ISSN 1004-9037,CODEN SCYCE4
Journal of Data Acquisition and Processing Vol. 39,No. 3,May2024,pp. 559－576
DOI:10. 16337/j. 1004-9037. 2024. 03. 005

http://sjcj. nuaa. edu. cn
E-mail:sjcj@nuaa. edu. cn
Tel/Fax:＋86-025-84892742

基于大模型的联动处置多智能代理协同框架

吴晓宁[1],李瑞欣[1],王　浪[1],刘文杰[1],王宏伟[1],朱新立[1],宋江帆[1],袁　梦[2]

(1.北方自动控制技术研究所,太原 030006;2.武警工程大学反恐指挥信息工程教育部重点实验室(立项),西安 710086)

摘　要: 针对指挥员应对重大突发情况时的处置决策难题,提出一种基于大模型的联动处置多智能代理协同框架。该框架通过智能代理角色生成、多层级蒙特卡洛树与交互式提示学习等策略,优化群体决策效率与动作规划,同时引入分层机制与工作流管理理念,通过强化学习奖励函数共享提升协同效率,设计显式与隐式通信模式确保节点状态一致。实验表明,该框架在多种场景下表现优异,与传统任务分配手段相比,大大提高了面对突发事件时的反应速度和处置效率。

关键词: 大模型;联动处置;多智能代理;处置规划

中图分类号: TP183　　**文献标志码:** A

Coordination Framework for Collaborative Disposal of Multi-intelligent Agents Based on Large Language Models

WU Xiaoning[1], LI Ruixin[1], WANG Lang[1], LIU Wenjie[1], WANG Hongwei[1], ZHU Xinli[1], SONG Jiangfan[1], YUAN Meng[2]

(1. North Automatic Control Technology Institute, Taiyuan 030006, China; 2. Key Laboratory of Counter-Terrorism Command & Information Engineering of Ministry of Education (Approval), Engineering University of PAP, Xi'an 710086, China)

Abstract: Addressing the decision-making conundrum faced by commanders in response to major sudden incidents, this paper proposes a coordination framework for collaborative disposal of multi-intelligent agents based on large language models. The framework optimizes collective decision-making efficiency and action planning through strategies such as agent role generation, multi-level Monte-Carlo tree and interactive prompt learning. It introduces hierarchical mechanisms and workflow management concepts, enhancing collaboration efficiency through the reward function shared among agents. A transparent and implicit communication model ensures node status consistency. Experimental results demonstrate that the framework performs well under various scenarios, significantly improving reaction speed and response efficiency compared to traditional task allocation methods.

Key words: large language models (LLMs); collaborative disposal; multi-intelligence agent (MIA); disposal planning

基金项目: 山西省重点研发计划(202102150401013)。
收稿日期: 2024-03-05;**修订日期:** 2024-04-24

引　言

随着多域作战和联合作战的发展，现代战争中，指挥员在应对重大突发情况的处置决策时，往往面临着战场态势认知困难、演进趋势研判困难和处置软件运用繁杂等问题。

ChatGPT（Chat generative pre-trained transformer）的出现，前所未有地提升了自然语言处理的技术水平，在军事及情报领域具有巨大的潜在价值，以ChatGPT为代表的生成式人工智能（Generative AI）技术在5个智慧维度实现重大突破[1-4]：（1）海量信息的参数化全量记忆；（2）任意任务的对话式理解；（3）复杂逻辑的思维链推理；（4）多角色风格长文本生成；（5）即时新知识学习与进化。据美国C4ISRNET报道，2023年1月美国国防信息系统局（DISA）已将ChatGPT等生成式人工智能技术添加到观察名单；同年4月，美国知名大数据分析公司Palantir最新推出人工智能平台AIP（Artificial intelligence platform），使用大模型综合多个相关数据，为指挥官提供军事决策、命令下达和作战监控，实现优化决策流程、缩短决策时间、获得最优作战方案和保障作战质量的目标。

智能代理[5]本质是一个控制大模型来解决问题的代理系统。大模型的核心能力是意图理解与文本生成，但在应对突发情况联动处置的现实问题涉及了很多超越语言模型之外的能力，如战略目标联动处置、实时数据分析和可视化结果，以及各类要素自动态势标绘等。如果能让大模型学会使用工具，那么大模型本身的能力也将大大拓展。目前，让大模型解决这些问题的一个最有前景的方向就是建立大模型驱动的智能代理，也就是让大模型作为核心控制者来学会使用不同工具，进而完成最终任务。

同时，国内外关于智能代理的研发也层出不穷[6-9]。Open AI开发的GPTs（Generative pre-trained transformers），以及推出的GPT-4Turbo和可定制智能代理，使得每个人都可以打造自己的大模型应用。微软全新工具AutoGen[10]允许多个LLM（Large language model）智能体通过聊天来解决任务。英伟达推出的基于GPT-4的最新版开源智能代理Eureka[11]，已在短时间内教会机器人完成30多个复杂任务，在超过80%的任务中都超越人类专家，更让机器人的平均性能提升超过50%。百度将文心大模型[12]应用到智能搜索、自动驾驶。阿里将通义千问模型[13]应用到高德地图、优酷和盒马等产品。华为将其盘古模型[14]应用到智能气象、语音识别等。在医疗领域，聚焦亚健康管理领域的医者AI（清华创业团队创建的公司）基于最前沿的自研MoE（Mixture-of-experts）架构大模型[15]，构建了健康管理Agent-Healthy Care Agents。而智能代理＋BlockChain[16]也已成为加密货币交易新战场。

传统突发事件处置业务场景中，指挥系统多采用基于规则引擎构建事件集成框架[17]、网络计划方法[18]的实时异常处置等手段对突发情况进行决策，这些传统手段往往采用串行的方式，处置的各个阶段都需要人工参与，每个阶段花费时间长，且无法发挥多角色联动处置的优势。随着多智能体技术在协同指挥[19]、协同决策[20]等场景中得到了应用，考虑将其与联动处置场景进行结合，充分发挥基于大模型的单智能代理可记忆、可理解和可规划的作用，同时结合多智能体不同角色划分，模拟事件处置场景中的多个席位角色，采用协作的形式共同完成突发事件处置场景。

1　框架设计

本文面向突发情况联动处置业务领域，从情况联动处置角度出发，将基于大模型的通用智能代理框架在情况联动处置认知决策领域上进行适配和改造，设计了具备处置决策最优规划、软件工具互操作和多智能代理协同重组为核心特征的开放式情况联动处置智能代理体系框架。

1.1　基于大模型的智能代理框架

基于大模型的智能代理框架[21]：智能代理＝大模型＋记忆＋规划程序＋工具使用，如图1所示。其中，大模型是智能代理中智能体的核心，具备良好的意图理解和文本生成能力。AI智能体的记忆模块

包括短期记忆和长期记忆,短期记忆包括从外界感知到的感觉记忆和用户输入的提示词,长期记忆指大模型中编码的知识以及存储在数据库中的信息。短期记忆受到大模型的Transformer类神经网络有限上下文窗口长度的限制,长度和隐含信息通常是有限的。规划模块通过提示词工程,引导智能体进行任务理解和分解,将复杂的目标任务根据思维链分解成子任务和可直接执行的指令。规划模块主要包括反思、自我批评、思维链和子目标分解。工具模块通过配备一些工具来延伸大模型生成和理解之外的能力,可以通过智能代理调用外部API以获取模型权重中缺少的额外信息,将其由理解和生成答案的语言模型转化为具备理解、调度和执行能力的AI智能体。工具主要包括情况发现与告警、情况处置组织协调、情况处置过程综合监控和情况信息综合查询等接口。行动模块是AI智能体依据规划的子目标去完成工具接口的调用,执行规定的指令动作。

