

专业学位硕士学位论文

机器人集群动态联盟围捕任务规划研究

Hunting Mission Planning for Dynamic Alliances of Robot Swarms

作者姓名: 孙傲

工程领域: 控制工程

学 号: 1120201202

指导教师: 刘中常 副教授

学位类别: 电子信息硕士

培养单位: 船舶电气工程学院

答辩时间: 2023 年 6 月

大连海事大学

Dalian Maritime University

摘 要

多机器人协同围捕任务规划是一个目标匹配的优化问题, 现有规划策略往往需要一个局部领导者进行协调, 而且要用到全局先验信息。其次, 针对动态实时的围捕任务场景的相关研究较少, 需要在任务重规划步骤保证围捕机器人的效能增加。另外, 在多种资源载荷约束下的围捕任务规划也是一个待解决的问题。因此, 本文针对这些问题展开研究, 利用不同的任务分配算法解决围捕任务规划问题, 从而提高任务分配的效率和实时性。

本文首先采用拍卖算法对围捕任务场景进行规划。拍卖算法可以迭代出任务目标所需的围捕机器人, 而围捕机器人可以通过选择性的投标形成目标所需的最佳联盟。围捕联盟通过人工势场法导航对目标进行围捕。基于改进的拍卖算法, 对拍卖商效能进行优化, 在保证顺利完成围捕任务的同时进一步提高了全局效能。此外, 还提出了一种动态实时的在线任务重规划算法, 对围捕任务过程中出现的新任务进行重规划, 进一步提升了围捕算法对动态场景的适应能力。本文还提出了围捕点的分配算法, 使机器人能够准确的匹配围捕点, 从而对任务目标进行有效围捕。

针对机器人只具有局部感知能力的围捕场景, 基于一致性联盟算法对机器人集群进行任务规划, 使得可以在任务规划时无需拍卖者的角色, 通过各围捕机器人的协商信息得出围捕联盟的分配结果, 考虑了机器人的里程限制、资源载荷等限制, 使算法更贴近实际应用。对原始算法中只能处理静态全局预分配的问题进行改进, 设计了具有实时特性的在线动态分配算法, 提出一种基于局部先验信息的一致性联盟算法, 避免了原算法对全局先验信息的依赖, 减少了机器人之间的通信交互频率, 并提升了任务分配系统的效能。

关键词: 围捕; 任务规划; 资源负载; 拍卖算法; 一致性联盟算法

Hunting Mission Planning for Dynamic Alliances of Robot Swarms

Abstract

Multi-robot cooperative hunting mission planning is an optimization problem of target matching. Existing hunting mission planning strategies often require a local leader to coordinate and use global prior information. Secondly, there are few relevant studies on dynamic and real-time hunting mission scenarios, and it is necessary to ensure that the efficiency of the hunting robot increases in the task re-planning step. In addition, the hunting mission planning under the constraints of various resource loads is also a problem to be solved. Therefore, this thesis conducts research on these issues, and uses different task allocation algorithms to solve the hunting mission planning problem, thereby improving the efficiency and real-time performance of task allocation.

In this thesis, the auction algorithm is firstly used to plan the hunting mission scene. The auction algorithm can iterate out the hunting robots required by the task target, and the hunting robot can form the best alliance required by the target through selective bidding. The hunting alliance rounds up the target through artificial potential field method navigation. Based on the improved auction algorithm, the efficiency of the auctioneer is optimized, and the overall efficiency is further improved while ensuring the successful completion of the hunting task. In addition, a dynamic and real-time online task re-planning algorithm is proposed to re-plan new tasks that appear during the hunting task, which further improves the adaptability of the hunting algorithm to dynamic scenarios. This thesis also proposes the allocation algorithm of the hunting points, so that the robot can accurately match the hunting points, so as to effectively hunting the task target.

For the hunting scene where the robot only has local perception ability, the mission planning of the robot cluster is carried out based on the consensus-based coalition algorithm(CBCA), so that the role of the auctioneer can be eliminated during mission planning, and the distribution result of the hunting alliance can be obtained through the negotiation information of each hunting robot. Considering the limitations of the robot's mileage and resource load, the algorithm is closer to practical applications. To improve the problem that the original algorithm can only deal with static global pre-allocation, an online dynamic allocation algorithm with real-time characteristics is designed, and a consensus-based coalition algorithm based on local prior information is proposed, which avoids the original algorithm's global prior information. The dependence of the robot reduces the frequency of communication and interaction between robots and improves the performance of the task distribution system.

Key Words: Hunting; Mission Planning; Resource Load; Auction Algorithm; Consensus Based Coalition Algorithm

目 录

1 绪论.....	1
1.1 研究背景及意义.....	1
1.2 国内外研究现状.....	1
1.2.1 机器人任务规划研究现状.....	1
1.2.2 面向围捕的任务分配问题研究现状.....	4
1.3 论文的内容安排.....	6
2 多机器人协同围捕任务规划问题	8
2.1 问题分析与描述.....	8
2.1.1 围捕任务类型.....	8
2.1.2 围捕机器人类型.....	8
2.1.3 多机器人任务分配模型.....	8
2.1.4 机器人运动学模型及导航算法.....	10
2.2 任务规划的相关知识.....	12
2.2.1 多机器人任务规划体系结构.....	12
2.2.2 经典分配算法.....	13
2.2.3 经典算法解决围捕任务的问题.....	15
2.3 本章小结.....	16
3 基于拍卖算法的机器人协同围捕任务规划算法	17
3.1 模型描述.....	17
3.1.1 拍卖商成本目标优化函数模型.....	17
3.1.2 围捕机器人效能目标优化函数模型.....	19
3.2 分布式拍卖算法.....	20
3.2.1 算法的整体流程.....	20
3.2.2 拍卖商策略.....	21
3.2.3 机器人投标策略.....	23
3.2.4 围捕点任务分配.....	23
3.2.5 任务重规划.....	25
3.3 仿真分析.....	26
3.4 本章小结.....	34
4 基于 CBCA 算法的机器人协同围捕任务规划算法	35
4.1 模型描述.....	35
4.1.1 问题描述与目标函数.....	35
4.1.2 机器人约束条件.....	36
4.2 一致性联盟算法.....	37

4.2.1 算法的关键参数.....	38
4.2.2 任务包构建.....	39
4.2.3 基于一致性的冲突消解.....	41
4.2.4 机器人忠诚度策略.....	42
4.2.5 基于局部视觉的部分任务重规划.....	43
4.3 仿真与分析.....	44
4.4 本章小结.....	51
结论.....	52
参考文献.....	53

图表目录

图目录

图 2.1 势场力方向和目标逃逸方向	12
图 3.1 拍卖策略流程图	20
图 3.2 围捕点分配示意图	24
图 3.3 无重规划任务分配	28
图 3.4 重规划围捕过程	29
图 3.5 协助未完成任务联盟	30
图 3.6 迭代奖励值	31
图 3.7 拍卖商的成本	32
图 3.8 机器人总效能	32
图 3.9 不同任务数量下拍卖商成本	33
图 3.10 机器人的总效能	33
图 4.1 CBCA 算法流程图	37
图 4.2 无重规划围捕过程	45
图 4.3 机器人效能	46
图 4.4 重规划围捕过程	48
图 4.5 机器人效能	49
图 4.6 任务目标数 $n_e = 30$ 时平均系统效能和执行时间	50
图 4.7 机器人数量 $n_r = 13$ 时平均系统效能和执行时间	50

表目录

表 3.1 任务目标参数	27
表 3.2 机器人初始参数	27
表 3.3 拍卖商实际支付奖励	31
表 4.1 任务目标初始参数	44
表 4.2 机器人初始参数	44
表 4.3 任务目标初始参数	46
表 4.4 机器人初始参数	47

1 绪论

1.1 研究背景及意义

随着时代的发展,机器人已经在多个领域得到了广泛的应用^[1-2],而机器人集群系统^[3-4]也一直是研究的热点,其中对群体机器人围捕问题的研究是一个十分典型的代表,其核心是通过多个机器人(围捕者)之间的协作完成对目标的围捕^[5-6]。围捕问题含有很多值得研究的内容,比如机器人的动力控制^[7],对目标的分配^[8],求解算法策略的优化等等。其中,针对多个围捕目标的机器人集群任务分配问题是尤为重要的。多机器人的围捕任务分配问题就是在一些约束限制^[9]的情况下,通过对目标函数的优化,得到一个最优或较优的分配结果,使得每个围捕机器人在不同时间、不同的位置围捕不同的任务目标。通过优化任务分配系统的效能或者成本代价可以影响任务分配结果,而且更关系到每个围捕机器人能否合理地利用自身的资源。因此,如何使多个异构围捕机器人通过高效协商来完成任务,从而获得更高的效能,减少机器人系统的资源消耗,并且在不同约束条件下实现有序地多机器人任务分配是目前必须要解决的问题。机器人任务规划性能的好坏影响着围捕机器人任务执行的质量和效率。因此,研究更高效的多机器人围捕任务规划技术对于机器人的发展具有重要意义^[10]。

面向围捕的任务分配问题在现实中有着广泛的应用,例如在军用领域,可以用于对敌人的捕获,使用多个导弹来拦截击毁敌人的导弹^[11-12],或者用多个无人机对军事目标进行围歼等。在民用领域也有广泛的需求,例如海上无人艇队对非法进入国境者进行围捕驱逐;在自动化物流仓储中^[13],多个机器人协同将货物运转到目标位置,利用分配算法防止发生任务冲突,提高仓库运行效率;在游戏领域^[14]也会大量用到围捕任务规划的方法。这些实际应用场景都需要多机器人在协同围捕场景下进行任务规划,因此面向围捕任务的多机器人任务规划具有重要研究意义,能够有效实现多机器人高效地去协同执行任务。

1.2 国内外研究现状

1.2.1 机器人任务规划研究现状

任务规划方法可分为集中式分配规划和分布式规划方法^[15-16]。集中式任务规划算法中需要有一个总控中心对任务环境中所有的信息进行收集,并通过总控中心的调配获得最优结果。但是,集中式规划算法在规模比较大的集群,并且在全局信息不完全已知的任务环境中的效果较差。因为集中式任务规划算法依赖于全局的任务信息,对

机器人之间的通信能力的要求较高。同时，规模较大的集群中利用集中式任务规划算法的求解速度也十分慢，当任务环境中出现新任务或者有某些机器人损坏退出任务时，都需要总控中心重新计算、调配。

分布式任务规划算法是由任务环境中的多机器人相互协同和互相影响决策的方法，比较适用于任务环境先验信息已知，并且任务空间动态实时变换的任务场景。分布式规划算法中多个机器人可同时对信息进行处理，同时进行决策，并且允许每个机器人只与附近的邻居相互通信，这种通信机制可以使任务分配的过程有着很好的实时性。分布式规划策略具有很强的鲁棒性、可拓展性、容错性。但是，分布式分配方法获取的信息是局部信息，因此，得到的分配结果不一定是最优结果，影响任务分配的效能。

1)集中式任务规划算法:

(1)传统最优化算法

传统的最优化算法大多是集中式求解方法，是从运筹学中的指派问题发展而来，能够解决简单的优化问题。包括穷举法^[17]、矩阵作业法^[18]、整数规划法、约束规划法^[19]、图论法^[20]。整数规划法又可细分为单纯型法^[21]、匈牙利法^[22]、分支定界法^[23]，这些求解算法都是先通过建立目标函数以及约束条件然后进行求解。然而这些求解算法的求解难度都会随着任务和机器人数量的增长呈指数性的增长，求解难度会大大增加。对于多机器人的协同任务规划的应用场景，集中式求解方法已经不能满足结构复杂任务的需求，因此，相关近似算法应运而生。

(2)智能仿生算法

智能仿生算法是模拟自然界的动物群体的行为，例如：蚁群^[24]、鸟群^[25]、鱼群^[26]等，而提出的一系列仿生算法。智能算法在性能上具有灵活性和抗干扰能力强等特点，常用的有蚁群算法、鲸群算法、狼群算法等。智能仿生算法的核心思想是在一些规则限制下寻找目标函数的最优解，其中包括两类搜索算法，基本搜索算法以及启发式搜索算法。基本搜索方法与穷举法比较类似，虽然可以得到最优解，但计算求解的时间比较长。启发式搜索算法是依据启发信息搜索解，虽然可以在计算时间上提高效率，但求解效率的高低与启发函数的设定有关，因此，选择适合的启发函数非常重要。

2)分布式任务分配算法:

(1)基于阈值响应的算法

基于阈值响应算法^[27]，各个机器人首先获取到任务空间内信息，信息通过映射函数映射到机器人的决策上，不同的机器人在执行任务时对任务都有一个初步定义，任务空间出现了需要分配的任务时，由于机器人初始的设定，机器人对于任务的重视程度大小不一。因此，各机器人基于阈值响应就会产生不同的响应。机器人也会随着响

应开始后，初始的设定值也会变化，如果任务目标对于机器人激励影响超过某一阈值时，机器人则会对任务进行选择。阈值响应算法虽然采用的是隐式通信，对任务分配的通信连接要求低且鲁棒性强。但是，阈值响应算法的阈值的设置直接影响着任务分配结果的质量。

(2) 基于市场机制的算法

市场机制方法更适合在任务环境任务先验信息不能获取的条件下进行任务分配，但市场机制方法对于任务分配系统中的通信连接具有一定的要求，这就导致在执行任务分配时，通信链路会出现拥堵的现象。

合同网算法：

合同网^[28](Contract Net)是上世纪80年代提出的任务分配算法，是基于市场机制的算法中最简单有效的算法之一，各机器人通过通信的信息传递，协商出任务分配结果。合同网算法中两个基本的要素，一个是任务发布者，另一个是投标者，任务发布者需要首先通过招标在系统内发布任务信息，然后各机器人进行投标，发布者通过投来的标确定中标机器人。在合同网算法之中，各个机器人都有机会成为任务发布者，无需一个固定的集控中心，同一个机器人既可以是任务发布者对周边机器人发布任务，也可以是投标者对任务进行投标。这种算法依旧不能解决依赖高质量通信的特点，任务分配系统的任务发布者计算量比较大，并且没有对任务的独立的属性进行拆解划分，导致任务分配场景具有局限性。

拍卖算法：

基于市场机制的另一种算法为拍卖方法^[29]，其算法的核心内容就是投标方在根据某些拍卖规则下，对任务目标进行出价，而拍卖商要在拍卖规则下，将目标分配给报出价格最高的出价者。拍卖策略的主要构成元素为拍卖商方、出价者、拍卖目标、效能函数、拍卖规则5个要素，在多机器人协同任务规划的场景中，将目标作为被拍卖的任务，机器人有拍卖方作为分配者，和出价方作为出价者，并且各自的利益由于是对立的，所以可以优化拍卖方和出价方各自的收益。拍卖法是基于真实的市场基本概念模拟出的算法策略，在严格的拍卖规则下，买卖双方进行目标的交易，拍卖算法具有简单的操作流程的特性，可以高效的将目标进行规划，并且合理地规划资源配置，很容易得到最优分配或者次优分配。在多机器人作战和资源配置等规划方面应用广泛。

(3) 博弈论

博弈论的核心思想是需要机器人对环境信息和自身信息进行处理^[30-31]，然后对行为有着独立的决定，自私的使自身收益最大。博弈论比较使用于各种分布式的任务问题中。但是，由于博弈论中机器人的自私性，很难让各机器人进行协同操作。因此，

需要提出一种策略解决无法协同的问题,使机器人自己的利益与其他机器人协作的利益达到一种平衡的状态,所以必须解决如下两个主要问题:1)建立适当的收益函数,2)采用适当的协商策略。由于机器人通信往往不能有完美的通信系统,因此,需要保证机器人间的信息传递量少。目前已有的收益函数包括范围约束收益函数、均等共享收益函数、最优收益函数等。协商策略求解的结果很多时候都不太理想,有时不能最优结果,空间自适应对策是可以近似得到最优结果,但是求解周期很长。博弈论算是一个新兴领域,博弈论的任务规划策略会是接下来的热门方向。

1.2.2 面向围捕的任务分配问题研究现状

近年来,针对围捕场景的任务分配研究也十分丰富。文献[32-33]基于一致性包算法^[34](Consensus-Based Bundle Algorithm, CBBA),提出了一种可以使多个机器人共同执行同一目标的一致性联盟算法(Consensus-Based Coalition Algorithm, CBCA),可以获得一个多机器人的无冲突分配的一致结果,并在算法的收敛性上证实了其算法收敛于改进顺序贪婪算法,提高了异构智能体任务分配问题的可行性和快速性。但是,分配系统依旧不能实现在不确定性环境下的任务分配应用。文献[35]则在未知和不断变化的环境下,基于一致性拍卖算法(Consensus-Based Auction Algorithm, CBAA),使其能够在多个用户之间动态地重新分配任务,并且能够灵活地找到多个动态目标的路径,同时避免遇到意想不到的障碍,尽可能地保持机器人之间紧密的联系。实现随机分配的多目标任务分配场景,在此基础上提出了一致性拍卖联盟算法,在维护任务的同时进行任务的重新分配。文献[36]提出基于负载均衡任务分配方法,假设智能体在执行任务期间具有相同的负载,每个智能体可以分配多个任务。允许智能体在时间限制内执行多个任务。并加入任务的优先级及任务需求。文献[37]基于一致性包算法,根据任务空间中有需要无人机的在线分配和资源消耗限制的需求下,提出了一种基于任务序列执行和资源限制的实时在线任务分配算法。该分布式算法分配所有已经获知的任务和新发现的任务,每个无人机分离必要的任务时间和同步等待时间。对于发现的新目标,每个无人机快速确定其可用时间段,可以充分利用群中各无人机的可用时间和资源,提高无人机群的效率。

在基于贪婪算法的围捕任务分配中, Martin Braquet 提出了一种基于分布式拍卖的动态任务求解算法^[38]。在其算法中,每个机器人只会分配给一组空间分布的任务中的最多一个任务,而多个机器人可以分配给同一任务。智能体通过贪婪选择自身最优任务,结果就会形成一个被动联盟。任务分配是动态的,因为它是在离散时间阶段迭代进行更新。基于此算法,文献[39]考虑更现实的问题,如任务的截止时间、任务的耦合逻辑约束以及智能体的载荷能力。文献[40]通过对生物群体围捕行为的研究,对目标和

围捕者之间进行了受力分析,构建了目标与围捕者的受力数学模型,并提出了一种多机器人实时自组织任务分配算法,提出了群机器人协同自组织动态多目标围捕算法,这两个算法只需多目标和个体两最近邻位置信息以及个体面向多目标中心方向的两最近邻任务信息,计算简单高效,易于实现。文献[41]为了实现多飞行器高效围捕未来战场上的静态和动态目标,提出了一种考虑通信距离和通信时延约束的实时动态网络模型。此外,通过使用改进的粒子群优化算法进行联盟组队,使 UAV 资源得到最佳利用,联盟规模最小化。文献[42]研究了联盟机器人以最大化全局效能函数来选择任务。并提出一种以曲率为界的分布式贪婪算法

文献[43]中构建了一种多层环状伏击围捕模型,使得机器人能够充分利用自身的特点更好地执行任务。在机器人围捕任务目标时,分配系统的整体的能力会相对对等的去消耗能量,因此,文章提出了一种可以使能量平衡消耗的算法。有效提高围捕成功率,延长系统寿命。文献[44]研究了异构机器人在城市中搜索和配送给同一地点的实时任务分配问题。考虑到任务的资源需求和机器人的资源约束,当单个无人机的资源无法满足时,根据任务要求,提出了一种基于合同网算法的机器人联盟组建方法。从任务完成时间和机器人能量消耗两个方面研究了联盟形成问题。同时,提出了一种两阶段联盟形成算法。为了最大限度地降低机器人的能耗,文章中将算法是转化为一个 0-1 整数规划问题,并提出了一种基于资源树的联盟形成算法。但文献并未考虑。文献[45]研究了多无人机的定位和调度过程中考虑无人机能耗,一项任务可能需要几个具有特定能力的无人机的合作。量化了无人机和任务区域的任务属性。综合考虑无人机任务分配中的耦合关系,无人机通过任务的自有特性,形成需要完成的任务集合。基于联盟博弈理论,将无人机的分布式任务分配问题建模为一个联盟编队。在有限迭代中,无人机通过决策完成网络能耗和任务的平衡,并提高任务完成度。文献[46]研究了异构无人机联盟动态任务问题,引入了基于贪婪策略的算法,有效利用有限资源,联盟对不可预测的动态事件具有良好的鲁棒性。无人机团队可获得全局更高资源,并提高了执行任务响应能力,可以适应各种任务环境,但算法并没有加入无人机的搜索探测能力。

