Modeling and Simulation 建模与仿真, 2025, 14(5), 472-487
Published Online May 2025 in Hans. https://www.hanspub.org/journal/mos
https://doi.org/10.12677/mos.2025.145408

Hans汉斯

面向延迟容忍的时序约束多无人机任务分配方法

陈九澳[1]，何杏宇[1,2]

[1]上海理工大学光电信息与计算机工程学院，上海
[2]上海理工大学出版学院，上海

收稿日期：2025年4月19日；录用日期：2025年5月12日；发布日期：2025年5月19日

摘　要

现有的面向时序约束的多无人机任务分配方法大多假定任务执行是非延迟容忍的情况，这不仅降低了任务被执行的机会，而且极大地限制了多无人机的竞争与协作灵活度，从而影响任务完成率。为此，文章将提出一种面向延迟容忍的时序约束多无人机任务分配方法，并解决延迟容忍带来的任务执行效益时变性问题。具体而言，首先设计了一种动态时间窗机制，为无人机任务执行提供弹性空间，允许无人机在任务可执行初始时间之前到达进行悬停等待或在预计截止时间后超时执行任务，并基于此定义了随动态时间窗变化的任务执行收益。接着，构建强化学习模型以引导无人机进行任务选择，然后定义无人机与任务的匹配因子，设计根据匹配因子、悬停和超时情况解决任务选择冲突的共识机制。最后采用多智能体深度确定性策略梯度算法(MADDPG)对强化学习模型进行求解。实验结果表明，与现有方法相比，本文所提的方法在系统效益和任务完成率方面均表现出显著优势。

关键词

多无人机系统，任务分配，深度强化学习，时序约束，动态时间窗

A Delay-Tolerant and Temporal-Ordering-Constrained Task Allocation Method for Multi-UAV Systems

Jiu'ao Chen[1], Xingyu He[1,2]

[1]School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai
[2]School of Publishing, University of Shanghai for Science and Technology, Shanghai

Received: Apr. 19th, 2025; accepted: May 12th, 2025; published: May 19th, 2025

文章引用: 陈九澳, 何杏宇. 面向延迟容忍的时序约束多无人机任务分配方法[J]. 建模与仿真, 2025, 14(5): 472-487.
DOI: 10.12677/mos.2025.145408

Abstract

The majority of existing multi-UAV task allocation approaches under temporal-ordering-constrained are based on the assumption of non-delay-tolerant task execution, which not only reduces the task execution chances but also greatly limits the competition and collaboration flexibility among multiple UAVs, thereby affecting the task completion rate. To address this problem, this paper will propose a delay-tolerant and temporal-ordering-constrained task allocation method, then solve the time-varying problem of task execution benefit related to delay tolerance. In this method, firstly, a dynamic time window mechanism is designed to provide an elastic space for UAV task execution, which allows UAVs to hover and wait for a task before its initial executable time and execute the task after its expected deadline and the time-varying task execution benefit related to the dynamic execution window is also defined; Secondly, a reinforcement learning model is constructed to guide the task selection of UAVs, a matching factor between task and UAVs is defined, and a consensus mechanism is provided to solve task selection conflicts between UAVs based on the matching factor, the waiting and delay time of UAVs. Finally, the multi-agent deep deterministic policy gradient algorithm is used to solve the reinforcement learning model. Experimental results show that, compared with existing methods, the method proposed in this paper exhibits significant advantages in terms of system benefit and task completion rate.

Keywords

Multi-UAV Systems, Task Allocation, Deep Reinforcement Learning, Temporal-Ordering Constraint, Dynamic Time Window

1. 引言

随着多智能体系统在军事领域和民用领域的广泛应用[1]-[3]，多智能体之间的协作与协调问题日益成为研究热点。与单智能体相比，多智能体系统在容错性、灵活性和可扩展性方面展现出显著优势。然而，随着任务环境复杂性的不断提升，任务之间可能存在时序约束关系，这显著增加了多智能体任务分配的复杂性。为此，针对时序约束的多智能体任务分配方法成为一个研究热点。

针对时序约束的多智能体任务分配，现有研究主要可分为集中式与分布式两类[4]。集中式任务分配方法通过中央控制器收集全局信息并统一决策，将任务分配给各执行单元，适用于小规模、静态且对全局优化要求较高的场景[5]-[16]；然而，在大规模、动态且实时性要求高的环境中，其性能可能受限。相比之下，分布式任务分配采用多智能体自主决策与协作的方式，各智能体基于局部信息和通信结果独立做出决策，具有容错性强、可扩展性高和灵活性好等优势[17]-[21]。

然而，上述针对时序约束的多无人机任务分配方法大多假定任务执行是非延迟容忍的情况。一方面，要求非延迟容忍的任务必须在特定的时间内完成，否则就会被放弃；另一方面，任务分配过程中对无人机之间的协作也要符合严格的截止时间限制，这将进一步减少了任务被执行的机会。为了提高任务完成率，本文提出一种面向延迟容忍的时序约束多无人机任务分配方法，该方法通过设计任务执行的动态时间窗，允许无人机在任务可执行初始时间之前到达或在预计截止时间后执行任务，然后根据任务在截止时间后执行存在执行收益时变性问题，定义了随动态时间窗变化的无人机任务执行收益。在此基础上，

定义了无人机与目标当前可执行子任务的匹配因子计算方法，并结合匹配因子和任务单位时间收益构建了强化学习模型，以引导无人机进行任务选择，最后本文还设计了冲突共识方法解决多无人机任务决策的冲突问题。

2. 相关工作

2.1. 集中式时序约束任务分配方法

集中式时序约束任务分配方法主要采用群智能优化算法和强化学习算法。例如文献[5]提出了一种基于改进遗传算法的元启发式算法，设计了针对时序约束任务分配的编码方式和遗传算子。文献[6]将基于有向图的优化方法与改进的两部分狼群搜索算法相结合，提出了一种新的任务分配问题求解框架，使用基于图的死锁检测与解决方法保证时序约束任务分配中的死锁检测与解决方案的有效性。文献[7]提出了一种自适应遗传算法，通过设计多类型基因染色体编码方案，生成满足无人机异构性和任务耦合约束的可行染色体，同时设计了相应的交叉和变异算子增强算法的搜索能力。文献[8]将异构无人机协同任务分配问题转化为约束多目标优化问题，提出了一种新的信息素更新机制和四个新定义启发式信息的多目标蚁群优化算法以提高搜索效率和解决方案的多样性。文献[9]考虑弹药库存的跨区域联合作战多无人机协同任务分配开发了一种改进的遗传算法，采用了一种新的染色体编码格式确定目标，针对算法中的交叉和变异操作，设计了一种高效的基于逻辑的时序约束的解锁机制。文献[10]引入了异构无人机载荷约束和任务费用约束，分析了无人机及其约束，提出了一种改进的和声算法。文献[11]提出了一种改进的多策略人工大猩猩部队优化器算法，采用 Halton 序列保证种群的多样性，采用信息共享的搜索策略跳出局部最优解，有效地解决无人机集群中的任务分配。文献[12]针对负载不同的异构多无人机的任务分配问题建立了多无人机任务分配模型，通过自适应机制将混沌优化引入到猴群算法中，提出了一种改进的混沌自适应猴群算法避免解的局部最优。文献[13]采用目标内任务调整策略和评估任务调整策略，提出基于强化搜索策略的自适应大邻域搜索方法，解决了来自不同基站的异构无人机时序约束任务的分配问题。文献[14]-[16]将图神经网络与强化学习相结合，解决了灵活作业车间调度问题的具有依赖约束的多任务分配。

2.2. 分布式时序约束任务分配方法

分布式时序约束任务分配方法主要采用合同网方法和拍卖算法。文献[17]针对多目标搜索问题，设计了一种基于狼侦察行为的无人机群协同搜索算法。其次通过分析狼群的灵活分工行为，提出了一种无人机集群协同攻击目标的分布式自组织任务分配算法。文献[18]提出了一种分布式编队结构下基于时间窗机制的多无人机动态任务规划算法，通过引入特定的粒子编码方法和竞争协同进化种群更新策略，对粒子群优化算法进行改进，提高了模型的求解速度和精度，实现了多无人机协同时序约束任务的最优分配策略。文献[19]提出了一种基于协商机制和遗传算法的异构无人机分布式协同优化方法，基于启发式规则选择优先完成的侦察任务，每个无人机基于感知到的环境信息和与相邻节点通信获得的历史状态信息，利用遗传算法生成局部任务分配和时间协调计划，并与相邻节点协商解决冲突，侦察节点发现敌方目标后，局部优化打击方案，请求相关节点协调执行。文献[20]考虑了时序约束任务的时间窗口，提出了一种改进的基于共识的分组算法来解决多无人机协同任务分配问题。文献[21]提出了基于 Q-Learning 的改进蚁群优化算法，为每架无人机生成一个任务序列。

