

面向网络化无人机集群的计算换通信分布式任务调度方法*

李杰, 陈润丰, 彭婷

(国防科技大学智能科学学院, 湖南长沙 410073)

摘要: 针对网络化无人机集群任务自主协同问题以及市场竞拍法的优缺点, 提出“计算换通信”思想及其相应的分布式任务调度方法。通过对显式和隐式冲突任务的分析, 建立任务相关智能体集合。提出基于任务抑制的局部优化方法, 用于提前消解部分任务冲突, 以减少算法迭代次数。设计基于历史竞标信息的智能体位置推断法, 为局部优化提供必要的信息输入。基于组网仿真平台与集群救援场景开展蒙特卡罗仿真实验, 结果表明, 相比于市场竞拍法中具有代表性的基于共识的捆绑算法和性能影响算法, 所提方法能够获得更少的迭代次数、更短的收敛时间以及更优的调度性能。

关键词: 分布式任务调度; 计算换通信框架; 市场竞拍法; 无人机集群

中图分类号: V19 **文献标志码:** A **开放科学(资源服务)标识码(OSID):**

文章编号: 1001-2486(2023)04-045-10



听语音
聊科研
与作者互动

Distributed task scheduling method for networked UAV swarm based on computation-for-communication

LI Jie, CHEN Runfeng, PENG Ting

(College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, China)

Abstract: Aiming at the problem of autonomous coordination of networked UAV swarm and the advantages and disadvantages of market auction method, the idea of “computation-for-communication” and its corresponding distributed task scheduling method were proposed. By analyzing explicit and implicit conflicting tasks, a set of task-related agents was established. A local optimization method based on task suppression was proposed to resolve some task conflicts in advance, so as to reduce the number of algorithm iterations. An agent position inference method based on historical bidding information was designed to provide necessary information input for local optimization. Monte Carlo simulation experiments were carried out based on the networking simulation platform and the swarm rescue scenario. The results show that compared with the representative consensus-based bundle algorithm and performance impact algorithm in the market auction method, the proposed method can obtain fewer iterations, shorter convergence time and better scheduling performance.

Keywords: distributed task scheduling; computation-for-communication framework; market auction method; UAV swarm

近年来, 无人机集群及其运用备受关注, 在灾后搜索救援、污染监测溯源等领域具有广阔的应用前景^[1-2]。集群是一组具有个体动力学的无人机, 需要依靠群体智能算法来驱动, 通过交互信息协作完成任务^[3-5]。对于编队飞行等单一或简单协作任务, 通常可采用基于共识主动性的群体智能算法^[6-7]; 对于搜索救援等具有时-空-任务约束的复杂任务, 须考虑细致的任务时间和时序, 依赖于多智能体多任务调度算法^[8-10]。

无人机集群具有分布式、无中心、自组织等特点, 鉴于这些特点, 研究人员倾向于采用分布式方法, 而非存在较高脆弱性和计算复杂度的集中式方法^[11-13]。分布式方法大致分三类: 冗余中心计

算法、分布式优化法和市场竞拍法。①冗余中心计算法是将集中式算法备份至每架无人机, 通过交互全局感知态势作为算法输入, 接着同步运行集中式算法生成并取出各自的任务调度序列^[14-16]。该方法优点在于算法运行仅需一次通信, 但缺点在于难以保证一致的全局态势输入, 这将造成调度存在冲突, 且随着节点规模增加, 算法计算量呈指数增长。②分布式优化法是各架无人机分布式并行优化生成各自的调度序列^[16-18]。该方法优点是单纯分布式方式获得较好的全局优化性能, 单点计算量小, 但缺点是严重依赖于智能体之间高频度的通信, 这对于集群而言要求过于苛刻。③市场竞拍法是各架无人机以类似市场

* 收稿日期: 2023-03-19

基金项目: 科技创新 2030“新一代人工智能”重大资助项目(2020AAA0108200)

作者简介: 李杰(1984—), 男, 湖南涟源人, 副教授, 博士, E-mail: lijie09@nudt.edu.cn

竞拍机制竞价生成各自调度序列^[8,19-20]。该方法优点在于以较低通信频度、较少通信量计算生成各自调度序列,具有适用于集群任务调度的天然优势。

市场竞拍法一般包含两个阶段:阶段 1 为计算阶段,其间各个智能体独自采用贪婪策略选择任务添加至本地调度序列;阶段 2 为通信与一致性阶段,各智能体与通信连接的智能体分享本地维护中标者列表,并以此为参照移除本地调度序列中的流标任务。两阶段循环迭代,直至全部智能体的调度序列无冲突,视为算法收敛。基于共识的捆绑算法(consensus-based bundle algorithm, CBBA)^[18]和性能影响(performance impact, PI)算法^[7]是两种最具代表性的市场竞拍类算法。前者侧重于在获得如分配任务数、任务执行时长等全局优化性前提下寻求更高的调度效率,具体表现为较少迭代轮次(或通信次数);后者侧重于进一步提升全局优化性能,但不可避免带来迭代次数增加。通常,需要在算法效率和性能之间折中,这对于两阶段市场竞拍法来说,意味着更好的任务调度性能需要更多的迭代次数,更好的调度效率则不得不损失任务调度性能。

从工程经验中发现,低成本集群对于机载计算、存储及通信组网等设备设定了更为严苛的成本要求。不过相比而言,其对于计算设备带来的制约远不及通信设备。随着芯片水平不断攀升,小型化、低功耗、高性能计算单元层出不穷,计算性能足以满足单节点分布式运算要求;但对于通信而言,即便自组网技术近年取得长足进步,但由于受制于环境因素甚至外部施加的干扰,可靠通信也难以保证,如时延、丢包、误码等问题仍客观存在^[21-23]。因此,如何在两阶段市场竞拍机制下,充分利用计算阶段客观存在的计算和存储余量,尽可能降低通信频度,以获得更短的算法收敛时间,同时保持甚至提升任务调度性能,是本文主要考虑的问题。

为此,提出一种面向网络化无人机集群的分布式任务调度方法。首先提出“计算换通信(computation-for-communication, CFC)”思想,并设计相应的实现框架,以期在维持甚至提升调度性能前提下减少通信次数、提高调度效率;其次提出任务相关局部优化方法,从通信协议中提取任务冲突节点,并设计抽样和估计策略提前消解部分潜在任务冲突;再次提出基于本地维护的历史竞标数据的智能体位置推断策略和任务相关智能体选择策略,以避免额外的通信协议;然后设计了

