): 2663-2673)

[37] Zhong Wei-Cai, et al. Combinatorial optimization using multi-agent evolutionary algorithm. Chinese Journal of Computers, 2004, 27(10): 1341-1353(in Chinese)
(钟伟才等. 组合优化多智能体进化算法. 计算机学报, 2004, 27(10): 1341-1353)

[38] Fan Bo, Pan Quan, Zhang Hong-Cai. A multi-agent coordination method based on Markov game and application to Robot Soccer. Robot, 2005, 27(1): 46-51(in Chinese)
(范波, 潘泉, 张洪才. 基于 Markov 对策的多智能体协调方法及其在 Robot Soccer 中的应用. 机器人, 2005, 27(1): 46-51)

[39] Zhang Qing-Jie, et al. Non-cooperative solving method of multi-UAV rendezvous problem in complex network. Journal of Southeast University (Natural Science Edition), 2013, 43(Supplement I): 32-37(in Chinese)
(张庆杰等. 复杂网络条件下多 UAV 集结问题非合作求解方法. 东南大学学报(自然科学版), 2013, 43(增刊 I): 32-37)

[40] Delgado J. Emergence of social conventions in complex networks. Artificial Intelligence, 2002, 141(1): 171-185

[41] Gaston M E, Desjardins M. Social network structures and their impact on multi-agent system dynamics//Proceedings of the 18th International Conference on Association for the Advancement of Artificial Intelligence (AAAI). Arlington, USA, 2005: 32- 37

[42] Xu Yang, et al. Effects of complex network characters on the coordination control of large-scale multi-agent system. Journal of Software, 2012, 23(11): 2971-2986(in Chinese)
(徐杨等. 复杂网络特性对大规模多智能体协同控制的影响. 软件学报, 2012, 23(11): 2971-2986)

[43] Li Xiao, Yang Hong-Yong. Complex network characteristics and consensus of multi-agent systems. Complex System and Complexity Science, 2011, 8(3): 38-43(in Chinese)
(李晓, 杨洪勇. 复杂网络特性与多智能体的一致性. 复杂系统与复杂性科学, 2011, 8(3): 38-43)

[44] Ng B, et al. Towards applying interactive POMDPs to real-world adversary modeling//Proceedings of the National Conference on Innovative Applications of Artificial Intelligence (IAAI). Atlanta, USA, 2010: 1814-1820

[45] Tian L, et al. Modeling and algorithms for multiagent communication through interactive dynamic influence diagrams. Applied Artificial Intelligence, 2016, 30(4): 352-377

[46] Doshi P, Zeng Y. Improved approximation of interactive dynamic influence diagrams using discriminative model updates//Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS). Budapest, Hungary, 2009: 907-914

[47] Zeng Y, et al. Utilizing partial policies for identifying equivalence of behavioral models//Proceedings of the 26th International Conference on Association for the Advancement of Artificial Intelligence (AAAI). San Francisco, USA, 2011: 1083-1088

[48] Zeng Y, Doshi P. Exploiting model equivalences for solving interactive dynamic influence diagrams. Journal of Artificial Intelligence Research, 2012, 43: 211-255

[49] Pan Ying-Hui, Luo Jian, Zeng Yi-Feng. The exploration on modeling methods for interactive multi-Agent dynamic influence diagrams. Journal of Xiamen University (Natural Science), 2012, 51(6): 985-990(in Chinese)
(潘颖慧, 罗键, 曾一锋. 多 Agent 交互式动态影响图的建模方法. 厦门大学学报(自然科学版), 2012, 51(6): 985-990)

[50] Chen Y, Doshi P, Zeng Y. Iterative online planning in multiagent settings with limited model spaces and PAC guarantees//Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS). Istanbul, Turkey, 2015: 1161-1169

[51] Conroy R, et al. A value equivalence approach for solving interactive dynamic influence diagrams//Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS). Singapore, 2016: 1162-1170

[52] Chandrasekaran M, et al. Team behavior in interactive dynamic influence diagrams with applications to ad hoc teams//Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS). Paris, France, 2014: 1559-1560

[53] Albrecht S V, Crandall J W, Ramamoorthy S. Belief and truth in hypothesised behaviours. Artificial Intelligence Journal, 2016, 235: 63-94

[54] Shachter R D. Evaluating influence diagrams. Operations Research, 1986, 34(6): 871-882

[55] Tatman J, Shachter R. Dynamic programming and influence diagrams. IEEE Transactions on Systems, Man and Cybernetics, 1990, 20(2): 365-379

[56] Doshi P, Zeng Y, Chen Q. Graphical models for interactive POMDPs: Representations and solutions. Journal of Autonomous Agents and Multi-Agent Systems, 2009, 18(3): 376-416

[57] Pynadath D, Marsella S. Minimal mental models//Proceedings of the 22nd International Conference on Association for the Advancement of Artificial Intelligence. Arlington, Virginia, 2007: 1038-1044

[58] Boyen X, Koller D. Tractable inference for complex stochastic processes//Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence. Madison, USA, 1998: 33-42

[59] Zeng Y, et al. Improved use of partial policies for identifying behavioral equivalence//Proceedings of the 11th International Conference on Autonomous Agents and Multi-Agent Systems. Taipei, China, 2012: 1015-1022

[60] Samir K, Anna M, Joseph S. The budgeted maximum coverage problem. Information Processing Letters, 1999, 70(1): 39-45

附录 X.

