基于动态规划的多无人机通信连通性保持研究

武云龙^{1,2}, 张博^{1*}, 任小广^{1,2}, 王彦臻^{1,2}, 易晓东^{1*}

1. 中国人民解放军军事科学院国防科技创新研究院, 北京 100071;

2. 天津(滨海)人工智能军民融合创新中心, 天津 300457

* E-mail: zhangbo10@nudt.edu.cn; xdong_yi@163.com

收稿日期: 2019-08-19; 接受日期: 2019-10-09; 网络版发表日期: 2020-04-14

国家重点研发计划(编号: 2017YFB1001900)和国家自然科学基金(批准号: 91648204, 61906212, 61601486, 61802426)资助项目

摘要 随着无人机技术的逐渐成熟, 制造成本不断降低, 无人机在多个领域得到广泛应用, 如农业、工业、安全、军事等. 而无人机在执行任务时, 往往需要与地面基站保持通信的连通性. 但在距离较远或有遮挡情况下, 信道质量将严重下降, 这对通信连通性的保持提出挑战. 针对该问题, 本文考虑部署多架无人机作为通信中继, 以串联方式构建任务无人机与地面基站稳定的通信链路. 本文以长期信道容量作为优化指标, 基于动态规划理论提出了两种中继无人机的规划方法: CMMP-AT和CMMP-OBO. CMMP-AT方法强调在任务执行初期就部署全部中继无人机, 该方法部署较为简便, 但计算复杂度较高. CMMP-OBO方法提出按照任务需求逐架部署中继无人机, 该方法较为灵活, 且计算复杂度低, 扩展性好, 可节约中继无人机的运动能耗. 实验结果表明, CMMP-AT方法针对一个两中继无人机场景, 需要40.03 h规划结果, 而CMMP-OBO方法只需57.66 s即可规划出结果, 并且可节约3.87%的运动能耗. 此外, 为了精确控制中继无人机遵循规划出的轨迹行进, 本文基于模型预测控制方法实现对规划轨迹的追踪, 并在V-REP环境中实现了多中继无人机场景的仿真. 仿真结果表明, 相比PID控制方法, 模型预测控制方法能够更精确地追踪规划轨迹. 在两种仿真场景下, 模型预测控制方法的追踪误差仅为PID控制方法的43.97%和41.42%.

关键词 通信连通性, 动态规划, 多无人机系统, 通信中继, 模型预测控制

1 引言

近些年, 无人机(unmanned aerial vehicle, UAV)相关技术快速发展, 广泛应用于工业、安全、军事、科研等各个领域^[1]. 特别是在军事领域, 无人机大量代替有人机执行重要任务, 如情报、监视和侦察(intelligence, surveillance and reconnaissance, ISR)^[2]. 随着制

造成成本的下降, 无人机开始应用于民用和商用领域, 如区域探索^[3]、巡逻监视^[4]、包裹递送^[5]等. 在无人机执行任务中, 往往要求其在地面基站之间保持通信的连通性, 用以实现信息的回传和任务指令的下达. 但在实际环境中, 无人机与地面基站间的无线信道往往受路径衰减(pathloss)、阴影衰落(shadowing)和多径衰落(multipath fading)等因素的影响, 降低了信道容量并使

引用格式: 武云龙, 张博, 任小广, 等. 基于动态规划的多无人机通信连通性保持研究. 中国科学: 技术科学, 2020, 50: 538–550

Wu Y L, Zhang B, Ren X G, et al. Research on multi-UAV communication connectivity maintenance based on dynamic programming (in Chinese). Sci Sin Tech, 2020, 50: 538–550, doi: 10.1360/SST-2019-0291

通信链路不稳定. 为了保证任务的高效执行, 需要设计针对性策略来维持无人机和地面基站之间的通信连通.