在异构机器人的联盟任务分配中,文献[47]研究了异构多无人机执行包含障碍物和禁飞区的搜索和攻击任务时的任务分配,然后与路径规划进行耦合。对目标的重要性和使用时间相关值进行测量,提出了获得最大的系统收益的任务分配算法。在系统收益函数中,考虑了目标的奖励、无人机的路径长度和同时执行任务的无人机数量。基于毕达哥拉斯速度曲线计算了无人机的路径长度,并将其作为任务分配问题的输入。采用无人机资源管理方法,实现无人机资源消耗的平衡。为满足在障碍物和禁飞区环

境中同时攻击的条件,提出了一种分布式协同粒子群优化算法,用于生成可飞行且安全的毕达哥拉斯速度曲线轨迹,以实现无人机的同时到达,蒙特卡罗仿真验证了所提出的任务分配和路径规划方法的性能。文献[48]建立了多无人机协同任务联盟数学模型,加入资源约束、任务时间约束和任务执行顺序约束。任务联盟规模以任务完成时间为目标函数,为加快完成任务时间,提出了一种改进的非支配排序遗传算法,通过增加有效性检验和修正规则来解决无人机资源异构条件下的多资源任务问题,采用均衡资源分配策略使各任务联盟中的无人机资源合理,并提高完成任务的效率。文献[49]提出一种基于联盟的解决资源受限的多机器人任务分配问题,任务执行所需的资源是机器人和任务的特征,对于在线任务分配,提出了一种顺序联盟方法,以有效地选择合适的机器人为分配的任务组成联盟。针对离线分配问题,提出了一种全局联合方法,对所有分配的任务进行全局优化。顺序联盟方法的优势在于在联盟形成过程中选择最适合的机器人加入,而整体联盟方法的优势在于能够有效地为所有任务找到全局最优解。

付光远则改进了合同网算法^[50],结合案例推理方法,在原合同网协议中引入了适配度和信誉度,通过与已有的案例中的历史案例进行匹配,迅速找到候选的承包商,结合信誉度,选出最适合的承包商并能够解决承包商故障问题。文献[51]提出在围捕目标时,进行预包围过程,使用具有闭环调节功能的动态任务分工方法进行任务分配、自组织地生成多个子群后,在子群中引入基于粒子群算法的机器人行为学习策略。在不需要学习全局经验的前提下,机器人能针对所属子群的共同意向目标进行协同作业,提高搜索效率。在文献[52]中,提出两种任务分配场景,一种是将猎物分配给围捕群体,以便捕获所有猎物,另一个是指围捕的同时包围目标的捕获位置。转化为一个动态优化问题,其每个时间步长分两个阶段求解,首先,将任务分配问题建模为二次分配问题,并提出了一个适应度函数来评估此类任务分配,通过这种方式,转化为几个单目标追踪问题。

经过研究这些任务规划文献发现,目前使机器人在未知环境中具备局部视觉探测能力并在有限通信条件下进行机器人围捕任务分配,同时将机器人的能力拆解,机器人能力与资源载荷,速度等能力相关,在资源限制条件下进行任务分配相关研究较少,因此,需要做进一步研究来解决这些问题。

1.3 论文的内容安排

本文分别针对不同场景下的协同围捕问题,设计围捕任务规划算法。首先,采用一种改进的拍卖算法进行围捕任务规划,并设计了围捕任务重规划算法。其次,设计了一种基于一致性联盟算法的规划算法,提高了任务效率和机器人的平均效能。

本论文的具体内容和结构安排如下:

第1章 主要介绍多机器人协同围捕任务分配问题的研究意义,以及多机器人任务分配的国内外研究现状和围捕任务规划算法的研究现状,并在最后给出本论文的整体结构安排。

第2章 简要地介绍了围捕任务规划中同构与异构机器人的区别、单一型任务与复杂结构型任务的区分,以及多机器人协同的控制体系,并建立了后续章节中需要用到的围捕任务分配算法的数学模型,然后对于将采用的拍卖算法和一致性包算法进行了总结,最后根据算法的弊端和现存的问题,引出下文待解决的问题。

第3章 针对动态围捕环境下的异构多机器人任务分配问题,采用了一种改进的拍卖算法。首先,建立了多机器人围捕任务规划的数学模型,针对现有拍卖算法中只重视中标者利益的问题,引入任务奖励反馈函数,最终在提高拍卖商利益的同时,中标者的利益也不会大幅度降低。拍卖商通过多轮迭代找到最适合于目标的围捕团队,围捕机器人通过择优投标,找到当前最适合自身执行围捕的目标。此外,针对围捕过程中出现新任务目标的情况,提出了一种任务重规划算法,提高了任务规划算法对不确定性的处理能力。最后通过仿真分析证明了所改进的算法的最优性和有效性。

第4章 针对协同围捕任务中异构多机器人系统任务分配问题,建立了多种约束条件下的异构多机器人系统的围捕任务分配模型,利用一致性联盟算法解决多异构机器人协同围捕任务的问题,该算法可以无需拍卖者的参与,仅仅通过邻居间的协商信息就可以得出分配结果。而针对围捕任务环境中出现新任务目标的情况,改进 CBCA 任务重规划算法。将环境中的围捕机器人进行分组,限制了围捕机器人整体的投标数量,降低了围捕机器人之间协商信息的复杂度和信息交互频率,提升了各围捕机器人执行围捕任务的效率,以及围捕机器人的整体效能,最后,通过仿真结果验证了所提出算法的有效性。

2 多机器人协同围捕任务规划问题

2.1 问题分析与描述

2.1.1 围捕任务类型

在多机器人协同围捕任务分配问题中, 根据任务场景不同, 围捕任务可以分为: 单一型围捕任务和复合型围捕任务^[53-54]。

(1)单一型围捕任务: 所有执行的任务目标均具有相同的结构特点和任务属性, 比如被围捕的目标具有相同的移动速度, 所需要的资源配置也都相同。由于任务目标的属性全部相同, 因此构建的围捕团队的机器人数量也是相同的。单一型围捕任务由于构成简单, 所以形成的围捕联盟的各机器人也不需要过高的能力, 围捕机器人一般只考虑和目标的距离远近即可。

(2)复合型围捕任务: 当任务目标的属性较为复杂多元化时, 需要对任务目标的各种属性进行拆解, 并形成合理的数学模型, 机器人需要针对各部分属性进行匹配。例如, 被围捕目标的运行速度不同, 并且需要机器人携带不同种类、不同数量的资源才能完成围捕。

2.1.2 围捕机器人类型

在多机器人协同围捕任务分配问题中, 根据机器人的速度和载荷资源携带能力等, 可以将机器人的类型分为同构型围捕机器人和异构型围捕机器人^[55-56]。

(1)同构型围捕机器人: 机器人所具备的速度和载荷资源携带能力完全相同, 在围捕任务中的执行能力也完全相同。同构围捕机器人的优势是价格低廉, 但在围捕的任务目标为个性化的多元复杂任务时就缺乏可扩展性和普适性。

(2)异构型围捕机器人: 针对复合型围捕任务目标, 需要每个围捕机器人对于不同类型的任务目标具有不同的执行能力。各围捕机器人根据任务所需的资源配置和围捕机器人的数量、能力等进行联盟组队, 使分配结果在保证完成围捕任务的同时, 合理地进行资源配置。

2.1.3 多机器人任务分配模型

本节对所研究的多机器人协同围捕任务规划问题进行描述。

在二维有限空间内, 设有 n_r 个围捕机器人和 n_e 个动态移动目标, 需要分配围捕机器人实现对多个移动目标的围捕。任务目标和围捕机器人的参数如下所示:

围捕机器人： $R = \{R_1, R_2, \dots, R_{n_r}\}$ 表示 n_r 个围捕机器人集合，用 R_i 作为机器人的唯一标识。机器人所包含的要素由三元素法表示为：

$$\langle X_i, E_i, l_i \rangle \quad (2.1)$$

其中， X_i 为机器人的位置坐标， E_i 为机器人所携带的资源列表， l_i 为机器人的可以接受的任务最大数量。

任务目标： $T = \{T_1, T_2, \dots, T_{n_e}\}$ ， $j \in \{1, 2, \dots, n_e\}$ 表示 n_e 个动态移动的任务目标， T_j 为任务目标的唯一标识。环境中的每个任务用四元素法表示：

$$\langle X_j, E_j, L_j, V_a \rangle \quad (2.2)$$

其中， X_j 为任务目标所处位置， E_j 表示该任务所需要的资源列表， L_j 为任务 j 所需的机器人数量，表示任务 j 需要多个机器人来完成围捕任务， $V_a = \{V_{a1}, V_{a2}, \dots, V_{an_e}\}$ 表示为任务目标的初始价值。

将任务目标的多种资源需求进行分类，用向量表示为：

$$E_j = [E_j^1, E_j^2, \dots, E_j^z], \quad j \in \{1, 2, \dots, n_e\} \quad (2.3)$$

其中 $E_j^p, p=1, 2, \dots, z$ 表示对任务目标 j 围捕所需的第 p 种资源的数量，比如 $E_j = [2, 5]$ 表示围捕目标 j 需要 2 个一型号资源和 5 个二型号资源。

在围捕打击任务中，对于围捕机器人所携带的不同种类的资源及数量，用如下 z 维向量来表示：

$$E_i' = [E_i'^1, E_i'^2, \dots, E_i'^z], \quad i \in \{1, 2, \dots, n_r\} \quad (2.4)$$

其中， $E_i'^p, p=1, 2, \dots, z$ 表示其所携带的第 p 种资源。

机器人的相关约束如下所示：

1) 机器人的里程约束

在任务规划中，机器人的油料约束能够影响机器人联盟的形成，其所携带的燃料约束可以等效为可运动的最大里程，表示为

$$d_i^c \leq d_{\max, i}^c, \quad i \in \{1, 2, \dots, n_r\} \quad (2.5)$$

其中， $d_{\max, i}^c$ 表示机器人可运动的最大里程。

2) 机器人联盟资源约束

考虑到目标的资源需求，针对被发现的目标 j 所构建的机器人联盟的多种不同可用资源需要满足：

$$\sum_{i \in \Phi_j} E_i^p \geq E_j^p \quad (2.6)$$

其中, Φ_j 为围捕任务 j 所形成的机器人联盟集合。

3) 机器人资源约束

$$E_i' \cap E_j \neq \emptyset \quad (2.7)$$

机器人 i 所具有的资源集合与任务目标 j 所需资源的集合不能为空。

当各个任务目标的速度属性各不相同, 需要不同数量的围捕者进行围捕, 移动目标被成功围捕的定义为机器人围绕目标形成正多边形的包围圈, 将目标围困而无法逃脱。在本章的协作围捕中做如下假设:

1. 围捕机器人和目标的外形和大小等信息忽略不计。
2. 在围捕过程中, 围捕机器人的最大运动速度小于目标的最大运动速度。

2.1.4 机器人运动学模型及导航算法

1. 机器人运动学模型:

本文研究的是宏观上的机器人的任务目标分配问题, 将机器人的运动轨迹离散化, 运动学模型如下所示:

$$x_i^{k+1} = x_i^k + v_i^k \Delta t \cos(\varphi_i^k) \quad (2.8)$$

$$y_i^{k+1} = y_i^k + v_i^k \Delta t \sin(\varphi_i^k) \quad (2.9)$$

$$\varphi_i^{k+1} = \varphi_i^k + w_i^k \Delta t \quad (2.10)$$

其中 (x_i^k, y_i^k) 表示第 i 个机器人在第 k 步时的位置坐标, φ_i^k 为第 i 个机器人的转向角, w_i^k 为第 i 个机器人的角速度, Δt 为控制时间步长。

2. 人工势场导航算法:

本文采用人工势场法对机器人的运动进行导航, 其中引力场产生的吸引力驱动机器人向目标点运动, 斥力场产生的斥力使机器人避免相互碰撞, 引力与斥力的合力引导机器人从起点运动到终点的同时避开运动轨迹中的其它机器人。Khatib 提出的常规引力场函数^[57]如下:

$$U_a = \frac{1}{2} k_a d^2(X_i, X_o^j) = \frac{1}{2} k_a (X_i - X_o^j)^2 \quad (2.11)$$

其中, k_a 是引力势场尺度因子, $d(X_i, X_o^j)$ 为机器人当前位置与围捕目标之间的距离, 其中 $X_i = (x_i, y_i)$ 为第 i 个机器人的位置坐标, 目标点坐标为 $X_j = (x_j, y_o^j)$ 。引力为引力势场的负梯度, 可表示为:

$$F_a = -\nabla U_a = -k_a(X_i - X_o^j) \quad (2.12)$$

引力方向是从机器人当前位置指向目标点。为解决机器人之间的避碰问题，建立斥力势场函数如下：

$$U_r = \begin{cases} \frac{1}{2}k_r(\frac{1}{d(X_i, X_q)} - \frac{1}{d_o})^2 d^n(X_i, X_o^j), & d(X_i, X_q) \leq d_o \\ 0, & \text{其它} \end{cases} \quad (2.13)$$

其中， k_r 是斥力系数， d_o 表示斥力场的影响半径， $d(X_i, X_q)$ 是机器人 i 与其他第 q 个机器人之间的距离， $X_q = (x_q, y_q)$ 为当前遇到的移动机器人坐标。由于人工势场法经常导致目标不可达的问题，因此在斥力函数中引入了因子 $d^n(X_i, X_o^j)$ ，使围捕机器人在前往目的地过程中，不会因为斥力过大而造成目标不可达。同理，斥力函数如下：

$$F_r^1(X_i) = -\frac{n}{2}k_r(\frac{1}{d(X_i, X_q)} - \frac{1}{d_o})^2 d^{n-2}(X_i, X_o^j)(X_i - X_o^j) \quad (2.14)$$

$$F_r^2(X_i) = k_r(\frac{1}{d(X_i, X_q)} - \frac{1}{d_o})^2 \frac{d^n(X_i, X_o^j)}{d^3(X_i, X_q)}(X_i - X_q) \quad (2.15)$$

由斥力函数可知，斥力是由两部分组成，但是方向指向不同， $F_r^1(x_i)$ 是从机器人指向目标的力，而 $F_r^2(x_i)$ 是从障碍物指向机器人的力。当围捕机器人到达目标点附近形成围捕队形时，为了继续向目标周围的围捕点移动需要将原先目标点的引力场改为围捕点的引力场，并得出围捕点的吸引力模型：

$$F_a = -\nabla U_a = -k_a d^2(X_i, X_m) \quad (2.16)$$

其中， $X_m = (x_m, y_m)$ 为围捕点的位置坐标。

3. 目标逃逸策略

被围捕目标在未被追捕之前保持静默状态，围捕机器人在开始围捕之后，任务目标开始逃逸，逃逸的方向为相邻围捕机器人的位置与目标位置所成的最大夹角的中线方向。逃逸运动方向的数学描述如下：

$$\varphi = \arctan((y_{i,k} - y_j)/(x_{i,k} - x_j)) \quad (2.17)$$

其中， $(x_{i,k}, y_{i,k})$ 为以目标为顶点的所成最大夹角的两个围捕机器人 R_i 和 R_k 的中点坐标，如图2.1所示。

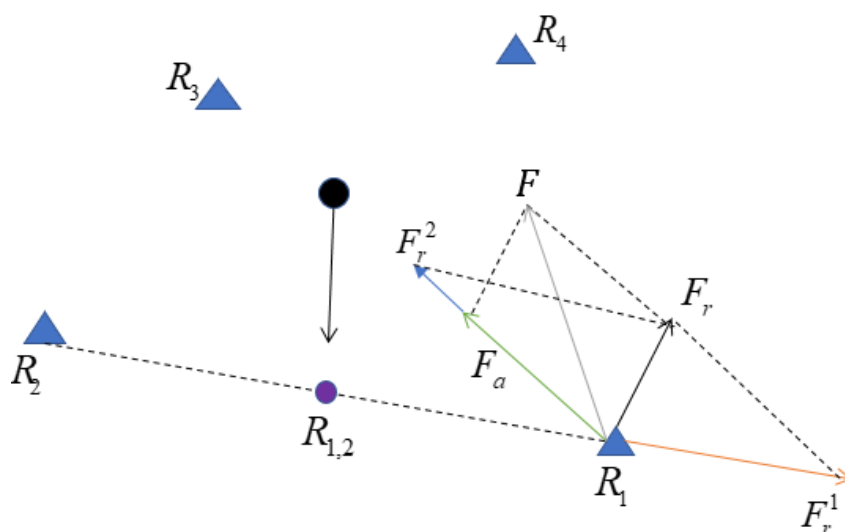


图 2.1 势场力方向和目标逃逸方向

Fig. 2.1 Potential field force direction and target escape direction

2.2 任务规划的相关知识

2.2.1 多机器人任务规划体系结构

多机器人任务规划的体系结构分为三种类型：集中式结构^[58]、分布式结构^[59]和分散式结构^[60]。

(1)集中式结构

在多机器人协同任务规划的应用场景之中，需要有一个总控平台作为指挥中心，它可以根据集中式的求解算法得出任务分配结果，并指派各机器人去执行任务目标。集中式结构的优点是一般可以得出最优结果，通过总控平台解决全局的分配结果具有可靠性^[61]。但是，这种结构在某些动态变化的任务场景中存在弊端，比如环境中出现了损坏的机器人，或者出现新的需要分配的目标，这种变化带来的运算和通信负担都施加到了总控平台，使得分配的效率大大降低。集中式结构中需由一个全局领导者对任务目标进行分配，当任务目标比较复杂，需要各机器人协同完成时，集中式结构的适用性较差。

(2)分布式结构

在分布式结构中，各个机器人可以依靠局部信息进行决策，但是还会存在一个机器人主控单元作为信息中枢来发布任务信息。各机器人在获取任务信息后，综合研判自己的条件是否匹配需要执行的任务目标，并进行信息共享，最后由主控单元对所要

执行的任务做出决策并发布给其他机器人。分布式结构相比于集中式结构可扩展性好、协调性强，但分布式结构比较复杂，对通信要求仍然较高^[62]。

(3)分散式结构

相比于集中式结构和分布式结构，分散式结构完全不依赖总控平台，每一个机器人都有通过获取的信息来进行独自决策的能力，各机器人之间相互独立，并且各个机器人可以与邻居机器人进行信息交互。分散式结构具有灵活性强、稳定性好等特点。但是，由于分散式结构的信息交互是在局部范围内进行的，所以往往不能得到最优的分配结果^[63]。

2.2.2 经典分配算法

在众多任务分配算法中，拍卖算法和一致性包算法比较适合于围捕场景的任务规划，对其介绍如下：

(1)拍卖算法

在拍卖策略的框架之下，需要有两个重要元素构成算法主体，一个是拍卖商，另一个是出价投标者。拍卖商作为协调者负责公布任务目标和拍卖成功后的获胜方，而作为出价的投标者负责向拍卖商出价并投递标书。如果出价合适，并被拍卖商确认为获胜方的机器人，就会被分配任务目标。一般是由拍卖商通过无线电广播的形式对任务进行发布，发布的任务信息包括任务类型、任务奖励、任务投标时间节点等要素。拍卖商首先通过发布任务信息，获取投标机器人的标书，拍卖商通过多轮迭代，找到最适合任务目标的多个机器人，从而得出初始围捕联盟。在进行任务重规划时，如果经过某一轮的迭代显示，分配的结果并不能保证完成任务，则通过广播再次发布，各机器人通过计算自身成本与收益的大小来决定是否参与投标，即可完成对任务目标的再次分配。拍卖算法的可扩展性强，但是比较依赖于机器人与拍卖商之间的通信质量，假如某些通信中断，则会非常影响分配结果的最优性。

拍卖算法就是在一定的拍卖规则下，通过投标机器人的出价，拍卖商将出价最高或者最低的机器人确认为中标机器人，经过多轮迭代最终获得最优分配结果^[64]。分配原则是对各个机器人的效能进行优化。机器人的效能目标函数为：

$$Y_i = \max_{j=1, \dots, n_e} (U_{ij} - C_{ij}) \quad (2.19)$$

其中， U_{ij} 为机器人 i 对于任务目标 j 的收益， C_{ij} 为机器人 i 对于任务目标 j 的成本。

机器人一般需要对拍卖商发布的所有任务进行投标，筛选出最适合于自身执行的任务。

在实际拍卖过程中，拍卖商与投标者的利益会发生冲突，往往只会关注投标者最后获得的利益，而忽略拍卖商的利益^[65]。本文将利用改进后的拍卖算法进行围捕任务的分配，在不损失过多的拍卖商利益的情况下，使得任务分配结果更加合理。

