

# 基于人工智能的多 AGENT 协同辩证逻辑推理方法

黄海

桂起权

**摘要:** 为消除 AGENT 内部信念冲突, 实现多 AGENT 的协同运行, 利用人工智能技术, 设计多 AGENT 协同的论辩式辩证逻辑推理方法。根据 AGENT 的组成结构以及多 AGENT 之间的通信方式, 建立多 AGENT 协同模型。在该模型下, 利用人工智能技术, 求解多 AGENT 协同任务, 消解 AGENT 间冲突。通过计算识别出微小变化会导致系统行为重大变化的关键变量, 找到推理依据, 在推理论据和规则的支持下, 实现多 AGENT 协同的论辩式辩证逻辑推理。实验结果表明, 设计论辩式辩证逻辑推理方法的加速率提高了 1.4, 多 AGENT 之间的协同程度提高了 4.12, 同时抵制恶意节点的成功率提升了 12.6%, 具有更好的应用效果。

**关键词:** 人工智能; 多 AGENT 协同; 论辩式辩证逻辑; 逻辑推理

**中图分类号:** B81

**文献标识码:** A

## 1 引言

AGENT 属于人工智能中的一个重要概念, 是一种具有自主、可移动、自适应性和智慧性的实体, 能够代表使用者或其他程序, 以积极的服务形式来执行一系列任务。AGENT 由环境、接收输入和构成输出的感测器三部分组成, 智能 AGENT 可以连续地执行三个任务, 分别为在环境中感知动态条件、执行对环境状况的影响、通过推理了解感知信息并作出推理和决策。([16]) AGENT 拥有自己的计算资源和行为控制机制, 可以通过自身的内部状况和对周围环境的感知来决定和控制自身的行为, 而不需要外部的直接操控。同时, AGENT 还掌握了其他 AGENT 的相关知识和信息, 能够与其他 AGENT 进行交互、协作和合作, 并使用行动者之间特定的通信语言。

收稿日期: 2022-09-21

作者信息: 黄海 许昌学院马克思主义学院  
同济大学移动源后处理研究院  
15022799968@163.com

桂起权 武汉大学哲学院  
qqgui@sina.com

多 AGENT 是指一组分布于逻辑或物理位置的 AGENT，在网络中实现共享和协作，从而构成了一个有组织的体系，以共同完成一项复杂的工作。多 AGENT 的目的是把大型的、复杂的任务分解成小的、相互通讯的、协调的、易于管理的任务。多 AGENT 的协同运行需要论辩式的辩证逻辑的支持，这种逻辑为消除 AGENT 内部信念冲突提供了途径，并通过论辩互动来协调多个 AGENT 的冲突。首先，系统需要识别并建模各个 AGENT 内部的信念。这通常通过一种形式化的表示来完成，例如使用谓词逻辑。一旦信念被建模，系统就可以检测是否存在冲突的信念。这通常通过逻辑推理来完成，例如，如果一个 AGENT 同时持有两个互相矛盾的信念，那么就存在冲突。当检测到冲突时，AGENT 可以通过论辩互动来解决这些冲突。这通常涉及到使用一种或多种论辩策略，例如提出反驳、提供证据或者通过逻辑推理来解决冲突。基于论辩的结果，AGENT 可能需要修正其内部的信念。这可以通过丢弃某些信念、采纳新的信念或者通过某种形式的信念合并来完成。论辩式的辩证逻辑允许 AGENT 在运行时动态地适应新信息或变化的环境。这是因为它提供了一种机制，使 AGENT 能够通过互动来修正或更新其信念。论辩式的辩证逻辑自然地支持社会互动和协调，因为它涉及到 AGENT 之间的互动和沟通。这有助于解决多 AGENT 系统中的协同问题。

这里有必要对“论辩式的辩证逻辑”的概念做一个说明。在 2005 年桂林召开的辩证逻辑会议上，晋荣东的论文《Dialectic：辩证逻辑的新形态》提出（[12]），20 世纪下半叶国际逻辑界出现了某种将辩证法（Dialectic）向“论辩术”的本真意义作“现代回归”的新趋向，它可以合理地看作一种新型的辩证逻辑，即“论辩术的辩证逻辑”。从词源和语义解释上看，日本人采用“辩-证-法”这三个汉字来翻译 Dialectic 这个词是非常确切到位的：通过论辩、对话来证明真理的方法 = 辩-证-法 = 论辩术。该论文分析了目前中国的辩证逻辑研究中存在的问题，阐述了辩证逻辑的产生过程，以更好地理解辩证逻辑。通过研究亚里士多德等人的论辩术，解决矛盾和冲突。以中国马克思主义辩证逻辑为基础，吸收论辩术中的积极因素，不断提高辩证逻辑研究的水平。

继晋荣东之后，周红艳在万小龙指导下，其博士论文专门研究了雷歇尔的“论辩式的辩证逻辑”。（[17, 23]）其重视辩证过程的形式结构：即反馈式的循环结构。他糅合了古希腊苏格拉底和柏拉图原初的对话辩证法的论辩术特性和黑格尔辩证否定的螺旋式上升的循环特征。他糅合了古希腊苏格拉底和柏拉图原初的对话辩证法的论辩术特性和黑格尔辩证否定的螺旋式上升的循环特征。同时汲取了怀特海“过程哲学”的理念，并且用 20 世纪系统科学的“反馈、自动调节”的图示和科学语言重新加以表述。因此，这种向“Dialectic”的回归，就带有当今时代的新意。

