面向多机器人环境中动态异构任务的细粒度动作分配与调度方法

王积旺，沈立炜

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.220500117

面向多机器人环境中动态异构任务的细粒度动作分配与调度方法

王积旺　沈立炜
复旦大学计算机科学技术学院　上海 201203
上海市数据科学重点实验室(复旦大学)　上海 201203
(fdwjw@qq.com)

摘　要　在多机器人环境中,具有不同能力的机器人相互协作以完成任务需求。现实情况下,这些任务动态发布,且具有不同的目标和紧急程度,因此需要为每个任务分解出的细粒度动作分配和调度合适的机器人来负责执行这些动作。现有的方法大多适用于静态和同构的任务分配场景,而针对动态异构任务的分配则大多采用独占式的分配策略,导致机器人频繁进入等待状态(即机器人处于被分配了任务到真正开始执行任务之间的闲置阶段)。由于任务存在不同的紧急程度和发布时间,这种分配方式将降低对更紧急任务的响应效率,同时导致更多的等待时间和更长的任务完成时间。针对该问题,提出了一种面向多机器人环境中动态异构任务的细粒度动作分配与调度方法。其中,分配与调度的对象是任务所分解出的细粒度的动作,且一个动作能够由机器人的一种能力承担。面对任务分解出的一组细粒度动作集合,本方法借鉴拍卖算法过程,根据机器人能力、状态及任务信息计算出机器人承担特定动作的最优分配方案。另外,在每一次新任务发布或某一机器人执行完动作时执行分配和调度过程,可以将处于普通任务等待状态的机器人调度至紧急任务,以保证紧急任务优先完成,且缩短机器人的总体等待时间。基于本方法,扩展实现了机器人执行框架(ROSPlan)的执行模块。围绕一组多机器人动态异构任务的模拟实验表明,所提方法相较于采用贪心策略的方法可得到更优的分配方案。

关键词:多机器人;动态异构任务;动作分配与调度;拍卖算法;ROSPlan
中图法分类号　TP311

Fine-grained Action Allocation and Scheduling Method for Dynamic Heterogeneous Tasks in Multi-robot Environments

WANG Jiwang and SHEN Liwei
School of Computer Science,Fudan University,Shanghai 201203,China
Shanghai Key Laboratory of Data Science(Fudan University),Shanghai 201203,China

Abstract　In a multi-robot environment,robots with different capabilities collaborate with each other to complete task requirements. Realistically,these tasks are dynamically issued and can have different goals and urgency levels,so it is necessary to allocate and schedule the appropriate robots responsible for executing the fine-grained actions decomposed for each task. Most of the existing approaches are suitable for static and homogeneous task allocation scenarios,while most of the dynamic heterogeneous tasks are assigned using exclusive allocation strategies,which causes the robot to frequently enter into waiting states(i.e.,robots are in the idle phase between being assigned a task and actually starting to execute it). Since tasks vary in urgency levels and release times,this allocation method will reduce the efficiency of response to more urgent tasks,while leading to longer waiting time and task completion time. To address this problem,this paper proposes a fine-grained action allocation and scheduling method for dynamic heterogeneous tasks in a multi-robot environment. In this paper,the object of allocation and scheduling is a fine-grained action decomposed by a task,and an action can be undertaken by one capability of a robot. Faced with a set of fine-grained actions decomposed by the task,this method draws on the auction algorithm process to calculate the optimal allocation scheme for a robot to undertake a specific action based on the robot capabilities,state and task information. In addition,by executing the allocation and scheduling process at each new task release or when a robot finishes executing an action,robots in the general task waiting state can be scheduled to the urgent task to ensure the priority completion of the urgent task and reduce the overall waiting time of the robot. Based on this approach,the execution module ROSPlan is extended and implemented. Simulation experiments around a set of multi-robot dynamic heterogeneous tasks show that the proposed methodr can obtain a better allocation scheme compared

到稿日期:2022-05-16　返修日期:2022-07-05
基金项目:上海市级科技重大专项(2021SHZDZX0100)
This work was supported by the Shanghai Science and Technology Major Special Project(2021SHZDZX0100).
通信作者:沈立炜(shenliwei@fudan.edu.cn)

to the method using greedy policies.

Keywords Multi-robot, Dynamic heterogeneous tasks, Action allocation and scheduling, Auction algorithms, ROSPlan

1 引言

当前，机器人已经在工业、服务业等领域得到广泛应用。在某些应用场景中，多种具有不同能力的机器人需要进行相互协作才能完成指定的任务，例如物体的搬运与运输、紧急救援等。为此，多机器人场景下的任务执行需要考虑机器人间的合作与竞争[1-2]、通信与沟通[3-4]以及任务规划[5-6]等问题。其中，面向多机器人的任务规划是保证任务完成的关键手段。任务规划一般被划分为任务分解和任务分配[7]两个阶段。多机器人任务分解（Multi-Robot Task Decomposition，MRTD）根据任务特点和约束将待完成的任务分解为多个细粒度的子任务，每个子任务由单个机器人独立承担[8]。任务分解一般由领域专家或设计人员手工完成[9]。在此基础上，多机器人任务分配（Multi-Robot Task Allocation，MRTA）可被看作最优分配问题的一个实例：存在多个代理和多个任务，代理可被分配去执行其能够完成的任务，执行任务的过程会消耗一些成本，这些成本会根据分配方案的不同而有所不同。在多机器人场景任务中，任务分配即给定一个多机器人系统和一个任务集合（即分解得到的一组细粒度子任务），为每个子任务寻找合适的机器人来负责执行该子任务[10]。

现实情况下的任务往往具有动态性和异构性。首先，动态任务是指多机器人系统无法预先获知所有任务的详细信息，即任务是随机产生的[11]。例如，在酒店餐饮领域，越来越多的商家使用机器人辅助送菜，由于食客的点单时间无法预知，因此配餐送餐任务也无法提前预知。相对而言，静态任务是指在任务分配开始前就已经确定了所有待分配的任务，并且任务数量不会增加或者改变。其次，异构任务指一个多机器人系统所接收的任务在目标和特点上具有差异。本文所针对的任务异构性主要体现在两个方面：1）不同的任务目标导致不同的子任务被分解，即涉及具有不同能力的机器人；2）不同的任务紧急程度导致出现不同的任务分配和调度决策，即更紧急的任务应该被优先、及时处理。

面对多机器人任务分配问题，大部分工作提出了针对静态任务的分配方法。例如，使用拍卖算法对给定的一组机器人和一组任务，执行一次具有多轮次投标的拍卖过程，以确定分配方案[12]。在每一轮投标中，所有机器人投标所有未分配的任务，特定的任务将被分配给出价最高的机器人。所有任务都被分配给机器人后，投标过程结束[13]。它们不适用于多机器人动态任务分配的场景。另外，也有许多工作针对同构的一组任务进行分配。例如，Khamis 等[14]和 Gong 等[15]同样把拍卖算法用于解决诸如移动等单一类型的任务分配问题。