(2)一致性包算法

一致性包算法(CBBA)是 Choi 等人在 2009 年提出的一种分散式算法^[34]，可以用于求解多机器人任务分配问题。算法是由任务包构建、冲突消解这两个阶段之间的迭代组成。

第一阶段采用基于市场的分布式拍卖策略作为任务选择机制，第二阶段使用基于局部通信的一致性策略作为冲突消解机制，结合了分布式拍卖算法和一致性算法的优势，快速产生无冲突的可行分配方案。

给定 n_e 个任务和 n_r 个机器人的列表，CBBA 算法的目标是寻找机器人和任务之间无冲突的匹配方式，从而最大程度地提高整体收益值。其中每个任务只可以分配给一个机器人，每个机器人最多可以分配 l_i 个任务。将多机器人任务分配的目标函数定义为：

$$\max \sum_{i=1}^{n_r} (\sum_{j=1}^{n_e} S_{ij}(p_i)x_{ij}) \quad (2.20)$$

其中 x_{ij} 是一个二进制决策变量， p_i 代表机器人 i 执行分配任务的顺序。 S_{ij} 表示机器人 i 执行任务 j 可以获得的效能。

算法第一阶段是任务选择阶段。在这个阶段中，每个机器人采用贪婪的方式进行局部任务包的构建。各个机器人都构建四个向量：任务包 b_i ，执行路径列表 p_i ，获胜者列表 Z^i ，获胜者出价列表 Y^i 。对于所有未分配的任务，机器人将其连续添加到自己的任务包中，并随着分配过程的进行不断更新四个向量，直到机器人的任务包已满或任务全部分配完毕为止。新任务添加到任务包中的方式：机器人自身计算出执行各个任务的收益，从中选取收益最大的任务作为目标任务，相对应的机器人作为目标机器人。将目标任务的收益值与当前获胜者出价列表 Y^i 中对应的收益值进行对比，若目标任务的收益值更大，则将目标任务添加至目标机器人的任务包中，并更新目标机器人的四个向量。机器人 i 沿着执行路径列表 p_i 执行任务获得的总收益值定义为 $S_i(p_i)$ 。 $S_i(p_i \oplus_n \{j'\})$ 表示当任务 j' 分配在机器人 i 的执行路径列表的第 n 个位置上时，机器人 i 沿着 p_i 执行任务获得的总收益值。当将任务 j' 添加到机器人 i 的任务包中时，任务 j' 的边缘收益计算公式为：

$$c_{ij'}[b_i] = \begin{cases} 0, & j' \in b_i \\ \max(S_i(p_i \oplus_n \{j'\}) - S_i(p_i)), & \text{其它} \end{cases} \quad (2.21)$$

可见，边缘收益值反映了任务添加到某个机器人的任务包前后，机器人的总收益值的变化情况。算法的方案是将一个新任务插入到收益值改善最大的位置，而且在给定的当前路径下，会影响已经分配好的任务的收益值。

在任务选择阶段中，各个机器人已经构建好了自身的任务包。在冲突消解阶段中，机器人之间通过局部通信彼此共享自身的任务信息结构，筛选出无效任务和冲突任务，实现无冲突的任务分配结果。相邻机器人之间共享的信息包括：获胜者列表 Z^i ，获胜者出价列表 Y^i ，时间戳向量 s_i 。

冲突消解阶段中引入的一个新的向量 s_i ，表示机器人 i 最后一次更新信息的时间集合，并作为机器人分配过程中存储的向量。例如，机器人 i 与机器人 k 是邻居关系，可以建立通信，则机器人 k 传递给机器人 i 信息的最后时间为 s_{ik} 。

当同一个任务出现在不同机器人的任务包中时，收益值最高的机器人竞拍到此任务，该任务在其余机器人的任务包中将不再有效。包含目标任务的机器人将对自身信息结构进行修改，即机器人会释放目标任务以及在目标任务之后添加的任务，被释放的任务将重新作为新的任务由多机器人竞拍。

当机器人 i 从另一个机器人 k 收到消息时，即收到机器人 k 的 Z^k 、 Y^k 、 s_k 时，机器人 i 会根据自身的 Z^i 和 s_i 来确定哪个机器人信息是任务的最新信息。机器人 i 可以对任务 j 采取三种可能的操作：

(1) 更新： $y_{ij} = y_{kj}$ ， $z_{ij} = z_{kj}$

(2) 重置： $y_{ij} = 0$ ， $z_{ij} = \phi$

(3) 离开： $y_{ij} = y_{ij}$ ， $z_{ij} = z_{ij}$

当完成冲突消解阶段后，算法会重新返回任务选择阶段，对于所有未分配的任务进行重新添加。两个阶段不断迭代，直至 Z^i 和 Y^i 不再发生改变。

2.2.3 经典算法解决围捕任务的问题

在拍卖算法中一方面要重视拍卖结构中拍卖商的利益，另一方面，也要对拍卖算法的重规划问题进行分析，因为在围捕过程中经常会出现需要更换目标^[66]或者某些围捕机器人需要退出当前联盟的情况，而拍卖算法往往只在任务分配中完成一次的分配，

忽略了围捕过程中的重规划情况，并且围捕任务规划中还需要对最后的围捕点位置进行分配，使机器人有序地进入围捕点也是围捕任务规划中的非常重要的一部分。

在一致性包算法的研究中，虽然已经有较多文献来利用其完成多机器人的协同任务分配，但是对于优化函数的相关约束不够完善，需要加入与围捕任务相关的约束限制，才能贴合实际需求。此外，一致性包算法的冲突消解阶段是基于机器人的局部信息交互^[67]，这就导致一致性消解的运行时间比较长，因此，需要对其进行改进，将各机器人对于先验信息全部已知改进为局部先验信息已知，使各机器人采取更少的通信次数，通信交互频率降低可以使消解的运行时间缩短。

2.3 本章小结

本章首先介绍了围捕任务中的任务类型、同异构围捕机器人分类，并且介绍了任务规划中需要用到的任务分配的三种结构特点。然后，根据围捕任务规划中的问题，建立了任务规划的数学模型，并选择了两种算法进行介绍。最后，针对目前围捕任务场景中出现的问题进行总结。本章为下面章节的研究内容做了铺垫，并且也针对要研究的机器人协同围捕任务规划的问题进行了分析。

3 基于拍卖算法的机器人协同围捕任务规划算法

在某些针对多个重要目标的围捕任务场景中，一般由一个发起人掌握所有需围捕目标的信息，通过招募成员来对目标进行协作围捕，而各参与者根据自身能力和需求来选择围捕的目标，最后由发起人根据各参与人的选择情况进行目标的分配。这种模式与市场中的“拍卖-投标”机制比较吻合。然而传统的“拍卖-投标”机制一般只考虑投标者（即参与者）的利益和代价，而拍卖商（即发起人）一般不参与利益分配，也忽略了发起人为完成任务所需要付出的代价。因此，本章采用改进的拍卖算法进行任务规划来解决这种围捕任务场景的目标分配问题。

本章首先建立了多机器人协同围捕任务分配拍卖模型，其中包括拍卖商的效能优化函数和投标围捕机器人的优化函数。然后采用了一种任务奖励反馈机制，对任务分配环境中拍卖商的效能进行优化，拍卖商可根据收到的标书的数量与任务所需要的机器人数量之间的差额，从而对所需围捕任务目标的奖励进行自适应调整，通过多轮增加或减少拍卖价格，最终得出对双方均有利可图的分配结果。在动态抓捕过程中，设计了围捕任务的重规划方法，使得任务分配可以获得更高的效能。建立了围捕点的分配策略以及围捕点重规划策略，使围捕点匹配到更适合的机器人。上述方法对于多机器人协同任务规划的实时分配能力具有一定程度的提升。最终在仿真分析中，证实了本章算法的有效性。

3.1 模型描述

3.1.1 拍卖商成本目标优化函数模型

在本章节建立的优化模型中，任务分配环境中的拍卖商是通过无线电广播手段发布每个任务的具体信息。对于完成任务的机器人可以获得相应的任务奖励，各个围捕机器人 R_i 接收到任务信息之后，需要先比对任务所需的资源等限制和自身能力的大小，然后计算投标后自身的效能并进行优化。拍卖算法是双向选择的过程，对于拍卖商来讲，需要在每一轮选择哪些中标机器人进行围捕。对于投标机器人 R_i ，在每一轮需要选择哪个任务进行投标。在拍卖过程中，需要分别考虑拍卖者和投标者的利益，发布任务的拍卖者需要给各投标机器人进行奖励，因而奖励本身就是拍卖者的一种支付成本。而对于投标者，获得的奖励即为投标者的利益，并且作为分布式的拍卖算法，所有的机器人都可以成为拍卖者进行任务发布。因此，本章所建立的多机器人任务分配

拍卖策略模型包括两个目标函数：拍卖商的成本函数和围捕机器人的效能函数，拍卖商的成本包含两部分：

1. 奖励成本 z_{ij}^r ：拍卖商的奖励成本 z_{ij}^r ：在拍卖过程中的第 r 轮的拍卖，拍卖商向第 i 个围捕机器人奖励的第 j 个任务的成本。

2. 时间成本 f_{ij} ：拍卖商预计围捕机器人 i 达到任务 j 的时间，这里将预计到达的时间等价于围捕机器人 i 到任务目标 j 的直线距离。

对以上两种成本进行叠加就可以得到拍卖商的总成本，可以得到如下的目标函数：

$$\min \sum_{j=1}^{n_e} \sum_{i=1}^{n_r} (z_{ij}^r + f_{ij}) x_{ij} \quad (3.1)$$

s.t. (2.5) (2.6) (2.7)

$$x_{ij} = \begin{cases} 1 \\ 0 \end{cases}, \quad \forall i \in R, \forall j \in T \quad (3.2)$$

$$\sum_{i=1}^{n_r} x_{ij} \geq L_j, \quad \forall i \in R, \forall j \in T \quad (3.3)$$

$$z_{ij}^r < z_{j\max}, \quad \forall i \in R, \forall j \in T \quad (3.4)$$

$$x_{ij} \in \{1, 0\}, \quad \forall (i, j) \in R \times T \quad (3.5)$$

其中， $x_{ij} \in \{0, 1\}$ 为决策变量，表示围捕机器人 i 被分配了第 j 个任务。在第 r 轮拍卖中，若围捕机器人对于某一任务目标 $j \in T$ 的投标数量 $L_{j\text{bid}}^r$ 大于等于该任务目标所需的围捕机器人数量 L_j ，而且围捕机器人被拍卖者确认为最终的中标者时，则 $x_{ij} = 1$ ，反之，则 $x_{ij} = 0$ 。

约束条件(3.3)保证了每个任务目标有足够数量的围捕机器人，约束条件(3.4)保证在第 r 轮拍卖中，拍卖商给予的第 j 个任务目标的任务奖励 z_{ij}^r 不超过任务奖励上限值 $z_{j\max}$ 。

在此目标函数中，机器人所需要花费的时间成本 f_{ij} 越低，那么拍卖者的利益就会越高。而对于奖励成本 z_{ij}^r 而言，奖励成本越低，拍卖者利益的也会越高，但是围捕机器人得到的收益就会越低，从而导致任务目标对于围捕机器人没有足够的吸引力，可能无法成功找到足够数量的机器人。相反地，奖励成本 z_{ij}^r 越高，则能够吸引更多的围捕机器人进行投标，但是这会损失拍卖商的利益。因此目标函数的设计需要既能保证任务成功执行，也不会使拍卖商损失太多利益。

3.1.2 围捕机器人效能目标优化函数模型

围捕机器人在进行任务分配时获得的效能需要进行优化。围捕机器人 i 的效能目标优化函数如下：

$$\max \sum_{j=1}^{n_e} (U_{ij}^r - C_{ij}) \quad (3.6)$$

s.t. (2.5) (2.6) (2.7)

$$U_{ij}^r - C_{ij} > 0, \quad \forall i \in R, \forall j \in T \quad (3.7)$$

其中， U_{ij}^r 表示在第 r 轮拍卖中第 i 个围捕机器人完成第 j 个任务目标所能获得的收益， C_{ij} 表示第 i 个机器人完成第 j 个任务目标所需要花费的时间成本。

约束中，效能大于0要保证在每一轮的迭代中，围捕机器人选择任务 j 而获得效能要大于围捕机器人不参与选择此任务时的效能。

1.收益函数：

不同的围捕机器人在执行任务中，可以获得不同类型的任务执行奖励，综合考虑第 j 个任务目标所获得的任务奖励 z_{ij}^r 和匹配系数 g_{ij} ，建立围捕机器人的收益函数 U_{ij}^r ：

$$U_{ij}^r = z_{ij}^r g_{ij} \quad (3.8)$$

其中 $g_{ij} \in [0,1]$ 代表围捕机器人 R_i 与任务目标 T_j 的匹配系数， g_{ij} 反映了围捕机器人 R_i 执行各种类型任务能力的差别。因为每个任务目标因其速度不同，导致需要不同能力的机器人去协作执行任务。而不同能力的围捕机器人在执行不同难度，不同价值的任务时具有不同的适合程度，因此，需要建立机器人 R_i 的能力与所有任务的匹配系数集合为 $G_i = \{g_{i1}, g_{i2}, \dots, g_{in_e}\}$ ，其中 $g_{i1} + g_{i2} + \dots + g_{in_e} = 1$ 。

2.成本函数：

每一个围捕机器人的成本需要进行计算，成本影响着最终效能，进而影响算法的投标结果。构建第 i 个投标机器人到 j 个任务目标所花费的成本函数 C_{ij} 如下：

$$C_{ij} = m_1 E_{ij} + m_2 T_{ij}' + m_3 (d_{ij} / (v_j - v_i)) \quad (3.9)$$

其中， T_{ij}' 表示第 i 个机器人预计达到任务目标 j 的有效围捕范围后形成包围圈所需要花费的时间， T_{ij}' 与任务目标所需机器人数量呈反相关，任务目标所需机器人越多，执行围捕阶段所花费的时间就会越少，并且当围捕机器人进入有效围捕范围后， T_{ij}' 与到围捕点之前的平均距离呈成正相关。

$$T'_{ij} = \frac{(r_e + r_s)}{2(v_j - v_i)L_j} \quad (3.10)$$

其中, r_e 为任务目标的有效围捕范围半径, 目标所需的围捕机器人越多, 则有效围捕半径设置地越小, r_s 为围捕任务目标所需队形半径。 d_{ij} 表示当前第 i 个机器人到第 j 个任务目标的欧氏距离。 v_i 为机器人的速度, v_j 为任务目标的速度。 m_1 、 m_2 、 m_3 为成本函数的调整系数, 其中 $m_1 + m_2 + m_3 = 1$, 通过调整各系数的数值来决定机器人成本函数中哪一项影响成本的因素更大。 E_{ij} 表示为机器人 i 在任务目标 j 上花费的资源。

3.2 分布式拍卖算法

3.2.1 算法的整体流程

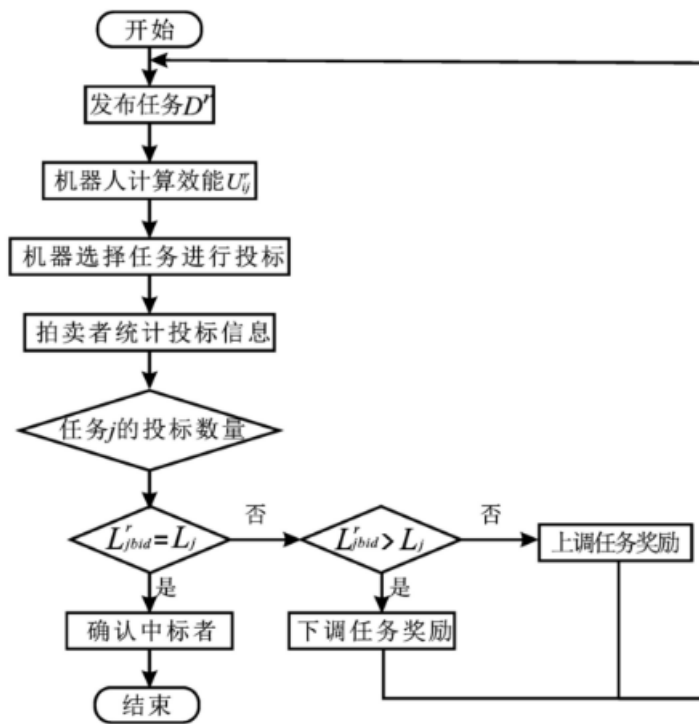


图 3.1 拍卖策略流程图

Fig. 3.1 Auction Strategy Flowchart

在拍卖任务分配策略中, 拍卖商需要在招标阶段发布任务的奖励和任务目标具体属性信息, 各围捕机器人需要根据招标信息对是否需要进行投标进行分析。在围捕任务分配环境中, 所有的信息都是通过无线广播方式进行交换, 另外, 拍卖商以及投标机器人均要对自身的效能目标函数进行优化, 最终完成分布式的拍卖任务规划策略。

由图 3.1 可知, 改进的拍卖算法需要进行四个阶段的处理: 拍卖商发布任务招标、投标机器人计算效能后投标、拍卖商统计投标数量并确定胜出者、拍卖商更新任务信息。针对各阶段需要完成的策略, 在下面内容中建立具体的数学模型进行阐述。

3.2.2 拍卖商策略

1. 任务招标阶段:

在任务规划中, 拍卖商向机器人宣告本轮还未被分配的任务信息集合表示为: $D^r = \{\theta^r, M^r, E^r, Z^r\}$, 其中 θ^r 为第 r 轮拍卖中未被分配的任务集合; $M^r = \{m_j^r, j \in \theta^r\}$, 其中 $m_j^r = (x_j^r, y_j^r)$ 表示在第 r 轮拍卖中第 j 个任务目标的坐标信息; $E^r = \{e_j^r, j \in \theta^r\}$, 其中 e_j^r 表示在第 r 轮拍卖中第 j 个任务目标的所需机器人能力信息; $W^r = \{w_j^r, j \in \theta^r\}$, 其中 w_j^r 表示在第 r 轮拍卖中第 j 个任务目标的类型信息; $Z^r = \{z_j^r, j \in \theta^r\}$, 其中 z_j^r 表示在第 r 轮拍卖中第 j 个任务目标的奖励信息。当 $r=1$ 时:

$$z_j^1 = \alpha z_{j\max} \quad (3.11)$$

其中, $\alpha \in (0,1)$ 是任务奖励系数, α 越大, 表示拍卖商在第一轮拍卖时需要支付的奖励成本越高, 奖励提高可以提升机器人的投标积极性。在本算法中 $\alpha \neq 1$, 因此, 拍卖商不会把任务 j 的任务奖励调整为上限值 $z_{j\max}$, 从而保留拍卖商与投标机器人博弈的空间。当 α 越小时, 表示在第一轮拍卖时任务分配中拍卖商支付的任务奖励越少。对于投标机器人, 如果可以获得的奖励小于自身成本, 那么将会导致自身不参与投标。在设置 α 时, 应保证机器人参与拍卖时所获得的利益大于不参与拍卖时所获得的利益, 从而使得所有机器人在第一轮都有机会参与投标。

2. 确认投标者阶段:

在任务分配中, 拍卖商收到各机器人投标以后, 需要统计在本轮拍卖中投标任务 j 的机器人数量 $L_{j\text{bid}}^r$ 大于任务所需机器人数量 $L_j, \forall j \in \theta^r$ 时的任务数量, 记为 L' 。若所有未分配的任务都在本轮中得到了充足的投标, 即 $L' = |\theta^r|$, 则对于 $j \in \theta^r$ 。