新的任务评分函数,既用于避免竞争更靠近其他智能体的任务,又用于作为任务相关智能体位置推断的必要输入;最后为了在更真实的模拟通信条件下验证算法的有效性,采用实物设备上运行的协议栈,并结合对无线信道传输特性的模拟,构建了一个自组网通信仿真平台并开展无人机集群搜救场景的蒙特卡罗仿真实验。

1 问题描述与基本方法

1.1 问题描述

多智能体调度是优化调度 n 个智能体分别执行 m 项任务,形成 n 组任务执行序列(简称为调度)。每个智能体 i 的调度 \mathbf{p}_i 包含任务执行次序和时间,满足任务之间偏序、任务起止时间等约束。通常可将其表述为如下带约束的优化问题:

$$\max \sum_{i=1}^n \sum_{k=1}^m s_{ik}(\mathbf{p}_i) x_{ik} \quad (1)$$

其中: x_{ik} 为二进制变量, $x_{ik} = 1$ 表示智能体 i 执行任务 k , $x_{ik} = 0$ 表示其不执行该任务; s_{ik} 是智能体 i 执行任务 k 的收益,通常与任务完成时间相关。

约束可以归纳为任务相关约束(式(2)~(3))和智能体相关约束(式(4)~(5))。

$$\sum_{i=1}^n x_{ik} \leq 1, \quad \forall k \in \mathcal{J} \quad (2)$$

$$t_{ik}(\mathbf{p}_i) \leq d_k, \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{J} \quad (3)$$

$$\sum_{k=1}^m x_{ik} \leq m, \quad \forall i \in \mathcal{I} \quad (4)$$

$$\sum_{k=1}^m x_{ik}(t) \leq 1, \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{J} \quad (5)$$

其中:约束(2)表示任意任务 k 最多只能由一个智能体执行;约束(3)表明智能体 i 执行其调度序列 \mathbf{p}_i 中任务 k 的时间不能晚于 k 的截止时间 d_k ;约束(4)表示智能体 i 最多可以承担 m 项任务;约束(5)表示智能体 i 最多同时执行一项任务。

1.2 市场竞拍法

市场竞拍法是一类优化求解上述问题的有效方法,它具有良好的分布式特性。如前所述,CBBA^[18]和 PI^[7]是市场竞拍类方法中最具代表性的方法,该研究领域内大多数其他方法都是对它们的扩展。计算换通信与市场竞拍机制的对比情况如图 1 所示。如图 1(a)所示,市场竞拍算法流程一般包括计算阶段(阶段 1,蓝色方块)和通信与一致性阶段(阶段 2,灰色方块)。如图 1(b)所示,计算通信算法流程中增加了增量计算环节(阶段 3,橙色方块)。

计算阶段:①任务移除:依据上一轮竞拍的任

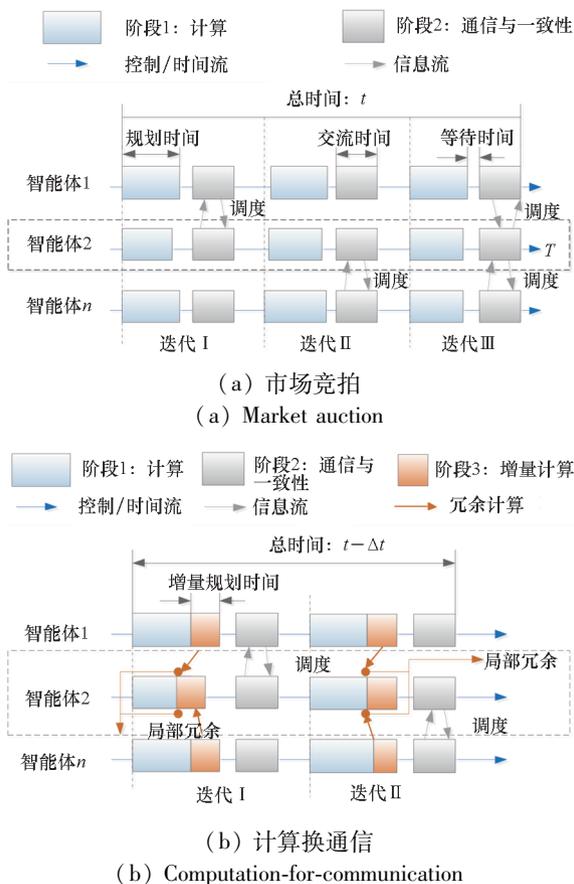


图1 计算换通信与市场竞拍机制对比

Fig. 1 Comparison between the computation-for-communication and the market auction

务价格差,按照由高到低的顺序,各个智能体逐项移除本地任务调度序列中落标的任务(出价低于上一轮中标价格);②任务包含:采用贪婪或者启发方式,各个智能体逐项将候选任务插入上一轮本地任务调度序列的最佳位置,任务的选择和插入的位置由定义的评分函数确定。

通信与一致性阶段:通信——智能体 i 将本地维护的中标者向量 $z_i \in \mathcal{G}^m$, 中标价向量 $y_i \in \mathbb{R}_+^m$, 时间戳 t 广播给通信拓扑连接的智能体(简称为邻近智能体),同时也接收其他智能体的数据。一致性——各个智能体使用统一的一致性规则表,判定各项任务的中标价格以及中标智能体,以消解任务冲突。

上述两个阶段不断循环迭代,直至智能体之间不存在冲突任务时算法终止。具体算法原理见参考文献[7,18],在此不赘述。

2 计算换通信机制设计

在计算阶段,每个智能体 i 选择某项任务 k 并计算其任务得分(即式(1)定义的评分函数 s_{ik})。对于搜索救援等时间敏感任务,各个智能体

通常以贪婪方式独立计算各自得分,当其在真实机载计算设备上运算时,计算量非常低,即使对于前期飞行实验中采用的树莓派,其计算性能也满足算法运算要求,如图2所示。随着计算设备性能的提升,相同算法所需的计算时间会越短,从而富余更多算力。