本文为了解决具有延迟容忍的时序约束多无人机任务分配问题，将设计动态时间窗为无人机任务执行构建悬停等待和超时执行的弹性空间，然后定义了无人机与任务的匹配因子计算方法，建立强化学习模型，接着设计冲突共识算法构建无人机任务决策的奖励。最后将多智能体深度确定性策略梯度算法和

共识算法结合进行问题求解。

3. 系统模型

本文定义的系统由三类无人机和多个目标构成。每个目标包含三个具有时序约束的子任务，依次为侦察、打击和评估子任务。无人机集群从基地启程，依次执行完成各目标的子任务后，最终返回基地。

无人机：多无人机集群定义为$U=\{U_1,\cdots,U_i,\cdots,U_N\}$，这里 N 为无人机总数，$N=N_1+N_2+N_3$，N_1 表示侦察类无人机的数量，N_2 表示打击类无人机的数量，N_3 表示综合类无人机的数量。其中，无人机 U_i 的属性特征可表示为$U_i=\{type_i,ability_i,pos_i,v_i,hc_i,fc_i\}$。$type_i$ 表示无人机 U_i 的类型，$type_i=1$ 为侦察类无人机，$type_i=2$ 为打击类无人机，$type_i=3$ 为综合类无人机。$ability_i=\{scout_i,attack_i\}$ 表示无人机 U_i 的能力，其中 $scout_i$ 为监察能力，$attack_i$ 为打击能力。pos_i 表示无人机 U_i 的位置坐标，$pos_i=\{x_i,y_i,z_i\}$。v_i 表示无人机 U_i 的移动速度。hc_i 表示无人机 U_i 单位时间内的悬停能耗，fc_i 表示无人机 U_i 单位时间内的飞行能耗。

任务：在目标任务环境中，具有 M 个动态地面目标，目标集合记为$Target=\{Tar_1,\cdots,Tar_j,\cdots,Tar_M\}$。$Tar_j$ 为第 j 个目标，目标 Tar_j 包含具有时序约束的三个子任务 Tar_j^1、Tar_j^2、Tar_j^3。用 Tar_j^k 表示第 j 个目标的当前可执行任务，每当分配一个子任务时，k 自动递增。$k=1$ 表示当前可执行任务为侦察任务，$k=2$ 表示当前可执行任务为打击任务，$k=3$ 表示当前可执行任务为评估任务。任务 Tar_j^k 的属性定义为 $Tar_j^k=\{loc_j,value_j^k,init_j^k,term_j^k,span_j^k,as_j^k,rate\}$，$loc_j$ 表示目标 Tar_j 的位置，$loc_j=\{x_j,y_j,z_j\}$。$value_j^k$ 为任务 Tar_j^k 的报酬。$init_j^k$ 为任务 Tar_j^k 的可执行初始时间，也是其前序任务 Tar_j^{k-1} 的完成时间，$term_j^k$ 为任务 Tar_j^k 的预计截止时间。as_j^k 为任务 Tar_j^k 的实际执行面积，其初始值 $as_j^{k'}$ 等于前序任务的执行面积 as_j^{k-1}。$span_j^k$ 为任务 Tar_j^k 信息稳定的时间跨度，如果在这个时间跨度内执行任务 Tar_j^k，则执行面积固定，否则执行面积的半径按照速率(rate)增加。$span_j^k$ 计算方法如公式(1)所示。

$$span_j^k=term_j^k-init_j^k \tag{1}$$

4. 动态时间窗与任务收益计算

4.1. 动态时间窗

为了提高任务完成率和无人机任务协作的灵活度，本文为任务执行定义了动态时间窗口。如图 1 所示，Tar_j^k 的动态时间窗用 D_j^k 表示，$D_j^k=\{start_j^k,end_j^k\}$。时间窗 D_j^k 虽然赋予了无人机任务执行的弹性空间，但为了减少无人机过多的无用悬停时间和过长的超时执行，所以为时间窗赋予了阈值 tp 的限制。因此，$start_j^k$ 为时间窗 D_j^k 的开始时间，可以在任务 Tar_j^k 的可执行初始时间 $init_j^k$（Tar_j^{k-1} 的实际完成时间)之前，若无人机提前到达，到达时间在阈值之后，则等于无人机到达任务 Tar_j^k 的时间，否则为 $init_j^k-tp$。end_j^k 为时间窗 D_j^k 的结束时间，可以在任务 Tar_j^k 的预计截止时间 $term_j^k$ 之后，当超时执行时，若任务 Tar_j^k 的实际执行完成时间在阈值内，则为任务 Tar_j^k 的实际执行完成时，否则 D_j^k 的结束时间为 $term_j^k+tp$。

如图 1(a)所示，时间窗 D_j^k 给予无人机任务执行悬停等待的弹性空间，允许无人机 U_i 悬停等待任务 Tar_j^k 前序任务 Tar_j^{k-1} 完成之后才执行任务。若无人机 U_i 在任务 Tar_j^k 的可执行初始时间 $init_j^k$ 之前到达，此时无人机的悬停等待时间 $hover_{ij}^k$ 为无人机 U_i 到达目标 Tar_j 的到达时间 $arrive_{ij}$（时间窗 D_j^k 的开始时间 $start_j^k$)和任务 Tar_j^k 可执行时间 $init_j^k$ 的差值。

$$hover_{ij}^k=\begin{cases}init_j^k-arrive_{ij} & \text{if } arrive_{ij}<init_j^k\\ 0 & \text{else}\end{cases} \tag{2}$$

无人机U_i的到达时间$arrive_{ij}$为无人机U_i执行前序任务Tar_g^k的完成时间$done_{ig}^k$和从前序任务Tar_g^k的目标位置loc_g飞行到目标Tar_j位置loc_j的时间fly_{gj}^i之和。

$$arrive_{ij} = done_{ig}^k + fly_{gj}^i \tag{3}$$

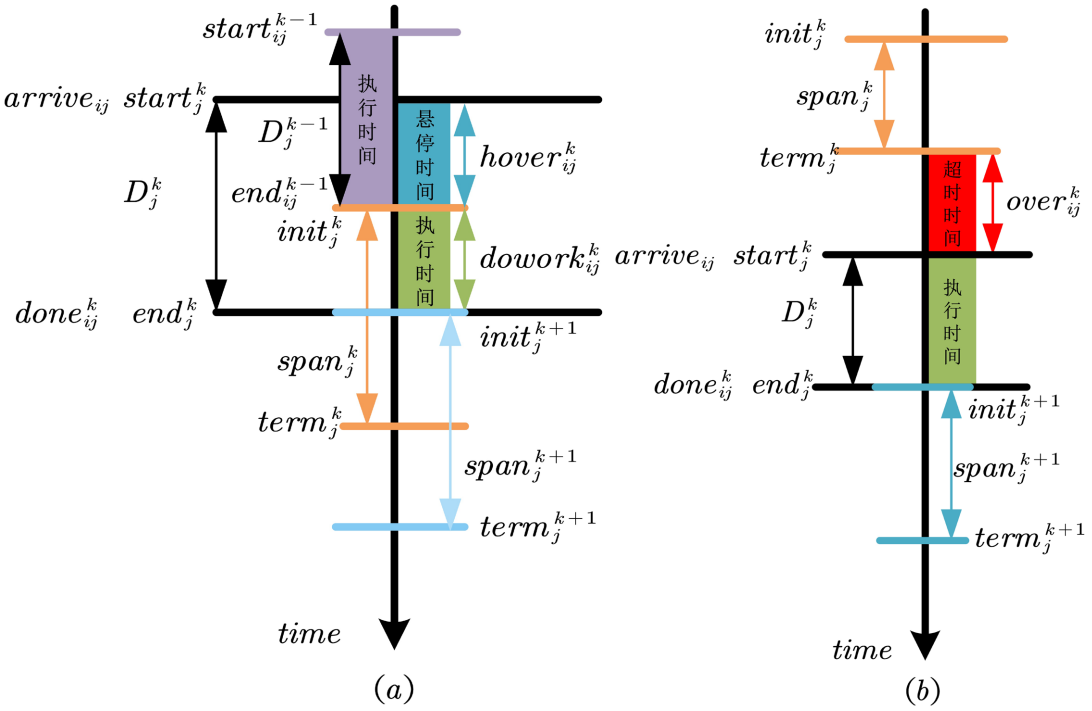

Figure 1. Dynamic time window description diagram
图 1. 动态时间窗描述图

如图 1(b)所示，时间窗D_j^k给予无人机任务超时执行的弹性空间。若无人机U_i在任务Tar_j^k的预计截止时间$term_j^k$之后到达，此时无人机U_i的超时时间$over_{ij}^k$为任务Tar_j^k的到达$arrive_{ij}$与任务Tar_j^k的预计截止时间$term_j^k$的差值。

$$over_{ij}^k = \begin{cases} term_j^k - arrive_{ij} & \text{if } arrive_{ij} > term_j^k \\ 0 & \text{else} \end{cases}$$