当前, 已有一些工作针对通信连通性保持问题展开研究, 并总结了较多有意义的结论. 将现有工作分为两大类: 一类是基于信道预测的方法, 另一类是基于通信中继的方法. 基于信道预测的方法专注于加强对实际通信环境中无线信道空间变化的预测, 并使无人机仅在具有高通信质量的位置传输数据^[6,7]. 该方法虽然可以保证数据在低误码率(bit error rate, BER)的条件下传输, 但数据在传输前, 无人机需要飞行到通信质量好的位置, 这将使数据在一段时间内暂存在本地, 降低了数据回传的时效性. 第二类方法是通过在任务无人机与地面基站之间部署中继来保证通信的连通性^[8]. 该方法的优点是可以同时保证通信质量和时效性的要求, 但需额外部署无人机充当中继. 除上述两类方法外, 还可采用更可靠的编码和调制方式, 以支持高质量的数据传输. 然而, 此类方法并没有改变无人机的信道环境, 因此只作为一种辅助方法. 在本文中, 我们的目标是在任务无人机与地面基站之间构建高容量的通信信道, 以高效支持数据的实时传输. 因此, 经过综合考虑, 本文更倾向使用中继无人机来保证任务无人机与地面基站间的通信连通性.

当前基于中继改善通信连通性的研究往往面临两个主要问题: 一个是如何量化表示通信连通性指标, 另一个是如何规划中继的运动以优化通信连通性指标.

第一个问题涉及两方面技术: 无线信道建模和连通性指标选择. 现有的关于信道建模的工作可以分为两类. 第一类是将无人机的通信范围建模为半径为 r 的球形区域, 当无人机之间(或无人机与地面基站之间)的距离小于 r 时, 则认为无人机可以无差错地相互通信. 相反, 如果距离超过 r , 则认为无法通信^[9]. 然而, 球形模型只是一种数学上的简化, 不能准确反映实际的通信环境. 而第二类信道建模方法则充分考虑路径损耗、阴影衰落和多径衰落的影响, 基于统计理论对无线信道进行建模, 使其更贴近真实的通信环境^[6]. 而在通信连通性指标的选取上, 往往需要根据实际任务的需要进行选择, 典型的通信连通性指标包括信道容量、误码率、包错误率(packet error rate, PER)、信噪比(signal-to-noise ratio, SNR)等^[10].

第二个问题关注如何计算中继的运动轨迹, 以实

现通信连通性指标的最优. 相关研究中, 采用图论来建模无人机网络是一种较为常用的方法. 在该模型中, 无人机与无人机以及无人机与地面基站之间的可达通信链路由图中的边表示, 无人机与地面基站表示为图的节点^[11~13]. 通过运用图论的方法, 对整个无人机网络的通信连通性进行求解和优化. 然而, 即使图论方法已应用于许多领域, 但它依然无法准确描述实际无线信道的特征. 因此, 针对该问题, 一些文献提出使用更为实际的信道模型, 并结合具体的算法进行求解, 如凸优化^[14]、混合整数线性规划^[15]、最优控制理论^[16]等. 然而, 上述方法一般适用于基于当前时刻的信息对下一时刻无人机的行为进行规划, 其局限是只关注如何根据当前时刻的信息获得当前时刻系统的最优解, 而不关心当前的解是否对后续时刻的解产生影响. 在一些情况下, 当前时刻的最优解从全局角度来看并不一定是最优的. 对于环境已知的情况, 中继无人机是可以提前“预测”后续动作产生的影响. 因此, 中继无人机可以根据全局信息, 选择一组最优的动作使得系统的连通性指标从长期(long-term)来看是最优的.

动态规划(dynamic programming, DP)理论最初提出就是用于求解多步决策问题, 并且经过长期的发展, 动态规划相关理论已经十分成熟. 因此, 本文拟基于动态规划理论, 对中继无人机在通信连通性约束下的运动规划问题(connectivity-maintaining motion planning, CMMP)展开研究, 并提出两种中继无人机运动规划方法, 以优化整个多无人机系统的长期平均信道容量. 本文的主要贡献如下.

(1) 考虑了一种多无人机中继场景. 该场景中, 一组无人机作为中继, 通过自主改变位置, 以改善任务无人机与地面基站之间的数据传输质量.

(2) 面向上述场景, 本文提出了一种基于动态规划的中继无人机轨迹规划方法——CMMP-AT (all together), 该方法通过在初始时刻就部署全部中继无人机以实现系统信道容量的优化.