拍卖商对于任务 j 的成本集合为 $C_j = \{c_{\rho j}, \forall \rho \in B\}$, 其中 B 表示投标该任务目标的机器人集合, $c_{\rho j}$ 表示当拍卖商选择第 ρ 个机器人完成该任务时, 拍卖商所需要花费的成本, $c_{\rho j}$ 表示如下:

$$c_{\rho j} = z_{\rho j}^r + f_{\rho j} \quad (3.12)$$

拍卖商需要在投标任务 j 的机器人集合 B 中找到前 L_j 个成本 $c_{\rho j}$ 最小的机器人子集。将挑选出来的机器人作为中标者，由拍卖者确认中标者身份，随后，拍卖算法结束、任务分配完成。拍卖商所需要支付的总成本 C 如下所示：

$$C = \sum_{j=1}^{n_e} \sum_{i=1}^{n_r} c_{ij} x_{ij}, \quad \forall i \in R, \forall j \in E \quad (3.13)$$

所有中标机器人的总效能 Y 如下所示：

$$Y = \sum_{j=1}^{n_e} \sum_{i=1}^{n_r} y_{ij} x_{ij}, \quad \forall i \in R, \forall j \in E \quad (3.14)$$

若所有未分配的任务都在本轮中没有得到充足的投标，即 $L^r \neq |\theta^r|$ ，则此时对于任务 $j \in \theta^r$ 的分配情况存在以下三种情况：

1) $L_{jbid}^r = L_j$ ，说明该任务目标的投标机器人数量刚好满足需求，不需要调整奖励，将任务 j 加入集合 Ω_1 中，按照上述方法选择出 L_j 个中标机器人并通知其执行任务，重新更新 θ^r 。

2) $L_{jbid}^r > L_j$ ，说明投标该任务目标的机器人过多，拍卖商需要继续调整任务奖励，放入 Ω_2 并进入第 3 阶段。

3) $L_{jbid}^r < L_j$ ，说明投标该任务目标的机器人过少，拍卖商需要继续调整任务奖励，放入 Ω_3 并进入第 3 阶段。

3.更新任务信息阶段：

针对 $j \in \Omega_2$ 的情况，拍卖商需要下调任务奖励，以达到节约拍卖商成本的目的；针对 $j \in \Omega_3$ 的情况，投标任务 j 的机器人数量不满足所需，拍卖商需要上调任务奖励，使更多的机器人在下一轮对任务 j 进行投标。针对上述问题，建立如下任务奖励反馈机制：

$$z_j^{r+1} = z_j^r - \frac{L_{jbid}^r - L_j}{n_r} (z_{jmax} - z_j^r) \quad (3.15)$$

其中，当投标任务 j 的机器人数量 $L_{jbid}^r > L_j$ 时，投标该任务目标机器人数量越多，

$\frac{L_{jbid}^r - L_j}{n_r}$ 的比值就会越大，拍卖商可以降低任务奖励 z_j^r 的值就越多，相应地，拍卖商

付出的成本也就会越少。反之，若 $L_{jbid}^r < L_j$ ，则投标该任务的机器人数量越少，需要上调更多的任务奖励 z_j^r ，从而满足下一轮的任务目标 j 的机器人数量需求。 $z_{jmax} - z_j^r$

的阈值约束，使拍卖商调整任务奖励后，支付的任务奖励不会超过任务奖励上限值 $z_{j\max}$ 。

拍卖商需要整理本轮的任务信息集合 D^r ，并反馈到下一轮作为下一轮招标的任务信息 $D^{r+1} = \{\theta^{r+1}, M^{r+1}, E^{r+1}, W^{r+1}, Z^{r+1}\}$ ，任务位置集合为 $M^{r+1} = \{m_j^{r+1}, j \in \theta^{r+1}\}$ ，任务所需机器人能力集合 $E^{r+1} = \{e_j^{r+1}, j \in \theta^{r+1}\}$ ，任务类型集合 $W^{r+1} = \{w_j^{r+1}, j \in \theta^{r+1}\}$ ，任务奖励集合 $Z^{r+1} = \{z_j^{r+1}, j \in \theta^{r+1}\}$ 。

通过拍卖商每轮拍卖中的任务奖励调整机制，达到任务分配过程中自适应调整奖励数值的目的，通过多轮市场竞拍，既能保证完成任务又不过多的付出任务奖励。

3.2.3 机器人投标策略

投标阶段：在第 r 轮拍卖中，每个收到任务信息 D^r 的机器人根据拍卖商发出的任务信息选择本轮要投标的任务，选择效能集合 $Y_i^r = \{y_{ij}^r, \forall i \in R, \forall j \in \theta^r\}$ 中对于机器人投标后效能最高的任务 j ，并向拍卖商投标 $d_i^r(b, y_{ib}^r)$ ， d_i^r 为机器人投递的标书， $b \in \theta^r$ 为机器人投标的任务， y_{ib}^r 表示与任务 b 相对应的效能。各投标机器人效能的具体计算方式如下：

$$y_{ij}^r = \begin{cases} z_{ij}^r g_{ij} - C_{ij}, & q_{ij} = 1 \\ -\infty, & q_{ij} = 0 \end{cases} \quad (3.16)$$

其中， $q_{ij} = 1$ 表示机器人 i 与任务 j 在满足约束条件下相匹配，并进行投标。

3.2.4 围捕点任务分配

通过上述拍卖算法的任务分配，可以确定围捕联盟，但还要解决如何将围捕点分配给机器人的问题。目前解决该问题的常用算法是贪婪算法，该算法主要通过最小化围捕机器人与目标的距离来实现分配，应用简单，但该算法存在死锁问题。本文采用距离均衡原理，首先通过计算机器人与围捕点的距离，对围捕点进行预分配，然后针对重复分配的围捕点利用协商法进行再分配，该方法在分配过程中可避免死锁现象。图3.2为围捕点分配示意图。

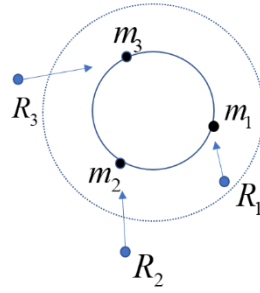


图 3.2 围捕点分配示意图

Fig. 3.2 Schematic diagram of encirclement point allocation

当围捕联盟中有机器人率先进入有效围捕范围时，首先进入范围的机器人开始计算并分配围捕点，有效围捕范围如下：

$$G = \{(x, y) : \sqrt{(x_g - x)^2 + (y_g - y)^2} \leq r_e\} \quad (3.17)$$

建立有效围捕范围后，为了可以有序地进入围捕位置，本文给每一个机器人都分配一个围捕点，在最后的围捕队形周围布设 n 个围捕点，以率先进入有效围捕范围的围捕机器人作为局部领导者，局部领导者、任务目标的连线与围捕圈的交点，作为开始执行围捕分配的第一个围捕点 s_1 ，从该点起始则围捕点 s_i 位置坐标计算：

$$x_{js} = x_j + r_s \times \cos(\pi - a_i) \quad (3.18)$$

$$y_{js} = y_j + r_s \times \sin(\pi - a_i) \quad (3.19)$$

$$a_i = \frac{2\pi(i-1)}{n_j} \quad i=1, 2, \dots, n_j \quad (3.20)$$

S_m 代表分配给第 m 个围捕点的围捕机器人数量，初始化 $S_m = 0$ ， $R = \{R_1, R_2, \dots, R_{n_r}\}$ 代表围捕机器人的集合，其中第 i 个机器人的位置坐标为 (x_i, y_i) ， $M = \{s_1, s_2, \dots, s_m\}$ 代表围捕点的集合，其中，第 m 个围捕点围捕坐标 (x_m, y_m) 。 n_j 为任务目标 j 需要建立的围捕点数量。

算法步骤如下：

- ① 首先进入有效围捕范围 r_e 的机器人为新的局部领导者，局部领导者进行围捕点预分配，初始化围捕机器人和各国捕点的位置。
- ② 通过当前各个机器人位置坐标和围捕点位置坐标计算出机器人 $R_i, i=1, 2, \dots, n_r$ 与围捕点 s_m 的距离 D_{im} ，其中 $D_{im} = \sqrt{(x_i - x_m)^2 + (y_i - y_m)^2} \quad i=1, 2, \dots, n_e$ 。
- ③ 针对机器人 $R = \{R_1, R_2, \dots, R_{n_r}\}$ ，计算 $D_{im}, j=1, 2, \dots, n$ 的最小值，将围捕点 s_m 分

配给机器人 R_i ，则该围捕点的分配到的机器人的数量 $S_m = S_m + 1$ 。

④ S_m 的值分为三种情况：

case1: $S_m = 0$ ，则围捕点 m_j 未分配

case2: $S_m = 1$ 则该点匹配成功，记录下匹配成功的围捕机器人 R_i 和期望围捕点 s_m ，将其从集合 R 和 S 中移出，令 $n_j = n_j - 1$ 。

case3: $S_m > 1$ ，围捕点 s_m 分配给距离最远的机器人，将 R_i 与 s_m 分别从集合 R 和 S 中移出，令 $n_j = n_j - 1$

⑤ 判断围捕机器人是否全部到达预定围捕点位置，若是，则算法结束，若否，则返回step2重新计算最佳围捕点。

在围捕任务中，围捕队形的形成是最后一个阶段，但是由于任务目标不断变化，因此需要在围捕过程中对于无法完成任务的联盟，以及更适合适配围捕点的机器人进行动态重规划。详细算法策略见3.2.5节。

3.2.5 任务重规划

若在设定周期 T 内，还未将目标 j 围捕成功，说明对于初始设定的任务所需机器人数量 L_j 有偏差，便告知其他已经完成围捕任务的联盟小组，对目标进行重新分配和重新规划围捕点分配，并加入到新目标 j_n 的围捕任务当中。而且在围捕的动态过程中，各机器人与目标的相对距离不断变化。需要不断更新围捕点所对应的机器人。

其次，若环境突然出现新任务为目标 j_n ，由于已经形成适配于任务目标 j 所需机器人数量的联盟。但是环境中突然出现收益 z_{ij}^r 很大的任务目标，在符合约束条件的情况下，

需要进一步判断整个联盟 Φ_s 是否需要更换围捕目标 j_n 。建立联盟整体效能模型：

$$U_{j_o}^{\Phi_s} = \sum_{i=1}^{|\Phi_s|} (U_{j_o} - C_{ij_o}) \quad (3.21)$$

其中， j_o 为联盟原来要围捕的任务， Φ_s 表示为分配给围捕任务 j_o 的机器人集合， $U_{j_o}^{\Phi_s}$ 为整个联盟在追捕任务 j_o 获得的绝对效能， C_{ij_o} 为此时第 i 个机器人执行任务 j_o 的成本。任务重规划算法步骤如下：

- ① 初始化各机器人和目标信息：
- ② 当环境中出现新目标 j_n ，拍卖商需要对新目标和原目标进行分析比较。
- ③ 若 $U_{j_o}^{\Phi_s} < U_{j_n}^{\Phi_s}$ ，联盟需要更换围捕目标，通过拍卖算法，拍卖商发布新任务，

联盟内的机器人 $i \in \Phi_s$ 进行投标，会有三种情况：

case1: $L_{jbid}^r = L_j$ ，说明刚好符合新任务所需求的机器人数量。

case2: $L_{jbid}^r > L_j$ ，通过奖励反馈公式，对投标机器人数量进行调整，原联盟中未中标新任务的机器人，继续保持原目标追随状态，等待其他新成员 $i \in \Phi_{else}$ 的加入。

case3: $L_{jbid}^r < L_j$ ，通过奖励反馈公式，对投标机器人数量进行调整，拍卖商在任务目标原有目标收益基础上，继续加码，直到投标机器人个数满足新任务目标需求。

若 $U_{j_o}^{\Phi_s} > U_{j_n}^{\Phi_s}$ ，保持联盟中的各成员围捕目标不变。

- ④ 在围捕过程中若有其他围捕机器人 $i \in \Phi_{else}$ 到围捕点的距离 D_{is} 小于联盟中的某一机器人到围捕点距离 D_{is} ，且满足目标约束条件，则对联盟成员进行调整。比较需要进行目标更换的两个机器人的效能和，例：机器人 i 原来的围捕目标是 j_1 ，机器人 k 原来的围捕目标是 j_2 ，若 $U_{ij_2}^r + U_{kj_1}^r > U_{ij_1}^r + U_{kj_2}^r$ ，两机器人交换围捕目标，若 $U_{ij_2}^r + U_{kj_1}^r < U_{ij_1}^r + U_{kj_2}^r$ ，则维持原来的围捕状态。
- ⑤ 如果在设定的围捕时间内，存在没有围捕住任务目标的联盟，则其他联盟已经完成任务的成员加入该联盟，拍卖商继续提高任务目标奖励 z_{ij}^r ，吸引更多机器人参与任务执行，并重新规划围捕点。
- ⑥ 若联盟中各成员追赶到围捕点附近即可认为机器人围捕准备就绪，联盟形成正多边形队形，围捕结束。

3.3 仿真分析

为了验证本章所提出的算法的有效性，在 Windows 10 计算机操作系统中利用 Matlab R2020b 软件进行仿真实验，通过改变初始条件进行重复仿真实验。另外，将本章所提出的算法策略与现有文献的算法进行比较，通过比较拍卖商成本，各机器人的总效能来验证本章算法的性能。

设置仿真场景如下：在环境中，随机生成 4 个不同类型的任务目标和 15 个异构机器人。不同的机器人和任务目标具备不同的速度，速度较大的目标，需要更多的机器人来执行任务；反之，目标速度小，则不需要太多机器人执行。任务相关参数设置如表 3.1 所示，机器人相关参数如表 3.2 所示：

表 3.1 任务目标参数

Tab. 3.1 Task Objective Parameters

任务序号 j	位置 (x_j, y_j)	需求机器人 数 L_j	任务奖励上 限 $z_{j\max}$	所需资源资源 E_j
1	1.94,9.11	3	520	{10,9,14,15,18,15}
2	2.83,4.48	3	640	{10,9,5,5,12,8}
3	7.32,9.45	4	820	{9,10,11,7,14,7}
4	7.21,5.36	5	1350	{9,10,12,8,12,6}

各任务目标 j 基本需要三个以上的机器人协同完成，各机器人的速度、能力各不相同。15 个异构机器人的相关参数如表 3.2 所示：

表 3.2 机器人初始参数

Tab. 3.2 Initial parameters of task objectives

机器人序号 i	位置位 (x_i, y_i)	任务匹配系数 g_i	能力资源 E_i'
1	1.54,6.88	0.4,0.1,0.3,0.1,0.1	{3,2,5,6,8,7}
2	4.13,8.30	0.4,0.3,0.1,0.1,0.1	{14,11,15,11,18,17}
3	3.92,6.44	0.1,0.1,0.5,0.2,0.1	{4,5,4,3,2,1}
4	1.32,3.64	0.1,0.4,0.2,0.2,0.1	{5,4,3,2,6,5}
5	4.43,4.36	0.1,0.1,0.5,0.2,0.1	{5,6,2,5,6,2}
6	2.09,1.37	0.1,0.4,0.2,0.2,0.1	{9,12,6,8,8,3}
7	3.53,2.68	0.1,0.4,0.2,0.2,0.1	{10,10,5,7,12,7}
8	6.92,8.28	0.1,0.1,0.4,0.1,0.1	{4,5,4,3,2,1}
9	8.54,6.99	0.1,0.1,0.4,0.1,0.1	{2,5,6,4,6,3}
10	7.58,3.75	0.1,0.1,0.5,0.2,0.1	{3,4,5,4,3,2}
11	7.03,2.11	0.1,0.1,0.5,0.2,0.1	{4,6,7,8,9,10}
12	9.04,2.56	0.1,0.1,0.4,0.1,0.1	{8,6,8,4,7,9}
13	8.92,5.44	0.1,0.1,0.4,0.1,0.1	{4,6,8,9,4,10}
14	5.89,7.82	0.1,0.1,0.5,0.2,0.1	{2,3,7,8,5,3,2}
15	4.82,5.12	0.4,0.3,0.1,0.1,0.1	{3,2,5,6,8,7}

仿真结果见图 3.3，其中图 3.3(a)为各机器人 i 与任务目标 j 的初始位置状态，截取任务环境中 $10\text{km} \times 10\text{km}$ 的区域，蓝色三角代表围捕机器人 i ，黑色实心圆为任务目标 j 。围捕机器人 i 和任务目标 j 的位置为系统随机生成，所有的围捕机器人和任务目标在初始时刻静默在环境中，在获得任务分配结果后开始执行抓捕。如图 3.3(b)所示，各机器人 i 已经完成初始分配，并通过人工势场导航算法开始移动，对目标 j 进行抓捕，

目标分配的结果为： $\Phi_1 = \{R_1, R_2, R_{15}\}$ ， $\Phi_2 = \{R_4, R_6, R_7\}$ ， $\Phi_3 = \{R_8, R_{13}, R_9, R_{12}\}$ ， $\Phi_4 = \{R_3, R_{10}, R_{14}, R_{11}, R_5\}$ ，从图中可知，各机器人 i 形成了4个围捕联盟。特别注意的是， R_3 虽然在初始时比 R_{15} 离任务目标 T_1 更近，但是由于 R_3 不符合资源需求，因此不能加入 T_1 的围捕任务中； R_{10} 虽然在初始时距离上比 R_{12} 离任务目标 T_3 更近，但 R_{10} 不符合任务目标的速度能力需求，因此 R_{10} 也不能加入到 T_3 的围捕任务中。图 3.3(c)为各机器人正在围捕目标的过程中，图中的虚线圆为目标有效围捕范围。图 3.3(d)为各围捕机器人在进入有效围捕范围后，开始进行围捕点的分配，形成目标的围捕队形。

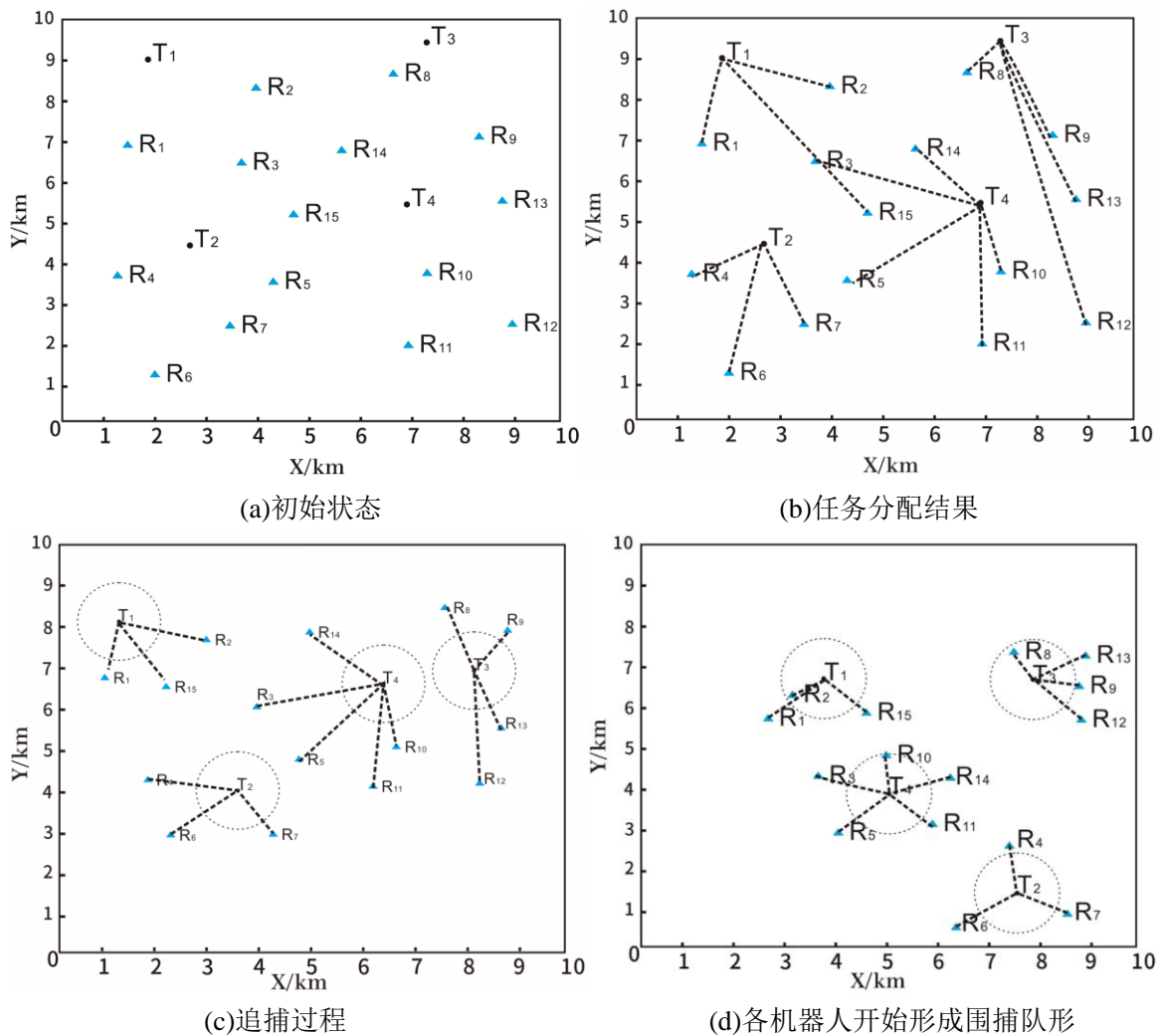


图 3.3 无重规划任务分配

Fig 3.3 No re-planning hunting process

接下来的仿真演示重规划的场景，仿真结果见图 3.4。在图 3.4(a)中，在围捕过程中出现了新的目标 T_5 ，并且 T_5 的价值收益对于其他正在围捕的机器人来说收益更高，则机器人通过任务重规划策略重新选择任务目标。通过重规划策略，只有 T_4 的围捕联盟 Φ_4 中的机器人从数量、能力和资源方面符合任务目标 T_5 的需求，则联盟 Φ_4 中的成员通过计算得出其中的 $\{R_{10}, R_{11}, R_{14}\}$ 需要从联盟中分离出来，这三个机器人若执行新目标获得的效能 U_{ij}^r 为 $\{164.39, 152.54, 158.76\}$ 均大于从原目标获得的效能，因此，加入到任务目标 T_5 的围捕。在图 3.4(b)中， T_4 联盟中剩余的两个保持原有的任务不变，并继续跟随目标，伺机等待其他合适的机器人加入来共同执行 T_4 的围捕任务。