整个辩证过程包含多个回合，每一回合都包含三个阶段：（1）初始假定；（2）反馈（反驳、回击）；（3）修正或调整（精致化、复杂化）。辩证过程通过初始一

反馈—修正—回归的循环使事物整体到达更高级的阶段。论辩辩证法完整地刻画了整个辩证过程的内容。言谈对话、论辩具有情境性,同时具有规范性。在对话双方的互动中,初始假定在回应挑战与反驳的同时得到修正和精致化。规范的论辩由辩护方、反对方和裁决者这三个部分组成。

现阶段多 AGENT 在执行与应用过程中,文献 [21] 提出了基于时序关系网络的论辩逻辑推理方法,利用卷积神经网络对输入的三幅图像进行空间特征提取,使用关系网络对四种空间特性进行了两两结合,再利用 LSTM 方法对输入的三幅图像进行时序特征提取,并将其时间序列特征与已有的空间特性结合起来,从而获得了时间-空间的融合特性,在此基础上,将前面三幅图像与各选项所获得的时间-空间融合特性进行进一步的推理,并利用 Softmax 函数对其进行评分,得出的结果为正确的。该方法的推理准确度较高。文献 [8] 提出了一种基于逻辑推理规则和冲突度量的证据网络推理方法。定义了两个逻辑推理规则,以表明父节点中某些前因的出现将导致子节点出现的子 BPA。然后通过父节点中的先验概率计算子 BPA 与子节点的发生概率。基于证据距离计算子 BPA 的支持度。并定义了新的推理规则—OR 规则,使推理过程更加直观。然而上述两种方法未考虑到不同 AGENT 之间的知识库不一致问题,影响了多 AGENT 的协同推理加速率。

为此,引入人工智能技术,设计了多 AGENT 协同的论辩式辩证逻辑推理方法。人工智能指的是使用计算机程序模仿人类的一些思维和行为的技术。论辩规范要求有论辩双方(辩护方,反驳方)+裁判,扩展到论辩各方( $n > 2$ ),多方,就是多 AGENT 情况。因为 AGENT 的工作原理和目的是符合人工智能技术的,所以人工智能技术与多 AGENT 技术之间的适配度较高,将人工智能技术应用到多 AGENT 协同论辩式辩证逻辑推理方法的设计中,以期能够提高多 AGENT 协同的论辩式辩证逻辑推理加速率以及应用价值。

## 2 多 AGENT 协同的论辩式辩证逻辑推理方法设计

### 2.1 建立多 AGENT 协同模型

作为智能体,AGENT 的作用决定了它的行为必须通过逻辑推理来实现。([14]) 例如,如果一个 AGENT 知道所有的红色方块必须放在蓝色方块上,并且它看到一个红色方块,那么它可以通过演绎推理得出这个红色方块需要被放在一个蓝色方块上的结论,或者,如果一个 AGENT 观察到每次它执行某个特定动作时,都会收到一个正面的奖励,它可能会归纳出这个动作通常是有益的。为了保证多个 AGENT 之间的互通性,多 AGENT 协同模型选择混合型 AGENT 作为组成个体,并在混合型 AGENT 的基础上对其进行改进,保证 AGENT 模型支持论辩协商,改装后的 AGENT 的内部结构如图 1 所示。

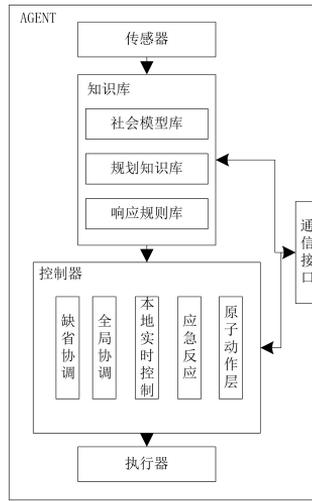


图 1: 改装 AGENT 内部结构图

图 1 所示的改装 AGENT 是通过感知周围环境变化，发现协作任务，采用远程过程调用的方式实现多 AGENT 之间的通信。客户端通过通信网络将报文传送给服务器，服务器收到数据后调用对应的 Server Stub，若存在返回值，则根据所需的传送和执行路径反向返回。([22]) 远程通信的核心内容是提高移动 AGENT 系统的互用性，以及新 IDL 的统一管理。AGENT 的整个生命周期内的信息是动态变化的，RTMAA 结构的 AGENT 的知识单位不仅要进行信息的存储，而且要注意信息的动态变化，保证信息的连贯性。支持辩论的多 AGENT 系统通常由参与者和仲裁方构成，其中仲裁方是独立于参与者 AGENT 的第三方，主要负责辩论过程的管理和控制，同时也担任结果的评判等责任。