此时虽然时间窗D_j^k允许无人机超时到达，但是由于无人机U_i的到达时间超过了Tar_j^k的预计截止时间$term_j^k$，任务Tar_j^k的实际执行面积as_j^k会随着超时时间$over_{ij}^k$持续增加，计算如公式(5)所示。

$$as_j^k = \left(\sqrt{as_j^{k'}/\pi} + over_{ij}^k * rate\right)^2 \tag{5}$$

其中$as_j^{k'}$为Tar_j^k的初始执行面积。因此无人机U_i执行任务Tar_j^k的时间开销$dowork_{ij}^k$的计算如公式(6)所示。

$$dowork_{ij}^k = \begin{cases} \dfrac{as_j^k}{scout_i} & \text{if } type_i \in \{1,3\} \\ \dfrac{as_j^k}{attack_i} & \text{if } type_i \in \{2,3\} \end{cases} \tag{6}$$

由此可见，Tar_j^k的实际执行面积as_j^k越大，无人机U_i执行任务Tar_j^k的时间开销$dowork_{ij}^k$也越大。

无人机U_i执行任务Tar_j^k的完成时间$done_{ij}^k$与无人机U_i的到达时间$arrive_{ij}$、悬停时间$hover_{ij}^k$及执行Tar_j^k的时间开销$dowork_{ij}^k$有关，计算如公式(7)所示。

$$done_{ij}^k = arrive_{ij} + hover_{ij}^k + dowork_{ij}^k \tag{7}$$

由于后续任务Tar_j^{k+1}的信息根据Tar_j^k的完成情况动态生成，任务Tar_j^{k+1}的开始时间$init_j^{k+1}$为无人机U_i执行任务Tar_j^k的任务完成时间。

$$init_j^{k+1} = done_{ij}^k \tag{8}$$

任务Tar_j^{k+1}的预计截止时间$term_j^{k+1}$会随着任务超时导致的面积变大而变得紧迫，计算如公式(9)所示，其中ξ为衰减速度参数。

$$term_j^{k+1} = init_j^{k+1} + span_j^k * \mathrm{e}^{-\xi*\left(as_j^k - as_j^{k'}\right)} \tag{9}$$

此时任务Tar_j^{k+1}的初始执行面积$as_j^{k+1'}$与前序任务Tar_j^k的实际执行面积相等，$as_j^{k+1'} = as_j^k$。

4.2. 无人机任务执行收益

报酬：无人机U_i选择目标点Tar_j执行任务Tar_j^k的任务报酬$value_{ij}^k$如公式(10)所示。

$$value_{ij}^k = value_j^k * e^{\beta\left(magin_{ij}^k\right)} \tag{10}$$

其中$magin_{ij}^k$为无人机U_i执行任务Tar_j^k的完成时间$done_{ij}^k$与任务Tar_j^k的预计截止时间$term_j^k$的差值，若$magin_{ij}^k > 0$表示无人机在任务预计截止时间之前完成任务，执行任务的报酬呈指数递增，若$magin_{ij}^k \le 0$表示无人机在任务预计截止时间之后完成任务，执行任务的报酬呈指数递减，β为控制参数。

$$magin_{ij}^k = term_j^k - done_{ij}^k \tag{11}$$

开销成本：无人机U_i执行任务Tar_j^k的开销成本为飞行开销$flight_{ij}$与悬停开销$wait_{ij}^k$。飞行开销$fight_{ij}$为飞行能耗fc_i与执行完其当前任务Tar_g^k飞往任务Tar_j^k时间fly_{gj}^i的乘积。

$$flight_{ij} = fc_i * fly_{gj}^i \tag{12}$$

悬停开销$wait_{ij}^k$为悬停能耗hc_i与悬停时间$hover_{ij}^k$的乘积。

$$wait_{ij}^k = hc_i * hover_{ij}^k \tag{13}$$

收益：无人机U_i执行任务Tar_j^k的总收益$profit_{ij}^k$的计算为报酬与开销的差值，如公式(14)所示。其中δ_1、δ_2、δ_3为权重参数。

$$profit_{ij}^k = \delta_1 value_{ij}^k - \delta_2 flight_{ij} - \delta_3 wait_{ij}^k \tag{14}$$

5. 基于共识强化学习模型的时序约束任务分配

在本小节中，本文首先定义了无人机与目标当前可执行子任务的匹配因子计算方法，然后建立了强化学习模型，利用匹配因子和效益引导无人机进行任务选择，最后设计了冲突共识方法，解决无人机任务选择的冲突问题。

5.1. 无人机与目标可执行子任务的匹配因子

无人机U_i与目标Tar_j当前可执行任务Tar_j^k的匹配因子$match\left(U_i, Tar_j^k\right)$的计算如公式(15)所示：

$$match\left(U_i, Tar_j^k\right) = \begin{cases} 0 & \text{if } k = 4 \text{ or } type_i \ne k \\ 1 & \text{if } type_i = k \text{ or } type_i = 3 \end{cases} \tag{15}$$

匹配因子为 0 的情况有两种，一种是当任务 Tar_j^k 的完成状态 $k=4$ 时，表示目标 Tar_j 的所有任务均已被完成，无人机 U_i 无需飞往目标点 Tar_j 执行任务，那么此次无人机 U_i 的目标任务选择是失败的，$match\left(U_i,Tar_j^k\right)=0$；另一种是无人机选择与自己类型不匹配的子任务导致无人机无法执行子任务，此时 $match\left(U_i,Tar_j^k\right)=0$。匹配因子为 1 的情况也有两种，无人机 U_i 的类型 $type_i$ 与可执行子任务 Tar_j^k 的状态 k 相等或者无人机的类型为综合类无人机时，那么无人机 U_i 与任务 Tar_j^k 完全匹配，$match\left(U_i,Tar_j^k\right)=1$。

5.2. 基于强化学习的时序任务选择引导

在本文的无人机集群协同任务分配场景中，每一个无人机被视为强化学习中的智能体，每个智能体根据自身的观测信息进行任务选择决策，并从系统环境中得到收益。于是本文将无人机集群协同任务分配场景建模为部分可观测马尔可夫博弈。部分可观测马尔可夫博弈使用元组(*N*, *S*, *O*, *A*, *R*, *P*)表示。其中 *N* 为智能体集合，*S* 代表所有智能体可能的状态空间，*O* 为每个智能体的观测空间，*A* 为所有智能体的动作集合，*R* 为所有智能体奖励函数的集合，*P* 是环境的状态转移概率。每个智能体的目标是最大化其期望累积奖励。具体的强化学习模型的基本定义如下：

状态空间：本文用 S_t^i 表示无人机 U_i 在时间步 t 的状态，保存自己当前的观测信息，观测信息为自身属性信息，观测环境下目标信息和其他无人机信息。$S_t^i=\left\{self_i\left(t\right),target\left(t\right),others\left(t\right)\right\}$，其中 $self_i\left(t\right)=\left\{type_i,pos_i\left(t\right),ability_i,ct_i\left(t\right)\right\}$，$type_i$ 为无人机 U_i 的类型，$pos_i\left(t\right)$ 为无人机 U_i 当前所处的位置，$ability_i$ 为无人机 U_i 的能力，$ct_i\left(t\right)$ 为无人机 U_i 的最早可用时间(完成前一个子任务的时间)。$target\left(t\right)=\left\{tar_1\left(t\right),\cdots,tar_j\left(t\right),\cdots,tar_M\left(t\right)\right\}$，表示所有目标当前可执行任务的信息，其中，$tar_j\left(t\right)=\left\{loc_j,as_j\left(t\right),init_j\left(t\right),term_j\left(t\right),span_j\left(t\right),D_j\left(t\right),k\left(t\right),value_j\right\}$。$loc_j$ 为目标 Tar_j 的位置，$as_j\left(t\right)$ 为目标 Tar_j 的当前执行面积需求，$init_j\left(t\right)$ 为目标 Tar_j 当前可执行任务的开始时间，$term_j\left(t\right)$ 为目标 Tar_j 可执行任务的预计截止时间，$span_j\left(t\right)$ 为目标 Tar_j 的当前子任务时间跨度大小，$k\left(t\right)$ 表示目标 Tar_j 的任务完成状态，$k\left(t\right)=1$ 表示目标 Tar_j 需要执行侦察任务，$k\left(t\right)=2$ 表示目标 Tar_j 需要执行打击任务，$k\left(t\right)=3$ 表示目标 Tar_j 需要执行评估任务，$k\left(t\right)=4$ 表示目标 Tar_j 所有任务均被完成，$value_j$ 为目标 Tar_j 的当前子任务报酬。$others\left(t\right)=\left\{info_{u_1}\left(t\right),\cdots,info_{u_e}\left(t\right),\cdots,info_{u_{N-1}}\left(t\right)\right\}$。$info_{u_e}\left(t\right)=\left\{type_e,pos_e\left(t\right),ability_e,ct_e\left(t\right)\right\}$ 表示其他无人机 U_e 的类型、位置、能力以及最早使用时间。