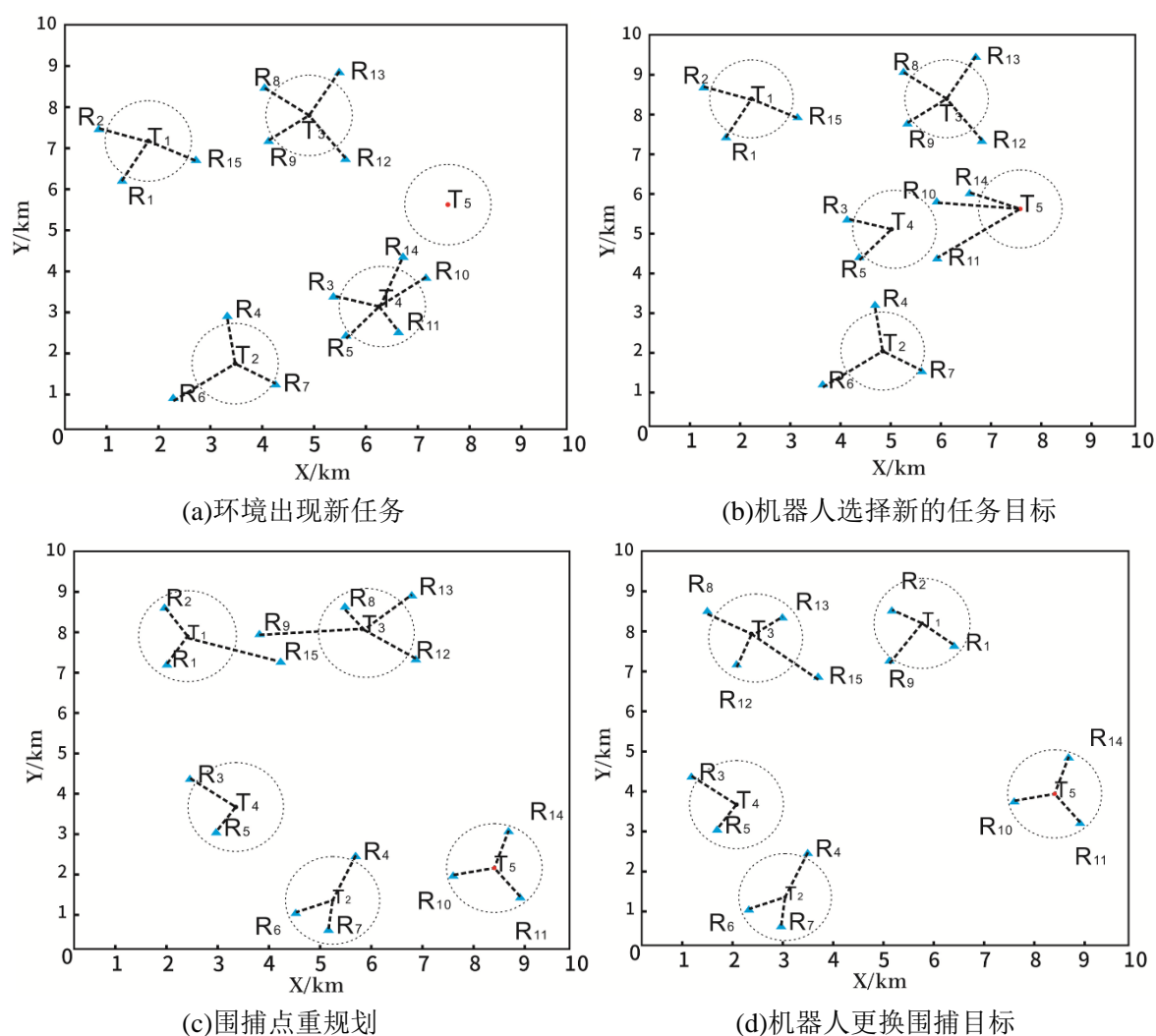


图 3.4 重规划围捕过程

Fig. 3.4 Re-planning the hunting process

任务目标 T_5 出现以后, 任务环境中的各围捕联盟通过重规划算法, 确定了新的需要执行的任务目标, 此时的任务分配结果为: $\Phi_1 = \{R_1, R_2, R_{15}\}$, $\Phi_2 = \{R_4, R_6, R_7\}$, $\Phi_3 = \{R_8, R_{13}, R_9, R_{12}\}$, $\Phi_4 = \{R_3, R_5\}$, $\Phi_5 = \{R_{10}, R_{14}, R_{11}\}$ 。各联盟继续执行抓捕任务目标, 但是在围捕点分配阶段, 围捕联盟 Φ_1 和围捕联盟 Φ_3 中的各机器人并没有达到围捕点位置, 此时, Φ_1 中的 R_{15} 比 Φ_3 中的 R_9 在距离上更接近 T_3 的围捕点位置, 如图 3.4(c) 所示。同时, R_{15} 和 R_9 满足 T_3 任务和 T_1 任务的速度和资源限制, 通过收益计算, R_{15} 执行任务 T_3 获得效能为 216.67, R_9 执行任务 T_1 获得效能为 198.86, 均大于原执行目标的效能, 因而 R_{15} 要加入到 T_3 的围捕任务执行中, R_9 要加入到 T_1 的围捕任务执行中, 如图 3.4(d) 所示。

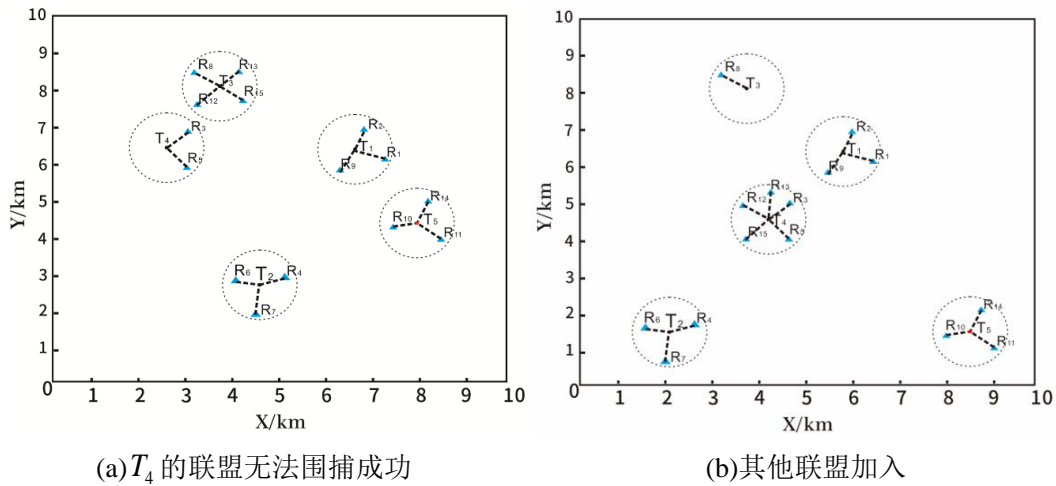


图 3.5 协助未完成任务联盟

Fig 3.5 Assist in unfinished task alliances

在图 3.5(a) 中, 各围捕联盟更换了任务目标继续执行任务, 在经历追逐后到达各自目标的围捕点。但是任务目标 T_4 并未被成功围捕, 因为其所需要的围捕机器人数量至少是 4 个, 但是机器人联盟 Φ_4 中只有 R_3 和 R_5 两个机器人, 不足以完成对任务目标 T_4 的围捕。于是 Φ_3 联盟中的成员 R_{15} 和 R_{12} 由于结束围捕任务, 可以加入到联盟 Φ_4 中去合力围捕目标, 并重新规划围捕点。直到将 T_4 围捕成功后, 所有围捕任务结束, 如图 3.5(b) 所示。

通过拍卖算法的多轮迭代, 优化出拍卖者需要支付给围捕机器人的任务奖励。由表 3.3 可知, 执行任务的上限值要大于拍卖者实际支付的奖励, 因此, 拍卖商节省了成本、拍卖者获得了更多的效能。

表 3.3 拍卖商实际支付奖励
Tab. 3.3 Auctioneer actually pays rewards

任务序号 j	实际支付奖励 z_{ij}	任务奖励上限值 $z_{j\max}$
1	490	520
2	580	640
3	740	820
4	970	1350
5	1450	1600

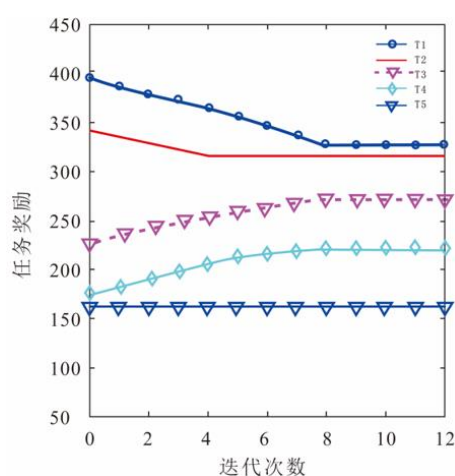


图 3.6 迭代奖励值

Fig 3.6 Iterations of reward values

在图 3.6 中，随着拍卖过程中迭代次数的变化，每个任务的实际任务奖励也在变化，不同线型表示不同任务点的任务奖励变化情况，上升的曲线为 T_3 、 T_4 ，说明这些任务在整个拍卖过程中投标该任务的机器人比较少，没有达到任务所需。因此，拍卖商在不断地增加任务的奖励，使更多的机器人进行投标。当曲线趋于平缓时，投标该任务的机器人已经满足任务所需，拍卖商不用继续增加任务奖励。下降的曲线为 T_1 、 T_2 ，表明投标该任务的机器人比较多，因此，拍卖商需要通过下调任务奖励来使多余的机器人离开本任务的投标，从而节省了自身成本。保持不变的曲线为 T_5 ，表明该任务点供需平衡，恰好满足分配条件，因此，拍卖商无需继续调整任务奖励。从图 3.6 可以看出，12 次迭代后，所有奖励曲线均不再发生变化，说明均达到了供需平衡。

对本章算法进行多次仿真验证，随机生成任务分配中的初始条件，对任务数量 $n_e = 12$ ，机器人 i 数量分别为 20、22、24、26、28、30 情况下的算法性能进行仿真对

比。仿真结果如图 3.7 和图 3.8 所示, 图 3.7 为在不同机器人数量下本章算法与文献[68]提出的 **Parallel** 拍卖算法拍卖商的成本对比图。从图 3.7 可以看出, 当 n_r 增大时, 本章所采用的算法的 C 值曲线值低于 **Parallel** 拍卖算法。因为本章引入了任务奖励反馈机制, 所以拍卖商可以对所需支付的任务奖励进行多轮试探, 既可以使任务目标顺利地完, 又不会使拍卖商支付过多的奖励, 基于这样的机制, 拍卖商的成本 C 一定会变低。

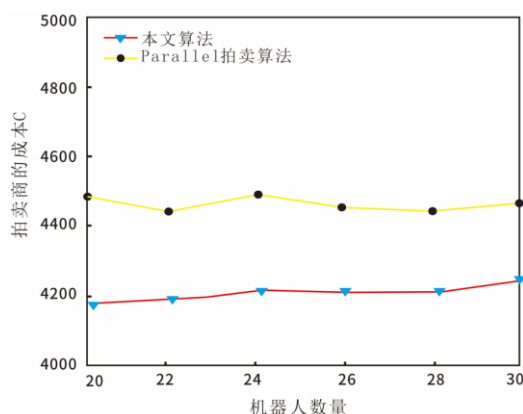


图 3.7 拍卖商的成本

Fig. 3.7 Auctioneer's costs

图 3.8 为两种算法机器人总效能 Y 的对比图。可以看出, 当 n_r 增大时, 本章所采用的算法的 Y 值高于 **Parallel** 拍卖算法, 并且随着机器人数量 n_r 的增加, 本章所采用的算法优势更加明显。在 **Parallel** 拍卖算法中, 机器人需要对本轮公布的所有任务都进行投标, 那么拍卖商与机器人利益会发生冲突, 这样会导致中标机器人被分配的任务不是最优的任务。而且随着机器人数量的增加, 上述情况的问题会变多, 所以当 n_r 增加时, 本章提出的算法优势会更加明显。

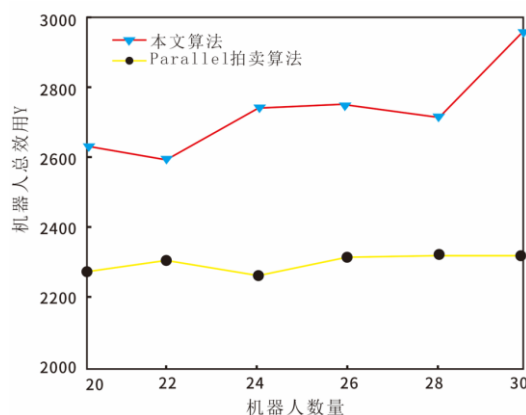


图 3.8 机器人总效能

Fig. 3.8 Total robot utility

不断增加任务环境中任务目标数，对机器人数量 $n_r = 30$ ，任务数量 n_e 分别为 10、13、16、19、22 和 25 的情况下的算法性能进行了仿真对比，如图 3.9 和图 3.10 所示。从图 3.9 可以看出，随着 n_e 的增加，两种算法的 C 值不断增加。这是由于投标机器人的可选择任务变多，拍卖商需要增大任务的奖励值才能对投标机器人有更多的吸引力，因此，拍卖商的成本 C 必然增多。图 3.9 还可以清晰地看出，与 Parallel 拍卖算法相比，本章采用的算法 C 值更小。这是任务奖励反馈规则降低了拍卖商的成本 C ，进而验证了算法的优越性。在图 3.10 中，随着 n_e 的增加，两种算法的 Y 值都是呈上升趋势。这是由于随着任务 n_e 的增加，就会有更多的机器人被确认为中标机器人，因此，机器人的总效能 Y 也会随之增加。从图 3.10 还可以清晰地看出，本章算法与 Parallel 拍卖算法相比较，由于本章算法能够保证中标者每轮被分配的任务都是对机器人而言最高效能的任务，因此，本章提出的算法的总效能 Y 更大，验证了本章采用算法的优势。

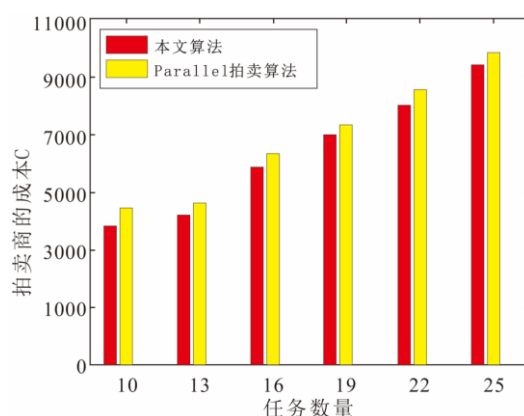


图 3.9 不同任务数量下拍卖商成本

Fig. 3.9 Auctioneers' costs under different number of tasks

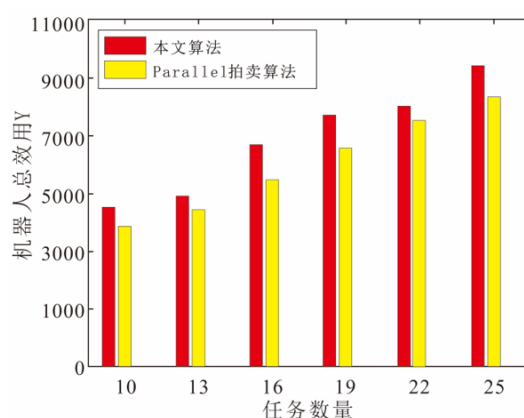


图 3.10 机器人的总效能

Fig. 3.10 Total utilities of robots

3.4 本章小结

本章针对多机器人协同围捕任务分配问题，采用了一种改进的拍卖算法。首先，建立多机器人任务分配拍卖模型，其次，对现有拍卖算法中的拍卖商成本以及投标机器人的效能进行优化，引入拍卖商的任务奖励反馈规则，使任务可以顺利完成，又降低了拍卖商所要支付的成本。最后，加入重规划算法策略和围捕点分配策略，使机器人在抓捕目标的过程中，动态实时地实现任务的重规划，使更适合的新机器人加入到目标的围捕之中。本章还详细介绍了形成围捕队形的过程，使围捕点有序地分配给围捕机器人。最后通过仿真实验验证了本文提出算法的有效性，并与现有算法进行了机器人效能和拍卖商成本的数值对比，进一步验证了算法的性能。

4 基于 CBCA 算法的机器人协同围捕任务规划算法

上一章采用的拍卖算法是分布式算法，需要一个协调者来进行分配，如果作为信息中枢的协调者损坏，则会大大影响任务分配的最优性。而一致性包算法(CBBA)不需要指定协调者，各参与者与邻居通过局部信息交互即可完成任务分配，因而可以分散式执行，适用于大规模集群的任务分配场景，但 CBBA 算法只能实现单个机器人对任务的执行，无法通过多机器人协同执行任务，因此本章采用基于 CBBA 算法改进的 CBCA 算法对围捕任务进行规划，CBCA 算法可以实现多个机器人协同执行同一目标的场景，因此，更适用于围捕任务规划。

然而，已有的基于一致性联盟的算法相关研究在解决多机器人任务分配策略时，考虑的多是静态目标的任务规划，未研究动态任务场景中的应用问题。而且，在基于一致性联盟算法的任务分配中，一般假设所有的任务目标的信息完全已知，并不完全适用于机器人感知能力有限的围捕任务场景。而且全局信息已知的假设增加了机器人获取信息的难度，因此，需要调整算法的分配机制来适应动态变化的围捕场景。

在本章中，首先建立异构多机器人围捕任务分配模型，包括机器人的收益目标函数和多种约束模型。然后设计分散式一致性联盟任务分配算法来解决多围捕目标的任务分配问题。针对出现新的任务目标的情况设计任务重规划算法，提高了多机器人分配算法应对突变任务环境的能力，最后通过算法仿真与对比验证本章算法的有效性和优势。

4.1 模型描述

4.1.1 问题描述与目标函数

在 CBCA 算法中，没有一个集中的信息中枢，因此需要各机器人通过邻居间的信息交互来实现自身决策，另外，由于算法本身的结构特性，需要两个阶段来对任务进行分配。首先是任务包构建阶段，机器人在满足资源约束条件下进行任务投标，选择对自身效能最大的任务收纳到自身任务包，并且每个任务目标都有所需的机器人数量限制，机器人通过对目标函数的优化，找出效能最大的目标形成围捕联盟。其次在一致性消解阶段，通过时间戳的更新和各机器人的中标矩阵来协商解决冲突，由此各机器人可以达成共识，获得最优分配结果。