## 2.2 利用人工智能技术求解多 AGENT 协同任务

在人工智能技术的支持下，利用多 AGENT 协同模型按照流程对协同任务进行求解，如图 2 所示。

定义多 AGENT 协同处理任务为，首先，要对任务进行分解，即把任务分成若干个区域，然后再把每个子任务组成一个集合，并将其分解为：

$$R = r_1, r_2, \dots, r_m \quad (1)$$

式中， $r_i$  为多 AGENT 执行的子任务。为了提高多 AGENT 的协同处理效率，可以把各个子任务分别指派给具有最大处理能力的分布式节点。([15]) 在问题知识库中，针对不同的任务，可以构建出各个辅助节点在问题知识库中的权值知识。假设协同处理任务需分配给  $n$  个 AGENT 节点，则可将人工智能技术分配任务的

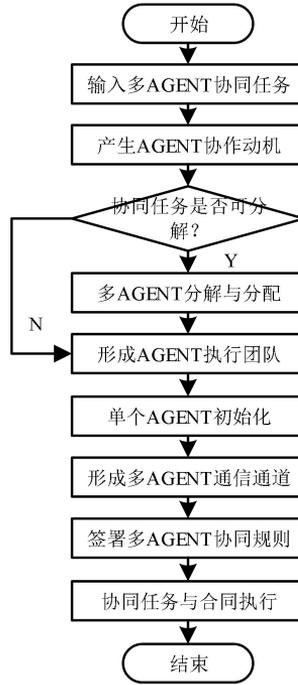


图 2: 多 AGENT 协同任务求解流程图

过程量化表示为:

$$y = \max \sum_{t=1}^m \sum_{j=1}^n \lambda_{ij} \tag{2}$$

式中,  $y$  为任务分配的目标函数, 变量  $m$  和  $n$  分别表示的是协同任务的分解数量和分配 AGENT 节点数量,  $\lambda_{ij}$  表示的是子任务  $r_i$  分配给第  $j$  个 AGENT 的适合度, 该指标的计算公式如下:

$$\lambda_{ij} = 1 - \sqrt{\frac{[1 - h_{\text{eff}}(\varphi_{ij})]^2 + [1 - h_{\text{cost}}(\phi_{ij})]^2}{2}} \tag{3}$$

式中, 变量  $h_{\text{eff}}(\cdot)$  和  $h_{\text{cost}}(\cdot)$  分别对应的是效能和费用的效用函数,  $\varphi_{ij}$  和  $\phi_{ij}$  表示的是第  $i$  个 AGENT 节点对任务  $j$  的求解能力和负载均衡度。在完成分解分配任务后, 在多 AGENT 模型中各成员的谈判是由一个代表谈判的结点组成的有限状态自动机, 由结点间的连线代表状态的转换, 而协商行为作为弧上的标签, 用以指明状态转换的原因。([19]) 节点上有叉形标志着谈判的终止。首先, 任务分配程序会将初步协商的提议发送出去, 再由双方进行反馈, 最终, 双方都可以根据各自的条件, 选择接受还是拒绝, 协商过程结束。在协商契约的约束下, 由形

成的 AGENT 组合完成协同子任务，最终通过多任务的融合，实现多 AGENT 协同任务的求解。

### 2.3 消解 AGENT 间冲突

多 AGENT 模型在协同任务处理过程中，由于各个 AGENT 之间的知识不完全、不相容，会引起 AGENT 之间的各种冲突，从而影响到 AGENT 之间的合作。为了降低 AGENT 间冲突对任务处理结果产生的影响，需对 AGENT 间的冲突进行消解处理。([10]) 将冲突动机组定义为  $C$ ，并判定同一动机是否存在于冲突动机组  $C$ ，如果判定的结论是“是”，那么用该动机生成目标，否则融合多个动机生成消除目标，利用 0-1 规划求解 AGENT 间冲突，冲突求解过程可以表示为：

$$g = \sum (\omega_1 \kappa_{\text{rep}} + \omega_2 \kappa_{\text{eco}}) e_i \quad (4)$$

式中， $\kappa_{\text{rep}}$  和  $\kappa_{\text{eco}}$  分别表示的是冲突消减过程中需要考虑的信誉度和经济因素， $\omega_1$  和  $\omega_2$  对应的是  $\kappa_{\text{rep}}$  和  $\kappa_{\text{eco}}$  的权重值，变量  $e_i$  表示的是冲突动机组  $C$  中的第  $i$  个动机。在上述冲突消解规则下，完成对 AGENT 间冲突的消解处理。

### 2.4 构建多 AGENT 协同的论辩式辩证逻辑推理论据

参与者 AGENT 都是围绕当前对话的临界元寻找攻击论据，而临界元集是由所有临界因子的元素构成。([13]) 因此，在目前的会话中，合法移动函数提供合法移动集的前提是能够明确对话中包含的临界因子和临界元。([18]) 临界依赖描述了临界因子中临界元的可替代关系，临界因子中任意临界元  $k$  被其临界依赖替换得到新的临界因子，可量化表示为：

$$\gamma_{\text{new}} = \delta \cup \left( \frac{\gamma}{\{k\}} \right) \quad (5)$$