(3) 为了节省中继无人机的运动能耗, 同时降低CMMP-AT方法的计算复杂度, 本文设计了一种面向阶段的规划方法——CMMP-OBO (one by one), 该方法通过增量式的部署中继无人机, 大大降低了计算的时间开销.

(4) 本文提出了一种基于模型预测控制(model predictive control, MPC)的轨迹追踪方法, 相比PID控

制方法, 该方法可更精确地实现规划轨迹的追踪。

2 系统建模

图1和2描述了本文关注的多中继无人机(以下简称“UAV”)场景。在图1中, 所有UAV(包括任务UAV和中继UAV)在初始时刻均位于地面基站周围。当接收到任务时, 任务UAV将起飞并飞往指定区域执行任务。为了保持任务UAV与地面基站之间的通信连通性, 将依据规划算法部署中继UAV到整个系统中, 如图2所示。

针对上述场景, 本文提出了两种中继UAV的运动规划方法, 分别为CMMP-AT和CMMP-OBO。在CMMP-AT方法中, 所有中继UAV同时起飞, 在任务初始时刻即全部加入到系统中。而在CMMP-OBO方法中, 初始时刻只有一架中继UAV加入到系统, 当任务UAV与地面基站之间的距离超过某个阈值时, 新的中继UAV起飞并加入系统, 以进一步改善通信链路的质量。在上述两种方法中, 均假设中继UAV可快速达到

图1 初始时刻, 所有UAV均位于地面基站周围

Figure 1 At the initial moment, all UAVs are located around the ground base station.

图2 当收到任务指令后, 任务UAV起飞执行任务, 中继UAV保证任务UAV与地面基站的通信连通性

Figure 2 When receiving task commands, the task UAV takes off, and the relay UAVs ensure the communication connectivity between the task UAV and the ground base station.

指定高度, 然后保持水平高度飞行^[17]。并且UAV的运动轨迹是离散化的位点序列, 相邻位点之间用直线连接。本文拟通过优化中继UAV的运动轨迹, 进而最大化系统的长期平均信道容量。

从信息论可知, 信道容量和接收信噪比相关, 而接收信噪比通常受三方面因素的影响: 路径衰减、阴影衰落和多径衰落。根据文献[18], 这三方面的因素常常可以分开考虑。而相比路径衰减和阴影衰落, 多径衰落的影响往往较小^[18], 因此, 本文忽略多径衰落的影响。另外, 在该场景下, 通信信道可以分为两类情况: 空-空信道和空-地信道。

对于空-空信道而言, 假设视距传输(line-of-sight, LOS)总是存在的, 所以信道可以建模为加性高斯白噪声(additive white Gaussian noise, AWGN)信道。UAV i 与 UAV j 间的单跳(per-hop)信道容量可基于香农公式^[19]表示为

$$C_{i,j} = B \log_2(1 + \gamma_{i,j}), \quad (1)$$

其中, B (Hz)和 $\gamma_{i,j}$ 分别表示信道的带宽和接收信噪比。 $\gamma_{i,j}$ 可认为只受路径衰减作用的影响, 且表示为 $\gamma_{i,j} = P_{tx,i} / \|x_i - x_j\|^\beta N_0 B$, 其中 N_0 (W/Hz)为噪声功率谱密度(power spectral density, PSD), $P_{tx,i}$ 表示位于位置 x_i ($x_i \in \mathbb{R}^2$)的UAV i 的发射功率, x_j ($x_j \in \mathbb{R}^2$)为接收数据的UAV j 的位置, β 为路径衰减指数。

对于空-地信道, 其通常会受到路径衰减和阴影衰落的共同影响。因此, 本文将UAV i 与地面基站 g 间的信道容量 $C_{i,g}$ 表示为两部分的和, 即

$$C_{i,g} = (1 - p_{sh}) B \log_2(1 + \gamma_{i,g}) + p_{sh} B \log_2(1 + \alpha \gamma_{i,g}), \quad (2)$$