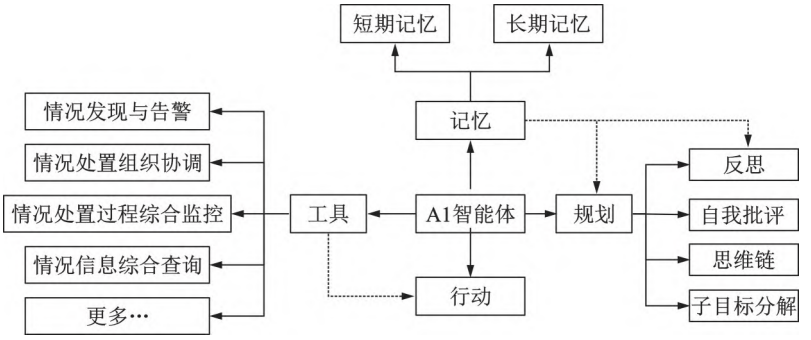

图1 基于大模型的智能代理框架

Fig.1 Intelligent agent framework based on large language models

1.2 情况联动处置智能代理框架

针对图1智能代理框架在情况联动处置问题上进行适配和改造,得到用于情况联动处置的智能代理框架,如图2所示。它包括了感知端、控制端和行动端。

(1)感知端。针对情报数据来源众多(谍报、技侦、航侦、各类开源情报等)、数据类型模态多样(结构化数据、流式数据、文本数据、图片、视频)等问题,值勤人员在情况联动处置业务活动中难以从多源异构数据中获取一致信息,感知端将智能代理的感知空间从纯文本拓展到包括文本、视频、图片和语音等多模态领域,使智能代理能够更有效地从复杂环境中获取与利用信息,突破多源异构数据信息识别抽取技术、多模态信息统一表征认知模型等关键技术,以支持对跨模态、多样式的要素信息的识别、抽取,多模态知识获取、表示与推理,形成多模态的统一语义表示,提升智能代理在多模态信息环境中的智能感知能力。

(2)控制端。主要是指为处置任务进行处置动作规划的单元,它可以决策处置动作需要运用哪些操作指令来完成,包括记忆、结构感知知识和处置动作规划3部分。记忆部分,提出一种面向冗长文本输入的自控制记忆增强机制,通过不断更新记忆实现演化,更好地为后续特定领域处置动作分解与规则提供历史依据;结构感知知识部分,通过动态图结构归纳自然语言文本和领域知识库,学习自然语言输入和领域知识库的语义表示,提升智能代理的认知能力;处置动作规划,引入多层级蒙特卡洛树搜索机制随机采样分解的处置动作,设计交互式提示学习最优规划策略不断地整合外部环境和自我反馈信息,进行处置动作最优规划。

(3)行动端。执行已规划的处置动作,生成相应软件操作路径,使用相应的软件工具,执行任务,完成指挥员事件处置任务。包括软件接口数据集和检索器;其中软件接口数据集,存储各个软件接口的相关属性,对各类接口进行管理,辅助模型与接口进行交互;检索器根据输入的操作指令从数据集中检索与操作指令相关的接口,帮助模型过滤无关信息,提高模型的训练效率;自主调用模型,根据操作指令与接口进行多轮交互,并返回交互结果。

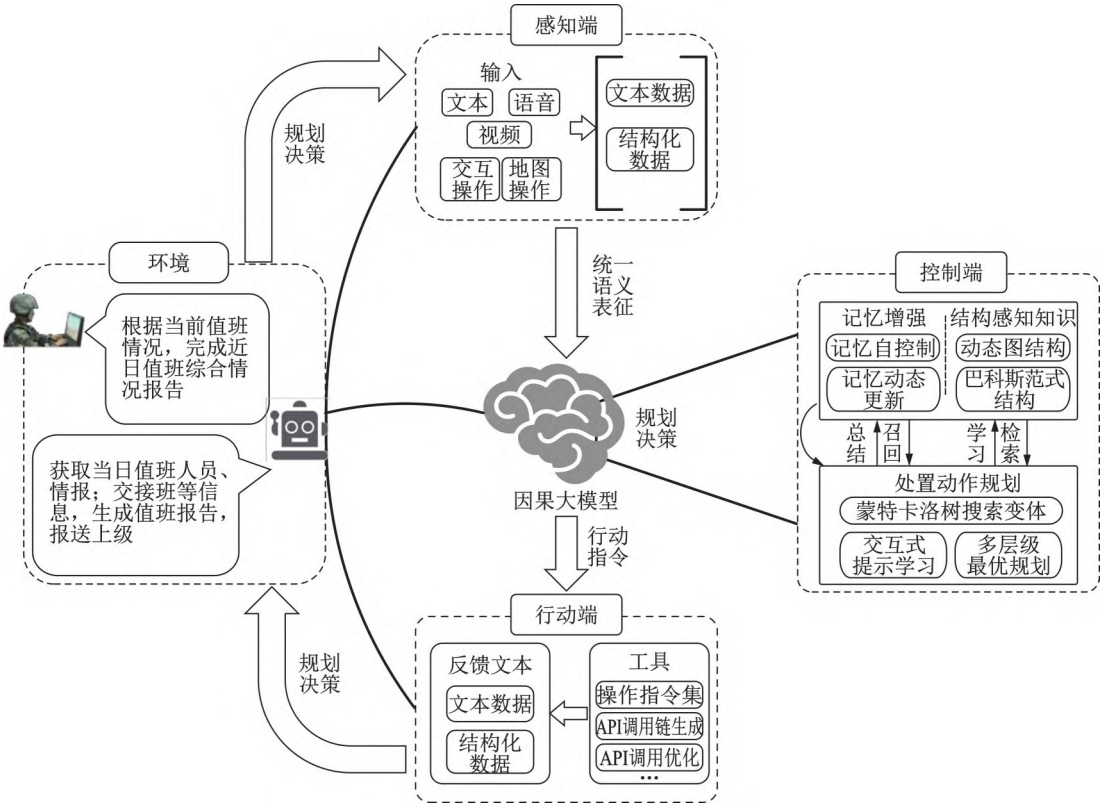

图2 情况联动处置智能代理框架

Fig.2 Intelligent agent framework for scenario-based collaborative disposal

1.3 多智能代理自组织协同运行机理

本文面向战略指挥过程的情况联动处置场景，以战略指控过程的事件处置条令和规范要求为导向，自动抽取生成情况联动处置流程，根据指挥作业意图和任务指令，理解指挥所任务情境，自动组织各业务要素协同完成任务。此过程需各种领域智能代理共同协作来完成，这些领域智能代理不是孤立的，而是需要与其他领域智能代理或人类进行交互和协作，形成一个多智能体协作框架。多智能体协作框架开启未来情况联动处置的全新模式。例如，在美军舰横穿台湾海峡的突发事件处置任务中，指挥员为完成处置任务，需要态势标绘、行动规划、装备性能分析和目标威胁分析等各作业要素协同，这样每个作业要素都是一个智能代理，它们需要根据战场的情况和事件的态势，选择合适的行动和策略，同时也需要与其他智能代理进行协作和竞争，以达成共同协作完成处置任务的目的。多智能代理协作框架使军事信息系统能够根据实时的情报和预警，自动地调整自己的参数和策略，能够与其他军事信息系统或人类进行有效的交互和协作。多智能代理协作框架如图3所示。

多智能代理协作框架的核心是如何实现智能代理之间的协作和竞争的平衡，即如何使每个智能代理都能达到自己的目标，同时也能促进整个系统的性能和效益。为了实现这一目标，多智能代理协作框架主要解决以下几个关键的问题：

(1)智能代理的角色建模和设计。定义智能代理的属性和行为，如目标、偏好、策略、动作、感知、学习和沟通等。角色定义采用多个单工具的智能体策略，针对指挥过程中的态势标绘、行动规划、装备性能分析和目标威胁分析等作战要素，按照各要素子任务分配对应的工具，结合工具的文档说明对智能