式中， $\gamma_{\text{new}}$  和  $\gamma$  为替换前后的临界因子， $\delta$  代表临界依赖。依据临界因子将合法移动集进行聚类，得出临界元的计算结果。将论证本体中的知识与规则转化成 PROLOG 语言，并利用 PROLOG 的推理机制建立论证。([9]) 充分利用论证本体，找出最不容易被攻击的论据：询问论证本体中是否存在与  $W$  有关的论点，如果论证主体中没有  $W$  作为结论的论点，那么，运用 PROLOG 推理机和论证本体来构造一个以  $W$  为根结点的推理树，把推理树转化为论证，存储在论证本体中，接着，对论证本体进行查询，得到了论证。若本体中有多个论点是基于  $W$  而得出的，则将其组成为：

$$L = L_f \cup L_s \cup L_y \cup L_k \quad (6)$$

式中,  $L_f$ 、 $L_s$ 、 $L_y$  和  $L_k$  分别为可废止论据子集、似然论据子集、严格论据子集和可靠论据子集。因为建立的论据储存在本体中的, 并且 PROLOG 推理机的知识库中也包括了关于本体论的结论, 因此, 在进行推理时, 不需要再构造原有的论据, 从而避免论据的重复构建。

### 2.5 实现多 AGENT 协同的论辩式辩证逻辑推理

在分布式的辩论架构下, 每个 AGENT 仅占有整个辩论框架的一小部分, 要考虑到这些争论的重点论点的可信度, 就必须综合考虑一切可能的攻击论点和支撑论点。为了实现这一目标, 通常要求所有的主体都参加辩论。([20]) 与仅有 presentation Pro 和 presentation Opp 的辩论逻辑不同, 在多 AGENT 协作论辩推理中, 每个论辩回合中都会有多个 AGENT 参与辩论, 并且每个 AGENT 都没有固定的立场: 在  $r + 1$  轮中, AGENT 会提出攻击或间接攻击焦点论点的反论点。多 AGENT 协同的论辩式辩证逻辑推理的基本框架如图 3 所示。

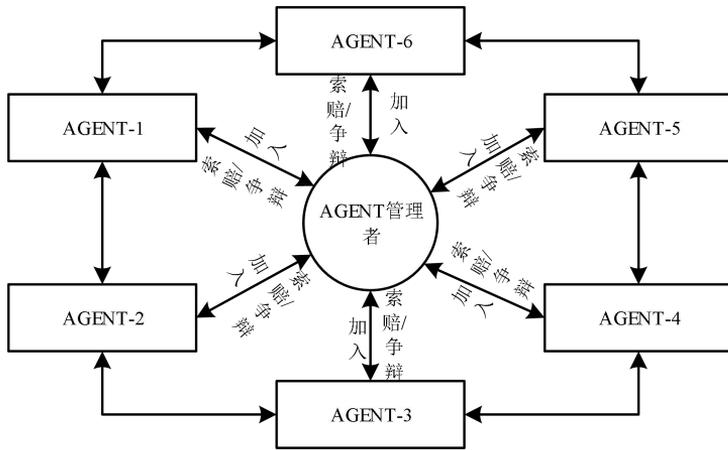


图 3: 多 AGENT 协同的论辩式辩证逻辑推理架构

为了保证多 AGENT 协同辩证推理的顺利进行, 必须建立一个管理者角色来协调不同行动者在论证博弈中的行为, 多 AGENT 协同的论辩式辩证逻辑推理管理者的形式定义如下:

$$G = \langle B_{dis}, f_{move}, B_{speech}, B_{stra}, f_{stru}, f_{rule}, f_{dist}, B_f, Y \rangle \tag{7}$$

式中  $B_{dis}$ 、 $B_{speech}$ 、 $B_{stra}$  和  $B_f$  分别表示的是论辩议题集合、管理论据博弈的言语行为集合、论据论辩过程的逻辑策略集合和执行逻辑函数活动集合。 $f_{move}$ 、 $f_{stru}$ 、 $f_{rule}$  和  $f_{dist}$  分别表示启动论辩博弈、检查论辩规则、改变参与者角色以及识别终止状态的函数,  $Y$  为论辩逻辑推理过程中的多方对话控制协议。([11]) 在管理者的

约束下,完成多 AGENT 协同任务的论辩式辩证逻辑推理。在多 AGENT 模式中,Referee 负责协调和控制整个辩论过程,Referee 会在初始化时将问题广播到已注册的 Participant,然后把竞技场的竞技者的申请资料保存到队列中。从竞技场的主人请求信息队列中选出一条消息,然后向所有的 Participant 进行广播,然后在一定的时间内接受 Participant 的挑战申请,将其保存到队列中,然后从队伍中选出一名挑战者,然后向播主发起挑战,并对其进行监督。发起播主-挑战者辩论,并监控这场辩论,参与辩论的所有 AGENT 都是 Participant。刚进入系统向 Referee 注册时,所有 Participant 的 role 属性值都为“spectator”。在收到 Referee 的议题后,Participant 试图根据议题建立一个主题,并为竞技场的拥有者提供一个请求,在建立好之后,将请求信息发送给 Referee。若选择了 Participant 作为竞技场的播主,role 值将变为“master”。如果 Participant 不是播主,且收到 Referee 发送了其它 Participant 申请播主的申请信息,则尝试构建反论据和用于申请挑战者的申请信息,如果构建成功就向 Referee 发送申请信息。如果选择了 Participant 作为一个挑战者,role 值将变为“challenger”。如果 Participant 的 role 值不是“spectator”,则表示 Participant 进行辩论。每次从 Referee 中接收到含有论点的内容时,会从其内容中提取论点,试图建立一个相反的论点,如果建立成功,就把它打包发给 Referee,反之,会沿着辩论线返回,去寻找另一方所用的相反论点,直到辩论一端发送承认失败的信息给 Referee 或者收到 Referee 发送辩论结束的消息。