动作空间：无人机的动作为选择目标点进行任务执行或者停留在原地等待，则动作空间为所有目标点和停留等待 $A=\left\{Tar_1,\cdots,Tar_j,\cdots,Tar_M,Stay\right\}$，例如，无人机 U_i 根据当前的状态 S_t^i 执行动作 $a_t=Tar_j/Stay$，表示无人机在当前状态准备选择目标 Tar_j 执行目标相应子任务或者停留在原地。

奖励计算：无人机的奖励由匹配因子和冲突共识方法得到。无人机的奖励分为两部分，若无人机执行 *Stay* 动作，则根据当前无人机与目标当前所有可执行子任务的总匹配度进行奖励设计。总匹配度的计算公式为：

$$Amatch\left(U_i\right)=\frac{\sum_{j=1}^{M}match\left(U_i,Tar_j^k\right)}{\left|M\right|} \tag{16}$$

如果总匹配度为 0 说明当前都是无法执行的任务，可以停留等待，否则给予惩罚。此时无人机执行 *Stay* 动作的奖励公式为(17)，其中 *C* 为一个时间步下的常数惩罚。

$$r_{u_i}\left(Stay\right)=\begin{cases}0 \text{ if } Amatch\left(U_i\right)=0\\ -C \text{ else}\end{cases} \tag{17}$$

若无人机选择的是目标子任务，此时无人机 U_i 此次动作选择的奖励计算如公式(18)。当无人机 U_i 与

所选目标的当前子任务 Tar_j^k 的匹配因子 $match\left(U_i,Tar_j^k\right)$ 为 0，无人机 U_i 无法执行任务，给予常数 R 的惩罚。当匹配因子为 1 时，此时分为两种情况，若竞争成功则为执行任务的单位时间效益，若竞争失败奖励为 0。单位时间效益的计算如公式(19)所示。

$$r_{u_i}\left(Tar_j^k\right)=\begin{cases}-R & \text{if } match\left(U_i,Tar_j^k\right)=0\\ putp_{ij}^k & \text{if } match\left(U_i,Tar_j^k\right)=1 \text{ and win}\\ 0 & \text{if } match\left(U_i,Tar_j^k\right)=1 \text{ and fail}\end{cases} \tag{18}$$

$$putp_{ij}^k=\frac{profit_{ij}^k}{\left(fly_{gj}^i+hover_{ij}^k+dowork_{ij}^k\right)} \tag{19}$$

5.3. 无人机任务选择的冲突共识

当无人机在每一个时间步进行动作选择后，无人机可能选择停留等待或者选择执行目标相应子任务，若无人机选择停留等待，此时无人机之间不构成任务冲突，若此时无人机选择执行目标相应子任务，此时多个无人机可能选择同一个目标子任务，无人机之间存在冲突需要抉择出最优无人机执行目标子任务。冲突共识的流程如图 2 所示。

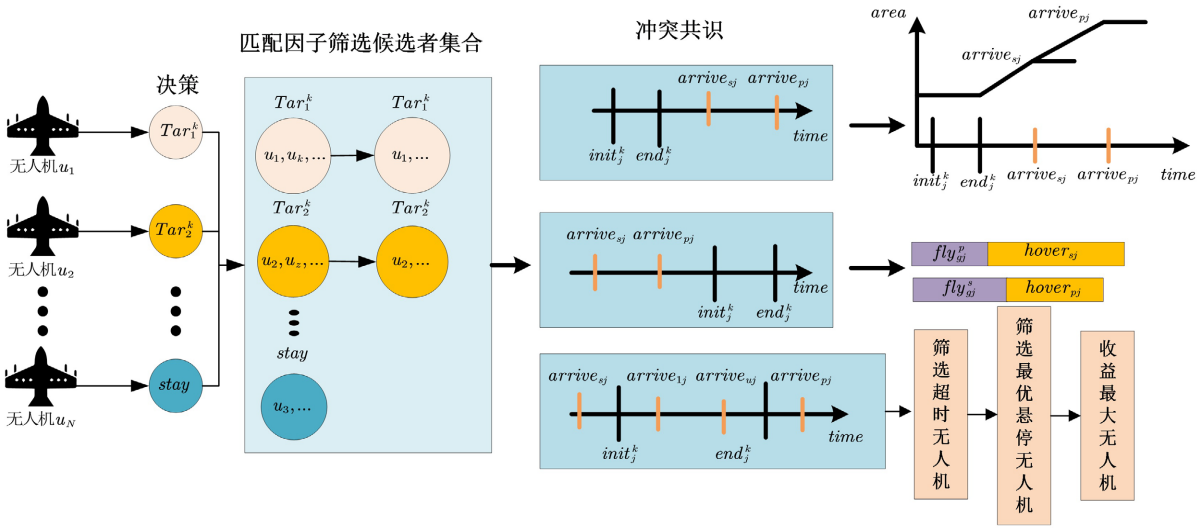

Figure 2. Conflict consensus process
图 2. 冲突共识流程

首先，本文为每一个目标当前选择的子任务构建执行候选者集合，目标 Tar_j 当前可执行任务 Tar_j^k 的执行候选者集合定义为 C_j^k，$C_j^k=\left\{U_s,U_p\right\}$，表示无人机 U_s 和无人机 U_p 选择了同一个目标子任务 Tar_j^k。接着，本文根据候选无人机与目标子任务的匹配因子对目标子任务的执行者候选集合进行优化，将匹配因子为 0 的执行候选者从集合中排除。若任务候选者集合中存在匹配因子为 0，此时存在两种情况，一种是当前目标所有子任务已全部完成，候选者集合中的无人机无法执行任务，另一种是候选者集合中无人机类型与任务类型不匹配导致无法执行当前任务。此时当前子任务执行候选者集合中只剩下匹配因子为 1 的无人机解决冲突。由于每个子任务的信息不同，无人机执行子任务的情况不同，可能悬停等待或者超时执行。此时需根据不同情况进行冲突共识。

1) 若所有无人机都超时到达，由于超时执行导致任务面积增加，使得后续子任务的执行增加，则任

务的开始时间越早越好，所以根据无人机的到达时间进行排序，最早到达的无人机竞争成功。

2) 若所有无人机都提前到达，此时都需要悬停等待，此时无人机执行任务的开始时间是一样的，则计算无人机的飞行时间和悬停时间，总时间越短的无人机竞争成功。

3) 若无人机的三种情况都有，首先将超时执行的无人机筛选掉，然后根据步骤 2 筛选出最优的悬停无人机与所有在时间跨度内到达的无人机进行竞争，最后根据公式(14)计算无人机任务执行收益，收益最大的无人机竞争成功。

5.4. 强化学习模型求解

本文采用多智能体深度确定性策略梯度(MADDPG)算法对强化学习模型进行求解。对于每一个智能体实现一个 DDPG 算法，每一个智能体都有一个策略(actor)网络和一个所有智能体共享的中心化的价值(critic)网络。采用中心化训练–去中心化执行方法，策略网络根据智能体的观测状态 $S_i(t)$，输出能够使得智能体获得最大预期收益的动作即去中心化的执行。价值网络则仅在进行中心化训练阶段使用，用来对智能体策略网络输出的动作进行指导，并反馈给 actor，实现 actor 的调优。由于 MADDPG 算法是用来解决连续动作空间的强化学习任务的，而在本文中智能体的动作空间是离散的，在离散动作空间中通常采用的 argmax 函数不满足多元函数连续且具有偏导数的条件，于是本文采用 Gumbel-Softmax 的方法来得到离散分布动作的近似采样。

采用 Gumbel-Softmax 方法生成离散动作的概率分布向量的算法流程如下所示：

1) 通过神经网络输出的 n 维向量 v，生成 n 个服从均匀分布 $Uniform(0,1)$ 的独立样本 $\epsilon_1,\epsilon_2,\cdots,\epsilon_n$；

2) 引入 Gumbel 噪声 $g_i=-\log\left(-\log(\epsilon)\right),\epsilon\sim Uniform(0,1)$；