结合本章任务分配需求，上述问题可描述为：

$$\max \sum_{i=1}^{n_r} (\sum_{j=1}^{n_e} S_{ij}(p_i)x_{ij}) \quad (4.1)$$

s.t. (2.5) (2.6) (2.7)

$$\sum_{j=1}^{n_e} x_{ij} \leq l_i \quad (4.2)$$

$$\sum_{i=1}^{n_r} x_{ij} \geq L_j \quad (4.3)$$

$$x_{ij} \in \{1, 0\}, \forall (i, j) \in R \times T \quad (4.4)$$

其中, $x_{ij} \in \{1, 0\}$ 为任务分配中决策变量, $x_{ij} = 1$ 表明机器人执行任务 j , 否则机器人不执行任务。 l_i 为机器人 i 所能接收的任务的最大数量。 s_{ij} 为机器人 i 执行任务 j 可以获得的效能。

在本章任务分配算法的执行过程中需要遵循下列原则:

1. 机器人优先选择效能高的任务。
2. 每个任务目标都有预计执行时间。
3. 每个任务目标对应的机器人数量不可以超过任务目标所需, 以免造成资源浪费。
4. 机器人需要在成功围捕住目标后对目标进行固定时间的围困, 因此, 在持续围困过程中, 机器人不可以在这段时间更换任务目标。

4.1.2 机器人约束条件

对于多机器人协同围捕任务规划中, 需要考虑到如下的约束条件:

1) 机器人的独立资源约束:

每个围捕机器人只能携带有限的资源, 且随着使用而消耗。考虑到每个机器人能够按分配顺序围捕多个目标, 其资源需满足:

$$E(p_i) \leq E_i' \quad (4.5)$$

其中, $E(p_i)$ 表示机器人 R_i 执行其任务序列上所有目标所消耗的资源。

2) 死锁约束:

考虑到每个机器人可以执行多个目标, 执行同一目标任务的各机器人需要同时到达目标。因此, 各机器人在完成其任务序列的过程中, 可能会因互相等待而陷入死锁状态无法完成任务。多个机器人的任务序列不发生死锁等价于以目标为顶点所形成的有向无环图(DAG)^[69]:

$$G(T, \varepsilon(A)) = DAG \quad (4.6)$$

$$T \triangleq \{T_1, T_2, \dots, T_{n_e}\} \quad (4.7)$$

$$\varepsilon(A) = \left\{ (A^{(i)}(k), A^{(i)}(k+1)) \mid i \in \{R_1, R_2, \dots, R_{n_r}\} \right\} \quad (4.8)$$

其中, T 为有向无环图中的顶点集合, $\varepsilon(A)$ 为边的集合, 用来描述各机器人的任务路径关系。

4.2 一致性联盟算法

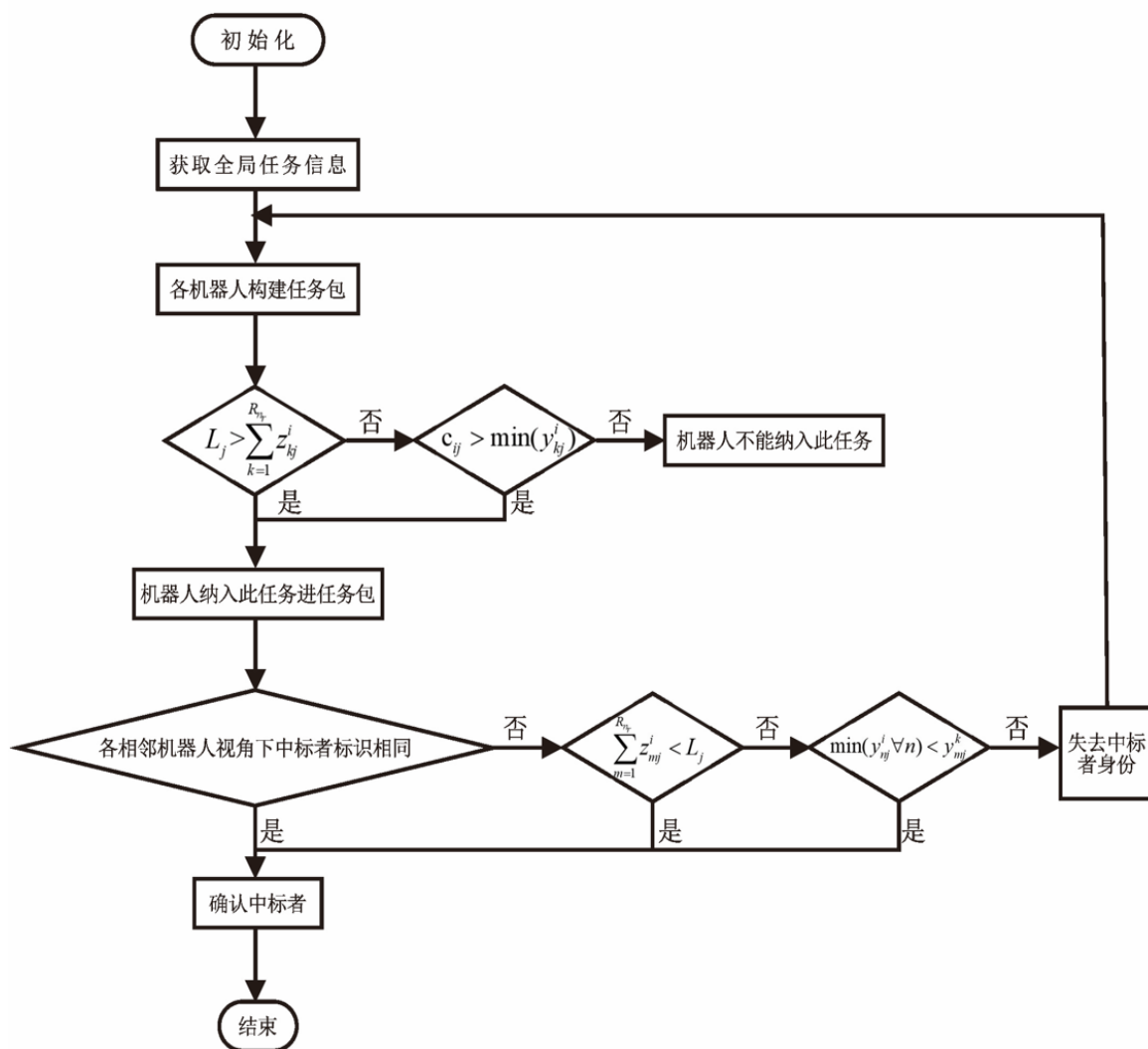


图 4.1 CBCA 算法流程图

Fig. 4.1 Diagram of CBCA algorithm

分组一致性联盟算法包括两个阶段, 预分配阶段和冲突消解阶段, 如图 4.1 所示。在初始预分配时, 各个机器人根据任务空间中的已知信息进行任务包构建, 构建原则为对于机器人自身效能最大的任务依次收纳入任务包中, 并且机器人在收集任务过程

中必须判断任务是否已经被足够的机器人选择, 如果已满足任务所需则需要比较投标值, 若机器人的投标值更大则可以收纳任务进包。

当任务包已满, 或探测到的任务空间中已无需要被分配的任务, 则任务包构建结束。第二阶段为一致性消解阶段, 各个机器人通过邻居信息, 得出一个自身视角的中标者矩阵, 并进行邻居间的协商消解, 最后得出一个无冲突分配结果。

4.2.1 算法的关键参数

本文利用 CBCA 算法来解决机器人围捕任务分配问题。在 CBCA 算法中, 有关集群机器人任务分配的关键参数介绍如下:

1) 任务路径列表 p_i

任务路径列表是分配给机器人 i 的一系列待执行任务, 表示为:

$$p_i \triangleq \{p_{i1}, \dots, p_{i|p_i|}\} \quad (4.9)$$

其中的任务元素按照机器人的执行计划进行排列, $|p_i|$ 表示列表的实际长度, 若机器人 i 的路径列表为空时定义 $p_i = \phi$ 。

2) 任务包 b_i

任务包是分配给机器人 i 的任务集合, 表示为:

$$b_i \triangleq \{b_{i1}, \dots, b_{i|b_i|}\} \quad (4.10)$$

集合中元素按任务加入包的先后顺序排列, $|b_i|$ 代表 b_i 列表的长度, 且有 $|b_i| \leq |p_i|$, $b_i = \phi$ 表示机器人 i 任务包为空。

3) 中标机器人矩阵 Z^i

Z^i 为 $n_r \times n_e$ 维矩阵, 其中 $z_{kj}^i = 1$ 表示机器人 i 认为机器人 k 是任务 j 的投标获胜者; 否则 $z_{kj}^i = 0$ 。 Z^i 中第 j 列非零元素数量之和表示为:

$$N_j^{sum} = \sum_{k=1}^{n_r} z_{kj}^i \quad (4.11)$$

其中, N_j^{sum} 代表了机器人 i 认为执行任务 j 的机器人总数。

4) 获胜投标值矩阵 Y^i

Y^i 存储机器人 i 视角下各项任务联盟成员的投标值, 矩阵中各元素与 Z^i 一一对应。当机器人 i 认为任务 j 没有获胜者时设 $y_{ij} = -\infty$ 。

5) 时间戳列表 s_i

时间戳列表 $s_i \triangleq \{s_{i1}, \dots, s_{in_r}\}$ 记录机器人 i 从相邻其他机器人中获得更新信息的时刻, 其中, s_{ik} 表示机器人 i 从机器人 k 获得最新信息的时刻。

4.2.2 任务包构建

CBCA 算法的核心就是每个机器人每次以最大化边际增益为原则选择任务, 即, 机器人 i 在每一次选择任务的时候, 选择收益增量最大的任务 j 加入到任务表 b_i 中。当任务包填满或者没有可以选择的任务时, 任务包构建过程结束。

本章采用一个随时间变化的收益函数 $r_{ij}(p_i)$, 假设第 j 个目标的初始价值为 V_{aj} , 机器人执行任务目标的收益和时间有关, 并随着时间的增大而衰减:

$$r_{ij}(p_i) = V_{aj} \cdot e^{-\beta_j(t_{(p_i)} - t_o)} \quad (4.12)$$

其中, t_o 为任务分配的起始时间, $t_{(p_i)}$ 为机器人沿着任务路径 p_i 预计执行目标 T_j 的时间, p_i 为机器人 i 需要按时间顺序完成的所有任务所构成的序列; $0 \leq \beta_j \leq 1$ 是任务时间折扣因子, 反映目标价值随时间下降的快慢。

机器人 i 沿路径 p_i 的效能为:

$$S_i(p_i) = \sum_{j \in p_i} S_{ij}(p_i) = \sum_j (w_1 r_{ij}(p_i) - w_2 \xi_{ij}) \quad (4.13)$$

式中, ξ_{ij} 为机器人的路程代价, 即:

$$\xi_{ij} = d_{ij} \gamma \quad (4.14)$$

其中, d_{ij} 为机器人 i 与任务 j 之间的直线距离, γ 为路程代价系数。 w_1 和 w_2 为效能函数中目标收益和机器人的路程代价的权值系数, 通过调整权值系数可以使得最终选定的联盟获得不同的效果。增大 w_1 , 适合于执行目标价值随时间下降较快的目标。增大 w_2 可使得机器人联盟代价较小, 节省机器人联盟的燃料

依据新任务 j' 加入 b_i 前后的收益, 可获得 j' 的边际增益为:

$$c_{ij'}[b_i] = \begin{cases} 0, j' \in b_i \\ \max S_i(p_i \oplus_n \{j'\}) - S_i(p_i), \text{ 其他} \end{cases} \quad (4.15)$$

其中, n 表示 j' 在 p_i 中所有可能位置, $|p_i|$ 表示为路径列表长度; $p_i \oplus_n \{j'\}$ 表示将 j' 插入到路径列表 p_i 中第 n 位, 原路径列表中第 n 位及其以后的元素保持原顺序不变, 依次后移。

围捕任务中每个目标必须要三个以上的机器人来协作完成，若竞标执行任务 j 的机器人数量已满足联盟机器人数量需求，即 $N_j^{sum} = L_j$ ，此时分配执行任务 j 的机器人数量已满荷，机器人将自身边际增益与当前任务最小获胜投标值作比较，若：

$$c_{ij} > \min(y_{kj}^i), \forall k \in I \quad (4.16)$$

机器人 i 可以投标 ($h_{ij} = 1$)，并将最小获胜投标值替换为自身投标值；否则，智能体 i 放弃任务 j ($h_{ij} = 0$)。当机器人 i 投标值与最小获胜投标值相同时，选择身份标识小的机器人。

由于本章考虑的是每个围捕机器人探测能力有限的场景，需要通过邻居传递任务的属性信息、投标信息等。因此，各围捕机器人在任务分配之前先根据探测范围形成多个临时小组，表示为 Ψ_s ，包括能形成通信链路的所有机器人，组内的机器人可以共享任务信息。

每一个小组基于当前已知的任务信息可以基于一致性联盟算法得出一个组内的分配结果，形成多个联盟：

$$\Phi = \{\Phi_1, \Phi_2, \dots, \Phi_{n_s}\}, i_s \in \Phi_{n_s} \quad (4.17)$$

针对初始阶段的预分配，对组内的各机器人先进行 CBCA 算法的分配，机器人不断构造自身任务包直至任务包已满或者没有满足约束条件的有效任务，下述算法为某一小组的初始预分配流程，具体算法如下：

算法 1. 任务包构建

输入：中标机器人矩阵 $Z^i(t-1)$ ，投标获胜值 $Y^i(t-1)$ ，任务包 $b_i(t-1)$ 和路径列表 $p_i(t-1)$

输出： $Z^i(t)$ ， $Y^i(t)$ ， $b_i(t)$ ， $p_i(t)$

While $|b_i(t-1)| < l_i$ do

$$c_{ij} = \max_{n \leq |p_i|} S_i(p_i \oplus_n \{j\}) - S_i(p_i), \forall j \in T$$

for $\forall j \in T$

$$\text{if } L_j > \sum_{k=1}^{R_{n_j}} z_{kj}^i \text{ then}$$

$$h_{ij} = 1$$

else if $c_{ij} > \min(y_{kj}^i), \forall k \in R$ then

```

         $h_{ij} = 1$ 
    else
         $h_{ij} = 0$ 
    end
end
 $T_i = \arg \max_j c_{ij} h_{ij}$ 
 $n_{i,j_i} = \arg \max_n S_i(p_i \oplus_n \{j\})$ 
 $b_i(t) = b_i(t-1) \oplus_{end} \{T_i\}$ 
 $p_i(t) = p_i(t-1) \oplus_{n_{j,j_i}} \{T_i\}$ 
 $y_{i,T_i}^i(t) = c_{i,T_i}$ 
 $z_{i,T_i}^i(t) = 1$ 
end while

```

4.2.3 基于一致性的冲突消解

任务包构建结束后，机器人需要与相邻机器人交换获胜机器人列表，获胜投标值列表和时间戳列表等信息，并按照一定的行动规则更新信息以获得无冲突任务分配。具体算法如下：

算法 2. 一致性联盟冲突消解

输入：中标机器人矩阵 $Z^k(t-1)$ ，投标获胜值矩阵 $Y^i(t-1)$ ，来自于 k 时间戳 $s_k(t-1)$

输出： $Z^i(t)$ ， $Y^i(t)$ ， $s_i(t)$

发送给邻居 k 的信息 $Z^i(t-1)$ ， $Y^i(t)$ ， $s_i(t-1)$

接收来自 k 的信息 $Z^k(t-1)$ ， $Y^k(t-1)$ ， $s_k(t-1)$

for $\forall j \in T$

for $\forall m \in R$ where $z_{mj}^i = 1$

if $k = m$ and $s_{km} > s_{im}$ then

$z_{mj}^i = z_{mj}^k$

$y_{mj}^i = y_{mj}^k$

end

end

```

for  $\forall m \in R$  where  $z_{mj}^k = 1$ 
    if  $m \neq i$  and  $z_{mj}^i = 0$  and  $s_{km} > s_{im}$  then
        if  $\sum_{m=1}^{R_{n_r}} z_{mj}^i < L_j$  then
             $z_{mj}^i = z_{mj}^k$ 
             $y_{mj}^i = y_{mj}^k$ 
        else
            if  $\min(y_{nj}^i \forall n) < y_{mj}^k$  and
                 $E_m \cap E_n \neq \phi$  then
                     $z_{nj}^i = 0, z_{mj}^i = z_{mj}^k$ 
                     $y_{nj}^i = -\infty, y_{mj}^i = y_{mj}^k$ 
            end
        end
    end
end
end
end
 $s_{ik} = t$ 
end

```

若各相邻机器人的视角下中标者标识相同，时间戳不同，则更新中标者矩阵，各邻居达成共识，确认出最后的中标者。若各相邻机器人视角下中标者标识不同，则判断发生冲突的机器人所投标任务的中标者是否已经达到上限，若不满足，则成为中标者，若任务所需数量已经达到上限，则和最小中标者进行效能比较，若大于最小中标者效能，则成为共识下的中标者，若小于最小者中标者效能，则失去中标者身份，重置此任务在机器人任务包中的序列所在位置，重新返回第一阶段进行任务包构建。

4.2.4 机器人忠诚度策略

在任务的执行过程中，若出现新的更有价值的任务，各机器人对任务进行动态重规划，为避免机器人出现频繁出现更换目标的情况，需要设置忠诚度策略模型，保证围捕机器人不会因为一个稍有价值的任务就脱离联盟去执行新的任务目标。每个机器人对于预分配的分配结果保有一定的忠诚度，建立机器人忠诚度函数模型如下：

$$\sigma_i = r_1 \bar{E}_i + r_2 \cdot d_{ic} + r_3 \cdot \mu_i \quad (4.18)$$

其中, r_1 、 r_2 、 r_3 为机器人的忠诚函数中的调整系数; \bar{E}_i 表示机器人 i 所具有的各项资源占联盟总资源的百分比, 即:

$$\bar{E}_i = \frac{1}{z} \sum_{w=1}^z \frac{E_i^w}{E_{C_j}^w} \quad (4.19)$$

其反映了机器人 i 对于联盟的重要性; d_{ic} 是机器人 i 在执行任务中所消耗的里程, 也就是机器人 i 在执行的过程中已经付出的沉没成本, 用下式计算:

$$d_{ic} = t_{ic} \gamma \quad (4.20)$$

其中, t_{ic} 为机器人已经行驶的时间; μ_i 是机器人 i 选择更换其他任务目标的违约成本, 表示为:

$$\mu_i = S(b_i) \quad (4.21)$$

其中, b_i 为机器人已分配的任务集合, $S(b_i)$ 表示机器人 i 所要执行的任务的收益。上式表明, 如果机器人 i 选择违约, 那么需要由其来承担其未执行的所有任务收益损失。而如果机器人更换任务路径, 选择放弃当前的任务路径, 就要满足:

$$S_i(p_i') > S_i(p_i) + \sigma_i \quad (4.22)$$

即, 只有新的任务可为机器人带来足够多的效能时才能进行任务路径的更换, 从而避免频繁的任务切换, 保证算法的收敛性。

4.2.5 基于局部视觉的部分任务重规划

受局部观测能力的限制, 在任务的执行过程中可能会探测到新的目标和其他的围捕联盟, 因此需要进行任务重规划。具体分为以下几种情况:

情况 1: 若通过视觉探测发现了新目标, 如果不确定是否有其它联盟正在围捕新目标, 则不进行重规划, 继续保持原有的任务路径。

情况 2: 若某个联盟 Φ_s 通信到了其它的联盟 Φ_{else} , 此时有两个情况:

情况 2.1: 如果通信到的联盟 Φ_{else} 已经围困住任务目标, 则不进行任务重规划。

情况 2.2: 如果通信到的 Φ_{else} 没有围困住目标, 则将 Φ_s 与 Φ_{else} 合并为一个新的联盟 Φ_{new} , 并共享任务目标信息。

对新形成的联盟进行 CBCA 算法求解, 建立新的任务包, 并进行一致性消解, 得到新的任务路径和中标收益。各机器人得到新的效能 $S_i(p_i')$, 为了避免频繁更换目标而使分配算法不收敛, 需要与 $S_i(p_i) + \sigma_i$ 进行比较, 判断是否需要按照重规划的任务路径