附表 1　ID 观察函数 O_i 的条件概率表

S	GL	GR
TL	0.85	0.15
TR	0.15	0.85

附表 2　ID 状态转移函数 S 的条件概率表

TL	TR
0.5	0.5

附表 3　ID 值函数 R_i 的收益表

A	TL	TR
OL	−100	10
OR	10	−100
L	−1	−1

附表 4　DID 观察函数 O_i^{t+1} 的条件概率表

A^t	S^{t+1}	GL	GR
OL	*	0.5	0.5
OR	*	0.5	0.5
L	TL	0.85	0.15
L	TR	0.15	0.85

附表 5　DID 状态转移函数 S^{t+1} 的条件概率表

A^t	S^t	TL	TR
OL	*	0.5	0.5
OR	*	0.5	0.5
L	TL	1.0	0
L	TR	0	1.0

附表 6　I-DID 状态转移函数 S^{t+1} 的条件概率表

$\{A_i^t, A_j^t\}$	S_i^t	TL	TR
$\{OL, *\}$	*	0.5	0.5
$\{OR, *\}$	*	0.5	0.5
$\{*, OL\}$	*	0.5	0.5
$\{*, OR\}$	*	0.5	0.5
$\{L, L\}$	TL	1.0	0
$\{L, L\}$	TR	0	1.0

附表 7　I-DID 观察函数 O_i^{t+1} 的条件概率表

$\{A_i^t, A_j^t\}$	S_i^{t+1}	$\{GL, CL\}$	$\{GL, CR\}$	$\{GL, S\}$	$\{GR, CL\}$	$\{GR, CR\}$	$\{GR, S\}$
$\{L, L\}$	TL	0.85 * 0.05	0.85 * 0.05	0.85 * 0.9	0.15 * 0.05	0.15 * 0.05	0.15 * 0.9
$\{L, L\}$	TR	0.15 * 0.05	0.15 * 0.05	0.15 * 0.9	0.85 * 0.05	0.85 * 0.05	0.85 * 0.9
$\{L, OL\}$	TL	0.85 * 0.9	0.85 * 0.05	0.85 * 0.05	0.15 * 0.9	0.15 * 0.05	0.15 * 0.05
$\{L, OL\}$	TR	0.15 * 0.9	0.15 * 0.05	0.15 * 0.05	0.85 * 0.9	0.85 * 0.05	0.85 * 0.05
$\{L, OR\}$	TL	0.85 * 0.05	0.85 * 0.9	0.85 * 0.05	0.15 * 0.05	0.15 * 0.9	0.15 * 0.05
$\{L, OR\}$	TR	0.15 * 0.05	0.15 * 0.9	0.15 * 0.05	0.85 * 0.05	0.85 * 0.9	0.85 * 0.05
$\{OL, *\}$	*	1/6	1/6	1/6	1/6	1/6	1/6
$\{OR, *\}$	*	1/6	1/6	1/6	1/6	1/6	1/6

附表 8　I-DID 值函数 R_i 的条件概率表

$\{A_i^t, A_j^t\}$	TL	TR
$\{OR, OR\}$	10	−100
$\{OL, OL\}$	−100	10
$\{OR, OL\}$	10	−100
$\{OL, OR\}$	−100	10
$\{L, L\}$	−1	−1
$\{L, OR\}$	−1	−1
$\{OR, L\}$	10	−100
$\{L, OL\}$	−1	−1
$\{OL, L\}$	−100	10

PAN Ying-Hui, born in 1981, Ph.D., associate professor. Her current research interests include multiagent reasoning and decision making, artificial intelligence.

ZENG Yi-Feng, born in 1976, Ph. D., professor. His current research interests include multiagent decision making, artificial intelligence, social networks, and computer games.

Background

This work addresses an important research issue in the areas of multiagent systems particularly on the topic of sequential multiagent decision making. Over the last ten years, interactive dynamic influence diagrams have played an important role in the multiagent planning research. This could be observed from around a dozen of research articles in the flagship AI conferences like AAMAS, AAAI and IJCAI. Most of the papers are written by the authors in this article, and have attracted much attention in the agent community.

The I-DID research mainly focuses on the model development and algorithms that are based on the behaviorally equivalent principle. The algorithms have successfully solved a set of multiagent decision making problems and moved towards solving practical applications. This article reviews the most recent development of I-DID research and proposes a freshly new technique. The new algorithm provides another view on solving I-DIDs and expects to elicit new I-DID research. This article is the first time of formally presenting I-DID models in the research community of China and provides a new research angle to Chinese researchers in addressing agent-planning problems.

This research is supported by the NSFC projects (No.61375070, No.61562033, No.61772442 and No.71361011) and summarizes the recent work development in the projects. Yinghui would like to thank the projects (No.16GJ20 and No.20171BAB202022) from Jiangxi Province, China.