3) 通过 Softmax 函数得到各动作的选择概率，如公式(20)所示。

其中 τ 为温度参数，控制着 Softmax 函数的 soft 的程度。τ 越大，生成的分布越趋向于均匀分布，τ 越小，生成的分布越趋向于 $onehot\left(argmax_i\left(\log\epsilon_i+g_i\right)\right)$ 的结果。在本文中，采用线性退火方式，如公式(21)所示，在训练初期使用较高的温度参数以增强探索，随着训练的进行逐渐降低温度参数以增强利用。

$$y_i=\frac{\exp\left(\frac{\log\epsilon_i+g_i}{\tau}\right)}{\sum_{j=1}^{n}\exp\left(\frac{\log\epsilon_i+g_i}{\tau}\right)},\quad i=1,\cdots,n \tag{20}$$

$$\tau(t)=\tau_{init}-\frac{t}{T}\left(\tau_{init}-\tau_{final}\right) \tag{21}$$

基于 MADDPG 的强化学习模型求解算法如表 1 所示。

Table 1. Reinforcement learning solution process
表 1. 强化学习求解过程

输入：初始化各个智能体策略网络参数 $\theta=(\theta_1,\theta_2,\cdots,\theta_N)$ 和价值网络参数 $\phi=(\phi_1,\phi_2,\cdots,\phi_N)$
输出：训练后的最优参数 θ^*，ϕ^*
For 序列 e = 1→E do
重置环境，初始化一个随机过程 χ，用于动作探索
获取所有智能体的初始观测 $o=(o_1,o_2,\cdots,o_N)$
For t = 1→T do:

续表

对于每个智能体 i，根据当前策略网络选择动作 $a_i = \mu_{\theta_i}(o_i) + \chi_t$

执行联合动作 $a = (a_1, a_2, \cdots, a_N)$，得到奖励值 $r = (r_1, r_2, \cdots, r_N)$ 和新的观测 $o' = (o'_1, o'_2, \cdots, o'_N)$

将各智能体生成的数据元组 (o, a, r, o') 存储到经验回放池 $\mathcal{D}$ 中

从 $\mathcal{D}$ 中采样出一批次的数据样本 $\mathcal{B}$

For agent i = 1 to N:

中心化训练 critic 网络，计算智能体的价值网络梯度值：

$$\mathcal{L}(w_i) = \mathbb{E}_{o,a,r,o'}\left[\mathcal{Q}_i^{\mu}(o, a_1, a_2, \cdots, a_N) - \left(r_i + \gamma \mathcal{Q}_i^{\mu'}(o', a'_1, a'_2, \cdots, a'_N)\Big|_{a'_j = \mu'_j(o_j)}\right)^2\right]$$

训练自身 actor 网络，计算智能体的策略网络梯度值：

$$\nabla_{\theta_i} J(\mu_i) = \mathbb{E}_{o\sim\mathcal{D}}\left[\nabla_{\theta_i}\mu_i(o_i)\nabla_{a_i}\mathcal{Q}_i^{\mu}(o, a_1, a_2, \cdots, a_N)\Big|_{a_i=\mu_i(o_i)}\right]$$

For agent i = 1 to N:

更新各智能体的 actor 网络参数：$\theta_i \leftarrow (1-\eta)\theta_i + \eta\mathrm{Adam}(\mathrm{d}\theta_i)$

更新各智能体的 critic 网络参数：$\phi_i \leftarrow (1-\eta)\phi_i + \eta\mathrm{Adam}(\mathrm{d}\phi_i)$

End for

End for

6. 实验模拟与分析

6.1. 仿真环境与参数设置

为了验证本文算法的性能，本文对无人机集群时序任务分配场景进行仿真实验，系统由一个无人机基地、多台无人机以及多个目标点组成。首先，我们对算法的收敛性进行了分析以评估其稳定性；随后，为了赋予时间窗合适的阈值，对不同阈值下总单位时间效益和任务完成率进行了分析，最后通过与现有方法在总单位时间效益和任务完成率两方面的对比，进一步验证了本文算法的有效性。本文实验采用 python 语言和 pytorch 深度学习库实现，实验所用 CPU 为 Inter-i5-13490F，显卡为 NVIDIA GeForce TRX 4060 Ti。实验中的仿真环境参数见表 2。

Table 2. Experimental parameters
表 2. 实验参数

参数	值
目标任务范围 zone	10 km × 10 km
(综合，打击，侦察)无人机速度 v	[60 m/s, 70 m/s, 80 m/s]
飞行能耗 fc	0.2/s
悬停能耗 hc	0.1/s
侦察、打击能力(scout, attack)	[4 m^2/s, 6 m^2/s]
任务价值 value	200
任务初始面积 as	24 m^2
面积增长率 rate	0.1 m/s
时间跨度 span	100 s
时间窗阈值 tp	20 s

此外，需要为训练定义各种超参数，超参数见表 3。

Table 3. Hyper-parameters
表 3. 超参数

参数	值
学习率 α	0.002
回放经验池 replay_buffer	10000
批次大小 batch_size	64
折扣因子 γ	0.95
软更新率 tau	0.01
最初温度 *init_temp*	1
最终温度 *final_temp*	0.1

6.2. 算法收敛性能

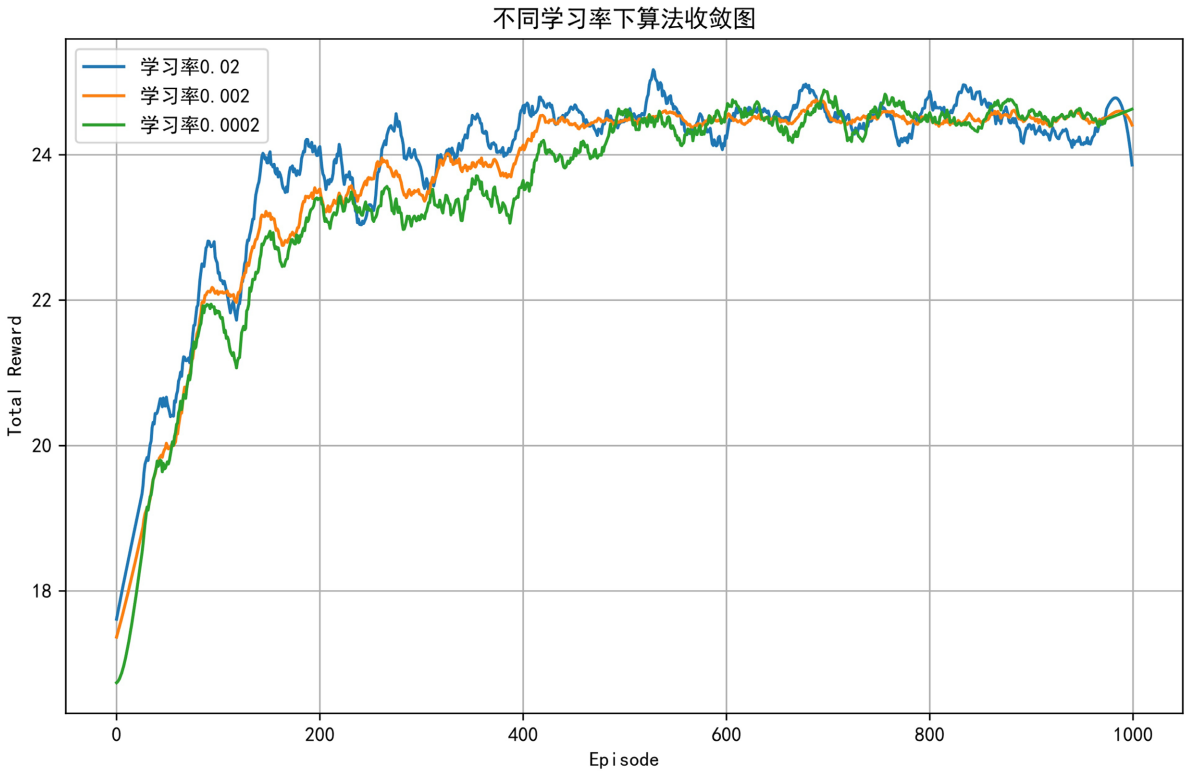

Figure 3. Convergence graphs at different learning rates
图 3. 不同学习率下的收敛图

为了评估超参数对所提算法性能的影响，采用不同的学习率验证对本文算法收敛性和稳定性的影响。在实验中，本文将学习率分别设定为 $2e^{-2}, 2e^{-3}, 2e^{-4}$，其中无人机的数量为 3 (1 架侦察类无人机、1 架打击类无人机、1 架综合类无人机)，目标的数量为 5。