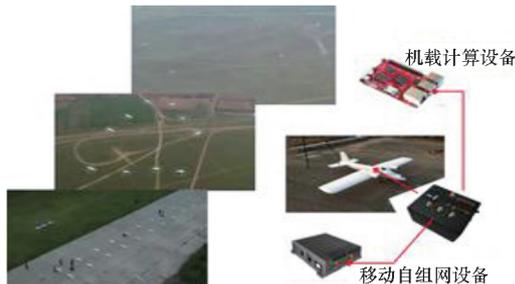


图2 前期集群飞行试验场景及机载系统

Fig. 2 Early swarm flight test scenario and airborne system

在通信与一致性阶段,每个智能体 i 将自己的调度 p_i 与邻居智能体通信,以消解智能体之间的任务冲突。尽管单次通信量不大,但多轮次通信避免不了客观存在的误码、丢包、延时甚至中断等通信问题,需要重传消息,因而通信次数增加。更糟糕的是,延时过长产生丢包,智能体之间的任务冲突无法得以消解,会出现一些任务由多个智能体重复执行,而某些任务不被执行的任务冲突情景。

鉴于机载计算设备算力过剩而通信问题难以避免,为此尝试“扬计算之长,避通信之短”(简称以计算换通信),通过适当增加计算量来寻求通信次数的减少,从而维持甚至减少算法收敛总时长。

2.1 基本思想

在无死锁情况下,影响市场竞拍类算法效率(即:算法收敛速度)的因素一般有三个:一是冲突任务数,这与智能体或任务的分布和算法的性能有关;二是任务评分的单调性,如果评分不单调,智能体偏好的任务可能会因为分数低而丢失,需要重新选择任务进行迭代,甚至会导致任务选择循环陷入僵局;三是任务分数的更新,删除或添加任务后,一旦任务分数更新,其属性可能会发生变化,需要再次迭代以消解冲突。

因此,如何加速收敛,可从以下三方面着手:一是减少任务冲突,通过计算并比较其他智能体的任务分数,从而提前消解冲突;二是保证任务分数的单调性,这可以通过评分函数的设计来保证,但通常难以实现;三是避免分数波动,建议减少任

务分数的更新,如 CBBA 的捆绑集中任务评分维持不变。

基于上述分析,提出计算换通信的基本思想:引入局部优化来替代原有的贪婪式计算策略,延长计算阶段运算周期(图 1(b)橙色方块),提前消解部分潜在的任务冲突,以期在维持甚至提升算法性能的前提下,减少算法迭代(或通信)次数,并缩短总的任务调度时间。

2.2 流程设计

图 3 给出了 CFC 流程的总体设计,它仍然沿用了计算和通信两阶段基本步骤。在任务包含运算和任务移除运算之间增加了一个局部优化算子(绿色部分),该算子将其他智能体的任务得分估计作为参考来选择任务,提前消解潜在的任务冲突,从而减少算法迭代次数,它也等于冲突消解的通信次数。此外,由于局部优化算子需要选择相关智能体并获取其位置,设计了智能体位置推断法,它仅需少量的存储空间用于记忆从其他智能体接收到的历史竞拍数据。

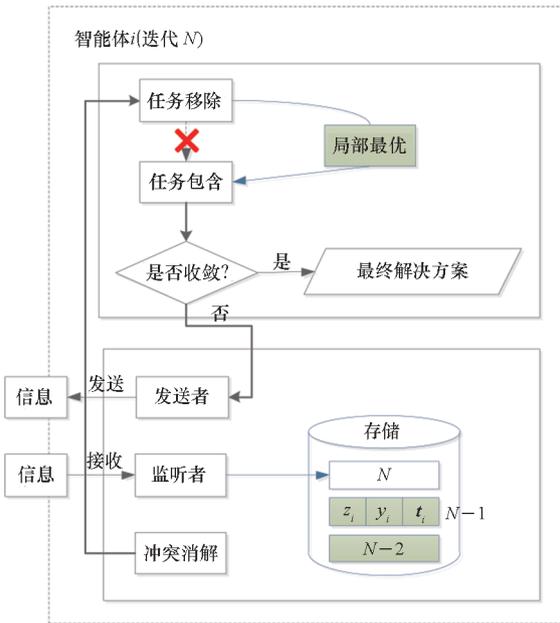


图 3 计算换通信流程设计

Fig. 3 Process design for CFC

3 任务相关局部优化方法

在市场竞拍法中,本地计算通常是自私的,即智能体只考虑对自身调度的优化。由于没有考虑其他智能体的同步调度,智能体之间竞争相同的任务时可能会抢更适宜于由其他智能体来执行的任务,需要更多次数的迭代,收敛时间较长。该优化方式虽然简单,计算量少,但冲突较多、性能较

差,且造成算力过剩和通信风险。

针对上述问题,提出一种任务相关局部优化方法,用于取代 CBBA 或 PI 等方法普遍采用的贪婪选择方式。首先,通过对与智能体 i 存在显式和隐式冲突的任务分析,构造任务相关智能体集合,它限定了参与本地局部优化的智能体范围,避免了不相关智能体造成对计算效率的影响;然后,基于上述集合可以计算得到对应的冲突任务集,将其作为本地局部优化的主动抑制任务集,当其用于局部优化时,遭抑制的任务就有可能被其他智能体竞拍得到,这样为产生更优性能的调度更加有益;最后,基于对相关智能体位置的预测,为各个智能体在本地虚构一个局部竞拍过程,并参照抑制任务集合尝试禁止本地智能体选择单项或者多项任务,用于调整智能体本地调度序列,而非仅贪婪地追寻本地优化最大化。

由于相当于预测了任务相关智能体的调度,上述过程使得智能体有可能提前避免与存在竞争关系的智能体发生任务选择冲突,而不是单纯依靠通信与一致性来消解冲突,并且有助于避免陷入个体贪婪的局部最优。

3.1 任务相关智能体选择

如果将过多的智能体全部纳入局部优化,迭代(或通信)次数很可能显著下降,但单个迭代周期内计算耗时过长,从而造成算法效率不升反降,这违背了计算换通信的初衷。因此,计算增量应该聚焦那些可能造成实际或潜在竞争的智能体。通过以下定义和计算来选择这些智能体:

定义 1(任务相关智能体集合 L) 集合内的智能体对同一任务感兴趣,可能存在冲突,其由与智能体 i 显性相关智能体集合 L_e 和隐性相关智能体集合 L_i 组成:

$$L = \{i\} \cup L_e \cup L_i \quad (6)$$

定义 2(显性相关智能体集合 L_e) 由当前存在显性任务冲突的智能体组成,智能体 i 从本地存储空间 ϕ 中获得当前迭代轮次 φ 中智能体 j 的中标者向量 z_j ,然后求出所有与智能体 i 的调度 p_i 有任务冲突的智能体 j 。

$$L_e = \{j | z_j[p_i] \neq z_i[p_i], z_j \in \phi_\varphi, j \in \mathcal{J}\} \quad (7)$$