动态异构任务的分配是一个随时间和环境变化的动态决策问题[16]，其复杂度进一步提升。针对该问题，已有的工作将拍卖算法和机器人能力模型结合来将异构任务分配给不同能力的机器人，这类算法一般都采取了贪心的策略[17-18]。然而，这些解决方案大多采用独占式的分配策略，即一旦给某一个机器人分配到任务，在完成任务之前，新发布任务的分配都不再考虑它。此时，机器人容易频繁进入等待状态，即机器人处于被分配了任务到真正开始执行任务之间的闲置阶段。由于任务存在不同的紧急程度和发布时间，这种分配方式将降低对更紧急任务的响应效率，同时也降低了总体任务的完成效率（即更长的等待时间和任务完成时间）。

针对以上问题，本文提出了一种面向多机器人环境中动态异构任务的细粒度动作分配与调度方法。在本方法中，分配与调度的对象是任务所分解出的细粒度的动作，且一个动作能够由机器人的一种能力承担。具体而言，面对任务分解出的一组细粒度动作集合，本方法借鉴拍卖算法过程，根据机器人能力、状态及任务信息扩展算法的输入和计算过程，通过竞价和分配的迭代计算出机器人承担特定动作的最优分配方案。在此基础上，针对一组动态发布且具有不同目标和紧急程度的任务，本方法在每一次新任务发布或某一个机器人执行完动作时，根据实时的多机器人状态执行分配和调度过程。该方法可以将处于普通任务等待状态的机器人调度至紧急任务，以保证紧急任务优先完成，从而提高任务的总体完成效率，缩短机器人的总体等待时间。基于本方法，本文扩展实现了机器人执行框架（ROSPlan）的执行模块，使得该模块可以支持多机器人动态异构任务场景下的动作分配和有序调度。最后，为了验证本文算法的有效性，本文基于贪心策略设计了一个对比算法，并使用本文方法和对比算法执行了 Gazebo 仿真实验和数据模拟实验。实验结果表明，本文方法可以计算得到更优的动作分配方案，并有效减少机器人的等待时间。

2 相关工作

Gerkey 等[19]从 3 个维度描述 MRTA 问题，并给出了如下定义。

（1）单任务机器人（Single-task Robots，ST）与多任务机器人（Multi-task Robots，MT）：单任务机器人表示每个机器人在一个时刻最多只能执行一个任务；多任务机器人表示每个机器人在一个时刻能够同时进行多项任务。

（2）单机器人任务（Single-robot Tasks，SR）与多机器人任务（Multi-robot Tasks，MR）：单机器人任务表示只需要一个机器人就能完成的任务；而多机器人任务表示需要多个机器人参与才能完成的任务。

（3）即时分配（Instantaneous Assignment，IA）与扩展分配（Time-Extended Assignment，TA）：即时分配意味着有关机器人、任务和环境的可用信息只适用于将任务瞬时分配给机器人，只考虑单个任务而不考虑其他任务；扩展分配意味着可以获得更多的任务信息，比如需要分配的所有任务的集合以及每个任务的发布时间等信息。

基于这 3 个维度，可以将 MRTA 问题分为 8 类：ST-SR-

IA, ST-SR-TA, ST-MR-IA, ST-MR-TA, MT-SR-IA, MT-SR-TA, MT-MR-IA, MT-MR-TA。目前研究最多的是 ST-SR-IA 问题，即给定一组任务集合 T、一组机器人集合 R 和每个机器人执行任务的代价函数，找到一个任务分配方案使得全局代价函数的数值最小[20]。虽然学者们针对 ST-SR-IA 问题提出了遗传算法[21]和蚁群系统[22]等启发式算法，但它们不适合动态环境。只有少数 MRTA 问题采用纯粹的 ST-SR-IA 一次性分配结构，大多数都采用它的 2 个变形结构：迭代分配和在线分配。在迭代分配结构下，所有任务是同时分配的，典型方法包括 BLE[23]和 M+[24]等。Werger 基于 PAB(Port-Arbitrated Behavior) 交互技术设计了 BLE(Broadcast of Local Eligibility)，任务发布后，机器人根据自身能力确定效用值，而任务将被分配给效用值最高的机器人。但是该方法只在多机器人多目标观测任务中有很好的控制效果，其在其他任务场景中的效果并未得到验证。M+是第一个基于市场交易机制实现多机器人任务分配的算法，它需要一个机器人扮演任务管理者的角色，将任务作为公告信息发布在多机器人系统中，收到信息的机器人计算完成该任务的效用值，并将效用值报告给任务管理者，任务管理者根据效用值将任务分配给最合适的机器人。由于迭代分配不考虑临时发布的新任务，因此该办法不适用于本文所讨论的动态异构任务分配的场景。

本文的研究内容属于在线分配。在线分配采用串行分配任务的策略[25]，典型方法包括 ALLIANCE[26]，MURDOCH[27]和 First-price auctions[28]。Parker 设计了一个面向多个异构移动机器人合作控制的软件结构 ALLICANCE，ALLICANCE 采用一种自适应动作选择机制来实现多机器人任务的分配。该选择机制基于不耐烦(Impatience)和默许(Acquiescence)两个数学模型，它们作为激活开关选择相应的任务动作。该方法具有较强的自适应能力和容错能力，对于系统中动态添加任务的情形也能够自适应运行。但是 ALLICANCE 难以实现复杂任务的分配，并且系统完成任务的效率较低。MURDOCH 和 First-price auctions 这两个方法均借鉴了拍卖算法的思想，本质上采取贪心的策略。在这类方法中，每个任务总是由当时最有能力的机器人来认领，尽管有足够的机器人资源来完成一组给定的任务，但是任务的不同呈现顺序导致机器人资源可能以非最优的方式被利用，因此仍然存在优化空间。在多任务分配的场景下，以任务最少完成时间作为优化目标是推导出更优机器人分配方案的可行策略[29]。然而，在动态任务场景下，任务发布的时间是随机且未知的，晚发布的任务会延长所有任务的完成时间，因此，任务完成的总时间不适合作为本文的优化指标。本文将全部机器人的等待时间作为优化指标，将更短的等待时间作为本文的优化目标[30]。

相关学者对任务分配中的任务优先级展开了讨论，Han 等考虑到了任务具有不同优先级的属性，提前指定待分配任务的优先级，但是只讨论了同构机器人和静态任务的场景[31]。Elango 等虽然考虑到了动态任务和任务的优先级属性，但是其研究的任务仅仅是移动这一简单类型任务，未考虑需要多个机器人参与才能完成的复杂任务，也没有考虑到机器人任务的切换[32]。为了满足高优先级的任务先被执行的需求，Das 等通过重新分配正在执行的低优先级任务来释放被占用的机器人资源[33]。这些工作虽然讨论了需要将任务拆分为子任务并对子任务进行分配，但没有给出具体办法。Elsefy[34]和 Zheng 等[35]基于复杂任务的线性时序逻辑(Linear-time Temporal Logic, LTL)描述，将复杂任务分解为层级子任务，并分配给同构机器人。不过其考虑的应用场景过于单一，只讨论了诸如地图探索等单一类型的任务。本文同时考虑了不同类型复杂任务的任务分解和任务分配，并给出了具体可行的设计方案。