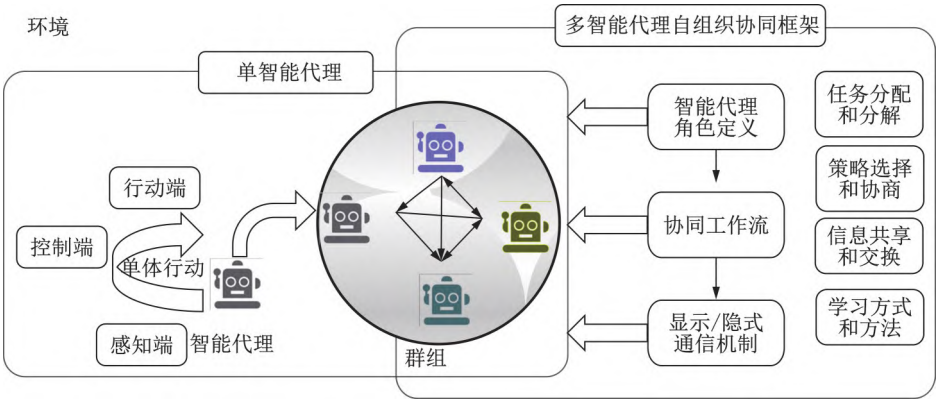

图3 多智能代理自组织协同框架

Fig.3 Self-organization coordination framework for multi-intelligent agents

体进行训练,使智能体达到理解工具或独立调用接口等目标。

(2)任务分配和分解。将一个复杂的任务分配和分解给多个智能代理,使得每个智能代理都能承担合适的子任务,同时也能保证任务的完整性和一致性。任务分配阶段通过提示词进行任务的明确和分解,形成完成复杂任务的思维链,提取思维链中各序列化的节点作为子任务,形成任务序列清单。根据各智能体角色分工将各子任务分发下达给各智能体。

(3)动作处置和规划。将高层次的任务目标分解为可执行的低层次动作,并规划反思动作的执行顺序,通过环境感知反馈来进一步适应多层级搜索,以实现整体目标。在此阶段,多个智能体获得初始任务,通过多轮与环境的交互,优化智能体所分配的子任务和执行顺序,得到最优的规划。

(4)智能代理的协同工作流。设计智能代理之间的交互和协调的机制和协议,如定义工作流程、交互信息等。本阶段针对指挥流程定义了领域化的业务流程,结合OODA环设计完整的各智能体交互流程。

(5)智能代理的通信机制。设计基于协议的显式通信和基于感知的隐式通信两种信息交互模式,采用压缩、缓存和批处理等技术优化,降低通信延迟和开销,保证多智能代理框架的信息同步和状态一致。

2 跨层级多智能代理协作技术

2.1 面向作业要素的智能代理角色生成方法

为提升群体决策中单智能代理的工作效率,对智能代理中的角色进行分工,定义不同智能代理的角色技能和行为逻辑,利用提示词工程生成特定场景下的角色代理配置,同时能够支持在多智能代理协作过程中,结合当前环境状态信息,动态地对智能代理角色身份进行调整。重点开展以下3个部分研究:

(1)面向作业要素的智能代理角色定义

角色定义是多智能代理框架构建时至关重要的一步,面向作业要素构建角色可以为多智能代理协作提供基础,以能力和任务需求为基础对角色进行分类,每个智能代理都被赋予与其能力相匹配的任务和责任。例如,某些智能代理可能具有强大的计算能力,适合进行复杂的数据分析,而某些智能代理可能具有良好的通信能力,适合与其他代理进行协调和信息交换。基于大模型的角色划分可以更好地分析任务的要求和目标,然后匹配和确定智能代理角色划分方案。

(2)智能代理角色信息管理

智能代理角色信息管理重点对代理自身的知识库数据和当前状态数据进行管理,保证协同工作时不出现信息差。在多智能代理协作过程中,每个智能代理都将自己的知识和经验存储在共享的知识库中,其他代理可以通过查询和更新这个知识库来获取或贡献知识。同时,多智能代理中的信息是分散的、多样的,需要从多源、多格式、多层次的信息中提取有用的信息并进行融合,更好地支持系统的决策和协调。

(3)智能代理信息动态调整

针对多智能代理协作过程中存在环境变化和突发事件的情况,本框架利用大模型来自动学习和优化角色分配。通过收集不同场景下的角色及任务相关的训练数据和模拟实验数据,可以学习智能代理在不同任务和场景下的表现,并根据学习结果来动态调整角色分配。本文采用更适合上下文学习的AdapterFusion方法对大模型进行微调。

2.2 面向作业要素的动作处置和规划算法

针对特定领域任务目标多样、环境信息复杂导致大模型动作规划效果不佳等问题,结合多层级搜索和交互式提示学习思想,引入多层级蒙特卡洛树搜索机制随机采样分解的处置动作,设计交互式提示学习最优规划策略,不断地整合外部环境和自我反馈信息,通过训练目标选择器对候选动作进行排序,允许智能代理在不同的层级上进行搜索,从宏观到微观逐步细化问题,同时允许代理与外部环境进行交互,得到系统反馈并及时对规划方案进行调整,从而实现处置动作最优规划,提高决策的准确性和效率。多层级搜索交互式处置动作最优规划流程如图4所示。

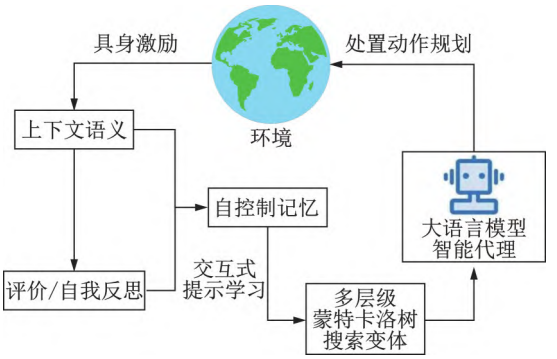

图4 多层级搜索交互式处置动作最优规划流程图

Fig.4 Flowchart for optimal planning of interactive search and response actions at multiple levels

(1)多层级蒙特卡洛树搜索变体

多层级蒙特卡洛树搜索变体[22]支持顺序推理或规划任务。在时间t,智能代理从环境中接收到观察,并按照策略$\pi(a_t|x, o_{1,\cdots,i-1}, a_{1,\cdots,i-1})$采取行动,其中$x$由任务指令和一些小样本示例组成,对大模型初始化代理,以利用大模型的有用语言表示作为基本决策者。遵循REAC T[23],本文中动作空间由当前状态动作空间和语言空间两部分组成。前者定义了在当前状态下可执行的所有合法动作集合,直接影响环境并导致观察反馈;后者通过对当前状态的上下文进行推理来编写有用的信息,并更新上下文信息,以支持未来的推理或行动,不影响外部环境,从而导致没有观察反馈。动作空间的具体实例取决于特定的环境,规划任务中动作空间可能包括网站上的命令,而推理任务中动作空间可能仅限于一些外部工具或API。

多层级蒙特卡洛树搜索变体的主要组成部分是一种搜索算法,它通过深思熟虑的规划来控制整个问题解决过程。为了找到最优的规划并保证系统探索的平衡性和全局性,本技术采用了蒙特卡洛树搜索的一种变体,它将决策过程转化为树搜索问题,其中每个节点$s=[x, a_{1,\cdots,i}, o_{1,\cdots,i}]$包含了原始输入、动作序列和观察序列。多层级蒙特卡洛树搜索变体将大语言模型当作状态评估器和反馈生成器,利用现代大语言模型的语言先验知识来促进规划。多层级蒙特卡洛树搜索变体搜索过程依赖环境交互的

反馈信息,由一系列操作组成,具体包含了选择、扩展、评估、模拟、反向传播和反馈,连续执行直到任务规划成功完成或达到计算极限。

(2)交互式提示学习最优规划策略

交互式提示学习最优规划策略是基于大型语言模型的交互式规划方法,通过对执行过程中计划进行描述,并在计划的扩展阶段遇到故障时提供反馈的自我解释,促进对大语言模型初始生成的规划进行错误纠正。此外,它还包括一个目标选择器,这是一个可训练的模块,根据当前计划下的完成步骤对候选子目标集合进行优先级排序,从而细化初始规划。