定义 3(隐性相关智能体集合 L_i) 当前没有显性任务冲突情况下,由将来可能发生冲突的智能体组成。有很多可采纳的探索隐性相关智能体的策略 π ,基本思路都是利用其他智能体 j 的中标价向量 y_j 来推断调度 p_j 或位置 l_j 。

$$L_i = \{j | \pi_i(p_j, l_j) \leftarrow y_j \in \mathbb{R}^m, j \notin L_e, j \in \mathcal{J}\} \quad (8)$$

这里给出寻找隐性相关智能体的策略:从智能体 i 当前调度 p_i 开始,遍历以某项任务 k 为中心、距其累积距离 $d[p_i^k]$ 为半径的圆周范围内的智能体。

$$\pi_i(p_i) = \{d[p_j^k] \leq d[p_i^k], p_j \leftarrow y_j \in \mathbb{R}_+^m\} \quad (9)$$

图4描绘了一个简单的任务相关智能体选择示例。寻找显性相关智能体,智能体1的调度序列为 $O_1 \rightarrow A \rightarrow B \rightarrow C$,并与智能体2通信,告知其任务 C 的总价比智能体2更高,因此智能体2是智能体1的显性相关智能体;接着寻找隐性相关智能体:以 A 为中心,以 O_1A 为半径的圆内没有智能体,继续探索;智能体3在以 B 为圆心、 $O_1A + AB$ 为半径的圆内进行搜索,因此智能体3是智能体1的隐性相关智能体。

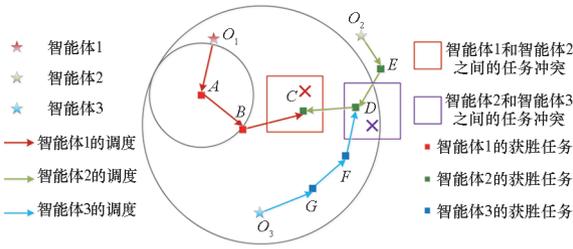


图4 任务相关智能体选择示例

Fig. 4 Example of task-related agent selection

任务相关智能体选择具体过程如算法1所示。

算法1 任务相关智能体选择

Alg. 1 Selection of the task-related agents

输入:本地维护的智能体 j 的中标者向量 z_j

输出:任务相关智能体集合 L

接收智能体 j 的轮次 φ 的中标者向量 z_j

计算显性集合 $L_e = \{j | z_j[p_i] \neq z_i[p_i], j \in \mathcal{J}\}$

for each 任务 $k \in p_i$

for each 其他智能体 $j \neq i, j \in L_e, j \in \mathcal{J}$

获得智能体 j 的调度 $p_j \leftarrow \text{cal}(y_j[z_i=j], I_i)$

if $d[p_j^k] \leq d[p_i^k]$

将智能体 j 插入隐性集合: $L_i \leftarrow L_i \cup \{j\}$

end if

end for each

end for each

3.2 基于任务抑制的局部优化

基于上述定义的任务相关智能体集合,本节计算得到相对应的候选抑制任务集合,它将在局部优化时作为本地禁止选择任务使用。

定义4(抑制任务集 U_i) 其由与智能体 i 存

在显性冲突和隐性冲突的任务组成,

$$U_i = U_i^e + U_i^i \quad (10)$$

其中:

1) U_i^e 为显性冲突任务集,它由 L_e 中与本地调度序列 p_i 发生冲突且竞价更高的任务组成。

$$U_i^e = \{p_i^k | z_j[p_i^k] \neq z_i[p_i^k], y_j[p_i^k] > y_i[p_i^k], j \in L_e\} \quad (11)$$

2) U_i^i 为隐性冲突任务集,它由 L_i 中智能体沿着各自调度序列到达 p_i 中任务 k 的估计到达时间 $d[p_i^k]$ 早于智能体 i 的任务组成。

$$U_i^i = \{p_i^k | d[p_j^k] \leq d[p_i^k], j \in L_i\} \quad (12)$$

接下来,通过逐项抑制集合 U_i 中的任务来开展局部竞拍。为了控制计算量,只允许对每项任务实施一次抑制操作 u_{ik} ,即智能体 i 被禁止选择任务 k 。其中,最佳抑制任务 k^* 是最大化估计局部得分的任务。

$$u_{ik^*} = \operatorname{argmax}_{u_{ik} \in U_i} S_L \quad (13)$$

式中, S_L 为局部优化性能评价指标,它是包括智能体 i 自身在内的与之相关所有智能体的任务回报。

$$S_L = S_i^{p_i} + \sum_{j_1=1}^{|L_e|} S_{j_1}^{p_i} + \sum_{j_2=1}^{|L_i|} S_{j_2}^{p_i} \quad (14)$$

式中, $S_i^{p_i}$ 是智能体 i 执行调度 p_i 时间折扣回报^[19]。

$$S_i^{p_i} = \sum \lambda_k^{\tau_k^i(p_i)} \bar{c}_k \quad (15)$$

式中, $\lambda_k < 1$ 是任务 k 的折扣因子, $\tau_k^i(p_i)$ 通常为智能体 i 沿着调度序列 p_i 到达任务 k 所在位置的估计到达时间, \bar{c}_k 是执行任务 k 的静态得分。

基于任务抑制的局部优化具体过程如算法2所示。

算法2 基于任务抑制的局部优化

Alg. 2 Local optimization based on task disqualification

输入:抑制任务集合 U_i

输出:智能体 i 新的调度序列 p_i

for each $k \in U_i$

智能体 i 主动抑制任务 k

估计调度 $p_j, \forall j \in \mathcal{J} \subset L$, 其中 L 由算法1得到

计算任务相关智能体总回报 S_L

end for each

计算 $u_{ik^*} = \operatorname{argmax}_{u_{ik} \in U_i} S_L$, 寻找最佳抑制任务 k^*

基于最佳抑制,智能体 i 虚构局局部竞拍得到调度 p_i

3.3 任务相关智能体位置推断

为了实现上述任务相关优化,需要获得相

关智能体的位置,这通常只能通过通信得到。但是,由于市场竞拍法的数据协议中不包含智能体的位置,因此需要在原有竞拍方法的通信协议上进行扩充,但这将产生额外的协议开销。构思不扩充协议维持原有协议的开销,尝试从前几轮通信接收到的历史竞拍中推断其他智能体的位置。