3 方案设计

3.1 拍卖相关概念

拍卖(Auction)概念来源于经济学，是人类社会中一种常见的经济现象，即卖方与买方遵照事先规定的拍卖规则达成物品交易[36]。受此启发，本文提出了“积分”的概念，从而将实际生活中的拍卖映射到多机器人动态异构任务分配的领域，将任务拆解为相互独立的动作，并将这些动作作为拍品。机器人作为竞争者参与到拍卖中，对机器人而言，积分相当于现实世界中的金钱。机器人总是希望尽一切所能提高自己的积分，这意味着机器人只对能够给其带来净利润较多的动作感兴趣。如果同时存在多个净利润为正的动作，机器人首先选择竞拍净利润最大的动作。

表 1 给出了在多机器人动态异构任务分配领域中对拍卖的全新解读。首先，每个任务都有初始积分，初始积分与完成任务的预估时间有关。其次，针对由每个任务分解得到的动作，每个机器人的能力都有与之相应的预估积分，预估积分表示如果执行该动作，机器人将会获得的预期积分回报。预估积分与机器人特定能力承担动作执行的适合程度、任务的初始价格、任务的优先级和机器人距离任务实施位置的距离远近有关。例如，针对“夹取玻璃杯”场景，带有力反馈的机械手比普通的机械手适合程度更高，所以其预估积分更高。与生活中的拍卖类似，动作的执行权将会被分配给在竞拍环节中出价最高的机器人。机器人在竞拍过程中给出的价格表示机器人竞拍该动作付出的积分，也就是成本。预估积分减去成本，是机器人执行动作所能获得的净积分。对机器人而言，它总是希望竞拍到可以为其带来最大净积分的动作。

表 1 拍卖概念在多机器人任务分配领域的映射

Table 1 Mapping of auction concept in the field of multi-robot task allocation

	传统拍卖	多机器人动态异构任务分配领域
拍品	具有金钱价值的物体	具有积分价值的动作
竞拍者	人	机器人
收益	每个人对物体的金钱价值预估	每个机器人对动作的积分价值预估
成本	获得拍品付出的金钱	获得拍品付出的积分
净利润	拍品带来金钱上的增加	拍品带来积分上的增加

3.2 多机器人动态异构任务规划流程

多机器人动态异构任务规划主要分为发布任务、任务

分解、动作分配和动作执行 4 个步骤，详细流程如图 1 所示。黑色实线表示多机器人动态异构任务规划的主要流程，红线实线表示对分配任务步骤的触发，黑色虚线表示机器人根据实际情况适时地执行动作。

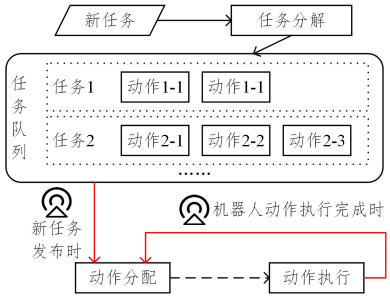


图 1 多机器人动态异构任务规划流程(电子版为彩图)

Fig. 1 Multi-robot dynamic heterogeneous task planning process

(1)发布任务：任务发布者向多机器人系统发布新的任务。对系统而言，任务是不断随机出现并增加的，系统无法事先获取所有的任务信息，不同任务是相互独立的。每当一个新任务被发布，系统就对其进行分解与分配。所发布的任务可以是由多机器人合作完成的复杂任务，也可以是由单机器人独立完成的简单任务。复杂任务是由一组动作组成的序列。以“递送咖啡”场景为例，需要移动机器人移动到咖啡机附近，机械手准确抓取咖啡杯，递送给移动机器人，移动机器人再移动到指定地点，从而完成任务。该复杂任务可以拆分为递送咖啡和夹取咖啡两个动作，分别由移动机器人和机械手完成。在机器人领域，PDDL(Planning Domain Definition Language)是用于描述机器人基本行为和任务目标的一种语言。基于 PDDL 的机器人程序可分为领域描述(Domain Description)和问题描述(Problem Description)两个部分。机器人的基本能力在领域描述中定义，任务目标则在问题描述中指定。诸如 ROSPlan[37] 等规划工具可根据 PDDL 程序将复杂任务分解成独立的动作序列，本文方法依赖于规划工具直接得到任务的动作序列。

(2)任务分解：将复杂任务拆解为相互独立的动作，这个步骤是借助 ROSPlan 实现的。在发布任务环节，任务发布者已经遵循 PDDL 语法规则将任务编写成域文件和问题文件。ROSPlan 将会使用 Knowledge Base，Problem Interface 和 Planner Interface 这 3 个节点，将复杂任务拆解为独立的动作组成任务队列。

(3)动作分配：将任务队列中的动作分配给合适的机器人。有两种事件会触发该流程，分别是新任务发布时和机器人动作执行完成时，如图 1 红线部分所示。当新任务发布时，机器人根据自身当前状态和能力信息主动竞拍这些动作，获胜的机器人将获得对应动作的执行权。机器人完成动作后，主动对未被分配的动作进行竞拍，从而保证当某些动作在临时找不到合适机器人的情况下最终也能被分配给合适的机器人。

(4)动作执行：根据分配方案驱动机器人依次执行其能力，从而完成任务。本文将任务划分为普通任务和紧急任务，正在执行普通任务的机器人，还可以参与紧急任务的竞拍。如果机器人成功竞拍紧急任务，机器人就暂停执行普通任务的动作，先完成紧急任务的动作后再恢复执行普通任务的动作。

4 模型构建

定义一个任务集合 $T=\{Task_1, Task_2, \cdots, Task_m\}$ 和一个机器人集合 $R=\{Robot_1, Robot_2, \cdots, Robot_m\}$，接下来分别对任务、机器人以及在算法中出现的数据结构进行详细解释。

4.1 任务模型

集合 T 表示所有任务，其中的每一个元素 $Task$ 是一个由任务序列号 $TaskID$、动作序列 $Action$、任务发布时间 $Time$、期望完成时间 $Deadline$、任务描述 $Description$ 和任务优先级 $Priority$ 构成的六元组，如式(1)所示：

$$Task=\langle TaskID, Action, Time, Deadline, Description, Priority\rangle \tag{1}$$

$TaskID$ 是任务序列号，用来标识任务，不同任务的序列号互不相同且唯一。$Action$ 表示动作序列，是由机器人动作构成的集合，每一项操作 $action$ 可以由机器人的能力在特定条件下实现。本文将执行动作的地理位置 l 作为约束条件，动作需要在地理位置 l 才能执行，l 是根据对复杂任务的分解得到的。

$$Action=\{action_1, action_2, \cdots\} \tag{2}$$

$$action=\langle capability, l\rangle \tag{3}$$

Time 是任务的发布时间，Deadline 是任务的期望完成时间，竞拍该任务的机器人需要在规定的截止时间前完成任务。Description 是任务的相关描述，它给出与任务相关的信息，例如，需要告知机器人到哪里获得咖啡，再将咖啡递交到哪里。任务发布者需要结合 Description 提供的信息，将其翻译为任务的 PDDL 形式化描述。如果任务对某个能力有更多的要求，例如需要机器人具有更快的移动速度或者更精细的夹取能力，那么该要求也被记录在 Description 中。Priority 是任务的优先级表示，如果任务为紧急任务，其值为 2，否则为 1。紧急任务需要尽快被执行，更高的 Priority 数值将会使任务的预估积分更高，吸引机器人优先执行紧急任务。