如图5所示,交互式提示学习最优规划策略由1个情况触发的描述器、1个大语言模型、1个目标选择器和1个目标条件策略控制器组成。利用一个大语言模型作为代理的零样本规划器来完成任务,给定一个目标命令作为任务,基于大语言模型的规划器将这个高级任务分解为一个子目标序列,作为初始规划。然后调用控制器,通过目标条件策略按顺序执行所提供的子目标。但是,规划器提供的初始规划通常包含错误,这将导致控制器的执行失败。当弹出失败时,描述符将最近一个目标的当前状态和执行结果汇总为文本,并将其发送到大语言模型。大语言模型将首先尝试通过自我解释来定位之前规划中的错误。然后,它将重新规划当前的任务,并根据解释生成一个修改后的规划。在这个过程中,除了规划器角色之外,大语言模型还被视为一个解释器。为了过滤掉低效的规划,训练选择器来预测在给定当前状态的一组并行目标中实现每个目标的剩余时间步数。

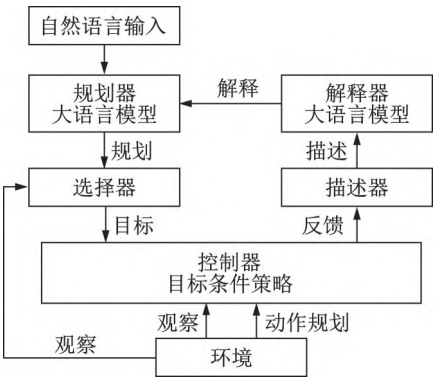

图5 交互式提示学习最优规划策略框架图

Fig.5 Framework diagram for learning optimal planning strategy using interactive prompts

2.3 跨层级多智能代理协同工作流

本文基于SOP(Standard operating procedure)工作流管理理念,引入分层机制,通过强化学习奖励函数共享,保证最大化地发挥单智能代理的联合价值,提升协同工作效率。

(1)SOP工作流程

在本框架中,指定代理角色的画像,包括他们的姓名,每个角色的概况、目标和约束,并初始化每个角色的具体背景和技能。例如,在处置突发事件的系统软件研制过程中共定义了4个角色:值勤首长、值勤参谋、数据保障人员和开发人员。其中值勤首长可以查看下级上传的报告信息,而开发人员可以执行代码。每个代理人都会监视环境以发现重要的观察结果(例如,其他代理人的消息)。这些消息能直接触发行动或协助完成任务。跨代理人的工作流程通过定义代理人的角色和操作技能,可以建立基本的工作流程。跟随标准作业程序,使全部的智能代理可以用顺序方式完成工作。

(2)合作策略和学习

本框架采用基于协商的多智能代理协作策略,每个智能代理根据自身的资源和能力,提出自己的合作意向和条件。这些条件可以包括任务分配、资源共享、利益分配等方面的要求。然后,智能代理之间进行协商和讨论,以寻找最优的合作方案。协商过程中可以采用各种协商机制和算法,如协商层次模型、协商框架、协商代理等。通过协商,智能代理可以达成一致的合作协议,明确各自的责任和义务,以及合作的具体方案和实施细节。此外,主控制器来促进智能代理之间的协商和合作。控制器可以提

供中立的协调服务,帮助智能代理解决合作中的矛盾和冲突,促进合作的顺利进行。

同时,在智能代理工作过程中,采用基于大模型的分布式任务协调策略进行任务调度,通过并行处理将问题分解并分配给多个智能代理处理,提高处理速度和效率。加入负载均衡策略,通过合理的任务分配,可以确保各智能代理的负载均衡,避免资源的浪费或过度占用。在任务执行过程中,存在由于通信成本导致的集中式决策模式难以实施的情况,采用基于多智能代理深度强化学习的分布式任务分配算法,为各API接口单元均设计一个独立的策略网络,并采用集中式训练、分布式执行的方法对智能代理的策略网络进行训练,经过学习训练后的各业务代理具备一定的自主协同能力,即使在没有中心指挥控制节点协调的情况下,依然能够独立地实现任务的高效分配。

2.4 一致性约束下的多智能代理协作通信机制

为保证多智能代理之间交换信息、协调行动、解决冲突和共同实现系统目标,本文开展一致性约束下的多智能代理协作通信机制研究,设计基于协议的显式通信和基于感知的隐式通信两种信息交互模式,采用压缩、缓存和批处理等技术优化,降低通信延迟和开销,保证多智能代理框架的信息同步和状态一致。同时提出多智能代理协作通信机制,构建以权重为主线的一致性网络,保证每个节点在通信过程中的状态一致、分组一致和聚类一致,实现在多智能协作过程中的行为和决策一致。

(1)显式通信

为了提升多智能代理之间的通信效率,针对信息传递方式和意图设计基于协议的显式通信和基于感知的隐式通信两种信息交互模式。显式通信适用于目的明确、发送方和接收方明确的场景,通过预定义的通信协议以规定消息的格式和交换规则,智能代理通过明确的消息传递机制来交换信息,用于请求信息、协调行动或解决冲突场景;隐式通信利用大模型的推理能力,不需要预先定义的通信协议,通过感知环境的变化来推断其他智能代理的意图或状态,降低通信成本。

当异构消息交换过程中存在源消息协议XML不一致时,采用协议版本协商机制,支撑协议动态更新、协议自适应,提高信息系统的版本前后兼容能力。为了保证向前以及向后的兼容性,显式通信涉及从协议版本号标识、消息类型、协议版本协商机制和协议版本更新4个方面进行设计。

协议版本号标识。协议版本有两个字段,分别为主版本号和次版本号,二者取值范围均为0到15,即0x0到0xF,可以序列化为主版本号。次版本号的形式,如协议版本为0.1。在二进制消息中,一个版本号序列化为1字节长度的信息,其中前4位为主版本号值,后4位为次版本号值。

消息类型。消息分类两种类型,不同的消息类型可能包含的字段及含义有所不同,详细如下:①请求消息。此消息表示发送方请求接收方返回某一个资源,如果在指定的时间内未收到接收方的回复,则放弃等待,并向上层应用返回一个状态为超时的回复,表示请求超时;②回复消息。此消息表示发送方向接收方发送一个回复消息以回复对方曾经发送的某一条请求消息,此消息的ID为接收方发送的此条请求消息的ID。如果上层应用在指定的时间内未返回消息,则向发送方发送一个状态为超时的回复消息,表明上层应用处理超时。

协议版本协商机制。以软件模块A和软件模块B交互为例:①当软件模块A向软件模块B发送消息时,软件模块A必须发送其协议版本号;②软件模块B对接收到的协议版本号进行协议版本检测,如果是版本号相同,则解析;如果是版本号不相同,则按步骤3进行协议版本协商;③软件模块B需要发送一个协议协商的请求消息给软件模块A,负载内容为软件模块A发送的协议版本号;④软件模块A在收到此消息后,则回复一个状态为ok的回复消息,负载内容为所选择的协议版本号的协议模板,由于考虑到协议信息的安全性以及通信带宽,需要对协议版本号模板进行加密压缩等操作,如果不能处理,则返

回一个错误消息,负载为空,并且关闭连接;⑤软件模块B收到回复消息后,按照协议模板对发送的消息进行解析,同时更新协议模板索引。

(2)隐式通信

隐式通信直接在智能代理之间共享动作和/或局部观测。与此同时,类似于显式通信,每个智能代理处理这些全局信息的模型也必须保持对环境的相同理解,以实现全局协作。本文通过MARL(Multi agent reinforcement learning)方法[24],结合上下文和环境信息,进行隐式通信建模,将所有智能代理的动作作为联合动作,环境会将联合状态S和联合奖励R反馈给MARL,目的是学习一种最优的策略集合$\pi^*=\{\pi_1^*,\pi_2^*,\cdots,\pi_n^*\}$,如图6所示。每个智能体在与环境和其他智能体的交互中,根据当前的状态和观察到的信息,选择适当的动作,并隐含地传达特定的信息给其他智能体。这种隐式通信建模涉及到智能体对环境的感知、学习和适应能力的不断提升,以及在交互过程中通过奖励信号来引导通信行为的优化,从而实现更有效的协作与通信。其中每个智能体之间主要通过共享动作、局部观测,实现全局协作,它们之间的主要关系可能是合作、竞争或混合模式,具体取决于任务和环境的性质以及系统设计的目标:

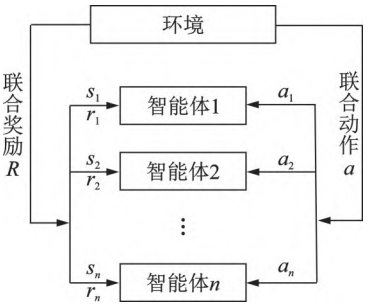

图6 MARL智能代理联合动作示意图

Fig.6 Schematic diagram of joint action of multiple intelligent agents

①在合作模式中,多个智能体之间相互协作,共同达到一个或多个共同的目标。它们可能会分享信息、资源和技能,以最大程度地提高整个系统的性能。

②在竞争模式中,多个智能体之间争夺有限的资源或者优势地位,以获得最大的奖励或者避免惩罚。它们可能会采取竞争性的行动,试图超越其他智能体并达到个体的目标。

③在混合模式中,智能体之间的关系既包含合作又包含竞争。它们可能在某些方面合作,共同完成任务,同时在其他方面竞争资源或者地位。这种模式下,智能体需要平衡合作和竞争的关系,以达到最优的解决方案。

MARL联合值函数表示为

$$Q_{\text{tot}}^{\pi}(s,a)=E\left[\sum_{t}^{\infty}\gamma^t r_t|s_0=s,a_0=a,\pi\right] \tag{1}$$

式中:s表示初始时刻状态,a表示初始时刻动作,π表示初始联合策略,代理接收奖励函数$r_t=R(s_t,a_t)$,$\gamma^t\in[0,1]$表示折扣因子,决定了对更远距离的奖励程度。

联合策略就是组成联合值函数的动作集合,即

$$\pi(s)=\arg\max_a Q_{\text{tot}}^{\pi}(s,a) \tag{2}$$

式中:$\pi(s)$表示s状态的联合策略,Q_{tot}^{π}表示MARL联合值函数。

(3)交互协议设计

为了实现多智能代理之间的高效通信,开展交互协议设计,定义智能代理之间的通信规则和格式,以确保信息的准确传递和有效交换。首先设计统一的消息格式,包括消息的类型、头信息、载荷和校验等。确定消息的编码和解码规则,以实现消息的高效传输和解析。其次定义通信协议,包括连接建立、消息交换、错误处理和连接关闭等过程。定义协议的状态机和流程,以确保通信的顺序和完整性。同时根据系统的需求,设计不同的交互模式,包括代理之间请求-应答、发布-订阅等,确定交互模式的触发条件和执行流程,以满足不同的通信场景。同时采用压缩、缓存和批处理等技术,以降低通信的延迟和

开销。设计高效的路由和转发机制,以减少网络拥塞和提高通信效率,提高单智能代理之间的交互效率。

当前大多数基于大模型的多智能代理框架使用自然语言作为通信接口,但纯自然语言通信是否足以解决复杂任务存在疑问。受人类社会结构启发,提出使用结构化通信格式来规范智能代理之间的通信。建立了每个角色的模式和格式,并要求个体根据其特定的角色和上下文提供必要的输出。本框架中的智能代理通过文档和图表进行通信,而不是对话,以避免不相关或缺失的内容。在协作中,信息共享至关重要,引入了发布-订阅机制来简化通信拓扑,提高效率。

本框架中每个智能代理通过从共享环境日志中检索相关历史信息,来主动策划个性化知识。智能代理不是被动地依赖对话,而是利用基于角色的兴趣来提取相关信息。同时,每个智能代理都维护了一个内存缓存,并对其角色相关的订阅消息进行索引,实现个性化的知识策划。具体来说,消息的集中复制创建了一个统一的数据源。智能代理可以注册订阅,自动从该数据源接收与其角色相关的消息。在内部,智能代理通过内容、来源和属性将内存缓存索引,以便在相关上下文中实现快速检索。①消息共享。当一个智能代理生成一条消息时,它会被复制到共享的环境日志中,创建一个真实的单一数据源。从而确保所有智能代理都可以获取相同的信息。②基于角色的订阅。智能代理可以根据其角色对其有意义的消息类型进行注册订阅,根据与智能代理的责任和任务相一致的预定义标准进行。③消息分发。当有新的消息符合订阅条件时,它会自动分发通知给相关的智能代理。这种主动传播信息的方式可以防止智能代理错过重要的更新。④内存缓存和索引。智能代理会维护一个内部的记忆缓存,其中订阅的消息会被存储并按内容、发送者和接收者建立索引,从而保障高效的信息存储和检索。⑤上下文检索。环境会维护一个支持缓存和索引的共享内存池,智能代理可以根据需要查询其内部内存,以获取与其当前任务相关的上下文细节。这有助于改进其理解并做出更好的决策。⑥更新同步。对消息进行的任何更新或更改都会在所有链接的智能代理内存中同步,以保持信息的一致视图。这确保所有智能代理都可以访问最新的数据。

通过在智能代理角色周围对信息流进行组织,确保多智能代理之间的协作。通过结合中心化的知识共享与基于角色的个性化内存缓存相结合,实现定制化的知识管理。这减少了无关数据的存在,并提供了共同的上下文,从而在团队协作和个人效率之间达成平衡。

3 情况处置场景下的多智能体协作设计应用案例

3.1 智能代理角色定义

针对突发情况处置业务需求,以岛礁周边领海维权巡航事件处置案例为例,细分各业务层级角色职能。本案例定义了3种角色:首长、值勤参谋和数据保障人员,每个角色都遵循React-style行为,并监视环境以便及时发现重要信息。其中,担任值勤参谋的执行代理收到某中心的报告后,立即标绘海上情况图,收集相关情况,查找处置依据和以往处置案例,结合当前战略形势,分析研究情况,形成判断结论,提出处置建议。担任某指挥机构值勤首长的执行代理做出强势驱离的命令指示,基于当前国家战略利益,争取外交主动。担任值勤参谋的执行代理及时跟踪情况,协调处理情报、外交等工作,确保驱离行动始终有统一的方向和目标。行动结束后,及时组织参与要素进行复盘总结,梳理成功经验,查找存在的问题,拟制总结报告,形成处置案例。

以岛礁周边领海维权巡航事件处置为例,包括掌握情况、研定决心、组织处置和复盘总结4个阶段。表1详细描述了不同角色在各阶段的工作及职责。

表1 不同角色于各阶段的工作及职责

Table 1 Roles and responsibilities of different characters at different stages

序号	角色	阶段	信息
1	某指挥机构值勤首长/主任	掌握情况	异常情况掌握、首长指示
2		研定决心	研判方案合理性、首长指示
3		组织处置	首长指示
4		复盘总结	首长指示
5	某指挥机构值勤参谋	掌握情况	异常情况掌握、研判结论
6		研定决心	处置决心建议、研判方案合理性
7		组织处置	提出调整处置行动建议
8		复盘总结	总结经验教训、复盘总结报告
9	某指挥机构保障人员	掌握情况	异常情况掌握、研判编队动向、跟监任务执行情况
10		研定决心	查阅历史经验、研判海上形势
11		组织处置	跟踪敌舰船情况、跟踪监控舆论动向
12		复盘总结	整理资料、生成案例知识

单个智能体是基于大模型的智能体,由大模型经过面向任务的微调和训练得到,具备使用工具和文本生成的能力。本案例中,启元实验室提供的九格大模型满足输入文本长度要求,在参数量为70 B左右的量级时,仍然具备良好的逻辑推理能力和涌现能力,此外在军事领域有较多基于九格大模型的军事项目落地应用案例,因此选择九格大模型作为智能代理的基础大模型。

针对值勤首长、值勤参谋和保障人员等不同角色,对3种角色的大模型智能代理进行PEFT(Parameters-efficent fine tuning)。相较于全量微调(Full fine tuning, FFT),只调整部分参数的PEFT能够避免全量微调带来的灾难性遗忘,同时训练成本相对较低,效果明显。以值勤参谋角色的智能代理为例,在基础的九格大模型上,使用P-tuning v2方法对大模型进行训练和微调(图7)。采用的训练数据包括掌握情况阶段收到的上报文书和研判结论报告,研定决心阶段的处置建议、研判方案,组织处置阶段的行动建议和复盘总结阶段的复盘总结报告等。