由于不考虑通信获得相关智能体位置,通过改进评分函数来隐含智能体与任务之间的距离信息,它为智能体位置推断提供必要的信息输入。评分函数是智能体选择任务的依据,它驱动各个智能体选择并排列出更为合适的待执行任务序列。提出的评分函数为:

$$s_{ik} = \max_{l \leq |p_i|} |(S_i^{p_l \oplus i^k} - S_i^{p_i}) + S_i^{p_{ik}}| \quad (16)$$

其中: $S_i^{p_{ik}} = \lambda_k \tau_i^k(p_{ik}) \bar{r}_k$ 是智能体依照 p_i 执行任务的回报; $\lambda_k < 1$ 为任务 k 回报的折扣因子,表明回报递减速度; $\tau_i^k(p_{i*})$ 为智能体 i 沿 p_i 到达任务 k 位置的估计到达时间, \bar{r}_k 为任务 k 的初始任务回报。

式(16)中加号前一项是常见的评分函数^[8,18],其表示任务 k 对 p_i 的影响,驱动智能体 i 在前一项任务基础上探索新任务,但不顾及新任务是否离自己较远或者离其他智能体较近,这将导致盲目的探索。为此补充加号后一项来引导智能体选择其自身位置附近任务,使智能体尽量避免竞争离其他智能体较近的任务,这有助于减少智能体之间任务冲突。并且每个智能体尝试选择离自己较近的任务,这样可以避免距任务较近的智能体盲目等待较远的智能体所造成的任务失败。

任务竞价是由式(16)评分函数生成,其不受外部环境等因素的影响,为此只要在评分函数中纳入智能体与任务的空间分布关系,就可利用收到的历史竞拍数据进行推断,得出智能体位置。智能体 i 对任务 k 的竞价由式(16)计算,智能体 i 到任务 k 的估计到达时间为智能体 i 沿调度序列到达任务 k 的累计距离除以智能体 i 的速度,因此其距离可由评分函数的反函数得到:

$$d_{ik} = s_{ik}^{-1}(I_k, \lambda_k, \bar{r}_k, v_i) \quad (17)$$

其中,对于所有智能体,一般认为任务 k 的位置 I_k 、值 \bar{r}_k 、折扣因子 λ_k 以及速度 v_i 均为已知。

在计算得到智能体 i 到任务 k 的距离 d_{ik} 后,智能体 i 的位置在以任务 k 的位置为圆心、距离 d_{ik} 为半径的圆周上。从 3 个圆来确定一个交集,可以看出,获得智能体 i 的位置只需要 3 个不同

的任务 k 。图 5 显示了一个简单的例子来说明推断方法。遵循竞拍数据协议,智能体 i 并不知道智能体 j 的位置。不过,在前序迭代轮次中,任务 T_1, T_2, T_3 被依次加入智能体 j 的调度,且这几项任务由式(16)计算所得的评分已通信至智能体 i 。为此智能体 i 可以根据式(17)计算智能体 j 与任务 T_1, T_2, T_3 之间的距离 d_1, d_2, d_3 。以 T_1, T_2, T_3 为圆心,以 d_1, d_2, d_3 为半径的 3 个圆的交点就是智能体 j 的位置。

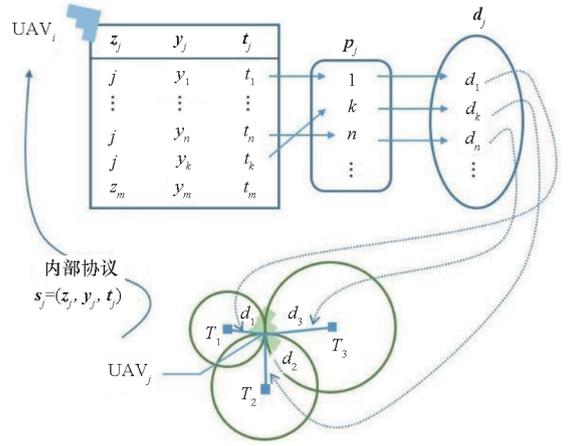


图 5 基于历史竞拍数据的智能体 j 位置推断
Fig. 5 Location inference of agent j based on historical auction data

任务相关智能体位置推断的具体过程如算法 3 所示。

算法 3 智能体位置推断方法

Alg. 3 Agent's location inference method

输入:本地存储的相关智能体 j 的历史竞标 (z_j, y_j)

输出:智能体 j 的估计位置

接收智能体 j 第 φ 轮次中标者向量 z_j 及价格列表 y_j

获得智能体 j 的调度: $p_j(\varphi) \leftarrow y_j[z_j[\varphi] = j]$

从智能体 j 调度 $p_j^k(\varphi)$ 解析任务 k 与智能体 j 的距离:

$d_{jk} \leftarrow p_j^k(\varphi), k \in \phi_j(\varphi)$

添加距离 d_{jk} 至本地缓存 $\phi_j \leftarrow \phi_j + d_{jk}$

if 存储器 ϕ_j 中的任务数满足 $|\phi_j| \geq 3$

通过圆交叉推断智能体 j 的位置

end if

4 算法流程

算法 4 概括了整个算法的流程,区别于 CBBA,主要是在计算阶段的任务移除和包含环节之间插入局部优化算子,通信与一致性阶段沿用已有算法^[19]。

算法4 算法流程

Alg. 4 Algorithm flow

输入:智能体位置、任务位置等

输出:任务调度序列 p_i 计次器 $T \leftarrow 1$ 和收敛标志量 $flag \leftarrow 0$ **while** $\omega = 0$

阶段1:计算阶段

任务移除步骤

本地局部优化

任务包含步骤

阶段2:通信与一致性阶段

冲突消解,见文献[19]

 $flag \leftarrow \text{CheckConvergence}$ $T \leftarrow T + 1$ **end while**

算法开始,每个智能体初始化计次器 T 和收敛标识符 $flag$ 。接着持续循环计算和通信阶段,直至收敛,在循环期间检查收敛性并更新计次器。

4.1 任务移除

任务移除阶段的主要工作是移除智能体本地调度与全局获胜调度之间存在冲突的任务,从而消除任务冲突,实现全局调度一致性。其具体运算过程如算法5所示。

算法5 任务移除步骤

Alg. 5 Task removal step

输入:当前本地调度序列 p_i 及其对应捆绑集 b_i 输出:移除运算后的任务调度序列 p_i **for each** 智能体 i 的 b_i 中的任务 k **if** 冲突 $z_i[k] \neq i$ 从 p_i 和 b_i 中移除任务 k