4.2 机器人模型

集合 R 表示环境中的机器人集合，其中的每一个元素 $Robot$ 是一个由能力集合 C、状态元组 S 和预估积分集合 V 构成的三元组，如式(4)所示：

$$Robot=\langle C, S, V\rangle \tag{4}$$

C 是机器人的能力集合，记录了机器人的具体能力。例如，$TurtleBot_2$ 的能力集合 C 是{移动，导航，巡检，…}。只有具备对应能力的机器人，才能成功竞拍到动作。

S 是机器人的状态元组，表示机器人的当前状态，并且能够实时更新。S 具体包含机器人的当前坐标 L_R、机器人当前是否可用的标记 A 和机器人参与的任务 T_{Robot}，如式(5)所示：

$$S=\langle L_R, A, T_{\text{Robot}}\rangle \tag{5}$$

L_R 记录了机器人当前的坐标信息，该数据实时更新。

A 的取值有 3 种，分别是“Available”“InNormal”和

"In-Emergency",其取值与机器人的状态有关。机器人的状态有4种:未被分配动作、等待、正在执行普通任务的动作和正在执行紧急任务的动作。当机器人未被分配动作或者处于等待状态时,可用标记A为Available,表示机器人可以参与普通任务与紧急任务的竞拍;当机器人正在执行普通任务的动作时,A为InNormal,表示机器人不再参与其他普通任务的竞拍,但是可以参与紧急任务的竞拍;当机器人正在执行紧急任务的动作时,A为InEmergency,表示机器人不再参与任何任务的竞拍,保证了紧急任务可以优先被执行。为了方便验证算法的可行性,本文暂不考虑机器人出现故障和电量等状态。

T_{Robot}表示机器人当前参与的任务集合,集合元素是$TaskID$。

V是机器人的预估积分集合,包含了机器人对不同动作的预估回报积分,每个元素v_{action}表示机器人对具体某个$action$的预估积分数值。假定action是移动,v_{action}表示机器人如果完成移动动作就可以获得数量为v_{action}的积分。如果任务发布者在$Description$对移动能力提出了要求,那么具有更高移动能力的机器人就会给出更高的v_{action}。

$$V=\langle v_{\text{action}_1}, v_{\text{action}_2}, \cdots\rangle \tag{6}$$

4.3 相关数据结构

本文定义了在机器人分配过程中出现的数据结构$Price$和$Assignment$,这些数据结构将在算法中被使用。因为一次拍卖可能有多轮竞价,所以这些数据结构有助于追踪算法的执行情况。

$$Price=\{price_{action_1}, price_{action_2}, \cdots\} \tag{7}$$

$Price$是当前竞价的动作价格集合,记录每个动作的当前价格。假设一个任务被分解为移动和夹取两个$action$,那么$Price$记录这两个动作的当前价格。当拍卖环节结束,$Price$记录的价格则为动作最终价格。

$$Assignment=\{action_1:robot_1, action_2:robot_2, \cdots\} \tag{8}$$

动作的分配方案由$Assignment$表示,其中的每一个元素是动作与机器人键值对,记录了每个动作及其分配的机器人。与$Price$类似,$Assignment$随着每一轮次拍卖的进行而不断改变,当拍卖结束后,$Assignment$记录的方案则为最终确定的方案。

5 算法设计与实现

5.1 普通任务与紧急任务

本文将任务分为普通任务与紧急任务。紧急任务是指必须立刻完成的任务,它的优先级比普通任务高。因此,任务发布者发布紧急任务时,必须立刻找到合适的机器人来完成。能越快到达任务地点并完成任务的机器人被认为是合适的机器人,但是可能存在机器人此时恰好正在执行其他动作的情况。在实际生活中,暂停正在执行的低优先级任务,先完成高优先级任务是处理紧急任务的合理办法。因此,本文使用机器人状态元组S中的可用标记A来帮助讨论该问题,并作出如下阐述。

(1)当A的取值为Available时,表示机器人没有被分配动作,或者机器人此时处于等待状态。这两种状态下的机器人可以被认为是完全可用的机器人,因此可以直接参与普通任务与紧急任务的竞拍。

(2)当A的取值为InNormal时,表示机器人已经竞拍到动作,并且正在执行动作,而且该动作是属于普通任务。换句话说,处于该状态下的机器人,不再参与其他普通任务的竞拍,但是可以参与紧急任务的竞拍。如果成功竞拍到紧急任务,那么机器人就会停止当前正在进行的普通任务动作,先执行紧急任务的动作。等完成紧急任务的动作后,再恢复原先任务动作的执行。

(3)当A的取值为InEmergency时,表示机器人成功竞拍到紧急任务,并且此时正在执行紧急任务的动作。机器人执行完紧急任务后,如果还有之前竞拍的普通任务未执行完,则继续执行普通任务。

综上所述,当机器人未被分配动作或者机器人处于执行普通任务动作这两种状态时,机器人可以竞拍紧急任务的动作。当机器人可以竞拍紧急任务的动作时,只要能够获得比当前状态更高的净积分,机器人就会选择先执行紧急任务。

5.2 预估积分的计算

每个机器人对动作都有不同的预估积分v_{action},该预估积分与机器人特定能力承担动作执行的适合程度c、动作的初始价格$price_{\text{base}}$、任务的优先级$Priority$和机器人距离动作实施位置的距离d有关。

$$v_{\text{action}}=c*price_{\text{base}}*Priority-d \tag{9}$$

如果机器人不具备完成该动作所需要的能力,则c为0,这会导致v_{action}的值为负数,此时机器人会放弃竞拍该动作,因为机器人只想竞拍高净积分的动作;$price_{\text{base}}$是动作的初始积分,与任务的完成时间有关,由发布者指定;$Priority$是任务的优先级;d是机器人当前位置距离特定动作指定实施位置的距离,距离越远,预估积分v_{action}越小。