不同学习率对算法收敛性的影响如图 3 所示，随着训练轮数的增加，算法趋于收敛，学习率太小时，算法需要更多的训练回合来达到收敛状态；当学习率过大时，虽然最后趋于收敛但波动很大。在本算法中学习率为 $2e^{-3}$ 时，算法收敛效果较好。

在学习率 $2e^{-3}$ 的基础上，本文以 18 个目标为基础，针对 6 架无人机、9 架无人机和 12 架无人机验证算法的收敛性。如图 4 所示，随着训练轮数的增加，奖励值逐渐增加最后趋于稳定水平，这反映模型已基本收敛。同时奖励值随着无人机数量的增加，这是因为无人机的数量越多，单位时间内任务的完成数量越多，使得后续子任务可以在时间跨度内有效完成，减少了任务完成时间，从而使得奖励值增加。

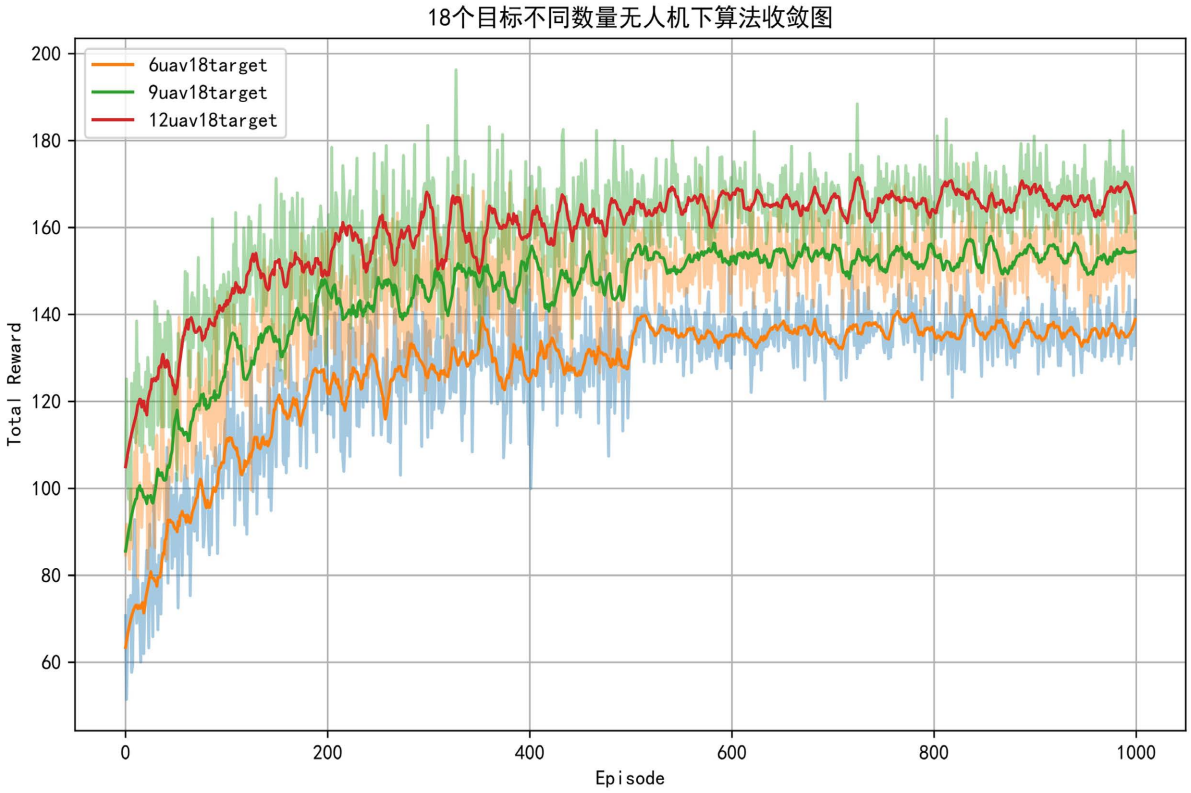

Figure 4. Convergence graph under different numbers of UAVs
图 4. 不同无人机数量下收敛图

6.3. 性能比较

为了验证本文算法的性能，本文以 12 架无人机(4 架侦察类无人机、4 架打击类无人机、4 架综合类无人机)为基准，针对不同数量的目标，以总单位时间效益和任务完成率为性能指标进行评估。

- 单位时间效益为无人机集群执行子任务的总效益与执行完最后目标子任务的完成时间。

$$\text{总单位时间效益} = \frac{\text{执行的子任务总效益}}{\text{最后子任务完成时间}} \tag{22}$$

- 任务完成率为无人机集群成功执行的子任务数量与所有目标子任务数量的比值。

$$\text{任务完成率} = \frac{\text{成功执行的子任务数量}}{\text{总子任务数量}} \tag{23}$$

为了给动态时间窗赋予合适阈值 *tp*，本文首先分析了阈值 *tp* 对系统性能的影响。

图 5、图 6 分别展示了在不同动态时间窗阈值下总单位时间效益和任务完成率的对比，由图可知阈值在 20 s 下，总单位时间效益和任务完成率相对于阈值在 15 s 和阈值在 25 s 高。这是由于阈值在 15 s 的情况下，无人机任务执行的弹性时间过短，导致时序约束任务的完成时间整体增加，任务完成率和总单位时间效益下降。阈值在 25 s 下，虽然提供了较大的任务执行弹性空间，但导致无人机过多飞行与悬停消耗，使得时序约束任务的任务完成率和总单位时间效益下降。阈值在 20 s 的情况下，平衡了悬停等待的弹性空间和超时执行的弹性空间，在任务完成率和总单位时间效益下相对较优。

为了体现实验结果的客观性和准确性，本文以文献[13]和文献[21]中的算法进行比较。对比算法具体如下：

- RSALNS [13]：该算法以模拟退火算法为搜索框架，采用任务调整策略和评估任务调整策略实现时序约束任务的分配。
- Q-learning + ACO [21]：将 Q-learning 和蚁群算法相结合，使用 Q-learning 来加速 ACO 算法的收敛速度并扩展搜索能力，实现时序约束任务的分配。

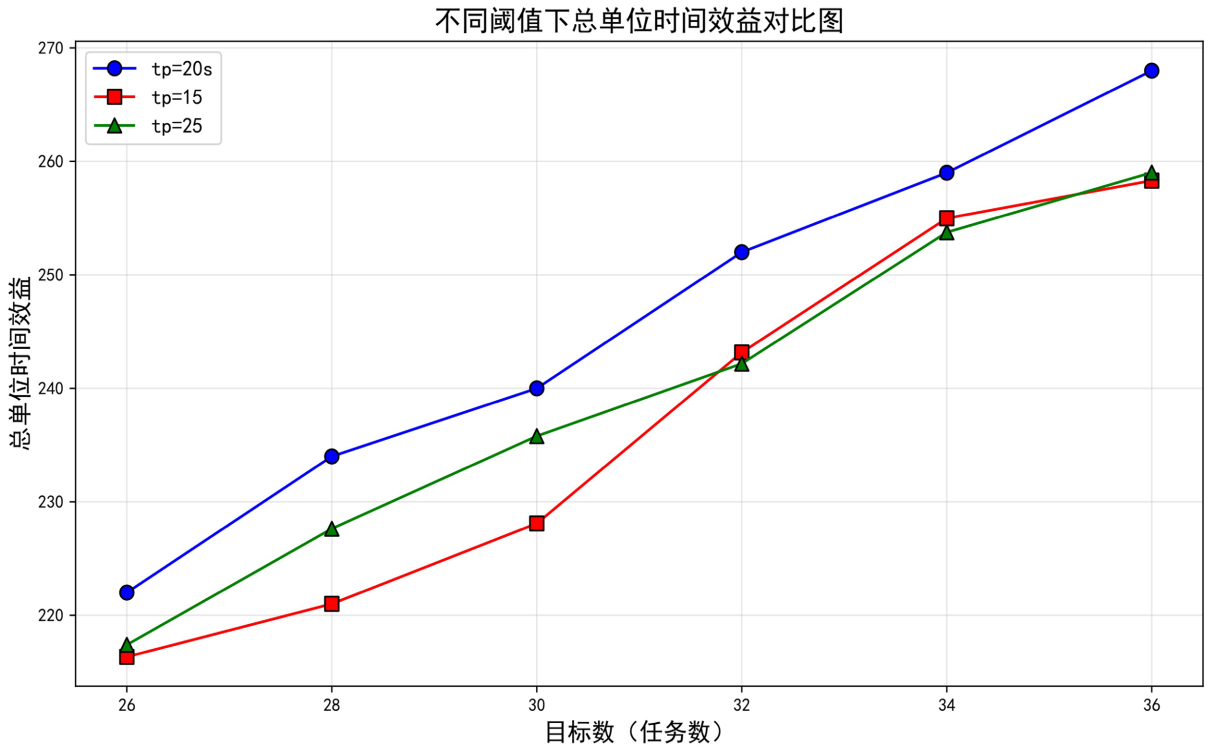

Figure 5. Total unit time efficiency under different thresholds
图 5. 不同阈值下的总单位时间效益图