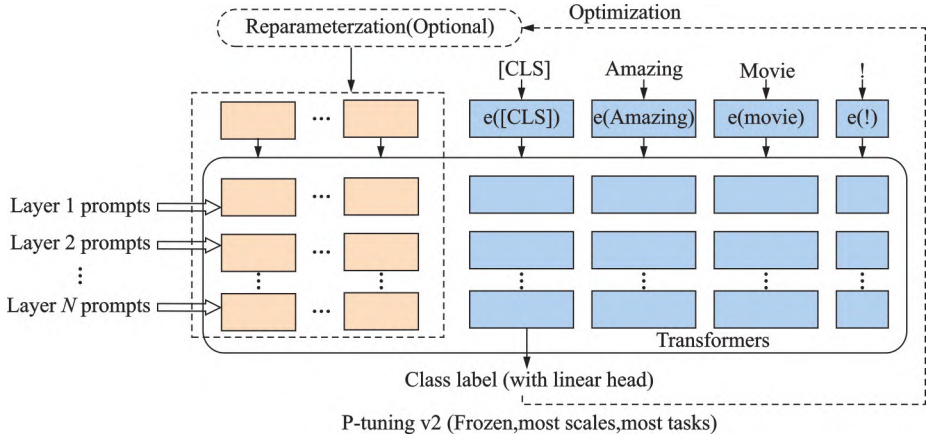

图7 P-tuning v2微调方法示例

Fig.7 Sample of P-tuning v2 micro-tuning method

本方法基于监督学习思想通过构造业务相关的prompt对模型部分参数进行更新,其中右侧深色部分的参数为已冻结参数,虚线框浅色部分为可训练参数。通过构造Reparameterization和任务标签(Class label)进行监督学习。其中,Reparameterization构造时,通过对比常用的MLP和Embedding方法,使用Embedding方法进行Reparameterization生成在不同任务下表现更好(图8)。根据经验,简单分类任务下提示词长度(Prompt length)选择20效果最好。

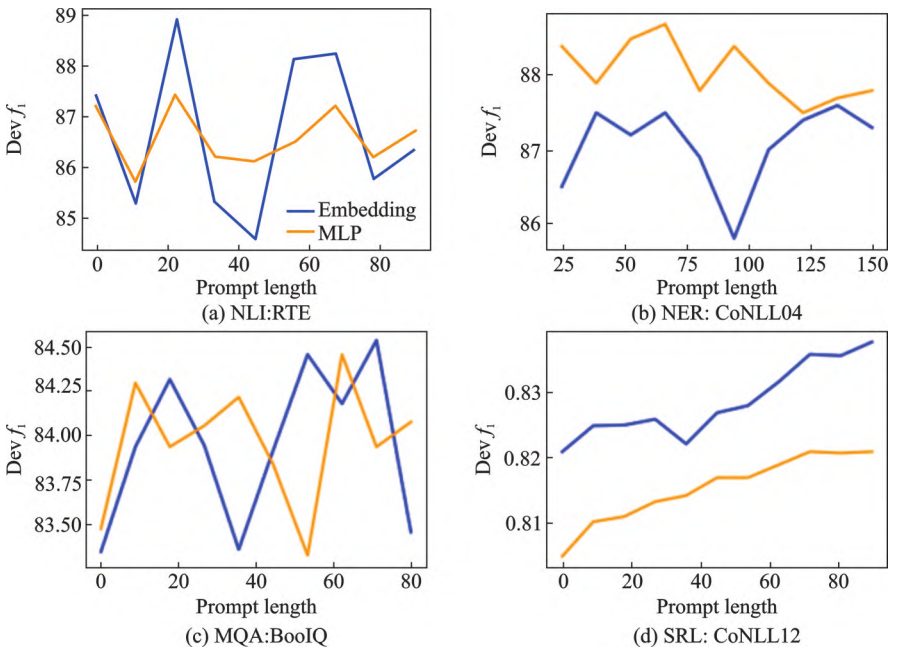

图8 Embedding方法和MLP方法在不同任务下生成Reparameterization的对比

Fig.8 Comparison of reparametrization generation using Embedding method and MLP method under different tasks

经过微调训练后,3种角色的智能体在各自任务场景下的文本序列预测和任务处置中效果显著提升。为验证本智能体代理训练后的有效性,与Meta-GPT进行对比消融实验,测试在突发事件处置场景下面向不同任务时的表现,实验结果如表2所示。其中,0表示任务失败,1表示基本可行,2表示基本符合预期,3表示完全成功。从实验结果来看,本文基于九格大模型训练微调后的训练情况联动处置智能代理框架能完成绝大部分场景下的处置任务,在全部值勤任务场景下表现都优于Meta-GPT。

表2 与Meta-GPT在处置任务下的对比实验结果

Table 2 Comparative experimental results with Meta-GPT under disposal tasks

任务	Meta-GPT	情况联动处置智能代理框架
掌握情况(值勤首长)	1	2
研定决心(值勤首长)	1	2
组织处置(值勤首长)	0	1
复盘总结(值勤首长)	1	2
掌握情况(值勤参谋)	1	0
研定决心(值勤参谋)	0	1
组织处置(值勤参谋)	0	1
复盘总结(值勤参谋)	1	2
掌握情况(保障人员)	1	1
研定决心(保障人员)	0	1
组织处置(保障人员)	1	1
复盘总结(保障人员)	1	2

3.2 多智能代理协同

在岛礁周边领海维权巡航事件处置场景中,如图9所示,几个事件处置单智能代理分别位于处置空间中的不同位置,每个单智能代理都是一

个基于大模型的智能体,都具备一定的逻辑推理能力和使用工具的能力。同时,由于不同席位的任务侧重和分工不同,因此每个席位的智能代理需要对大模型进行面向任务的针对性微调和训练,使其在本席位任务领域具有专家级别的能力。每个单智能代理都具备一定的能力,由于不同的智能代理部署在不同席位,导致单智能代理与中心决策节点难以进行有效通信,而各实体间只能进行有限的通信或者不能通信。因此,在进行决策时每个单智能代理都只能根据自身所能获取到的信息独立地进行决策。这种分布式的决策方式可能会带来一系列问题,例如由于没有中心决策节点来协调任务分配,各实体在进行独立决策时可能会出现多个单智能代理都选择去完成同一个任务,从而造成某些任务没有单智能代理来完成的现象。因此,利用多智能代理强化学习技术为每一个单智能代理都训练出来一个能够进行独立的分布式决策的策略网络,并且根据这些策略网络得到的智能代理策略,能够实现一定程度上的自协同。

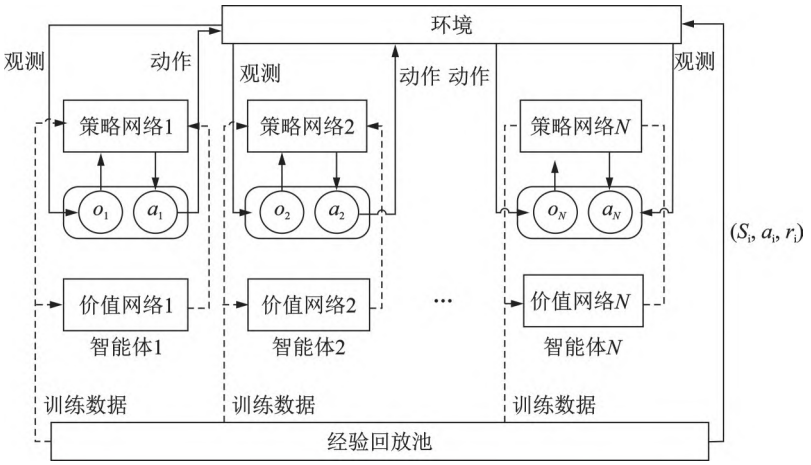

图9 多单智能代理协同任务示意图

Fig.9 Schematic diagram of collaborative tasks for multiple single intelligent agents