在以往的研究中,研究者发现管理器模型的一些缺陷。首先是集中化,在管理器模型中,所有数据和命令通常都通过单个点(管理器),如果管理器不能足够快地处理信息,则可能会减慢整个系统的速度。其次是有限的带宽,管理器的通信带宽可能有限,当代理数量或数据量增加时,这可能成为瓶颈。还有资源限制,管理者可能受到计算资源的限制,从而难以有效地处理大量代理或复杂的任务。([3, 4])

在发生故障时,整个多 AGENT 系统也会面临一定的风险。一是单点故障,如果管理器发生故障,可能会破坏整个系统,因为所有代理都依赖它进行协调。二是数据丢失,管理器故障可能会导致数据丢失或数据传播不正确,从而导致代理采取不正确的操作。三是安全风险,如果管理器受到威胁,可能会对整个系统造成安全风险,因为它通常控制所有代理。([1, 7])

以上问题是管理者模型所面临的共通的问题,但是研究者也继续研究了一些解决这些问题的方法和措施。例如实施备份管理器可以帮助减轻与单点故障相关的风险,而且一些任务可以去中心化,以减少管理者的负担,从而降低瓶颈的风险。还有研究指出适当的资源分配策略可以帮助管理者有效地处理更多的座席和任务。([2, 5, 6])

我们所设计的多 AGENT 系统可以利用分布式管理系统来解决上述问题。在分布式管理系统中,可以实施多个备份管理器(Backup Managers)以减轻与单点

故障 (Single Point of Failure) 相关的风险。这样, 即使主管理器出现故障, 备份管理器可以立即接管, 确保系统的持续运行。该系统也是去中心化的, 这意味着没有单一的管理实体负责所有任务。这有助于减轻任何单一管理者的负担, 从而降低成为瓶颈的风险。去中心化还增加了系统的灵活性和可扩展性。同时, 资源分配通常是动态进行的, 并且可以根据各个 AGENT 的能力和当前负载来进行调整。这样, 管理者可以更有效地处理更多的任务。此外, 该系统可以通过负载均衡技术来分散请求, 从而减少任何单一节点或管理者的压力。这有助于提高系统的整体性能和可靠性。最后, 该系统可以采用一致性算法 (如 Paxos、Raft 等) 来确保所有管理实体都有一致的系统视图, 即使在网络分区或其他故障情况下也是如此。通过这些方式, 分布式管理系统能够缓解多代理系统中的各种问题, 从而提供更高的可靠性、可扩展性和效率。这些特点使其成为多代理系统中的一个有用工具, 特别是在需要高度可靠和可扩展解决方案的场合。

由于本文主要的研究对象是辩证推理方法下的管理者 G 和多 AGENT 协同工作的效率等问题, 所以在本文接下来的内容中暂时不考虑分布式系统的问题, 相关内容会在下一阶段的工作中进行展开研究。

### 3 逻辑推理效果测试实验分析

多 AGENT 协同的论辩式辩证逻辑推理方法是多 AGENT 协同运行工作中的辅助方法, 无法通过可视化的推理结果判定论辩逻辑推理效果, 因此需要将其应用到多 AGENT 协同任务处理工作中, 从而反映出论辩式辩证逻辑推理方法的推理结果。为了体现出设计的基于人工智能的多 AGENT 协同的论辩式辩证逻辑推理方法在推理效果方面的优势, 此次实验采用对比测试的方式, 实验对比方法为文献 [21] 方法 (基于时序关系网络的论辩逻辑推理方法) 和文献 [8] 方法 (基于逻辑推理规则和冲突度量的证据网络推理新方法)。

#### 3.1 配置多 AGENT 研究对象

此次实验选择 6 个相同的 AGENT 作为研究对象, 分别从引擎、执行器、AGENT 管理等多个方面对多 AGENT 研究对象进行配置, 多 AGENT 研究对象连接结构的配置结果如图 4 所示。

图 4 中的引擎是多 AGENT 的核心, 用来负责协调各 AGENT 的动态调整, AGENT 探测器和验证器分别用来检索配置请求并验证 AGENT 协同执行任务的合法性, 另外 AGENT 执行器为协同论辩式辩证逻辑推理方法的运行环境, 将上述组成元件进行连接, 并布设各个 AGENT 之间的通信环境。将包含 AGENT 构件静态信息、接口信息、执行规则、协同协议等内容的数据导入到对应的代理模