其中, p_{sh} 表示阴影衰落的概率, 参数 α 表示阴影衰落对信噪比的影响。

令 $\mathbf{x} = [x_1, x_2, \dots, x_N]$ ($\mathbf{x} \in \mathbb{R}^{2N}$)表示中继UAV的位置向量, 其中, x_i 表示中继UAV i 的位置, N 为中继UAV的个数。假设中继UAV均采用“解码-转发(decode-and-forward)”协议, 并且每个UAV具有相同的时隙和发射功率。因此, 该系统的多跳信道容量可以表示为

$$C_s(\mathbf{x}, q) = \frac{\min[C_{1,2}(\mathbf{x}, q), C_{2,3}(\mathbf{x}), \dots, C_{N+1,g}(\mathbf{x})]}{N+1}, \quad (3)$$

其中, $(N+1)$ 表示UAV的数量, 包括一架任务UAV(位于

位置 q , $q \in \mathbb{R}^2$)和 N 架中继UAV. 通过获得任务UAV在下一时间步的位置, 规划算法可以计算得到下一时间步使得信道容量最优的中继UAV的位置向量. 面向单时间步的优化方法, 当前已有较为成熟的理论, 这里不再赘述. 本文专注于求解使系统长期(多时间步)信道容量最优的中继UAV最优轨迹. 为求解该问题, 本文将基于动态规划理论展开建模和分析.

3 基于动态规划的信道容量优化

在讨论具体算法前, 首先对本文关注的问题进行建模. 设 $C_s^i(\mathbf{x}^i, q^i)$ 表示第 i 个时间步的系统多跳信道容量, 其中 $\mathbf{x}^i$ 和 q^i 分别为第 i 个时间步中继UAV和任务UAV的位置向量. 接着, 假设共有 T 个时间步, 状态转换过程可以用图3进行表示. 该状态转换过程可看作一个多步决策问题, 通过优化一系列的状态转换动作(决策), 实现多个时间步的多跳信道容量之和最优.

设 $\pi(\mathbf{x}^i, q^i)$ 表示从状态 $(\mathbf{x}^i, q^i)$ 到具体动作的映射. 例如, $\pi(\mathbf{x}^i, q^i) = \mathbf{a}$ 表示当状态为 $(\mathbf{x}^i, q^i)$ 时, 遵照策略 π , 中继UAV应采取的行动为 $\mathbf{a}$ ($\mathbf{a} \in \mathbb{R}^{2N}$). 在动态规划理论中, 其规划目标是求解使多步目标函数最优的行动策略. 在本文中, 将目标函数 V 建模为系统的平均信道容量, 具体表示为

$$\max_{\pi} V' = \frac{1}{T} [C_s^1(\mathbf{x}^1, q^1) + \dots + C_s^T(\mathbf{x}^T, q^T)] \quad (4)$$

其中, T 通常是固定且有限的, 因此, 最大化 V' 等价于最大化累积信道容量. 此外, 考虑到最大速度的限制, 在每个时间步中, 应约束中继UAV的运动范围. 因此, 本文将中继UAV的行动空间定义为 $\mathcal{U} = \{\mathbf{a} = [a_1, \dots, a_N] \mid \|a_i\| \leq \sigma_i, a_i \in \mathbb{R}^2, \forall i\}$, 其中 σ_i 表示中继UAV i 的最大运动范围, 并且满足 $\mathbf{x}^i = \mathbf{x}^{i-1} + \mathbf{a}$. 基于上述分析, 本文将构建如式(5)所示的最优化问题:

图3 状态转换过程
Figure 3 State transition process.

$$\begin{aligned} \max_{\pi} V &= C_s^1(\mathbf{x}^1, q^1) + \dots + C_s^T(\mathbf{x}^T, q^T) \\ \text{s. t. } \pi(\mathbf{x}^i, q^i) &\in \mathcal{U}, i = 0, 1, \dots, T-1, \\ \mathbf{x}^i &= \mathbf{x}^{i-1} + \pi(\mathbf{x}^{i-1}, q^{i-1}), i = 1, 2, \dots, T, \end{aligned} \quad (5)$$

其中, V 表示累积信道容量, 在动态规划理论中称为“值函数(value function)”.