上述场景中的单智能代理决策过程,可以被建模为一个部分可观测的马尔可夫决策过程。场景中的每一个执行单元都可以被视为一个决策智能代理,智能代理的状态空间也即观测空间包含自身的位置信息和能力值信息、其他智能代理的位置信息,以及任务节点的位置信息和能力需求信息。智能代理的动作是选择哪一个任务节点作为自己的目标,因此智能代理的动作空间是离散的。

在上述岛礁类突发事件的任务分配场景中,任务分配的目标是体系中所有的任务节点都被分配了合适的单智能代理来完成,因此通过合作型的多智能代理强化学习手段,各智能代理共享一个相同的奖励值,相关奖励函数的设计可以根据任务节点的覆盖程度以及任务的完成效果来进行设计。

本文的岛礁类突发事件的任务分配场景属于通信受限条件下的体系“任务-席位”匹配的任务场景。在该任务场景中,设计体系中拥有相同数量的作战席位节点和作战任务节点,每个席位对应于一个单智能体,各席位和作战任务节点分别位于场景中一个随机生成的位置上,该位置的坐标在范围内随机生成,此外每个席位都拥有3种类型能力,各能力的取值采用均匀分布的形式在一定的数据范围内随机生成,同样的每个任务目标也有一定的能力需求对应于作战席位所能提供的3种能力,任务目标的能力需求也采用均匀分布的形式在一定的数据范围内随机生成。由于敌方通信干扰等因素的影响,各作战席位间不能进行通信,同时场景中也不存在一个中心决策节点来协调各作战单元的决策,因此各作战单元需要根据自身的状态信息和观测信息独立地进行决策,决策内容是选择哪一个任务目标作为自己

的目标节点。由于所设计的体系任务分配场景属于是合作型的任务，任务分配的目标是体系中所有的任务节点都被分配了合适的单智能代理来完成，各作战席位希望通过合作达到体系总体决策效果最优。因此，通过合作型的多智能代理强化学习手段，各智能代理共享一个相同的奖励值，将各作战单元的任务分配整体效果作为各智能体的奖励值。相关奖励函数的设计可以根据任务节点的覆盖程度以及任务的完成效果来进行设计：

(1)如果有任意一个任务节点没有被分配席位来完成，那么奖励值－5，任务节点的覆盖程度越低，则智能体所获得的奖励值越低。

(2)任务完成的效果可以根据席位与任务节点的距离以及席位的能力取值与任务实体的能力需求的匹配程度来确定。席位与任务节点的距离越小，任务完成的时效性越高，智能体获得的奖励值相应也越高，同时任务节点的能力需求与席位所能提供的能力值匹配度越高，则任务完成的效果越好，相应的智能体所能获得的奖励值越多。

智能体所包含的信息可以用一个元组进行表示，其中(x_i,y_i)表示智能体当前所处的位置坐标，c_i表示智能体在能力上的取值，n为能力类型的数量。同时任务节点包含的信息也可以用一个元组来表示，(x_j,y_j)表示任务节点的位置坐标，c_1表示任务节点对能力1的需求。那么智能体i与任务节点j之间的距离可以根据两者的坐标计算得到，如式(3)所示。智能体与任务节点j的能力匹配值也可以根据式(4)计算得到，其中co_{ij}表示能力匹配系数。对于任意一项能力来说，智能体i所能提供的能力值与任务节点i的能力需求值之间的比值越大说明采用智能体来完成任务在该项能力上取得的效果越好，将各项能力的效果进行累加，可以得到完成该任务的整体效果评估结果，累加得到的取值越大，则该项任务的整体完成效果越好；同时考虑如果智能体所提供的所有能力值都大于该任务节点的需求值，那么表示该任务节点的所有需求都得到了较好的满足，则将上述累加得到的匹配值乘以一个系数，而如果有一项智能体所提供的能力值小于任务节点的需求值，则认为任务节点的需求没有得到很好的满足，因此将上述累加得到的匹配值乘以一个系数。

$$d_{ij}=\sqrt{\left(x_i-x_j\right)^2+\left(y_i-y_j\right)^2+\left(h_i-h_j\right)^2} \tag{3}$$

$$\mathrm{eff}_{ij}=\left(\frac{c_{i1}}{c_{f1}}+\frac{c_{i2}}{c_{f2}}+\cdots+\frac{c_{in}}{c_{fn}}\right)\times \mathrm{co}_{ij} \tag{4}$$

$$\mathrm{co}_{ij}=\begin{cases}2 & \forall 1\leqslant k\leqslant n\colon c_{ik}>1\\ \dfrac{1}{2} & \text{其他}\end{cases} \tag{5}$$

各智能体独立地进行决策后输出的决策结果共同构成一个完整的体系任务分配方案$a=(a_1,a_2,\cdots,a_N)$，其中a表示智能体i的决策结果，也即该智能体的目标任务节点的索引，N为智能体的数量。

各智能体奖励函数的设计如式(6)所示，其中rew_d为各智能体与任务节点距离的倒数，rew_e为个智能体与任务节点的能力匹配之和，n_0为没有被分配对应的作战单元的任务节点的数量。

$$\mathrm{rew}=\mathrm{rew}_\mathrm{d}+\mathrm{rew}_\mathrm{e}-5*n_0 \tag{6}$$

$$\mathrm{rew}_\mathrm{d}=\frac{1}{\sum_{i=1}^{N}d_{ij}} \tag{7}$$

$$\mathrm{rew}_\mathrm{e}=\sum_{i=1}^{N}\mathrm{eff}_{ij} \tag{8}$$

本次实验所有算法都采用Python进行实现，并在同一台配置了GeforceRTX3090显卡、Intel

16-Core i9-11900K CPU的计算机上运行。其中多智能体强化学习算法的体系任务分配模型网络主要超参数如表3所示。为验证多智能体强化学习算法的有效性,与当前实际场景下值勤参谋的作业流程相对比,列出完成相同任务的情况掌握、研判以及任务分配中各阶段所需时间。

表3 模型网络超参数

Table 3 Parameters of model networks

参数名称	参数取值
价值网络层数	4
价值网络隐藏层	64
价值网络学习率	0.01
策略网络层数	4
策略网络隐藏层	64
策略网络学习率	0.01
训练周期	80 000
最大训练步数	1
更新周期	800
参数更新批次大小	1 024
策略网络优化器	Adam
价值网络优化器	Adam

分析实验结果,如图10所示,横坐标表示训练的回合数,纵坐标表示智能体得到的平均奖励值。可以看到,随着训练进程的推进,采用集中式训练的多智能体强化学习算法进行训练的智能体所得到的奖励值不断增大,最终稳定在0.6左右的水平,曲线收敛。在模型训练刚开始时,智能体所得到的奖励值小于0,也就是智能体还没有学会与其他智能体进行任务协同分配,导致体系的任务分配出现有的任务被多个智能体选择而有的任务没有被选择的现象,而随着训练进程的推进,由于环境反馈作用的影响,智能体逐渐学会了与其他智能体进行任务协同分配,即使在没有中心决策节点进行协调的情况下,各智能体依然能够根据自身的状态信息和观测到的信息采用分布式决策的方式独立地做出使体系效能最大的任务分配方案。

与传统任务分配手段相比,本算法完成各子任务的时间从传统的小时级缩短到分钟级,大大提高了面对突发事件时的反应速度和处置效率。智能代理框架和传统方法完成任务分配的结果如表4所示。

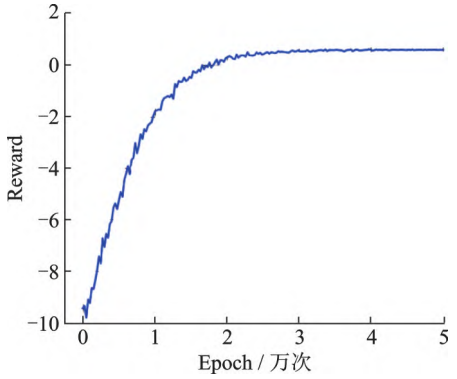

图10 多智能体强化学习算法训练的智能体平均奖励曲线

Fig.10 Reward curve of average agent rewards for multi-agent reinforcement learning algorithms trained on various tasks