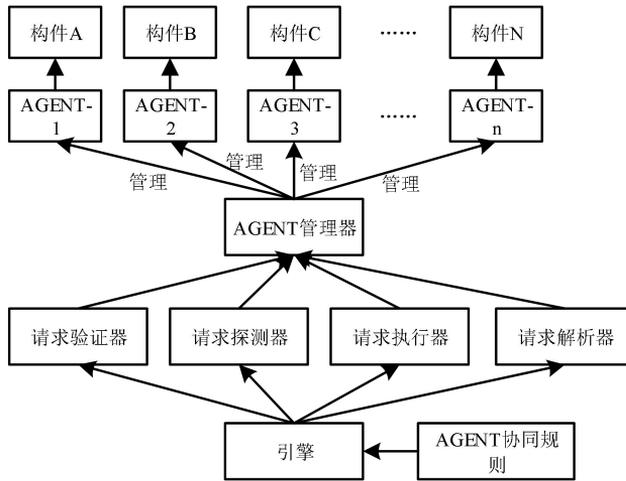


图 4: 多 AGENT 拓扑结构配置结果

块, 构成一个独立的 AGENT, AGENT 管理器提供 AGENT 总线负载对多 AGENT 的统一管理。

### 3.2 准备多 AGENT 协同处理任务

将设计的基于人工智能的多 AGENT 协同的论辩式辩证逻辑推理方法应用到在面向 P2P 社区的协同任务中, 在实验环境中设置多个恶意节点, 测试多 AGENT 协同的论辩式辩证逻辑推理方法是否能够成功识别并抵制恶意节点对协同工作的影响。多 AGENT 协同处理任务以及恶意节点的设置情况如表 1 所示。

实验组别	协同处理子任务数量/个	正常节点数量/个	恶意节点数量/个
1	200	300	20
2	220	350	30
3	240	400	40
4	260	450	50
5	280	500	60
6	300	550	70
7	320	600	80
8	340	650	90

表 1: 多 AGENT 协同处理任务设置表

在协同处理任务正常节点位置上导入恶意攻击程序, 实现恶意节点的布设。为了保证实验协同处理任务中恶意节点的可控性, 需要在恶意攻击程序中添加一

个强制开关, 从而实现正常节点和恶意节点之间的转换。

### 3.3 描述测试实验过程

我们利用基于人工智能的多 AGENT 协同的论辩式辩证逻辑推理方法调整多 AGENT 的协同工作逻辑, 在完成多 AGENT 的初始化设置后, AGENT 管理器向各个 AGENT 发出开始指令, 并分配协同任务和恶意节点判定议题。AGENT 打开一个具有专业规则和知识库的论据本体, 在论据和推理规则的支持下得出节点类型的判定结果。在协同论辩式辩证逻辑推理方法的支持下多 AGENT 协同任务的运行情况如图 5 所示。

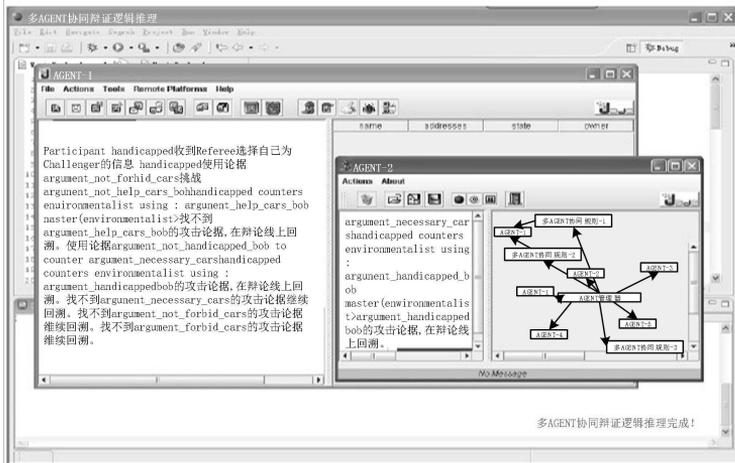


图 5: 应用论辩式辩证逻辑推理的多 AGENT 协同任务运行界面

按照上述操作流程运行三种逻辑推理方法, 得出三种方法下多 AGENT 协同任务的执行结果。

### 3.4 设置逻辑推理效果测试指标

此次实验分别从多 AGENT 协同推理加速效果和应用效果三个方面进行测试, 设置协作推理加速率作为多 AGENT 协同推理加速效果的量化测试指标, 其数值结果如下:

$$\mu = \frac{\sum \tau_{i+1} - \tau_i}{n_{\text{subject}}} \quad (8)$$

式中,  $\tau_{i+1}$  和  $\tau_i$  分别表示的是多 AGENT 协同推理第  $i+1$  和  $i$  个主体的运行时间,  $n_{\text{subject}}$  表示的是协同处理任务中推理主体的总数量。多 AGENT 协同的论

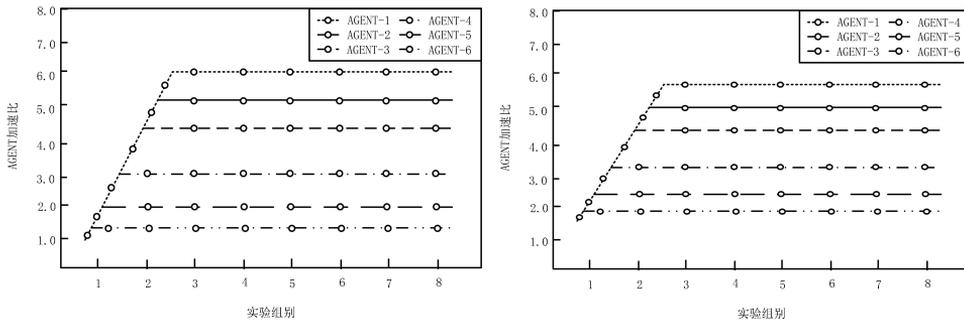
辩式辩证逻辑推理方法应用效果的测试指标为 AGENT 之间的协同程度和恶意节点的抵制成功率，AGENT 协同程度的数值结果为：

$$\varepsilon_{\text{coor}} = \frac{\mu_{\text{avg}}}{\sum_{k=1}^{N_{\text{AGENT}}} (\mu_{k+1} - \mu_k)} \tag{9}$$