针对式(5)所示的优化问题, 本文首先考虑使用CMMP-AT方法进行求解, 具体流程如表1的算法1所示. 初始时刻, 所有中继UAV均位于地面基站周围. 当任务UAV沿着轨迹 $Q = \{q^1, q^2, \dots, q^T\}$ ($Q \in \mathbb{R}^{2T}$)飞行时, 所有中继UAV也会同时起飞. 在CMMP-AT算法中, 对于值函数和策略的初始化, 较为常用的方法是随机初始化和全0初始化, 本文均采用全0初始化, 即 $V(\mathbf{x}^i, q^i) = 0$ 和 $\pi(\mathbf{x}^i, q^i) = 0$ (第1行). Δ 为判停的误差值, 当 Δ 大于阈值 ε 时则进行值函数和策略的优化(第2行). 在CMMP-AT方法中, 每次迭代优化, 需遍历整个状态空间 $\mathcal{X}$ (第4行). 在进行值函数和策略更新时需要进行判断, 如果当前时间步为最后一个($i = T$), 则只需对最后一个时间步的单步信道容量优化(第8~9行); 否则需要考虑后续时间步的值函数(第11~12行). 由于该算法考虑在初始时刻就部署全部的中继UAV, 因此算法1也称为CMMP-AT方法.

但是, 一方面, 在初始时刻就激活全部的中继UAV会造成不必要的运动能量消耗. 例如, 在任务UAV距离地面基站较近时, 此时, 没有必要部署全部的中继UAV. 因此, 使中继UAV在初始时刻就全部升空会造成额外的能耗损失. 另一方面, 遍历整个状态空间(算法1的第4行)在一些场景中是没有必要的. 例如, 对于状态空间 $\mathcal{X}$ 的某些区域, 中继UAV可能永远不会经过. 因此, 对这部分区域的遍历将造成额外的计算开销, 从而限制了该方法在大规模场景中的应用. 因此, 我们需要设计高效的算法, 以解决算法1的计算和能耗问题.

4 面向阶段的信道容量优化

为了降低中继UAV的运动能耗和算法的时间复杂度, 本文针对算法1提出了两种优化方法. 第一种优化方法的核心思想是将该多步优化问题进一步分成若干个阶段, 每个阶段可以看作是原问题的子问题, 每个子问题可以包含多个时间步. 在每个阶段中, 中继UAV的

表 1 CMMP-AT方法的算法流程**Table 1** The algorithm flow of CMMP-AT

算法1 CMMP-AT	
已知: $\mathcal{Q} = \{q^1, q^2, \dots, q^T\}$: 任务UAV的运动轨迹; T : 时间步数; $\mathcal{X}$: 状态空间; $\mathcal{U}$: 行动空间; ϵ : 算法中止条件.	
求: π : 最优策略.	
1:	初始化: $\Delta = \infty; V(\mathbf{x}, q^i) = 0, \pi(\mathbf{x}, q^i) = 0, \forall \mathbf{x} \in \mathcal{X}, \forall i \in [1, T]$
2:	while $\Delta < \epsilon$ do
3:	$\Delta = 0$
4:	for $\mathbf{x} \in \mathcal{X}$ do
5:	for $i \in [1, T]$ do
6:	$V_{\text{old}} = V(\mathbf{x}, q^i)$
7:	if $i == T$ then
8:	$V(\mathbf{x}, q^i) = \max_{\mathbf{a} \in \mathcal{U}} C(\mathbf{x} + \mathbf{a}, q^i)$
9:	$\pi(\mathbf{x}, q^i) = \operatorname{argmax}_{\mathbf{a} \in \mathcal{U}} C(\mathbf{x} + \mathbf{a}, q^i)$
10:	else
11:	$V(\mathbf{x}, q^i) = \max_{\mathbf{a} \in \mathcal{U}} [C(\mathbf{x} + \mathbf{a}, q^i) + V(\mathbf{x} + \mathbf{a}, q^{i+1})]$
12:	$\pi(\mathbf{x}, q^i) = \operatorname{argmax}_{\mathbf{a} \in \mathcal{U}} [C(\mathbf{x} + \mathbf{a}, q^i) + V(\mathbf{x} + \mathbf{a}, q^{i+1})]$
13:	end if
14:	$\Delta = \max(\Delta, V(\mathbf{x}, q^i) - V_{\text{old}})$
15:	end for
16:	end for
17:	end while
18:	return π

数量是固定的, 即没有新的中继UAV被添加到当前网络中, 这也意味着每个阶段中网络拓扑和路由策略是静态的. 当任务UAV和地面基站之间的距离持续增加并触发某个阈值时, 新的中继UAV将被添加到网络中, 从而标志着一个新阶段的开始, 该方法的思想如图4所示. 该优化方法的优点是在每个阶段只需关注固定数目中继UAV的运动规划, 同时其他中继UAV可以在基站原地等待, 进而省去了在空中飞行的运动能耗.