5.3 加价策略

假设机器人i想与其他机器人竞争动作j,因为完成动作j可以得到最大的净积分,净积分可以由机器人对该动作的预估积分v_{action}和该动作的当前价格$price_{\text{action}}$相减得到。机器人i将会在动作j的当前价格$price_{\text{action}}$上加价r_i得到新的竞价$price_{\text{new}}$。r_i是机器人在每轮拍卖中的竞价增量,其数值是机器人i所能得到的最高净积分和次高净积分之差。

$$price_{\text{new}}=price_{\text{action}}+r_i \tag{10}$$

$$r_i=\max\{V_i-Price\}-second\{V_i-Price\} \tag{11}$$

机器人都希望竞拍到净积分最高的动作,假设$action_1$对机器人i而言是净积分最高的动作,$action_2$是净积分次高的动作。机器人i对$action_1$最感兴趣,但是其他机器人也会对$action_1$竞价,如果$action_1$的价格不断上升,那么它给机器人i带来的净积分将会越来越小。当$action_1$的价格超过一定数值后,它就不再是净积分最高的动作。因此,机器人i不应该一直参与对$action_1$的加价,不合理的加价并不会让机器人i获得最大净积分。当$action_1$动作价格过高时,放弃继续加价,选择竞拍次高净积分动作$action_2$,才是正确的策略,因为此时原先次高净积分的动作$action_2$已经变成新的

净积分最高的动作。

该加价策略并不是最好的策略，如果使用该策略，有可能在拍卖中出现无效拍卖。所谓的无效拍卖，是指当存在多个动作为机器人提供相同的净积分时，竞价增量 r_i 为零。因此，可能会出现这样一种情况，即几个机器人争夺数量较少的动作，但却没有提高它们的价格，从而造成一个永无休止的循环。为了打破这样的循环，引入一个补偿量 ε，新的加价公式如式(12)所示：

$$r_i=\max\{V_i-Price\}-second\{V_i-Price\}+\varepsilon \tag{12}$$

增加了 ε 的加价策略更加符合现实生活中的拍卖。ε 是机器人在每次竞价中必须要提高的最小增量，其数值大小的选取需要遵循互补松弛定理，一般 $\varepsilon<\frac{1}{n}$，n 为环境中机器人的数量。当 ε 取值较大时，可以减少迭代次数，加快拍卖的速度。

5.4 竞拍过程与算法

当一个新任务被分解为一组动作后，就运行本文的竞拍算法进行动作分配，每个动作都有对应的初始化竞拍价格。算法以迭代的方式进行，如果所有的动作都功被拍卖，这个过程就会终止。拍卖规则如下：

(1)一开始将动作随机分配给具有对应能力的机器人，如果不存在多个机器人竞争一个动作的情况，可以认为该分配是最优解，流程结束。如果存在竞争情况，则进行步骤(2)。

(2)在动作的当前价格基础上，机器人加价 r_i。

(3)迭代步骤(2)，直到所有的动作都被分配到机器人。

分配算法的伪代码如算法 1 所示。算法依赖 4 项输入，分别是环境中的机器人集合 R、待分配的动作结合 $Action$、当前的分配方案 $Assignment$ 和当前的动作价格集合 $Price$。算法可以分为机器人竞价和确定当前获胜者两部分，机器人竞价部分为第 3－15 行，确定当前获胜者部分为第 16－21 行。

算法 1 多机器人动态异构任务分配拍卖算法

输入：环境中的机器人集合 R；待分配的动作集合 Action；当前的分配方案 Assignment；当前的动作价格集合 Price

输出：当前的分配方案 Assignment

```
1. function ALLOCATING(R,Action,Assignment,Price)
2.   foreach Robot in R do
3.       计算 Robot 的预估积分集合 V
4.     if Robot in Assignment then
5.       continue
6.     else
7.       Robot.profit={ }
8.       Robot.profit ← Robot.V-Price
9.       first_action ← max(Robot.profit)的动作
10.      second_action ← 除去 first_action,max(Robot.profit)的动作
11.      if first_action 当前价格 in(v_second_action, v_first_action) then
12.        Robot 对 first_action 的出价=first_action 当前价格+r_i
13.      end if
14.    end if
15.    end for
16.    foreach action in Action do
17.      if action 有对应的竞争者 then
18.        Price_action←max(所有竞拍者针对 action 给出的竞拍价格)
19.        Assignment←〈action:出价最高的竞争者〉
20.      end if
21.    end for
22. end function
```

在机器人竞价环节，首先确认本轮竞价的参与者，当前暂时竞拍到该动作的参与者不再参与报价。这符合拍卖规则，例如 A 在竞价中出价最高，只表示 A 获得了动作的暂时所有权，直到其他竞拍者结束竞价，才可以认为 A 最终竞拍到动作。既然 A 已经获得了动作的暂时所有权，就无需主动加价，因为竞拍者总是希望以较小的成本获得动作。随后，其他的竞拍者可以发起竞价，每个竞拍者根据上文提到的加价策略，给出新的竞价来竞争动作，从而更新动作价格 $Price$。确定当前获胜者环节的核心思想是分别找出每个动作出价最高的竞拍者，将该最高价赋值给当前动作，并认为该竞争者是本轮竞拍的优胜者。

分配算法更新了当前的分配方案 $Assignment$ 和当前的动作价格集合 $Price$，经过一轮又一轮地迭代运行分配算法，所有的动作都被分配到合适的机器人，从而得出最终的动作价格与竞拍成功的竞拍者。

5.5 扩展 ROSPlan 的任务执行

ROSPlan 框架为 ROS 系统中的 AI 规划提供了一组工具。ROSPlan 有各种各样的节点，它们封装了计划、问题生成和计划执行。通过 Knowledge Base，Problem Interface，Planner Interface 和 Parsing Interface 4 个节点，可以将任务发布者发布的任务分解为动作序列，并通过 Plan Dispatch 节点将动作指派给机器人。需要引起注意的是该实现方式是对动作的指派而非分配，原因在于依照传统 PDDL 的用法，任务发布者一开始就会确定好执行任务的机器人，并根据这些机器人的信息来求解方案。这种办法本质上是对任务采取“先分配后分解”的策略。本文采取另一种方式使用 PDDL，仅抽象描述在 PDDL 中出现的机器人类型，但不指定具体的机器人实例，例如使用 DeliverRobot 描述具有递送能力的机器人类型，而不使用 TurtleBot2 表示环境中特定的机器人。基于此描述，可在 Plan Dispatch 环节结合本文算法，确定最合适执行任务动作的机器人实例。此外，由于 ROSPlan 框架设计机制遵照 ROS 系统，ROS 系统不允许节点重复出现，每个节点的名字必须是独一的，受该特性限制，ROSPlan 只支持解决一个复杂问题，针对本文提出的多机器人动态异构任务问题，ROSPlan 并不直接适用。

为此，本文拓展了 ROSPlan 框架，使其支持任务执行，如图 2 所示。

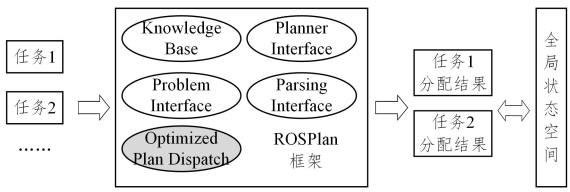


图 2 扩展 ROSPlan 框架

Fig. 2 Extending ROSPlan framework

该扩展主要体现在以下 3 个方面。

(1)运行时产生 ROSPlan 实例。将原先的 ROSPlan 框架抽象成为模板,每当任务发布者发布一个新任务,系统就调用一次 ROSPlan 框架产生实例,将任务名字作为该实例的名字。这种方式可以实现系统中同时存在多个实例,并且实例之间互相隔离。