重置中标者向量和中标价向量

end if更新当前调度序列 p_i 中任务时间**end for**

特别地,与传统的市场竞拍法不同,没有删除第一个冲突任务之后的所有任务,而是只删除冲突任务。这有利于提高收敛速度,也保证了局部优化中任务交换分配任务的一致性。

4.2 任务包含

任务包含阶段的主要工作是更新本地任务调度序列,在未分配或标价低的任务中选择最可行的任务并添加至智能体的调度序列中。其具体运算过程如算法6所示。

算法6 任务包含步骤

Alg. 6 Task inclusion step

输入:当前本地调度序列 p_i 输出:包含运算后的任务调度序列 p_i **while** $|p_i| \leq m$ 按照式(16)评分智能体 i 执行任务 $k \in \mathcal{J} - p_i$ $k^* = \text{argmax}(s_{ik} - y_k)$ **if** $s_{ik^*} - y_{k^*} > 0$ 智能体 i 添加最佳任务 k^* 至 p_i 和捆绑集 b_i 更新任务 k^* 中标者及标价: $z_i[k^*] = i, y_i[k^*] = s_{ik^*}$ 更新智能体 i 调度 p_i 中任务的时间戳**end if****end while**

与传统最优任务选择方法不同,不选择得分最高任务,而是选择得分增益最大任务,即与中标者标价相差最大任务。这有利于全局优化,在一定程度上避免陷入贪婪策略导致的局部最优。

5 仿真实验

为了验证 CFC 方法,不失一般性,采用文献[8-10]描述的搜救场景,使用无人机集群为伤员或被困人员运送药品和食品,以此为任务背景开展仿真实验。假定 n 架无人机集群在 $10 \text{ km} \times 10 \text{ km} \times 1 \text{ km}$ 三维空间同时发现地面有 m 名伤员或被困人员,随即以分布式方式同步运行调度算法生成各自待执行任务序列(即调度)。无人机分别以 30 m/s 和 50 m/s 的航速执行药品和食物运送任务,单处投放药品和食物的任务持续时间分别为 300 s 和 350 s ,各项任务结束的后墙时间随机分布在 $[0, 2000] \text{ s}$ 内。为了统计算法性能,在不同无人机数量 n (取值范围为 $6 \sim 16$)和任务-智能体的数量比值 $p = m/n$ (取值范围为 $3 \sim 5$)及随机分布飞机/任务位置条件下开展了 100 组蒙特卡罗仿真实验,并将 CFC 与 CBBA、PI 进行比较。

不同于大多数算法测试所采用的基于拓扑全连接、带宽不受限以及理想信道状态的假设,所有算法均运行于自组网通信仿真平台上,其协议栈与真实通信设备一致,传输层协议选择用户数据报协议(user datagram protocol, UDP),而非传输控制协议(transmission control protocol, TCP),尽管 TCP 可提供可靠的数据传输,但 UDP 允许广播,更适用于竞拍类算法;数据链路层选择动态时分多址(dynamic-time division multiple access, D-TDMA)协议,与带有冲突避免的载波侦听多路访

问 (carrier sense multiple access with collision avoid, CSMA/CA) 协议相比,当节点数量较大时, TDMA 具有更短的端到端延迟和更高的吞吐量; 路由协议选择以 Linux 内核模块的形式运行在数据链路层的 batman-adv 协议,通过介质访问控制 (media access control, MAC) 寻址代替传统的 IP 寻址实现数据透传。物理信道模拟采用 Bernoulli 模型模拟丢包^[24]、Rayleigh 衰落模型模拟路径衰减^[25]。值得注意的是,假定集群内无人机以 Mesh 方式通信连接,这一假设有助于呈现算法本身的特点,而非改变拓扑连接造成的性能差异。选取的通信仿真参数为:通信范围为 15 km,全网带宽为 2 Mbit/s,发射功率为 30 dB,接收灵敏度为 -25 dB, $d_0 = 1$ 处的路径损失为 40 dB,路径损失指数为 2.5。

5.1 时间性能

5.1.1 迭代次数 (或通信次数)

迭代次数是算法收敛 (即所有智能体的调度之间不存在任务冲突) 所需的迭代次数,其不受计算机性能的影响,是反映方法时间性能的重要指标。如前所述,由于每轮迭代都包含计算和通信两个环节,因而迭代次数等同于通信次数。

三种算法的迭代次数比较情况如图 6 所示,从图可以看到, CBBA 的迭代次数显著少于 PI,这是由于 PI 在任务包含和任务移除阶段都更新了任务的重要性,其产生了更多潜在的任务冲突,从而导致需要更多的迭代轮次实现算法收敛,实验结果与文献 [8] 结论一致。三种算法中, CFC 的迭代次数最低,得益于前文所述的任务相关优化策略提前消解了与其他无人机可能存在的部分任务冲突。在 n 取值为 6 ~ 16 和 p 取值为 3 ~ 5 范围内, CBBA 和 PI 所需的通信次数区间分别为 [3.4, 6.1] 和 [6.9, 14.4], CFC 的通信次数区间为 [2, 3.6]。相对于 CBBA 和 PI, CFC 分别降低了 34% ~ 63% 和 66% ~ 84% 的通信次数。

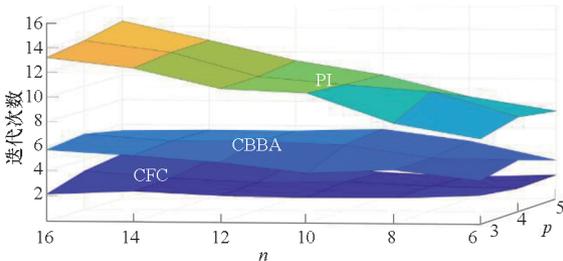
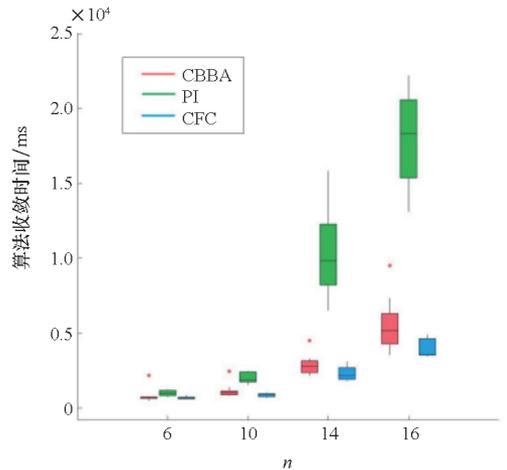