另一种优化方法是压缩状态空间 $\mathcal{X}$ 的大小. 在(每个时间步)最大速度的限制下, 中继UAV在单个阶段内都有一个最大运动范围. 例如, 如果当前阶段包括3个时间步, 那么中继UAV在3个时间步中到达的最大运动范围可以用一个矩形区域进行表示, 如图5所示. 基于

图 4 多步优化问题的阶段划分**Figure 4** Stage division of multi-step optimization problem. **图 5** 状态空间压缩**Figure 5** State space compression.

该优化方法, 可以有效压缩状态空间大小, 从而降低算法的计算复杂度.

基于上述考虑, 式(5)所描述的优化问题可建模为一个多阶段多步决策问题. 针对该问题, 本文提出一种面向阶段的规划方法——CMMP-OBO, 该方法强调增量式的逐架部署中继UAV. 算法2给出了CMMP-OBO方法的详细求解过程(表2).

在算法2中, $\mathcal{Q}$ 为任务UAV的运动轨迹, $\mathbf{x}^0$ 为0时刻中继UAV的初始位置向量. 所有中继UAV在初始时刻均停在地面基站周围. 算法2的输出同样是中继UAV的最优策略, 遵循该策略可最大化系统的长期信道容量. $\mathbf{x}_{\text{init}}^i$ 和 $\mathbf{x}_{\text{new}}^i$ 分别表示在阶段 i 时中继UAV的初始位置和最后一个时间步的位置. $\mathbf{x}_{\text{new}}^i$ 初始为空集. 在初始时刻, 轨迹 $\mathcal{Q}$ 将根据任务UAV与基站的距离划分为多

表 2 CMMP-OBO方法的算法流程

Table 2 The algorithm flow of CMMP-OBO

算法2 CMMP-OBO

已知: $\mathcal{Q} = \{q^1, q^2, \dots, q^T\}$: 任务UAV的运动轨迹; T : 时间步数;

$\mathbf{x}^0 = [x_1^0, x_2^0, \dots, x_N^0]$: 中继UAV的初始位置向量.

求: π : 最优策略.

```

1:      初始化:  $\mathbf{x}_{\text{new}}^0 = \phi, \zeta = \phi, \pi = \phi$ 
      将 $\mathcal{Q}$ 划分为 $P$ 个阶段:
2:       $\mathcal{Q} = \{\mathcal{Q}_1, \mathcal{Q}_2, \dots, \mathcal{Q}_P\}, \mathcal{Q}_i = \{q_i^1, q_i^2, \dots, q_i^{T^i}\}$ 
3:      for  $i \in [1, P]$  do
4:           $\mathbf{x}_{\text{init}}^i = [\mathbf{x}_{\text{new}}^{i-1}, x_i^0]$ 
5:           $[\mathbf{x}_{\text{new}}^i, \pi^i] = \text{stage\_opt\_policy}(\mathbf{x}_{\text{init}}^i, \mathcal{Q}_i, T^i)$ 
6:           $\pi = \pi \cup \pi^i$ 
7:      end for
8:      return  $\pi$ 