来执行任务：若 $S_i(p_i) > S_i(p_i) + \sigma_i$ ，则需要按照重规划的任务路径继续执行围捕任务；否则按照原先的任务路径继续执行围捕任务。

任务重规划算法是为了使机器人执行围捕任务获得更高的效能所提出的算法，由于可以降低对机器人信息获取能力的要求，使机器人可以在局部先验的任务信息下进行任务分配，提升了机器人在任务分配时的鲁棒性，相对于无重化的 CBCA 算法，获得了更高的机器人效能。

4.3 仿真与分析

在本章中，利用 MATLAB 2020b 仿真软件来对本章提出的算法进行仿真实验并和现有算法进行性能比较，以验证本章所提出算法的有效性。仿真实验在 $10\text{km} \times 10\text{km}$ 环境区域里进行，每个机器人的探测范围半径为 1.2km ，机器人间的通信半径为 1.5km ，机器人最大执行任务数 l_i 均为 3 个。在算例 1 中，基于改进的 CBCA 算法和机器人的观测能力和通信范围有限条件下，应用于一个围捕实例场景，用来说明多机器人协同合作围捕的全过程，环境中的任务目标和机器人的具体参数如表 4.1 和 4.2 下：

表 4.1 任务目标初始参数

Tab. 4.1 Initial parameters of the robot

任务序号 j	位置 (x_j, y_j)	初始价值 V_{aj}	所需资源资源 E_j	收益衰减系数 β_j	所需机器人 数 L_j
1	2.28,2.56	200	{10,9,14,15,18,15}	0.52	3
2	4.89,0.95	300	{10,9,5,5,12,8}	0.63	3
3	2.34,6.96	250	{9,10,11,7,14,7}	0.45	3
4	5.75,7.12	230	{9,10,12,8,12,6}	0.71	4

表 4.2 机器人初始参数

Tab. 4.2 Initial parameters of task objectives

机器人序号 i	位置 (x_i, y_i)	路程代价系 数 γ	携带资源 E_i'
1	1.23,0.76	0.5	{3,2,5,6,8,7}
2	1.28,2.58	0.6	{14,11,15,11,18,17}
3	2.46,1.86	0.7	{4,5,4,3,2,1}
4	3.46,2.34	0.5	{5,4,3,2,6,5}
5	4.32,1.79	0.5	{5,6,2,5,6,2}
7	5.64,7.82	0.6	{9,12,6,8,8,3}
8	6.28,8.73	0.4	{10,10,5,7,12,7}

9	7.36,9.45	0.6	{4,5,4,3,2,1}
10	0.75,1.68	0.5	{2,5,6,4,6,3}
11	6.85,6.97	0.4	{3,4,5,4,3,2}
12	7.78,7.95	0.5	{4,6,7,8,9,10}
13	8.87,8.45	0.6	{8,6,8,4,7,9}
14	8.05,8.74	0.7	{4,6,8,9,4,10}
15	7.03,5.99	0.8	{2,3,7,8,5,3,2}

在算例 1 中，对环境中的机器人进行 CBCA 无重规划任务分配策略的仿真验证，各个机器人的实时任务分配结果如图 4.2 所示。

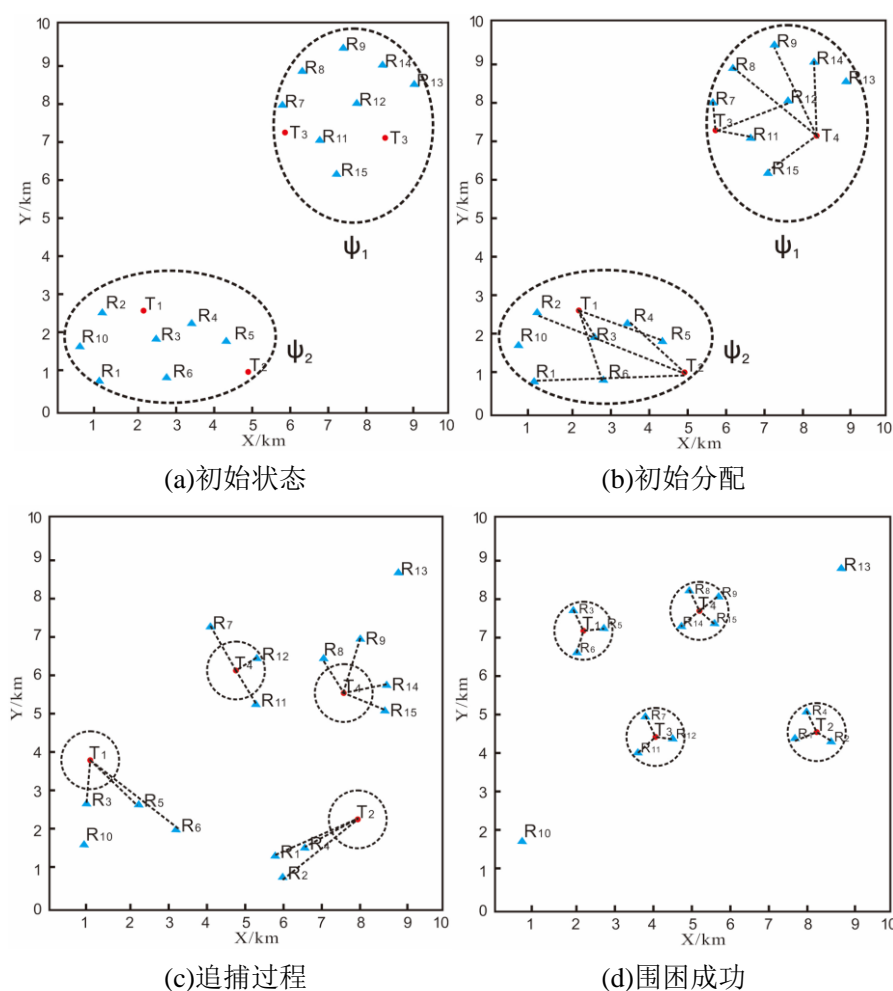


图 4.2 无重规划围捕过程

Fig. 4.2 No re-planning hunting process

图 4.2(a)为任务空间的初始状态，各个机器人 i 和任务目标 j 的位置随机生成，可见任务空间内形成了两个临时小组 Ψ_1 、 Ψ_2 。临时小组组内的成员是通过通信的连接而

形成小组,图 4.2(b)为分组进行 CBCA 算法的分配结果,各组同时进行算法求解,得出分配结果为 $\Phi_1=\{R_3,R_6,R_5\}$, $\Phi_2=\{R_1,R_2,R_4\}$, $\Phi_3=\{R_7,R_{11},R_{12}\}$, $\Phi_4=\{R_8,R_9,R_{14},R_{15}\}$ 。机器人 R_{10} 虽然距离 T_1 很近,但由于资源受限而未分配到任何目标。 R_{13} 虽然距离 T_4 也很近,但是由于任务所需的机器人投标数量已满,并且 R_{13} 也没有替换最小收益中标者的能力,所以也无法加入到 T_4 的围捕中去。图 4.2(c)中各围捕机器人已经开始了追捕,可以观察到 R_{10} 和 R_{13} 由于任务包里没有任务,因此停留在原地,没有执行任务。图 4.2(d)为各围捕机器人已将各个任务目标围困住。

如图 4.3 所示,各机器人的效能会随着时间进行衰减,而且在图像前段,各机器人效能值衰减的较快,这是由收益函数的递减速率决定的。如果想要使收益衰减的不那么快,可以适当调节系数 w_1 , 延缓目标收益的衰减。当各机器人 i 的效能趋于平缓之时,说明任务目标已经被执行完毕。机器人获得了最终目标效能。若没有其它需要执行的任务,效能曲线也就不再变化,则说明围捕任务已经结束。

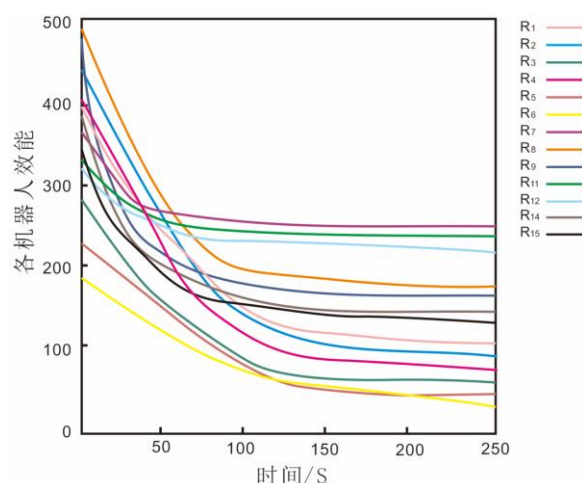


图 4.3 机器人效能

Fig. 4.3 Robot utility

在算例 2 中,利用本章所提出的基于 CBCA 算法的任务重规划算法进行仿真实验,各机器人 i 与任务 j 的初始参数如表 4.3 和表 4.4 所示:

表 4.3 任务目标初始参数

Tab 4.3 Initial parameters of task objectives

任务序号 j	位置 (x_j, y_j)	初始价值 V_a	所需资源 E_j	收益衰减系数 β_j	所需机器人数量 L_j
1	2.98,6.32	100	{11,14,10,5,8,9}	0.52	3

2	5.34,3.65	200	{12,11,9,6,5,4}	0.63	3
3	3.04,3.25	300	{9,10,7,9,10,5}	0.45	3
4	5.38,3.63	500	{10,7,9,13,12,14}	0.71	4
5	1.45,8.82	600	{9,8,10,0,13,12}	0.56	3

表 4.4 机器人初始参数

Tab. 4.4 Initial parameters of the robot

机器人序号 i	位置 (x_i, y_i)	路程代价系数 γ	携带资源 E_i'
1	2.12,9.56	0.5	{4,5,2,7,8,3}
2	3.21,8.76	0.6	{3,6,7,8,3,2}
3	1.89,7.95	0.7	{2,3,7,8,4,4}
4	5.05,8.64	0.5	{3,4,7,9,6,5}
5	5.76,8.09	0.5	{2,3,6,7,8,5}
6	4.32,7.54	0.6	{5,7,8,9,4,5}
7	1.46,0.95	0.4	{3,5,6,7,9,2}
8	2.46,2.03	0.6	{1,4,6,8,9,5}
9	4.32,2.23	0.5	{3,5,7,9,6,3}
10	6.28,1.92	0.4	{10,3,6,8,9,5}
11	4.32,0.71	0.5	{4,5,8,9,10,4}
12	6.81,0.76	0.6	{6,3,6,8,9,10}
13	3.11,0.91	0.7	{9,2,3,5,6,8}
14	1.21,2.56	0.7	{3,6,9,9,6,5}
15	2.01,6.59	0.8	{12,5,6,7,8,5}

图 4.4(a)为任务空间的初始状态，任务空间内形成了两个临时小组 Ψ_1 、 Ψ_2 ，图 4.4(b)为分组进行 CBCA 算法的分配结果，得出分配结果为 $\Phi_1 = \{R_1, R_{15}, R_5\}$ ， $\Phi_2 = \{R_2, R_4, R_6\}$ ， $\Phi_3 = \{R_7, R_8, R_9\}$ ， $\Phi_4 = \{R_{13}, R_{12}, R_{11}, R_{10}\}$ 。图 4.4(c)中各围捕机器人已经开始了围捕，并且两个临时小组的成员进入各自的通信范围，从而形成了一个新的大联盟。通过图 4.4(d)可以知道， R_9 通过重规划任务路径加入到了任务 T_5 的执行中， R_{15} 加入到了任务 T_3 的执行中， R_9 和 R_{15} 若执行新目标获得的效能为 $\{204.39, 104.53\}$ 均大于从原目标获得的效能 $\{110.56, 36.32\}$ ，而 T_1 的联盟中，由于原来执行 T_1 任务的 R_{15} 已经执行了其他任务，并且重规划时也没有符合 T_1 的资源需求的围捕联盟，因此 T_1 任务被搁置等待其他的机器人联盟发现并重新规划任务。图 4.2(e)为各围捕机器人已将各任务目标围困住。

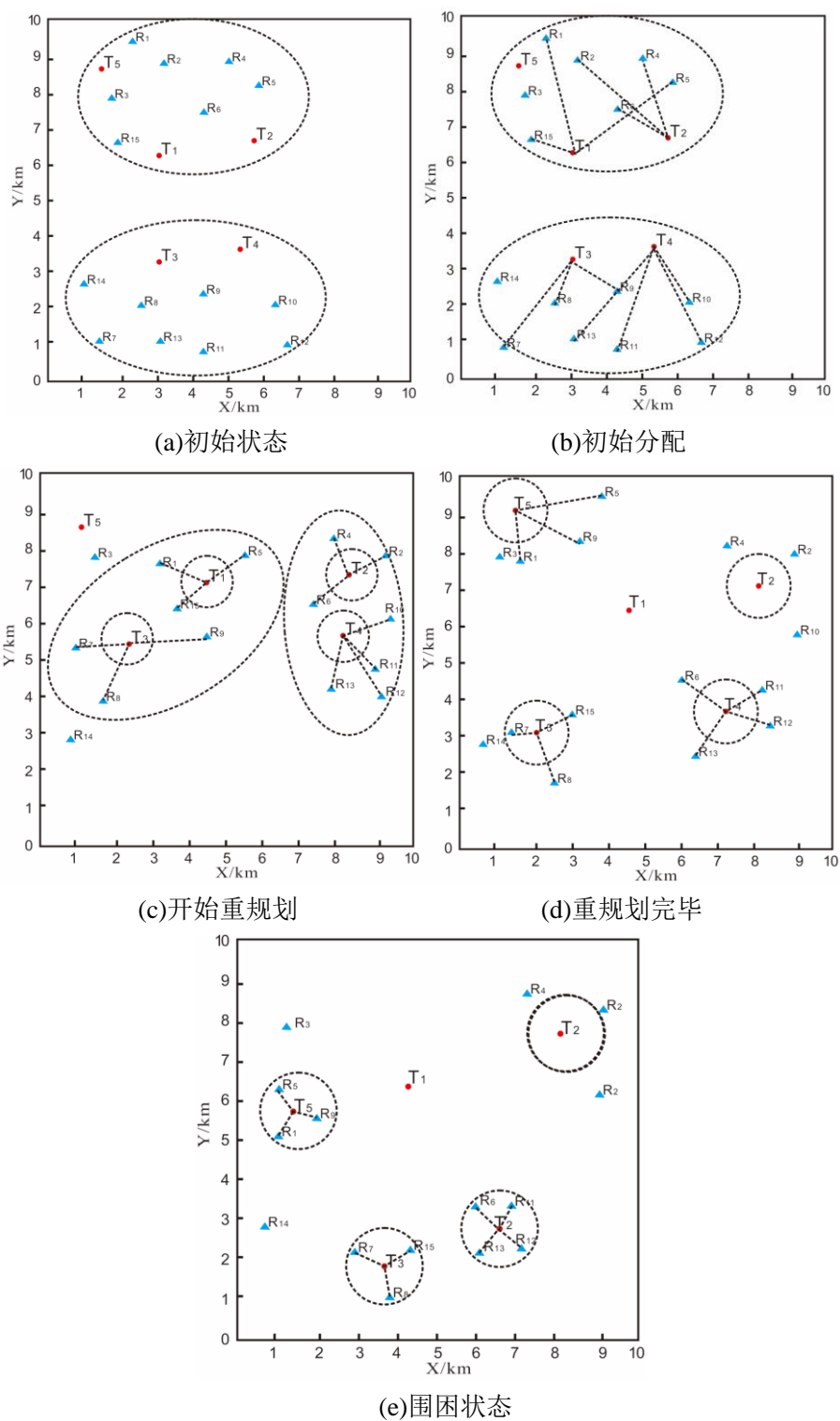


图 4.4 重规划围捕过程

Fig. 4.4 Re-planning hunting process

图 4.5 展示了各机器人效能会随着时间的变化情况。效能的两次突然提升发生在任务重规划阶段，这是由机器人选择新任务路径进行执行而产生的赔偿奖励。在重规划后，由于 R_6 比 R_{10} 更适合于 T_4 的执行而脱离原来围捕 T_2 的联盟。 R_{10} 脱离了原来的联盟后没有分配到新的目标，因此效能立即衰减为 0。而 R_4 和 R_2 由于 R_6 的违约获得了效能补偿，因此效能曲线陡增，而又因为 T_2 任务联盟解散，且没有新成员加入围捕的联盟，因此 R_4 和 R_2 的收益在此时刻后一直不变。

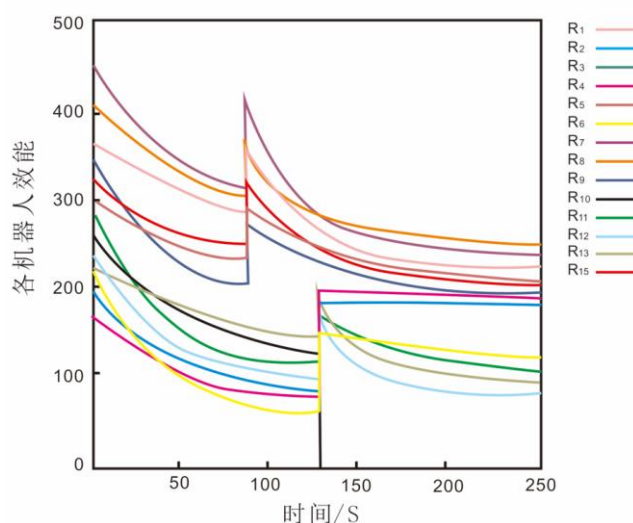


图 4.5 机器人效能

Fig. 4.5 Robot utility

当环境中的机器人数量 n_r 变化时，用本章算法与多项式时间联盟构建算法^[70]、资源福利联盟构建算法^[46]、改进的合同网算法^[71]的平均系统效能和平均任务完成时间进行比较，分别如图 4.6(a)和图 4.6(b)所示。从图 4.6(a)可以看出，当机器人数量变化时，本章所提出的算法的效能值高于其余三种算法，并且随着机器人数量的增加，本章所提出的算法效能一直大于其他四种算法。在图 4.6(a)中，机器人数量继续增大时，曲线呈现出趋于平稳的态势。这是由于在足够多的机器人数量下，任务分配算法渐渐找到最佳机器人组合。图 4.6(b)展示了四种算法的机器人平均完成时间的比较。可以看到四种算法的平均完成时间都是呈下降趋势，这是由于环境中的机器人数量不断增大，使得机器人所要任执行的任务的距离会比较近，因此，机器人执行围捕任务的完成时间会逐渐下降。而且其他算法的平均任务完成时间均低于本章算法，这是因为本章算法减少了机器人之间的交互频次，使得求解运行时间略有下降，从而提升了围捕效率。