式中， $\mu_k$  和  $\mu_{k+1}$  分别为第  $k$  和  $k + 1$  个 AGENT 的加速度，代表多 AGENT 加速度的平均值，参数  $N_{\text{AGENT}}$  为实验环境中设置 AGENT 的数量。另外通过论辩逻辑推理方法的应用多 AGENT 抵制恶意节点的成功率测试结果如下：

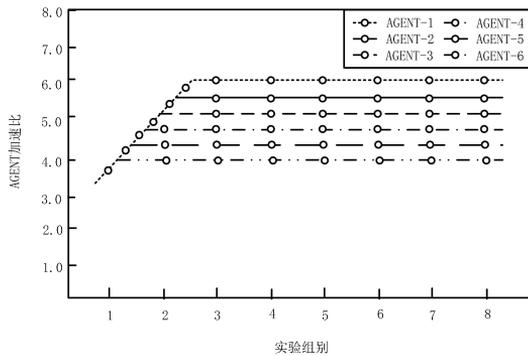
$$\sigma = \frac{\text{Num}_{\text{boycott}} - \text{Num}_b}{\text{Num}_{\text{malice}}} \times 100\% \tag{10}$$

式中， $\text{Num}_{\text{boycott}}$  和  $\text{Num}_{\text{malice}}$  分别表示的是多 AGENT 实际抵制的节点数量和实验环境中恶意节点的总数量， $\text{Num}_b$  为多 AGENT 抵制节点中非恶意节点的数量。最终计算得出协作推理加速度越高、逻辑推理复杂度越低、恶意节点抵制成功率越高，说明对应多 AGENT 协同的论辩式辩证逻辑推理方法的性能越优、应用价值越高。



(a) 文献 [21] 方法

(b) 文献 [8] 方法



(c) 设计多 AGENT 协同的论辩式辩证逻辑推理方法

图 6: 不同方法的加速度测试对比结果

### 3.5 测试实验结果分析

统计相关数据，通过公式 8 的计算得出反映论辩逻辑推理方法加速率的测试对比结果，如图 6 所示。

综合 6 个 AGENT 的运行结果可知，文献 [21] 方法和文献 [8] 方法的平均加速率分别为 3.4 和 3.8，而设计辩证逻辑推理方法的平均加速率为 4.8，高于文献 [21] 方法和文献 [8] 方法，证明设计的多 AGENT 协同的论辩式辩证逻辑推理方法的性能更好。另外多 AGENT 协同辩证逻辑推理方法应用效果的测试结果，如表 2 所示。

实验组别	恶意节点数量	文献 [21] 方法		文献 [8] 方法		基于人工智能的多 AGENT 协同辩证逻辑推理方法	
		抵制节点数量/个	非恶意节点数量/个	抵制节点数量/个	非恶意节点数量/个	抵制节点数量/个	非恶意节点数量/个
1	20	15	1	13	1	20	0
2	30	24	1	24	1	28	0
3	40	37	2	33	2	40	0
4	50	43	3	42	0	47	1
5	60	55	1	51	1	60	0
6	70	64	0	59	0	70	0
7	80	72	2	68	2	76	1
8	90	85	1	81	1	88	0

表 2: 不同推理方法应用效果测试数据表

将表 2 中的数据代入到公式 (10) 中，计算得出三种方法恶意节点抵制成功率平均值分别为 87%、82.5% 和 97.1%。综上所述，与文献 [21] 方法和文献 [8] 方法相比，设计的基于人工智能的多 AGENT 协同的论辩式辩证逻辑推理方法的加速率更高、推理复杂度更低，同时具有更高的应用效果。

## 4 结束语

多 AGENT 协同问题研究的关键是要建立一个协同的社会组织，对任务实现有效分解和分配，以及解决多 AGENT 协同中的通信开销问题。在人工智能技术的支持下，通过论辩式辩证逻辑推理方法的设计与应用，有效提高了多 AGENT 之间的协同度，进而提高多 AGENT 协同任务的执行效率，由此证明基于人工智能的多 AGENT 协同的论辩式辩证逻辑推理方法具有较高的研究与应用价值。