```

个阶段. 阶段划分的规则为: 按顺序遍历 $\mathcal{Q}$ 中的每个位点, 当位点与基站的距离超过阈值距离时, 则将前面的位点划分为一个阶段, 并将前面的位点从 $\mathcal{Q}$ 中提取出来保存为一条子轨迹; 当超过2倍的阈值距离时则再划分为一个阶段, 依此类推. 最终, 依据阈值距离将形成 P 条子轨迹 $\mathcal{Q}_i = \{q_i^1, q_i^2, \dots, q_i^{T^i}\}, \forall i \in [1, P]$. 我们假定, 当任务UAV与基站之间的距离超过阈值距离时, 一架中继UAV将从地面基站处起飞并加入到中继系统中. 在CMMP-OBO方法中, 由于需要根据中继UAV的初始位置设定待遍历的状态空间, 因此在每个阶段开始前, 需更新中继UAV的位置(第4行). 其中, x_i^0 为中继UAV i 在初始时刻的位置, $\mathbf{x}_{\text{new}}^{i-1}$ 为上一阶段所有中继UAV的位置向量. 由于每个阶段都将引入新的中继UAV, 因此新的中继UAV的初始位置向量为 $\mathbf{x}_{\text{init}}^i = [\mathbf{x}_{\text{new}}^{i-1}, x_i^0]$. 针对每个阶段的子问题, 我们基于动态规划方法对其进行求解, 从而找到当前阶段中继UAV的最佳运动轨迹. 在算法2中, 我们用函数stage_opt_policy()表示面向阶段的中继UAV最优策略的求解过程, π^i 表示求出的阶段 i 的中继UAV的最优运动策略, $\mathbf{x}_{\text{new}}^i$ 表示阶段 i 最后一个时间步中继UAV的位置向量. 每个阶段的最优策略保存在 π 中.

stage_opt_policy()函数的具体实现如算法3所示(表3). T^i 表示阶段 i 的时间步数. 在动态规划方法中, 遍

表 3 stage_opt_policy()函数的算法流程

Table 3 The algorithm flow of stage_opt_policy()

算法3 函数stage_opt_policy()的实现

已知: $\mathbf{x}_{\text{init}}^i$: 阶段 i 中继UAV的初始位置; T^i : 阶段 i 的时间步数; $\mathcal{Q}_i$: 阶段 i 任务UAV的轨迹; $\mathcal{U}$: 行动空间; $\mathcal{X}$: 状态空间; $\mathcal{X}^-$: 压缩后的状态空间; ε : 算法中止条件.

求: π^i : 阶段 i 的最优运动策略; $\mathbf{x}_{\text{new}}^i$: 阶段 i 中继UAV在最后一个时间步的位置向量.

```

1:      初始化:  $\Delta = \infty, V(\mathbf{x}, q_i^j) = 0, \pi(\mathbf{x}, q_i^j) = 0, \forall \mathbf{x} \in \mathcal{X}$ 
       $\forall j \in [1, T^i]$ 
2:       $\mathbf{x}_{\text{new}}^i = \mathbf{x}_{\text{init}}^i$ 
3:       $\mathcal{X}^- = \text{compress}(\mathbf{x}_{\text{new}}^i, \mathcal{X})$ 
4:      while  $\Delta < \varepsilon$  do
5:           $\Delta = 0$ 
6:          for  $\mathbf{x} \in \mathcal{X}^-$  do
7:              for  $j \in [1, T^i]$  do
8:                   $V_{\text{old}} = V(\mathbf{x}, q_i^j)$ 
9:                  if  $j = T^i$  then
10:                      $V(\mathbf{x}, q_i^j) = \max_{\mathbf{a} \in \mathcal{U}} C(\mathbf{x} + \mathbf{a}, q_i^j)$ 
11:                      $\pi(\mathbf{x}, q_i^j) = \arg\max_{\mathbf{a} \in \mathcal{U}} C(\mathbf{x} + \mathbf{a}, q_i^j)$ 
12:                  else
13:                      $V(\mathbf{x}, q_i^j) = \max_{\mathbf{a} \in \mathcal{U}} [C(\mathbf{x} + \mathbf{a}, q_i^j) + V(\mathbf{x} + \mathbf{a}, q_i^{j+1})]$ 
14:                      $\pi(\mathbf{x}, q_i^j) = \arg\max_{\mathbf{a} \in \mathcal{U}} [C(\mathbf{x} + \mathbf{a}, q_i^j) + V(\mathbf{x} + \mathbf{a}, q_i^{j+1})]$ 
15:                  end if
16:                   $\Delta = \max(\Delta, |V(\mathbf{x}, q_i^j) - V_{\text{old}}|)$ 
17:              end for
18:          end for
19:      end while
20:      for  $j \in [1, T^i]$  do
21:           $\mathbf{x}_{\text{new}}^i = \mathbf{x}_{\text{new}}^i + \pi(\mathbf{x}_{\text{new}}^i, q_i^j)$ 
22:      end for
23:       $\pi^i = \pi$ 
24:      return  $\mathbf{x}_{\text{new}}^i, \pi^i$ 