图 6 不同 p 和 n 组合下三种算法的迭代次数比较
Fig. 6 Comparison of iteration times of three algorithms under different combinations of p and n

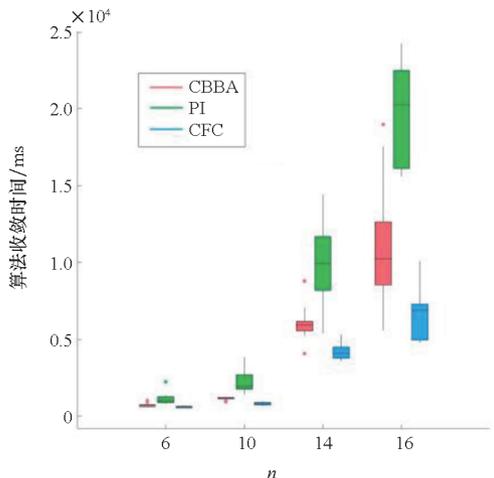
5.1.2 运算时长

运算时长是所有迭代运算周期内计算、通信以及同步等待的时间总和,也是算法的收敛时间。由于三种方法均需要通信同步,因此每个计算 + 通信周期,都有同步等待时间。该项指标尽管与计算机性能相关,但能够反映算法的时间性能。

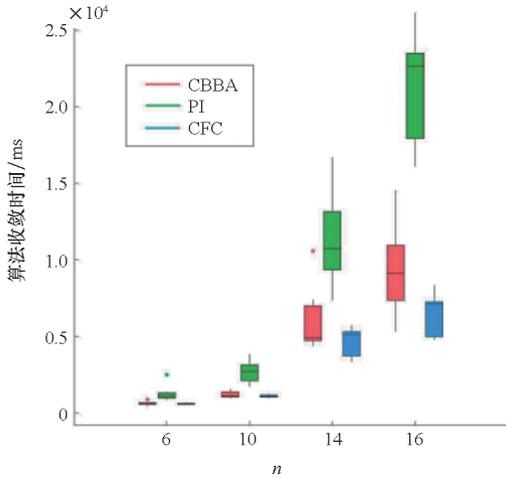
图 7 比较了三种算法在不同 p 和 n 组合下的收敛时间,即为计算生成任务调度结果的总时长。首先比较 CBBA 和 PI,二者分别需要 [628, 11 295] ms 和 [978, 21 418] ms 时间区间计算得到任务调度序列,可见 PI 收敛时间明显长于 CBBA,这是由于 PI 为了寻求更好的优化性能,客观需要更多的计算 + 通信时间,这与前述结论一致;而 CFC 仅需要 [586, 6 655] ms 时间区间,相比于 CBBA 与 PI, CFC 分别缩减了 5% ~ 41% 和 32% ~ 78%。由此看出,尽管增加了一定计算时间,但由于减少了迭代次数, CFC 总调度时间最短。



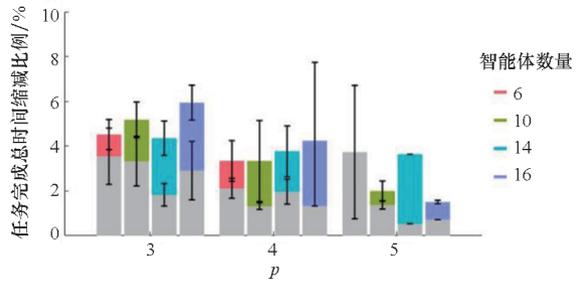
(a) $p = 3$



(b) $p = 4$



(c) $p = 5$



(b) 任务完成总时间缩减比例

(b) Reduction ratio of total task execution time

图7 不同 p 和 n 组合下三种算法收敛时间比较
Fig.7 Comparison of convergence time of the algorithms under different combinations of p and n

图8 不同 p 和 n 组合下集群任务完成总时间比较
Fig.8 Comparison of the total time of tasks under different combinations of p and n

5.2 优化性能

任务完成总时间是救援场景中最重要的一项评价指标,用于判定任务调度算法的优化性能。显而易见,完成药品和食物投送任务的总时间越短,伤员或被困人员所面临的风险就会越小。

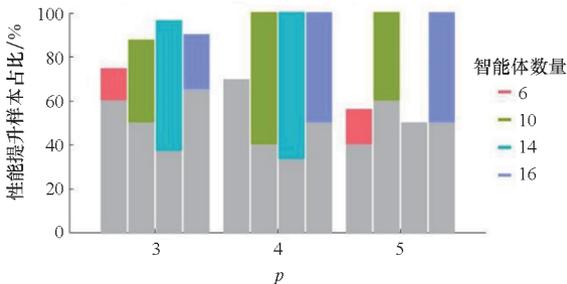
图8显示了CFC相比CBBA和PI在任务完成总时间性能提升情况。图8(a)为取得性能提升的样本数占蒙特卡罗仿真总样本数的比例,其中彩色立柱的高度表示CFC相对于CBBA取得的性能提升比例,灰色立柱的高度则表示CFC相对于PI取得的性能提升比例,可以看出,CFC较CBBA最低改善率在50%以上(位于 $p = 5, n = 6$ 立柱),特别是 p 和 n 较大时,改善率高达100%(如: $p = 5, n = 16$),这表明集群及其任务规模越大,改善效果越好;与PI相比,最低改善率也不低于35%(位于 $p = 4, n = 14$ 立柱),最高可达70%(位于 $p = 4, n = 6$ 立柱)。图8(b)为任务完成总时间缩减比例,可以看出,CFC比CBBA最多可减少约8%的任务完成总时间(位于 $p = 4, n = 16$ 立柱),平均减少约4%(见各根彩色立柱高度);与