(2)增加"全局状态空间"来记录全局机器人的状态信息。如前文所述,机器人的状态信息会影响其对动作的竞拍,本文使用"全局状态空间"记录所有机器人的实时状态信息。"全局状态空间"不仅记录了前文所示的机器人状态元组信息 S,还记录了机器人当前参与的任务编号。如果机器人当前未参与任务,那么任务编号为空。

(3)设计了 Optimized Plan Dispatch 节点,将原先的 Plan Dispatch 节点与本文提出的算法相结合。原有的 Plan Dispatch 节点本质上是依照 PDDL"先分配后分解"的策略指派动作,分解得到的动作序列可以直接指派给对应的机器人。Optimized Plan Dispatch 节点集成本文算法,作用在由 Knowledge Base, Problem Interface, Planner Interface 和 Parsing Interface4 个节点分解出的动作序列,对动作序列进行二次规划,实现对动作的分配。

6 实验与分析

为了对比验证本文算法的有效性,本文另外设计了一个基于贪心策略的算法作为对比对象,该算法的伪代码如算法 2 所示。算法的输入依赖环境中的机器人集合 R 和待分配的动作集合 $Action$,该算法的贪心策略主要体现在第 4—8 行,首先找到未被占用的,并且可以完成动作的机器人;然后将离动作坐标位置最近的机器人指派为执行动作的机器人,并根据动作需要完成的预计时间更新该机器人被占用的时间。算法最后给出当前的分配方案 $Assignment$。

算法 2 多机器人动态异构任务分配贪心算法

输入:环境中的机器人集合 R;待分配的动作集合 Action

输出:Assignment 当前的分配方案

```
1. function GREEDY(R,Action)
2.   while Action 不为空 do
3.     foreach Robot in R do
4.       if 机器人未被分配动作 and 机器人的能力可以完成该动作
         then
5.         distance ← 机器人当前位置距离特定动作指定实施位置
           的距离
6.       end if
7.     end for
8.     Robot ← distance 最小的机器人
9.     update Robot.S
10.    length(Action)--
11.  end while
12. end function
```

6.1 实验一

本文在 Gazebo 仿真平台搭建工厂环境,来验证算法的有效性。Gazebo 是一款 3D 动态模拟器,能够在复杂的室内和室外环境中准确有效地模拟多机器人[38]。工厂环境如图 3 所示,环境中有两台移动机器人 TurtleBot2 和两个机械手 UR5。

TurtleBot2 具有移动和运载的功能,UR5 可以夹取周围一定范围内的物体[39]。现有两个任务:任务一将传送带上的易拉罐运载到指定地点;任务二是让 TurtleBot2 移动到指定目的地,从而触发某种服务,任务二是紧急任务。任务一可以拆解为 3 个动作:1) TurtleBot2 移动到传送带旁边;2) UR5 夹取易拉罐并将易拉罐放置在 TurtleBot2 上;3) TurtleBot2 运载易拉罐到达目的地。任务二可以看作一个完整的动作,无需继续拆解。所以本实验的目标在于如何调度 TurtleBot2 与 UR5,从而最高效地完成任务一和任务二。

图 3 多机器人动态异构任务场景

Fig. 3 Multi-robot dynamic heterogeneous task scenario

在仿真环境中,1 号 TurtleBot2 的位置是(−4,0),2 号 TurtleBot2 的位置是(−6,1),1 号 UR5 的位置是(0.5,1.8),2 号 UR5 的位置是(−7.8,−1.5),易拉罐的位置是(1.2,1.8,0.92)。

将贪心算法与拍卖算法分别运行在图 3 场景中对任务一与任务二进行分配。图 4 记录了机器人运行的时间,横坐标是时间,纵坐标是执行动作的机器人。

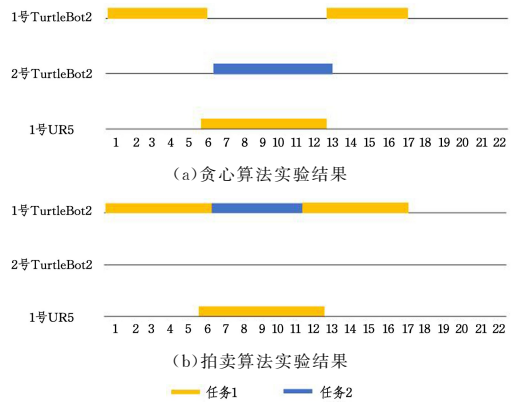


图 4 多机器人任务执行时序图

Fig. 4 Multi-robot task execution timing diagram

从图 4 中可以看出,贪心算法将任务一分配给 1 号 TurtleBot2 和 1 号 UR5,根据贪心策略,1 号 TurtleBot2 被占用,此时 2 号 TurtleBot2 距离任务执行地点最近,所以将任务二分配给 2 号 TurtleBot2。该分配方案使得 1 号 TurtleBot2 存在一段空闲等待时间。从图中也可以看出,将本文提出的算法应用在动态增长的多机器人任务场景中,可以取得较为不错的结果。在发布完任务一并将其拆解为 3 个动作后,环境中的所有机器人都参加了任务一的竞拍。1 号 TurtleBot2 竞拍到运载易拉罐的动作,1 号 UR5 竞拍到夹取易拉罐的动作。在任务一进行的过程中,才发布任务二。采取该方式是

为了拟合动态增长的任务，环境中的机器人无法提前知晓将要执行的任务。从实验结果可以发现，1 号 TurtleBot2 也成功竞拍到任务二，这是因为相比 2 号 TurtleBot2，正在等待 1 号 UR5 夹取易拉罐的 1 号 TurtleBot2 距离任务二更近，并且任务二为紧急任务需要尽快被执行，所以 1 号 TurtleBot2 赢得了任务二。1 号 TurtleBot2 完成任务二以后，返回原先的位置，此时 1 号 UR5 已经完成对易拉罐的夹取，1 号 TurtleBot2 完成剩下的易拉罐运载动作。

6.2 实验二

从机器人的能力来看，机械手具有夹取的能力，移动机器人具有移动的能力，类人机器人同时具备机械手和移动机器人的能力。在任务分配之前，本文先将任务分解为动作，动作必须是可以被 3 类机器人中的一类机器人独立执行的。例如，“导航”“巡检”“递送”都可以看作是对移动机器人的移动能力的使用。

由于在多机器人动态异构任务分配领域缺少相关数据集，因此本文随机生成任务数据与机器人信息作为数据，部分动作之间存在时间上的依赖关系。例如，“递送”任务就必须在“夹取”任务完成之后才能进行，因为需要机械手成功将目标物放置在移动机器人身上。场景的地图大小为 10 * 10，场景内有 5 个机器人，机器人型号和坐标分别是：1 号 TurtleBot2(7,5)；2 号 TurtleBot2(4,2)；1 号 UR5(6,10)；2 号 UR5(3,1)；Fetch(1,8)。每个动作的坐标信息也是随机生成的。总体而言，共有两类动作，分别是 P(Pick)和 M(Move)。为了便于比对算法在减少空闲等待时间上的有效性，本实验的所有任务具有一样的优先级，任务的详细信息如表 2 所列。