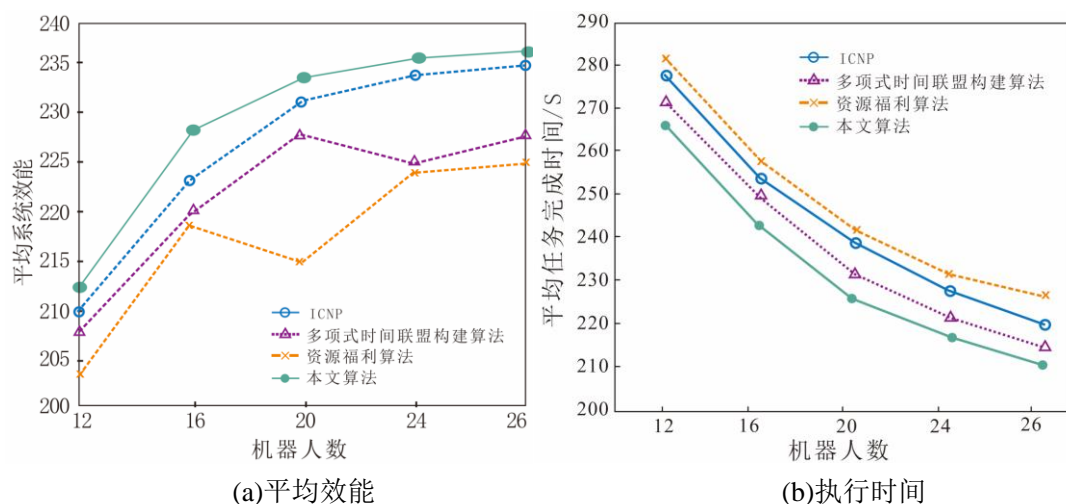

 图 4.6 任务目标数 $n_e = 30$ 时平均系统效能和执行时间

 Fig. 4.6 $n_e = 30$, Average system utility and execution time during tasks

当环境中的任务数变化时, 使用不同联盟构建算法, 机器人完成任务的平均效能和平均任务完成时间分别如图 4.7(a)和图 4.7(b)所示。

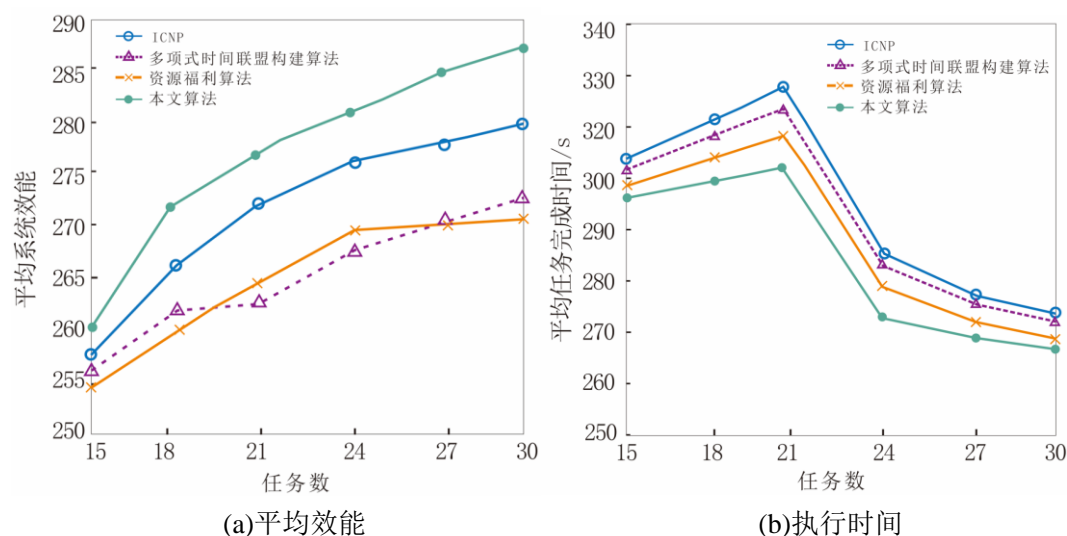

 图 4.7 机器人数量 $n_r = 13$ 时平均系统效能和执行时间

 Fig. 4.7 $n_r = 13$, Average system utility and execution time with the number of robots

从图 4.7(a)可以看出, 当任务数变化时, 本章所提出的算法的平均效能值高于其余三种算法, 并且随着任务数量 n_e 的增加, 本章所提出的算法优势更加明显。这是由于本章提出的算法为分散式结构, 仅仅通过机器人之间的协商决策就可以找到全局最优

分配结果，而其他算法只能在首次分配时获得最优结果，在围捕过程中的重分配时只能获得当前的局部最优。因此在平均系统效能这个性能上，本文提出的算法更佳。从图 4.7(b)可以看出，在机器人数量一定任务数不断增大时，各算法的曲线是先略有上升，而后呈下降趋势，这是由于环境中的任务数不断增多，一开始由于出现更适合的分配组合，因此可能会花费更多的执行时间，但任务数持续增多，任务目标和机器人会越来越密集，使得目标和机器人的距离越来越近，所以执行时间也会相应减少。

4.4 本章小结

本章研究了基于一致性联盟算法多种约束下的多机器人协同围捕任务分配问题，首先，考虑多机器人协同围捕任务目标的使用场景，建立了资源约束和路径约束等多机器人系统任务分配问题的优化模型。其次，提出忠诚度的概念及数学模型，考虑在任务执行过程中出现新目标的情况，利用一致性联盟算法进行重新规划，并将多机器人系统分组进行一致性联盟算法，减少了求解时间，围捕机器人由于预先分组，使得各机器人需要消解的冲突就会相对减少，从而可以在预分配阶段就有效地降低邻居间的信息交互信息频率和迭代次数，从而获得更高效的任务分配方法，最后通过仿真实验验证了所提出算法的有效性和优越性。

结 论

任务规划对于机器人学来说一个非常重要的分支，而多机器人围捕任务规划更是广泛应用于实际生活中。因此，设计一个高效的任务规划方法应用于围捕场景的任务分配是非常必要的。本文针对围捕任务场景的实际需求，采用拍卖算法以及一致性联盟算法求解围捕任务分配问题，主要完成的研究工作如下：

1.首先阐述了现有算法中常用的任务分配策略，找到更适合于围捕场景的算法。由于现阶段针对围捕场景的动态实时性任务规划的研究相对较少，首先对现有的围捕场景算法的缺点进行分析。其次对围捕任务类型和机器人类型进行介绍，并对任务分配的架构进行描述。然后建立了任务分配的数学模型。最后，对要采用的算法模型进行了简要阐述。

2.采用改进的拍卖算法，解决围捕任务分配问题，建立了围捕任务的优化目标函数，从拍卖商成本和投标机器人效能两方面分别优化，首先，解决了以往的拍卖算法中只对投标者进行优化的缺点，其次，提出了基于拍卖算法的重规划算法，使围捕机器人具有更高的实时性和鲁棒性以及更高的效能，解决了在围捕任务中，突然出现新的更有价值目标时的重规划问题，使机器人更具智能，更具抵抗环境中不确定性因素的能力。最后，设计围捕点分配方法，通过围捕点协同分配解决了围捕点分配中的死锁问题。

3.考虑更加接近于真实围捕任务的场景，加入更多资源约束对机器人任务分配进行限制，使机器人在有限里程和有限的异构资源下进行联盟组队。提出了一种基于一致性联盟的动态联盟任务分配算法，该算法改进了 CBCA 算法中全局先验信息已知的缺点，从而降低了机器人的通信频率，提升了任务分配算法的效率。另外，提出了一种机器人忠诚度模型，确保了算法的收敛性。仿真结果验证了所提出算法的有效性。

未来研究的展望如下：

1.由于机器人围捕目标应用的场景，并不是每一次都可以成功地执行完任务，因此，通过更合理的算法来提高机器人的围捕率是需要考虑的问题。

2.在环境中添加障碍物，使机器人配合障碍物进行围捕，减少围捕机器人的使用数量也是一个值得研究的方向。

3.对于一致性联盟算法，如果可以再进一步缩小环境中先验信息的范围，则更能降低算法中各机器人获取信息的需求。

4.对于在围捕过程中，如果有某些围捕机器人突然损坏，如何进行重规划也是在接下来研究中需要考虑的情况。

参考文献

- [1] Garrett C R, Chitnis R, Holladay R, et al. Integrated task and motion planning [J]. Annual review of control, robotics, and autonomous systems, 2021, 4: 265-293.
- [2] Khamis A, Hussein A, Elmogy A. Multi-robot task allocation: A review of the state-of-the-art [J]. Cooperative robots and sensor networks, 2015, 31-51.
- [3] Antonyshyn L, Silveira J, Givigi S, et al. Multiple Mobile Robot Task and Motion Planning: A Survey [J]. ACM Computing Surveys, 2023, 55(10): 1-35.
- [4] Rizk Y, Awad M, Tunstel E W. Cooperative heterogeneous multi-robot systems: A survey [J]. ACM Computing Surveys (CSUR), 2019, 52(2): 1-31.
- [5] 李瑞珍, 杨惠珍, 萧丛杉. 基于动态围捕点的多机器人协同策略 [J]. 控制工程, 2019, 26(3): 510-514.
- [6] 符小卫, 陈子浩. 基于一致性协议的多无人机协同围捕控制方法 [J]. 系统工程与电子技术, 2021, 43(9): 2501-2507.
- [7] 王玉杰, 唐钟南, 陈清阳, 等. 多无人机协同打击制导控制技术研究进展 [J]. 航空工程进展, 2023.
- [8] Johnson L B, Choi H L, How J P. The role of information assumptions in decentralized task allocation: A tutorial [J]. IEEE Control Systems Magazine, 2016, 36(4): 45-58.
- [9] Chen G. Pinning control and controllability of complex dynamical networks [J]. International Journal of Automation and Computing, 2017, 14: 1-9.
- [10] Sun X, Zhang Y. A review of domain knowledge representation for robot task planning [C]//Proceedings of the 2019 4th International Conference on Mathematics and Artificial Intelligence. 2019: 176-183.
- [11] Gao K, Yan X. Study on the optimal strategy of missile interception [J]. IEEE Access, 2021, 9: 22239-22252.
- [12] Sun Z, Yang J. Multi-missile interception for multi-targets: Dynamic situation assessment, target allocation and cooperative interception in groups [J]. Journal of the Franklin Institute, 2022, 359(12): 5991-6022.
- [13] Shirani B, Najafi M, Izadi I. Cooperative load transportation using multiple UAVs [J]. Aerospace Science and Technology, 2019, 84: 158-169.
- [14] Wang D, Peng Z. Pursuit-evasion games of multi-players with a single faster player [C]//2016 35th Chinese Control Conference (CCC). IEEE, 2016: 2583-2588.
- [15] 贾高伟, 王建峰. 无人机集群任务规划方法研究综述 [J]. 系统工程与电子技术, 2020, 43(1): 99-111.
- [16] Barca J C, Sekercioglu Y A. Swarm robotics reviewed [J]. Robotica, 2013, 31(3): 345-359.
- [17] 叶媛媛, 闵春平, 沈林成, 等. 基于满意决策的多 UAV 协同目标分配方法 [J]. 国防科技大学学报, 2005, 27(4): 116-120.
- [18] Dawei L, Lijuan L, Li W. On Network Dynamic Planar Location Problem Based on Matrix Operation [C]//2007 Chinese Control Conference. IEEE, 2007: 158-161.
- [19] Ekelin C, Jonsson J. Solving embedded system scheduling problems using constraint programming [J]. Chalmers University of Technology, Sweden, Report number TR 00-12, 2000.
- [20] Bollobás B. Modern graph theory [M]. Springer Science & Business Media, 1998.
- [21] 王皓轮, 倪宏, 郭志川. 基于连续效能子集的资源分配算法 [J]. 中国科学技术大学学报, 2013, 43(4): 287-294.

- [22] 周洪喜, 张进, 彭晨远, 等. 基于匈牙利算法的多导弹阵地拦截指派规划 [J]. 弹箭与制导学报, 2021.
- [23] Fujita S, Masukawa M, Tagashira S. A fast branch-and-bound algorithm with an improved lower bound for solving the multiprocessor scheduling problem [C]//Ninth International Conference on Parallel and Distributed Systems, 2002. Proceedings. IEEE, 2002: 611-616.
- [24] Cao R, Li S, Ji Y, et al. Task assignment of multiple agricultural machinery cooperation based on improved ant colony algorithm [J]. Computers and Electronics in Agriculture, 2021, 182: 105993.
- [25] Chen K, Sun Q, Zhou A, et al. Adaptive multiple task assignments for uavs using discrete particle swarm optimization [C]//Internet of Vehicles. Technologies and Services Towards Smart City: 5th International Conference, IOV 2018, Paris, France, November 20–22, 2018, Proceedings 5. Springer International Publishing, 2018: 220-229.
- [26] 凌文通, 倪建军, 陈颜, 等. 基于改进鲸鱼优化算法的多无人机围捕 [J]. 计算机与现代化, 2021 (06): 1.
- [27] de Lope J, Maravall D, Quiñonez Y. Response threshold models and stochastic learning automata for self-coordination of heterogeneous multi-task distribution in multi-robot systems [J]. Robotics and Autonomous Systems, 2013, 61(7): 714-720.
- [28] Lee J, Lee S J, Chen H M, et al. Composing web services enacted by autonomous agents through agent-centric contract net protocol [J]. Information and Software Technology, 2012, 54(9): 951-967.
- [29] Lee D H, Zaheer S A, Kim J H. A resource-oriented, decentralized auction algorithm for multirobot task allocation [J]. IEEE Transactions on Automation Science and Engineering, 2014, 12(4): 1469-1481.
- [30] 王训, 王兆魁, 张育林. 基于合作博弈的智能集群自主聚集策略 [J]. 国防科技大学学报, 2017, 39(2): 146-151.
- [31] Jang I, Shin H S, Tsourdos A. Anonymous hedonic game for task allocation in a large-scale multiple agent system [J]. IEEE Transactions on Robotics, 2018, 34(6): 1534-1548.
- [32] 李相民, 唐嘉钰, 代进进, 等. 异构多智能体联盟动态任务分配 [J]. 西北工业大学学报, 2020, 38(5): 1094-1104.
- [33] 唐嘉钰, 李相民, 代进进. 复杂约束条件下异构多智能体联盟任务分配 [J]. 控制理论与应用, 2020, 37(11): 2413-2422.
- [34] Choi H L, Brunet L, How J P. Consensus-based decentralized auctions for robust task allocation [J]. IEEE transactions on robotics, 2009, 25(4): 912-926.
- [35] Yu W Y, Huang X Q, Luo H Y, et al. Auction-Based Consensus of Autonomous Vehicles for Multi-Target Dynamic Task Allocation and Path Planning in an Unknown Obstacle Environment [J]. Applied Sciences, 2021, 11(11): 5057.
- [36] Shuo W, DongMei S H I, Yu D, et al. Research on distributed task allocation of loitering munition swarm [C]//2020 International Conference on Information Science, Parallel and Distributed Systems (ISPDS). IEEE, 2020: 162-166.
- [37] Fu X, Feng P, Gao X. Swarm UAVs task and resource dynamic assignment algorithm based on task sequence mechanism [J]. IEEE Access, 2019, 7: 41090-41100.
- [38] Braquet M, Bakolas E. Greedy decentralized auction-based task allocation for multi-agent systems [J]. IFAC-PapersOnLine, 2021, 54(20): 675-680.
- [39] Bakolas E, Lee Y. Decentralized game-theoretic control for dynamic task allocation problems for multi-agent systems [C]//2021 American Control Conference (ACC). IEEE, 2021: 3228-3233.

- [40] 张红强, 章兢, 周少武, 等. 基于简化虚拟受力模型的未知复杂环境下群机器人围捕 [J]. 电子学报, 2015, 43(4): 665.
- [41] Zhu M, Lv N, Chen K, et al. An algorithm for coalition formation of swarm aerial vehicles considering communication constraints [C]//Journal of Physics: Conference Series. IOP Publishing, 2020, 1693(1): 012225.
- [42] Qu G, Brown D, Li N. Distributed greedy algorithm for multi-agent task assignment problem with submodular utility functions [J]. Automatica, 2019, 105: 206-215.
- [43] 张子迎, 吕骏, 徐东, 等. 能量均衡的围捕任务分配方法 [J]. 国防科技大学学报, 2019, 2.
- [44] Zhai X B, Li L, Zhao X, et al. Real-time task allocation of heterogeneous unmanned aerial vehicles for search and prosecute mission [J]. Wireless Communications and Mobile Computing, 2021, 2021: 1-13.
- [45] Luan H, Xu Y, Liu D, et al. Energy efficient task cooperation for multi-UAV networks: A coalition formation game approach [J]. IEEE Access, 2020, 8: 149372-149384.
- [46] Kim M H, Baik H, Lee S. Resource welfare based task allocation for UAV team with resource constraints [J]. Journal of Intelligent & Robotic Systems, 2015, 77: 611-627.
- [47] Yan F, Zhu X, Zhou Z, et al. Heterogeneous multi-unmanned aerial vehicle task planning: Simultaneous attacks on targets using the Pythagorean hodograph curve [J]. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, 2019, 233(13): 4735-4749.
- [48] Ma Y, Bai S, Xu Z, et al. The formation method of task coalition for multi-UAV cooperation [C]//2020 IEEE 16th International Conference on Control & Automation (ICCA). IEEE, 2020: 713-718.
- [49] Chen J, Sun D. Coalition-based approach to task allocation of multiple robots with resource constraints [J]. IEEE Transactions on Automation Science and Engineering, 2012, 9(3): 516-528.
- [50] 付光远, 李源, 付文字, 等. 改进合同网在多机器人围捕任务分配中的应用 [J]. 收藏, 2019, 3.
- [51] 薛颂东, 张云正, 曾建潮. 目标围捕任务中搜索与预包围阶段的群机器人行为学习 [J]. 模式识别与人工智能, 2018, 31(4): 370-378.
- [52] Sun L, Lyu C, Shi Y, et al. Multiple-Preys Pursuit based on Biquadratic Assignment Problem [C]//2021 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2021: 1585-1592.
- [53] Bao-Fu F, Qi-Shu P, Bing-Rong H, et al. Research on high speed evader vs. multi lower speed pursuers in multi pursuit-evasion games [J]. Information Technology Journal, 2012, 11(8): 989.
- [54] Saha S, Samanta G P. A prey-predator system with disease in prey and cooperative hunting strategy in predator [J]. Journal of Physics A: Mathematical and Theoretical, 2020, 53(48): 485601.
- [55] Luo Y, Huang X, Yang J, et al. Auction mechanism-based multi-type task planning for heterogeneous UAVs swarm [C]//2020 IEEE 20th International Conference on Communication Technology (ICCT). IEEE, 2020: 698-702.
- [56] 吴子沉, 胡斌. 基于态势认知的无人机集群围捕方法 [J]. 北京航空航天大学学报, 2021, 47(2): 424-430.
- [57] Khatib O. Real-time obstacle avoidance for manipulators and mobile robots [J]. The international journal of robotics research, 1986, 5(1): 90-98.
- [58] Liu F, Liang S, Xian X. Multi-robot task allocation based on utility and distributed computing and centralized determination [C]//The 27th Chinese Control and Decision Conference (2015 CCDC). IEEE, 2015: 3259-3264.

- [59] Geng N, He Z, Rong M, et al. Distributed PSO for Rescue Task Allocation with Multi-Type Multi-Task [C]//2022 China Automation Congress (CAC). IEEE, 2022: 6743-6747.
- [60] Wang S, Liu Y, Qiu Y, et al. Consensus-Based Decentralized Task Allocation for Multi-Agent Systems and Simultaneous Multi-Agent Tasks [J]. IEEE Robotics and Automation Letters, 2022, 7(4): 12593-12600.
- [61] Hao N, Yi H, Tian C, et al. A Distributed-Centralized Dynamic Task Allocation Algorithm for UAVs Tracking Moving Targets [C]//2021 40th Chinese Control Conference (CCC). IEEE, 2021: 3774-3779.
- [62] Mazdin P, Rinner B. Distributed and communication-aware coalition formation and task assignment in multi-robot systems [J]. IEEE Access, 2021, 9: 35088-35100.
- [63] Ayari E, Hadouaj S, Ghedira K. A dynamic decentralised coalition formation approach for task allocation under tasks priority constraints [C]//2017 18th International Conference on Advanced Robotics (ICAR). IEEE, 2017: 250-255.
- [64] Du R, Chen C, Yang B, et al. Auction based task assignment for pursuit-evasion game in wireless sensor network [C]//Proceedings of the 10th World Congress on Intelligent Control and Automation. IEEE, 2012: 4435-4440.
- [65] Kim W, Cho D S, Kim H J. Sequential multi-agent task assignment using auction algorithm based on D* lite [C]//ICCAS 2010. IEEE, 2010: 938-942.
- [66] Talebpour Z, Martinoli A. Adaptive risk-based replanning for human-aware multi-robot task allocation with local perception [J]. IEEE Robotics and Automation Letters, 2019, 4(4): 3790-3797.
- [67] Yang S, He S, Song T, et al. Improved Consensus-Based Bundle Algorithm for Multi-to-Multi UAV Interception [C]//Advances in Guidance, Navigation and Control: Proceedings of 2022 International Conference on Guidance, Navigation and Control. Singapore: Springer Nature Singapore, 2023: 4522-4533.
- [68] Otte M, Kuhlman M J, Sofge D. Auctions for multi-robot task allocation in communication limited environments [J]. Autonomous Robots, 2020, 44: 547-584.
- [69] Yuan Y, Shen X, Pan W, et al. Constrained likelihood for reconstructing a directed acyclic Gaussian graph [J]. Biometrika, 2019, 106(1): 109-125.
- [70] Sujit P B, Manathara J G, Ghose D, et al. Decentralized multi-UAV coalition formation with limited communication ranges [J]. Handbook of unmanned aerial vehicles, 2014: 2021-2048.
- [71] 李明, 刘玮, 张彦铎. 基于改进合同网协议的多 Agent 动态任务分配 [J]. 山东大学学报 (工学版), 2016, 46(2): 51-56.