表4 智能体代理框架和传统方法完成任务分配结果对比

Table 4 Comparison of task allocation results between intelligent agent framework and traditional methods

子任务	智能体代理强化学习方法完成分配时间/ min	传统方法完成分配时间/ min
情况掌握	2	30
情况研判	3	40
事件研讨	5	70
处置建议	2	20
复盘分析	4	80

3.3 多智能体协同工作流

值勤首长在获得突发事件信息后,进行深入分析,制定详细的组织处置方案分解,其中包括每个席位执行什么功能。然后将结构化的执行任务传递给开发人员,将需求转化为系统设计组件,例如文件

列表、数据结构和接口定义。一旦在系统设计中捕获,信息就会直接发送给对应的值勤参谋以进行任务分配。开发人员继续按照概述执行指定的类和函数。在接下来的阶段,数据保障人员制定相关的方案来完成数据获取,整理资料并生成案例知识。工作流示意如图11所示。

图11 面向突发事件处置的SOP工作流示意图

Fig.11 Diagram of SOP workflow emergency response

4 结束语

本文研究了基于大模型的多智能代理协同框架在情况联动处置业务中的应用,旨在简化多席位协同任务的复杂性。通过对大模型智能代理框架的改造和专门设计,探讨了多智能代理的自组织协同机制,提出了智能代理角色生成方法、优化动作处置算法和降低通信开销的设计,并通过案例验证了其有效性。该框架可辅助指挥员自动理解和规划处置任务,调用现役系统功能接口,提升应对突发情况的效率。

参考文献:

[1] 夏润泽,李丕绩.ChatGPT大模型技术发展与应用[J].数据采集与处理, 2023, 38(5): 1017-1034.
XIA Runze, LI Piji. Large language model ChatGPT: Evolution and application[J]. Data Acquisition and Processing, 2023, 38(5): 1017-1034.

[2] 罗锦钊,孙玉龙,钱增志,等.人工智能大模型综述及展望[J].无线电工程, 2023, 53(11): 2461-2472.
LUO Jinzhao, SUN Yulong, QIAN Zengzhi, et al. Review and prospect of artificial intelligence big models[J]. Radio Engineering, 2023, 53(11): 2461-2472.

[3] 孙柏林.ChatGPT:人工智能大模型应用的千姿百态[J].计算机仿真, 2023, 40(7): 1-7.
SUN Bolin. ChatGPT: The various applications of large artificial intelligence models[J]. Computer Simulation, 2023, 40(7): 1-7.

[4] 王静仪.大型语言模型技术的影响,挑战与应对策略[J].华东科技, 2023(6): 96-98.
WANG Jingyi. Impact, challenges and countermeasures of large language model technology[J]. East China Science and Technology, 2023(6): 96-98.

[5] HONG S, ZHENG X, CHEN J, et al. MetaGPT: Meta programming for multi-agent collaborative framework[J/OL]. (2023). https://arXiv preprint arXiv:2308.00352.

[6] CHEN G, DONG S, SHU Y, et al. Autoagents: A framework for automatic agent generation[J/OL]. (2023). https://arXiv preprint arXiv:2309.17288.

[7] GUO T, CHEN X, WANG Y, et al. Large language model based multi-agents: A survey of progress and challenges[J/OL]. (2024). https://arXiv preprint arXiv:2402.01680.

[8] WANG L, MA C, FENG X, et al. A survey on large language model based autonomous agents[J/OL]. (2023). https://arXiv preprint arXiv:2308.11432.

[9] BEINEMA T, DAVISON D, REIDSMA D, et al. Agents united: An open platform for multi-agent conversational systems [C]//Proceedings of the 21st ACM International Conference on Intelligent Virtual Agents. [S.l.]: ACM,2021: 17-24.

[10] WU Q, GAGAN B. AutoGen: Enabling next-gen LLM applications via multi-agent conversation[J/OL]. (2023). https://arXiv preprint arXiv: 2308.08155.

[11] MA Y J, LIANG W, WANG G, et al. Eureka: Human-level reward design via coding large language models[J/OL]. (2023). https://arXiv preprint arXiv: 2310.12931.

[12] CHEN Yang. Legal risks of conversational artificial intelligence and its systematic governance[J]. Dispute Settlement, 2024, 10: 493.

[13] CHEN Zhuo. Analysis on the development of product design major in the context of AIGC era[J]. Advances in Education, 2024, 14: 516.

[14] 曾炜,苏腾,王晖,等.鹏程·盘古:大规模自回归中文预训练语言模型及应用[J].中兴通讯技术, 2022(2): 28.
ZENG Wei, SU Teng, WANG Hui, et al. Pengcheng Pangu: Large-scale autoregressive Chinese pre-training language model and its application[J]. ZTE Communications Technology, 2022(2): 28.

[15] ZUO S, ZHANG Q, LIANG C, et al. Moebert: From bert to mixture-of-experts via importance-guided adaptation[J/OL]. (2022). https://arXiv preprint arXiv: 2204.07675.

[16] ZHANG Wei, GUO Hongcheng. mABC: Multi-agent blockchain-inspired collaboration for root cause analysis in micro-services architecture[J/OL]. (2024). https://arXiv preprint arXiv: 2404. 12135.

[17] 张庆海,陈霖.基于规则引擎的事件集成框架[J].指挥信息系统与技术, 2016. DOI:10.15908/j.cnki.cist.2016.05.015.
ZHANG Qinghai, CHEN Lin. Event integration framework based on rule engine[J]. Command Information Systems and Technology, 2016. DOI:10.15908/j.cnki.cist.2016.05.015.

[18] 包林波,季新源,陈希林,等.空中异常情况网络计划处置方法[J].兵工自动化, 2015, 34(7): 4.
BAO Linbo, JI Xinyuan, CHEN Xilin, et al. Network planning method for abnormal situations in the air[J]. Ordnance Automation, 2015, 34(7): 4.

[19] 刘金星,佟明安.多智能体作战飞机协同空战指挥控制的若干技术问题[J].电光与控制, 2007, 14(3): 5.

LIU Jinxing, TONG Ming'an. Several technical issues on cooperative air combat command and control of multi-agent combat aircraft[J]. Electro-Optics and Control, 2007, 14(3): 5.

[20] WANG Hongjun , CHI Zhongxian. The research on fleet jamming plan decision-making based on collaboration[J]. Systems Engineering-Theory & Practice, 2007, 27(4): 171-176.

[21] HUA W, FAN L, LI L, et al. War and peace (waragent): Large language model-based multi-agent simulation of world wars[J/OL]. (2023). https://arXiv preprint arXiv: 2311.17227.

[22] YAO S, ZHAO J, YU D, et al. React: Synergizing reasoning and acting in language models[J/OL]. (2022). https://arXiv preprint arXiv:2210.03629.

[23] ZHOU A, YAN K, SHLAPENTOKH-ROTHMAN M, et al. Language agent tree search unifies reasoning acting and planning in language models[J/OL]. (2023). https://arXiv preprint arXiv: 2310.04406.

[24] LI Wenhao, JIN Bo, WANG Xiangfeng, et al. F2A2: Flexible fully-decentralized approximate actor-critic for cooperative multi-agent reinforcement learning[J/OL]. (2023). https://arXiv preprint arXiv: 2310.11145.

作者简介:

吴晓宁(1987-),**通信作者**,女,高级工程师,研究方向:大数据与人工智能,E-mail:568712145@qq.com。

李瑞欣(1992-),女,工程师,研究方向:人工智能与大模型,E-mail:dearxin032@163.com。

王浪(1998-),男,工程师,研究方向:机器学习与大模型。

刘文杰(1997-),男,助理研究员,研究方向:大模型与数据挖掘。

王宏伟(1987-),男,高级工程师,研究方向:人工智能与强化学习。

朱新立(1992-),男,工程师,研究方向:数据处理。

宋江帆(1998-),男,助理工程师,研究方向:强化学习。

袁梦(2001-),女,硕士研究生,研究方向:电子与通信工程。

(编辑:夏道家)