## 参考文献

- [1] J. Briot, S. Aknine and Z. Guessoum, 2007, “Multi-agent systems and fault-tolerance: State of the art elements”, *Deliverable, FTCAT Project, EuroControl INO CARE III Programme*, Brétigny-sur-Orge, France, **Vol. 67**.
- [2] D. Calvaresi *et al.*, 2018, “Multi-agent systems’ negotiation protocols for cyber-physical systems: Results from a systematic literature review”, *International Conference on Agents and Artificial Intelligence*, **Vol. 1**, pp. 224–235.
- [3] F. Cicirelli, A. Giordano and L. Nigro, 2011, “Distributed simulation of situated multi-agent systems”, *2011 IEEE/ACM 15th International Symposium on Distributed Simulation and Real Time Applications*, pp. 28–35.
- [4] F. Cicirelli, A. Giordano and L. Nigro, 2015, “Efficient environment management for distributed simulation of large-scale situated multi-agent systems”, *Concurrency & Computation Practice & Experience*, **27(3)**: 610–632.
- [5] A. Helsinger, H. Thome and T. Wright, 2004, “Cougaar: A scalable, distributed multi-agent architecture”, *2004 IEEE International Conference on Systems, Man and Cybernetics*, **Vol. 2**, pp. 1910–1917.
- [6] P. R. Lewis, P. Marrow and X. Yao, 2010, “Resource allocation in decentralised computational systems: An evolutionary market-based approach”, *Autonomous Agents and Multi-Agent Systems*, **21(2)**: 143–171.
- [7] D. Yazdansepas and M. R. Nami, 2010, “Multi-agent systems: A survey”, *International Conference on Parallel & Distributed Processing Techniques & Applications*, Las Vegas, Nevada, USA, **Vol. 2**.
- [8] H. Zhang *et al.*, 2020, “A novel method of evidential network reasoning based on the logical reasoning rules and conflict measure”, *IEEE Access*, **8**: 78015–78028.
- [9] 曹锋, 徐扬, 陈树伟等, “多元协同演绎在一阶逻辑 ATP 中的应用”, *西南交通大学学报*, 2020 年第 2 期, 第 401–408 + 427 页。
- [10] 陈凤, “基于 Agent 的数据链系统信息交互过程建模仿真”, *计算机仿真*, 2020 年第 8 期, 第 266–270 页。
- [11] 陈光, 蒋同海, 王蒙等, “自定义 SWRL 知识图谱推理补全插件的实现”, *计算机工程与应用*, 2021 年第 1 期, 第 261–270 页。
- [12] 晋荣东, “Dialectic: 辩证逻辑的新形态”, *湖南科技大学学报 (社会科学版)*, 2006 年第 6 期, 第 32–38 页。
- [13] 梁星星, 冯旻赫, 马扬等, “多 Agent 深度强化学习综述”, *自动化学报*, 2020 年第 12 期, 第 2537–2557 页。
- [14] 林颖, 卫龙, “多 Agent 的动态信念和知识推理”, *贵州工程应用技术学院学报*, 2022 年第 1 期, 第 54–59 页。
- [15] 潘俊峰, 唐敦兵, 张泽群等, “基于规则调整的多 Agent 制造系统调度研究”, *机械制造与自动化*, 2021 年第 5 期, 第 160–163 页。

- [16] 秦丽娜, “基于多 Agent 技术的混合微组装生产线协同控制方法”, 制造业自动化, 2022 年第 3 期, 第 176–179 页。
- [17] 万小龙, 周红艳, “雷歇尔的辩证法思想探析——过程的辩证法与辩证的过程论”, 马克思主义与现实, 2017 年第 4 期, 第 102–107 页。
- [18] 吴甜甜, 王洁, “基于可能回答集程序的多 Agent 信念协调”, 计算机科学, 2020 年第 2 期, 第 201–205 页。
- [19] 叶幸瑜, 刘玮, 王宁等, “基于马尔可夫模型的多 agent 自适应在线验证”, 计算机应用研究, 2021 年第 5 期, 第 1477–1481 页。
- [20] 张红颖, 周子林, 李彪, “基于多 Agent 的通航运力资源协同调度”, 交通运输系统工程与信息, 2020 年第 1 期, 第 214–221 页。
- [21] 张姝楠, 曹峰, 郭倩等, “一种基于时序关系网络的逻辑推理方法”, 计算机科学, 2021 年第 5 期, 第 239–246 页。
- [22] 郑延斌, 樊文鑫, 韩梦云等, “基于博弈论及 Q 学习的多 Agent 协作追捕算法”, 计算机应用, 2020 年第 6 期, 第 1613–1620 页。
- [23] 周红艳, 雷歇尔的辩证逻辑思想研究, 2018 年, 华中科技大学博士学位论文。

(责任编辑: 映之)

# Multi AGENT Cooperative Dialectical Logic Reasoning Method Based on Artificial Intelligence

Hai Huang    Qiquan Gui

## Abstract

In order to eliminate the conflict of beliefs within AGENT and realize the cooperative operation of multiple AGENTs, a multi-AGENT collaborative dialectical logic reasoning method is designed by using artificial intelligence technology. According to the composition structure of AGENT and the communication mode between multiple AGENTs, a multi-AGENT collaboration model is established. Under this model, artificial intelligence technology is used to solve multi-AGENT collaborative tasks and resolve conflicts between AGENTs. By calculating the critical factor and critical element, the dialectical logic inference evidence of multi-AGENT collaboration is constructed, and with the support of inference evidence and rules, the multi-AGENT collaborative dialectical logic reasoning is realized. The experimental results show that the acceleration rate of designing the dialectical logic reasoning method is increased by 1.4, the degree of cooperation between multiple AGENTs is increased by 4.12, and the success rate of resisting malicious nodes is increased by 12.6%, which has a better application effect.

---

Hai Huang    School of Marxism, Xuchang University  
Mobile Source Emission Aftertreatment Technology Henan Institute  
15022799968@163.com

Qiquan Gui    School of Philosophy, Wuhan University  
qqgui@sina.com