图 7 展示了在固定数量的无人机下，三种算法在不同目标数量下的总单位时间效益对比。随着目标数量的增加本文算法的总单位时间效益比其他两种算法更好，这是由于本文算法给予无人机悬停等待和超时执行的弹性任务执行空间，这使得无人机能够完成更多的目标子任务数量，同时能够以较短的实行时间执行超时的子任务，从而使得单位时间效益增加。

图 8 对比了不同目标数下任务的完成率。结果显示，本文算法的任务完成率整体比其他两种算法更高。随着目标数量的增加任务的完成率逐渐降低，并且目标数量越多任务完成率下降得越快，这是因为

在固定数量的无人机下，无人机集群的整体执行能力不变，导致无人机无法在最大的超时弹性空间内执行目标后续子任务。

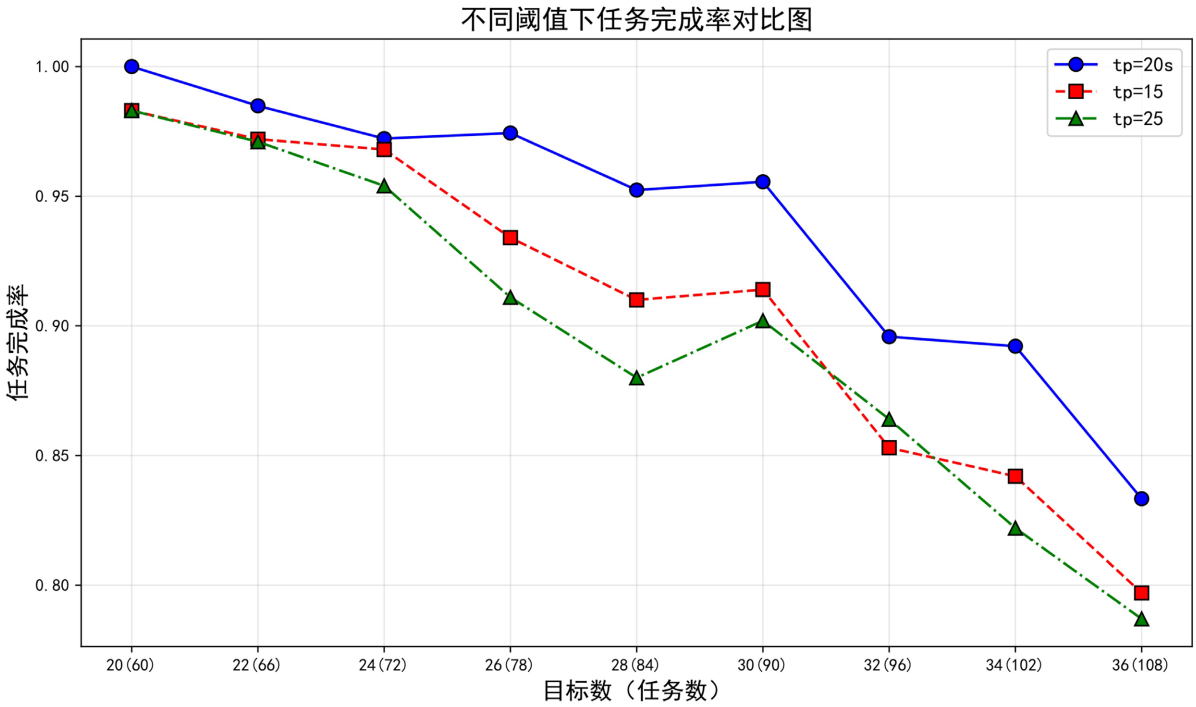

Figure 6. Task completion rate graph under different thresholds
图 6. 不同阈值下的任务完成率图