表 2 任务信息

Table2 Mission information

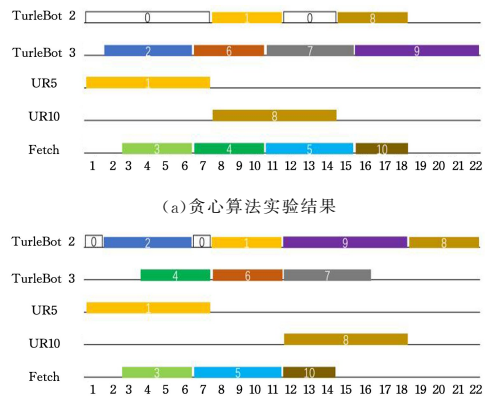
任务编号	任务描述	发布时间	任务所需能力	位置	耗费时间/min
1	递送	09:01	夹取	(5,10)	7
			移动	(6,9)	4
2	导航	09:03	移动	(8,10)	5
3	摆放物体	09:03	夹取、移动	(9,8)	4
4	巡检	09:04	移动	(2,7)	4
5	整理杂物	09:06	夹取、移动	(1,4)	5
6	运送	09:07	移动	(6,4)	4
7	导航	09:10	移动	(10,4)	5
8	递送	09:11	夹取	(4,1)	7
			移动	(3,2)	4
9	巡检	09:12	移动	(7,2)	7
10	运送	09:12	移动	(9,1)	3

前文所述的 *Task* 给出了对任务的描述，任务包含期望完成时间 *Deadline*，机器人根据表 2 中记录的耗费时间计算自己能否在 *Deadline* 之前完成任务。该时间不是严格准确的，因为动作的执行时间还会受到其他因素的影响，无法提前准确预估，只能给出一个大概的时间。例如实验一中，TurtleBot2 在接近目的地时，存在“原地打转”的现象。此时 TurtleBot2 并没有真正到达它所认为的目标，只是很接近了，所以通过“打转”来靠近目的地。因此，准确的移动时间不能通过移动距离和移动速度就能简单计算得到，还要考虑到机器人在运行中出现的不确定时间。本文认为任务时间的准确性对算法结果影响不大。

为了模拟动态增长的任务，本文并不是一次性将任务集中的动作进行分配，而是随机间隔一段时间发布新的动作。贪心算法采取的贪婪策略是选取距离动作坐标距离最小，并且此时未执行动作的机器人。贪心算法虽然也能找到合适的分配方案，但是该方案不一定是最优解。当执行顺序上存在依赖的动作时，机器人需要等待被依赖的动作完成，才能继续执行新的动作。在等待过程中，该机器人是被锁定的，无法继续参与新任务的分配，故而导致所求的分配方案未必是最优解。

与贪心算法相比，本文提出的分配算法在竞拍过程中考虑到了机器人在等待的这一状态。本文假设当任务被拆解为动作，并且每个动作都有对应的可以满足执行该动作所需能力的机器人时，分配算法就可以找到更合适的分配方案。根据贪心算法与分配算法在实验集上的任务分配情况，得到图 5 所示的结果，其中“0”表示机器人当前处于等待状态。不同颜色的条形图表示不同的动作，条形图上的数字表示该动作所属的任务编号，相同的数字表示动作是由一个复杂任务分解得到的。例如 1 号任务被分解为移动和夹取两个动作，因此有两个机器人执行 1 号任务。从实验结果可以看出，与贪心算法相比，拍卖算法可以更高效地调度机器人，将等待时间减少 80%。另一方面，拍卖算法对机器人的使用也比贪心算法更均衡，不容易出现将所有任务都分配给一个机器人的情况。

本文以 1 号任务和 2 号任务为例，解释为何拍卖算法可以有效减少机器人等待时间。1 号任务需要具备移动能力和夹取能力的机器人参与完成任务。贪心算法和拍卖算法都把 1 号 TurtleBot2 和 1 号 UR5 运用在了 1 号任务的完成中。1 号任务中的夹取动作需要依赖于移动动作，导致 1 号 TurtleBot2 存在等待时间。2 号任务需要具备移动能力的机器人参与完成任务。在 2 号任务的分配中，贪心算法将动作分配给 2 号 TurtleBot2，但此时 1 号 TurtleBot2 距离 2 号任务动作的执行地点更近，并且 1 号 TurtleBot2 处于等待状态，所以拍卖算法将 2 号任务分配给 1 号 TurtleBot2。因此，拍卖算法有效地利用了处在等待状态的 1 号 TurtleBot2，从而减少了机器人的等待时间。



(a)贪心算法实验结果

(b)拍卖算法实验结果

图 5 多机器人动态异构任务分配算法对比结果

Fig. 5 Comparison results of multi-robot dynamic heterogeneous task allocation algorithms

结束语 相比于传统的多机器人任务分配，本文针对的

多任务场景具有更广的实用性，更贴近现实场景。传统的多机器人任务分配并未针对动态增长的任务提出有效的任务分配解决方案，本文提出的分配算法旨在解决这类问题。首先，介绍了分配方案的流程并对分配中的要素进行了规范化的定义。其次，提出了多机器人动态异构任务的分配算法，该算法基于拍卖机制，将任务拆解为互相独立的动作，根据机器人对动作的出价进行动作分配，并充分考虑到了任务具有不同优先级的属性。最后，设计并实施了模拟实验，验证了所提算法相比于贪心算法能够在相同的场景中提高机器人的利用效率，并减少机器人的等待时间。

下一步的工作将围绕算法进行展开。首先，考虑更符合物理世界现实场景的空间拓扑结构；其次，考虑机器人在实际生活中使用的影响因素，如电量等。

参 考 文 献

[1] FIERRO R, CHAIMOWICZ L, KUMAR V. Multi-robot cooperation[M]// Autonomous Mobile Robots. CRC Press, 2018: 417-460.

[2] XUE F, DONG T, YOU S, et al. A hybrid many-objective competitive swarm optimization algorithm for large-scale multirobot task allocation problem[J]. International Journal of Machine Learning and Cybernetics, 2021, 12(4): 943-957.

[3] OTTE M, KUHLMAN M J, SOFGE D. Auctions for multi-robot task allocation in communication limited environments[J]. Autonomous Robots, 2020, 44(3): 547-584.

[4] CORAH M, O'MEADHRA C, GOELK, et al. Communication-efficient planning and mapping for multi-robot exploration in large environments[J]. IEEE Robotics and Automation Letters, 2019, 4(2): 1715-1721.

[5] HUANG L, DING Y, ZHOU M C, et al. Multiple-solution optimization strategy for multirobot task allocation[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2018, 50(11): 4283-4294.

[6] SHRIYAM S, GUPTA S K. Incorporating potential contingency tasks in multi-robot mission planning[C]// 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018: 3709-3715.