PI相比,CFC方法可使任务完成总时间最多减少7%(位于 $p = 5, n = 6$ 立柱),平均减少约2.5%(见各根灰色立柱高度)。

综上所述,相比于CBBA和PI,CFC可以有效减少迭代次数,因而减少通信次数,并减少算法收敛时间,即调度生成时间,与此同时还能够取得不同程度的优化性能提升。

6 结论

面向网络化无人机集群,提出基于计算换通信机制的分布式任务调度方法。①提出一种计算换通信思想,并设计形成相应实现框架,以期在维持甚至提升调度性能前提下减少通信次数、提高调度效率;②提出的任务相关局部优化方法可从如CBBA、PI等算法依托的通信协议中提取任务冲突节点,并使用抽样和估计策略提前消解部分潜在任务冲突;③提出基于本地维护历史竞标数据的智能体位置推断策略和任务相关智能体选择策略,以避免额外的通信协议开销及无关计算;④设计了新的任务评分函数,既用于避免竞争其他智能体附近的任务,又用于作为任务相关智能体位置推断的必要输入;⑤为了在更真实的模拟通信条件下验证算法的有效性,采用实物设备上运行的协议栈,并结合对无线信道传输特性模拟,构建了自组网仿真平台并开展集群搜救场景仿真实验;⑥蒙特卡罗仿真结果表明,与CBBA和PI相比(在 n 取值为6~16和 p 取值为3~5范围内),CFC算法分别降低了34%~63%和66%~84%通信次数,总调度时间分别缩减了5%~41%和32%~78%,这说明CFC机制可通过适当延长计算时间来减少通信次数,从而减少任务调度时间。当然上述结论都是在无须通信协议扩充和不增加协议开销情况下达成的。



(a) 性能提升样本占比

(a) Performance improvement sample quantity ratio

参考文献 (References)

- [1] 贾永楠, 田似营, 李擎. 无人机集群研究进展综述[J]. 航空学报, 2020, 41(增刊1): 723738.
JIA Y N, TIAN S Y, LI Q. Recent development of unmanned aerial vehicle swarms[J]. Acta Aeronautica et Astronautica Sinica, 2020, 41(Suppl 1): 723738. (in Chinese)
- [2] 王祥科, 刘志宏, 丛一睿, 等. 小型固定翼无人机集群综述和未来发展[J]. 航空学报, 2020, 41(4): 023732.
WANG X K, LIU Z H, CONG Y R, et al. Miniature fixed-wing UAV swarms: review and outlook[J]. Acta Aeronautica et Astronautica Sinica, 2020, 41(4): 023732. (in Chinese)
- [3] ŠOŠIĆ A, KHUDABUKHSH W R, ZOUBIR A M, et al. Inverse reinforcement learning in swarm systems [C]// Proceedings of 16th Conference on Autonomous Agents and Multiagent Systems, 2017.
- [4] MA A, OUMET M, CORTÉS J. Hierarchical reinforcement learning via dynamic subspace search for multi-agent planning[J]. Autonomous Robots, 2020, 44(3): 485 - 503.
- [5] 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, 2021, 43(1): 99 - 111.
JIA G W, WANG J F. Research review of UAV swarm mission planning method [J]. Systems Engineering and Electronics, 2021, 43(1): 99 - 111. (in Chinese)
- [6] ORDAZ-RIVAS E, RODRIGUEZ-LIÑAN A, TORRES-TREVIÑO L. Autonomous foraging with a pack of robots based on repulsion, attraction and influence[J]. Autonomous Robots, 2021, 45(6): 919 - 935.
- [7] WANG Y C, WANG Y F, SUN J W, et al. A stimulus-response-based allocation method for the circle packing problem with equilibrium constraints[J]. Physica A, 2019, 522(3): 232 - 247.
- [8] ZHAO W Q, MENG Q G, CHUNG P W H. A heuristic distributed task allocation method for multivehicle multitask problems and its application to search and rescue scenario[J]. IEEE Transactions on Cybernetics, 2016, 46(4): 902 - 915.
- [9] TURNER J, MENG Q G, SCHAEFER G, et al. Distributed task rescheduling with time constraints for the optimization of total task allocations in a multirobot system [J]. IEEE Transactions on Cybernetics, 2018, 48(9): 2583 - 2597.
- [10] TURNER J, MENG Q G, SCHAEFER G, et al. Fast consensus for fully distributed multi-agent task allocation[C]// Proceedings of the 33rd Annual ACM Symposium on Applied Computing, 2018: 832 - 839.
- [11] CHEN C H, CHOU F I, CHOU J H. Optimization of robotic task sequencing problems by crowding evolutionary algorithms[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2022, 52(11): 6870 - 6885.
- [12] KIM S, MOON I. Traveling salesman problem with a drone station [J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2019, 49(1): 42 - 52.
- [13] XIN B, WANG Y P, CHEN J. An efficient marginal-return-based constructive heuristic to solve the sensor-weapon-target assignment problem [J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2019, 49(12): 2536 - 2547.
- [14] CHEN W W, WANG D, LI K Q. Multi-user multi-task computation offloading in green mobile edge cloud computing[J]. IEEE Transactions on Services Computing, 2019, 12(5): 726 - 738.
- [15] MNIF S, ELKOSANTINI S, DARMOUL S, et al. An immune network based distributed architecture to control public bus transportation systems [J]. Swarm and Evolutionary Computation, 2019, 50: 100478.
- [16] MUDROVA L, HAWES N. Task scheduling for mobile robots using interval algebra[C]// Proceedings of IEEE International Conference on Robotics and Automation, 2015: 383 - 388.
- [17] TESTA A, RUCCO A, NOTARSTEFANO G. Distributed mixed-integer linear programming via cut generation and constraint exchange [J]. IEEE Transactions on Automatic Control, 2020, 65(4): 1456 - 1467.
- [18] CAMISA A, NOTARNICOLA I, NOTARSTEFANO G. Distributed primal decomposition for large-scale MILPs [J]. IEEE Transactions on Automatic Control, 2022, 67(1): 413 - 420.
- [19] 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.
- [20] BUCKMAN N, CHOI H L, HOW J P. Partial replanning for decentralized dynamic task allocation [C]// Proceedings of AIAA Scitech Forum, 2019.
- [21] GE X H, HAN Q L, DING L, et al. Dynamic event-triggered distributed coordination control and its applications: a survey of trends and techniques[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2020, 50(9): 3112 - 3125.
- [22] HAN Y, ZHANG K, LI H, et al. MAS-based distributed coordinated control and optimization in microgrid and microgrid clusters: a comprehensive overview [J]. IEEE Transactions on Power Electronics, 2018, 33(8): 6488 - 6508.
- [23] GARCIA E, CAO Y C, CASBEER D W. Periodic event-triggered synchronization of linear multi-agent systems with communication delays[J]. IEEE Transactions on Automatic Control, 2016, 62(1): 366 - 371.
- [24] OTTE M, KUHLMAN M J, SOFGE D. Auctions for multi-robot task allocation in communication limited environments[J]. Autonomous Robots, 2020, 44(3/4): 547 - 584.
- [25] ZHENG Y R, XIAO C S. Simulation models with correct statistical properties for Rayleigh fading channels[J]. IEEE Transactions on Communications, 2003, 51(6): 920 - 928.