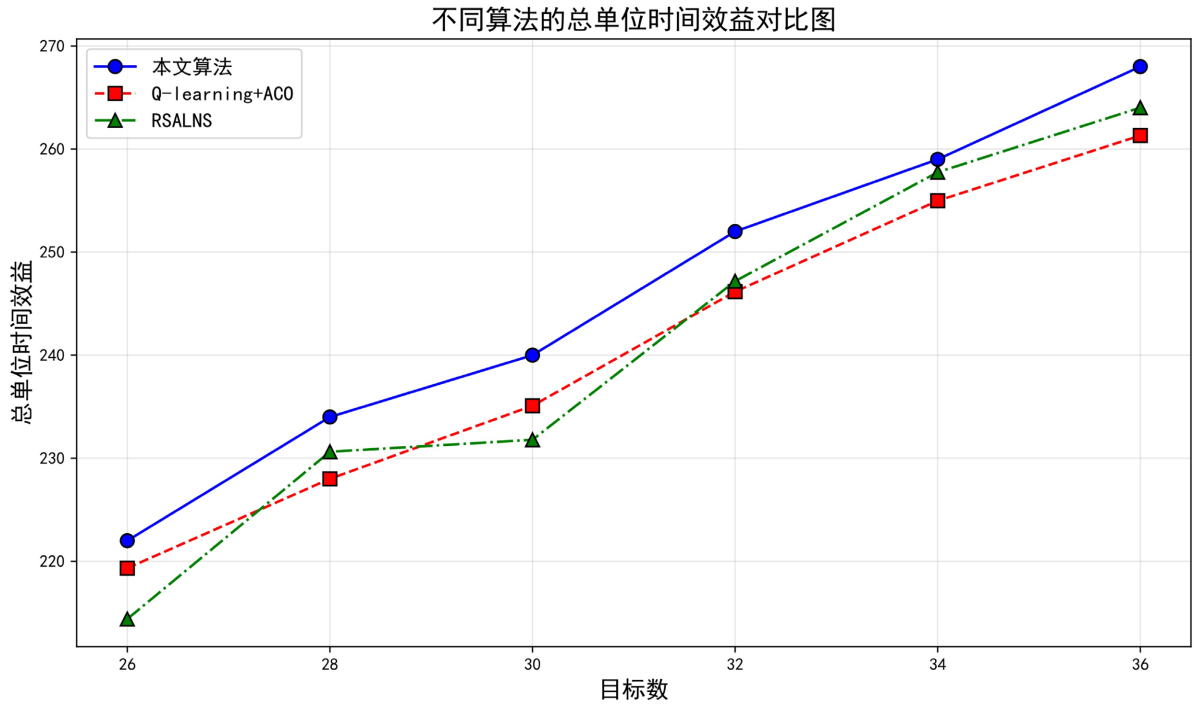

Figure 7. Comparison chart of total unit time efficiency
图 7. 总单位时间效益对比图

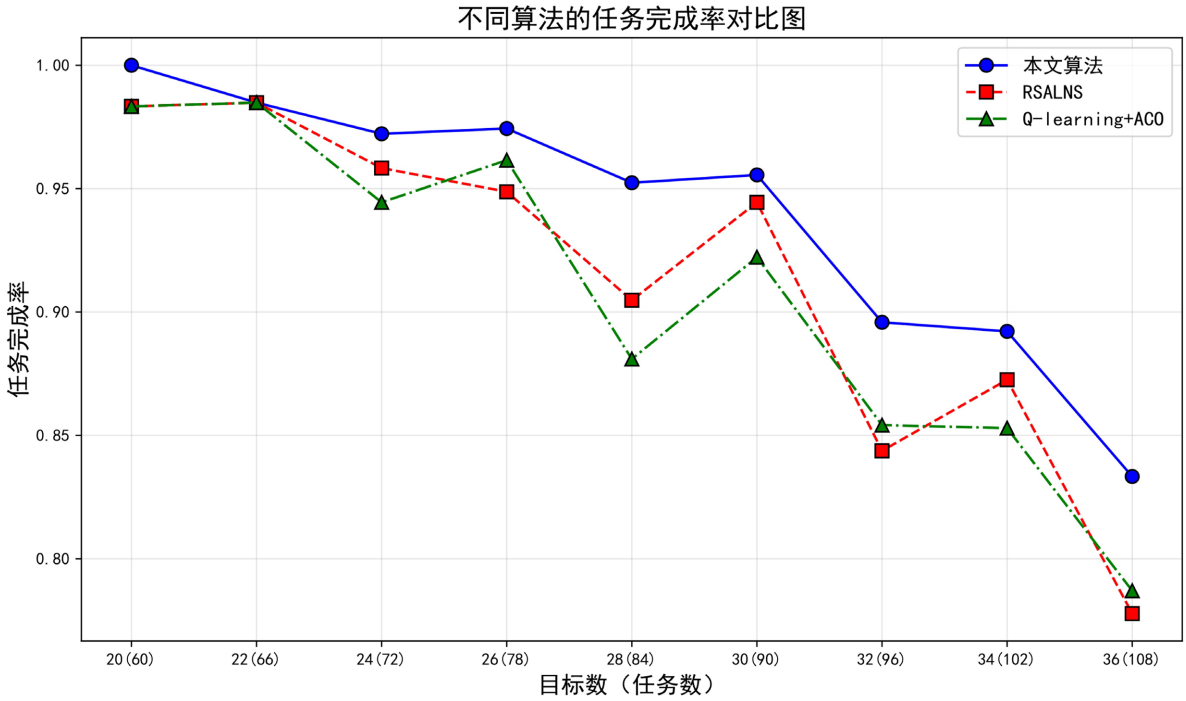

Figure 8. Comparison chart of task completion rate
图 8. 任务完成率对比图

7. 总结

本文针对面向延迟容忍的时序约束的多无人机任务分配问题，构建了无人机任务执行的动态时间窗，使得无人机可以提前到达目标点悬停等待或者超时执行任务，为无人机任务执行提供弹性空间。通过构建强化学习模型引导无人机进行任务决策，并设计了冲突共识机制解决无人机任务决策中的冲突问题。实验结果表明，本文所提出的方法展现了出色的稳定性和适应性，能够有效引导无人机进行任务决策，并且相比于现有算法，无人机的任务决策具有更高的效益、更优的任务完成率。未来的研究将主要探索任务时变性导致多个任务聚合的问题。

基金项目

国家自然科学基金资助项目(61602305, 61802257)；上海市自然科学基金资助项目(18ZR1426000, 19ZR1477600)。

参考文献

[1] Wang, Z., Liu, L., Long, T. and Wen, Y. (2018) Multi-UAV Reconnaissance Task Allocation for Heterogeneous Targets Using an Opposition-Based Genetic Algorithm with Double-Chromosome Encoding. *Chinese Journal of Aeronautics*, **31**, 339-350. https://doi.org/10.1016/j.cja.2017.09.005

[2] Wei, Z. and Zhao, X. (2022) Multi-UAVs Cooperative Reconnaissance Task Allocation under Heterogeneous Target Values. *IEEE Access*, **10**, 70955-70963. https://doi.org/10.1109/ACCESS.2022.3188859

[3] Yue, L., Yang, R., Zuo, J., Zhang, Y., Li, Q. and Zhang, Y. (2022) Unmanned Aerial Vehicle Swarm Cooperative Decision-Making for SEAD Mission: A Hierarchical Multiagent Reinforcement Learning Approach. *IEEE Access*, **10**, 92177-92191. https://doi.org/10.1109/access.2022.3202938

[4] Chakraa, H., Guérin, F., Leclercq, E. and Lefebvre, D. (2023) Optimization Techniques for Multi-Robot Task Allocation Problems: Review on the State-of-the-Art. *Robotics and Autonomous Systems*, **168**, Article ID: 104492. https://doi.org/10.1016/j.robot.2023.104492

[5] Jia, Z., Yu, J., Ai, X., Xu, X. and Yang, D. (2018) Cooperative Multiple Task Assignment Problem with Stochastic Velocities and Time Windows for Heterogeneous Unmanned Aerial Vehicles Using a Genetic Algorithm. *Aerospace Science and Technology*, **76**, 112-125. https://doi.org/10.1016/j.ast.2018.01.025

[6] Chen, Y., Yang, D. and Yu, J. (2018) Multi-UAV Task Assignment with Parameter and Time-Sensitive Uncertainties Using Modified Two-Part Wolf Pack Search Algorithm. *IEEE Transactions on Aerospace and Electronic Systems*, **54**, 2853-2872. https://doi.org/10.1109/taes.2018.2831138

[7] Ye, F., Chen, J., Tian, Y. and Jiang, T. (2020) Cooperative Task Assignment of a Heterogeneous Multi-UAV System Using an Adaptive Genetic Algorithm. *Electronics*, **9**, Article 687. https://doi.org/10.3390/electronics9040687

[8] Chen, L., Liu, W. and Zhong, J. (2022) An Efficient Multi-Objective Ant Colony Optimization for Task Allocation of Heterogeneous Unmanned Aerial Vehicles. *Journal of Computational Science*, **58**, Article ID: 101545. https://doi.org/10.1016/j.jocs.2021.101545

[9] Yu, X., Gao, X., Wang, L., Wang, X., Ding, Y., Lu, C., *et al*. (2022) Cooperative Multi-UAV Task Assignment in Cross-Regional Joint Operations Considering Ammunition Inventory. *Drones*, **6**, Article 77. https://doi.org/10.3390/drones6030077

[10] Cui, Y., Dong, W., Hu, D. and Liu, H. (2022) The Application of Improved Harmony Search Algorithm to Multi-UAV Task Assignment. *Electronics*, **11**, Article 1171. https://doi.org/10.3390/electronics11081171

[11] Zhang, R., Ren, H., Li, X. and Ding, Y. (2023) UAV Cluster Task Assignment Algorithm Based on Improved Artificial Gorilla Troops Optimizer. *IEEE Access*, **11**, 135133-135146. https://doi.org/10.1109/access.2023.3333912

[12] Cui, Y. (2024) An Improved Chaotic Self-Adapting Monkey Algorithm for Multi-UAV Task Assignment. *IEEE Journal on Miniaturization for Air and Space Systems*, **5**, 9-15. https://doi.org/10.1109/jmass.2023.3327721

[13] Xiao, Y., Li, Y., Liu, H., Chen, Y., Wang, Y. and Wu, G. (2024) Adaptive Large Neighborhood Search Algorithm with Reinforcement Search Strategy for Solving Extended Cooperative Multi Task Assignment Problem of UAVs. *Information Sciences*, **679**, Article ID: 121068. https://doi.org/10.1016/j.ins.2024.121068

[14] Lei, K., Guo, P., Zhao, W., Wang, Y., Qian, L., Meng, X., *et al*. (2022) A Multi-Action Deep Reinforcement Learning Framework for Flexible Job-Shop Scheduling Problem. *Expert Systems with Applications*, **205**, Article ID: 117796. https://doi.org/10.1016/j.eswa.2022.117796

[15] Song, W., Chen, X., Li, Q. and Cao, Z. (2023) Flexible Job-Shop Scheduling via Graph Neural Network and Deep Reinforcement Learning. *IEEE Transactions on Industrial Informatics*, **19**, 1600-1610. https://doi.org/10.1109/tii.2022.3189725

[16] Su, C., Zhang, C., Xia, D., Han, B., Wang, C., Chen, G., *et al*. (2023) Evolution Strategies-Based Optimized Graph Reinforcement Learning for Solving Dynamic Job Shop Scheduling Problem. *Applied Soft Computing*, **145**, Article ID: 110596. https://doi.org/10.1016/j.asoc.2023.110596

[17] Jinqiang, H., Husheng, W., Renjun, Z., Rafik, M. and Xuanwu, Z. (2021) Self-organized Search-Attack Mission Planning for UAV Swarm Based on Wolf Pack Hunting Behavior. *Journal of Systems Engineering and Electronics*, **32**, 1463-1476. https://doi.org/10.23919/jsee.2021.000124

[18] Zhang, J., Cui, Y. and Ren, J. (2023) Dynamic Mission Planning Algorithm for UAV Formation in Battlefield Environment. *IEEE Transactions on Aerospace and Electronic Systems*, **59**, 3750-3765. https://doi.org/10.1109/taes.2022.3231244

[19] Deng, H., Huang, J., Liu, Q., Zhao, T., Zhou, C. and Gao, J. (2023) A Distributed Collaborative Allocation Method of Reconnaissance and Strike Tasks for Heterogeneous UAVs. *Drones*, **7**, Article 138. https://doi.org/10.3390/drones7020138

[20] Ma, Y., Li, X., Wang, H., Kang, Y., Cao, T., Zhang, Y., *et al*. (2024) Multi-UAVs Collaborative Task Allocation Based on Improved Consensus-Based Grouping Algorithm. 2024 *IEEE* 18*th International Conference on Control & Automation* (*ICCA*), Reykjavík, 18-21 June 2024, 850-855. https://doi.org/10.1109/icca62789.2024.10591950

[21] Liang, Z., Li, Q. and Fu, G. (2023) Multi-UAV Collaborative Search and Attack Mission Decision-Making in Unknown Environments. *Sensors*, **23**, Article 7398. https://doi.org/10.3390/s23177398