[7] YAN Z, JOUANDEAU N, CHERIFA A. A survey and analysis of multi-robot coordination[J]. International Journal of Advanced Robotic Systems, 2013, 10(12): 399.

[8] MOTES J, SANDSTRÖM R, LEEH, et al. Multi-robot task and motion planning with subtask dependencies[J]. IEEE Robotics and Automation Letters, 2020, 5(2): 3338-3345.

[9] TANG F, PARKER L E. Asymtre: Automated synthesis of multi-robot task solutions through software reconfiguration [C]// Proceedings of the 2005 IEEE International Conference on Robotics and Automation. IEEE, 2005: 1501-1508.

[10] DAI W, LU H, XIAO J, et al. Multi-robot dynamic task allocation for exploration and destruction[J]. Journal of Intelligent & Robotic Systems, 2020, 98(2): 455-479.

[11] YAO L S, SU L Y, LI X P. Research and development of task assignment methods for multi-robot systems[J]. Manufacturing Automation, 2013, 35(10): 21-24.

[12] KOENIG S, KESKINOCAK P, TOVEY C. Progress on agent coordination with cooperative auctions[C]// Proceedings of the AAAI Conference on Artificial Intelligence. 2010.

[13] ZLOT R, STENTZ A. Complex task allocation for multiple robots[C]// Proceedings of the 2005 IEEE International Conference on Robotics and Automation. IEEE, 2005: 1515-1522.

[14] KHAMIS A M, ELMOGY A M, KARRAY F O. Complex task allocation in mobile surveillance systems[J]. Journal of Intelligent & Robotic Systems, 2011, 64(1): 33-55.

[15] GONG J W, HUANG W N, XIONG G M, et al. Genetic algorithm based combinatorial auction method for multi-robot task allocation[J]. Journal of Beijing Institute of Technology, 2007, 16(2): 151-156.

[16] VERMA J K, RANGAV. Multi-Robot Coordination Analysis, Taxonomy, Challenges and Future Scope[J]. Journal of Intelligent & Robotic Systems, 2021, 102(1): 1-36.

[17] SARIEL-TALAY S, BALCH T R, ERDOGAN N. Multiple traveling robot problem: A solution based on dynamic task selection and robust execution[J]. IEEE/ASME Transactions on Mechatronics, 2009, 14(2): 198-206.

[18] TANG F, PARKER L E. A complete methodology for generating multi-robot task solutions using asymtred and market-based task allocation[C]// Proceedings 2007 IEEE International Conference on Robotics and Automation. 2007: 3351-3353.

[19] GERKEY B P, MATARIĆ M J. A formal analysis and taxonomy of task allocation in multi-robot systems[J]. The International Journal of Robotics Research, 2004, 23(9): 939-954.

[20] ZLOT R, STENTZ A. Complex task allocation for multiple robots[C]// Proceedings of the 2005 IEEE International Conference on Robotics and Automation. IEEE, 2005: 1515-1522.

[21] CHEN J P, YANG Y M, WU Y B. Multi-robot task allocation based on robotic utility value and genetic algorithm[C]// 2009 IEEE International Conference on Intelligent Computing and Intelligent Systems. 2009: 256-260.

[22] ZHENG T, YANG L. Optimal ant colony algorithm based multi-robot task allocation and processing sequence scheduling[C]// 2008 7th World Congress on Intelligent Control and Automation. 2008: 5693-5698.

[23] WERGER B B, MATARIĆ M J. Broadcast of local eligibility for multi-target observation[M]// Distributed Autonomous Robotic Systems 4. Tokyo: Springer, 2000: 347-356.

[24] BOTELHO S C, ALAMIR. M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement [C]// Proceedings 1999 IEEE International Conference on Robotics and Automation. IEEE, 1999: 1234-1239.

[25] ZHANG Y, LIU S H. Research and progress of multi-robot task assignment[J]. Journal of Intelligent Systems, 2008, 3(2): 115-120.

[26] PARKER L E. ALLIANCE: An architecture for fault tolerant,

cooperative control of heterogeneous mobile robots[C]// Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS'94). IEEE，1994：776-783.

[27] GERKEY B P，MATARIĆ M J. Sold!：auction methods for multirobot coordination[J]. IEEE Trans. Robotics Autom，2002，18：758-768.

[28] ZLOT R，STENTZ A，DIAS M B，et al. Multi-robot exploration controlled by a market economy[C]// Proceedings 2002 IEEE International Conference on Robotics and Automation. 2002：3016-3023.

[29] SCHNEIDER E，SKLAR E I，PARSONS S，et al. Auction-based task allocation for multi-robot teams in dynamic environments [C]// Conference Towards Autonomous Robotic Systems. Cham：Springer，2015：246-257.

[30] SEENU N，CHETTY R M K，RAMYA M M，et al. Review on state-of-the-art dynamic task allocation strategies for multiple-robot systems[J]. Ind. Robot，2020，47：929-942.

[31] HAN Y，LI D，CHEN J，et al. A multi-robots task allocation algorithm based on relevance and ability with group collaboration [J]. International Journal of Intelligent Engineering and Systems，2010，3：33-41.

[32] ELANGO M，NACHIAPPAN S. Balancing multi-robot prioritized task allocation：A simulation approach[C]// 2011 IEEE International Conference on Industrial Engineering and Engineering Management. 2011：1725-1729.

[33] DAS G P，MCGINNITY T M，COLEMAN S，et al. A distributed task allocation algorithm for a multi-robot system in healthcare facilities[J]. Journal of Intelligent & Robotic Systems，2015，80：33-58.

[34] ELSEFY A E. A task decomposition using(hdec-posmdps) approach for multi-robot exploration and fire searching[J]. Journal of Robotics And Mechatronics，2020，7：22-30.

[35] ZHENG H，WANG Y. A distributed framework for dynamic task allocation of multi-robot symbolic motion planning[C]// 2019 American Control Conference(ACC). 2019：3291-3296.

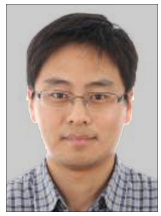
[36] XU Y G. An overview of the economic theory of auctions[J]. Economic Research，2006，175(3)：1605-1615.

[37] CASHMORE M，FOX M，LONG D，et al. Rosplan：Planning in the robot operating system[C]// Proceedings of the International Conference on Automated Planning and Scheduling. 2015.

[38] JIANG H N，ZHANG S，LIN Y F，et al. Simulation Optimization and Testing Based on Gazebo of MPI Distributed Parallelism[J]. Computer Science，2021，48(S2)：672-677.

[39] LI S P，HAN J B，LU B F，et al. Research on Grasping Technology of Cooperative Robot Guided by Vision[J]. Journal of Chongqing Technology and Business University(Natural Science Edition)，2022，39(1)：42-48.

WANG Jiwang，born in 1996，postgraduate. His main research interests include robotic systems and multi-robot applications.



SHEN Liwei，born in 1982，Ph.D，associate professor. His main research interests include the fusion of the ternary human-machine-thing，mobile application development and analysis，etc.

（责任编辑：柯颖）