```

历所有状态可能涉及较高的计算时间复杂度, 因此我们考虑使用空间压缩的方法, 将原始状态空间 $\mathcal{X}$ 压缩为 $\mathcal{X}^-$. 基于该优化方法, 算法3对原始状态空间 $\mathcal{X}$ 大小的依赖可以由 $\mathcal{X}^-$ 替代, 而 $\mathcal{X}^-$ 的大小只与中继UAV的最大速度有关. 因此, 对于大规模问题, 中继UAV在每个阶段的运动范围远小于整个状态空间 $\mathcal{X}$ 的大小. 算法3的整体流程与算法1类似, 不同点在于在迭代开始前, 需要依据中继UAV的初始位置对状态空间 $\mathcal{X}$ 进行压缩(第2~3行). 算法3的第4~19行是主要的值迭代过程. 在得到当前阶段的最优策略 π 后, 需要根据中继UAV的初始位置向量 $\mathbf{x}_{\text{init}}^i$ 计算出当前阶段最后一个时间步中继UAV的位置向量(第20~22行), 以供后续阶段使用. 最后, 算法3的输出是当前阶段的最优运动策略和最后一个时间步中继UAV的位置向量.

5 基于模型预测控制的轨迹追踪

CMMP-AT和CMMP-OBO方法只规划出了中继UAV的位点序列(轨迹). 但在实际场景中, 还需依据规划出的轨迹生成UAV的控制指令. 为了保证UAV轨迹追踪的准确性, 本文将基于MPC方法对UAV(包括任务UAV和中继UAV)的轨迹追踪过程进行建模和求解.

假设每架UAV的动力学特性相同, 且为离散时间系统(discrete-time system), 其运动学方程可表示为

$$\begin{aligned} \mathbf{x}_i^{k+1} &= f(\mathbf{x}_i^k, \mathbf{u}_i^k) = \mathbf{x}_i^k + \mathbf{u}_i^k \Delta t, \\ \mathbf{x}_i^k &\in \mathcal{X}, \mathbf{u}_i^k \in \mathcal{A}, \end{aligned} \quad (6)$$

其中, $\mathbf{x}_i^k$ 和 $\mathbf{u}_i^k$ 对应UAV i 在第 k 时间步的位置向量和速度控制向量, Δt 为采样时间步长度. $\mathcal{X}$ 和 $\mathcal{A}$ 为UAV的可行位置和可行速度集合. 由于各UAV是相互解耦的, 因此, 所有UAV的运动学方程均可由式(6)描述.

在本文场景下, 控制的目标是, 各UAV按照规划出的轨迹行进, 其数学问题是, 在位置约束和速度约束下, 最小化式(7)所示的代价函数.

$$\begin{aligned} \underset{\mathbf{u}_i^k, \forall k}{\operatorname{argmin}} \quad & \sum_{k=0}^{N_{\text{mpc}}-1} \left(\left\| \mathbf{x}_i^k - \mathbf{x}_{i,\text{ref}}^k \right\|_Q^2 + \left\| \mathbf{u}_i^k - \mathbf{u}_{i,\text{ref}}^k \right\|_R^2 \right) \\ & + \left\| \mathbf{x}_i^{N_{\text{mpc}}} - \mathbf{x}_{i,\text{ref}}^{N_{\text{mpc}}} \right\|_{Q_{N_{\text{mpc}}}}^2 \end{aligned}$$

$$\begin{aligned} \text{s. t.} \quad & \mathbf{x}_i^{k+1} = f(\mathbf{x}_i^k, \mathbf{u}_i^k), \\ & \mathbf{u}_i^k \in \mathcal{A}, \\ & \mathbf{x}_i^k \in \mathcal{X}, \\ & \mathbf{x}_i^0 = \mathbf{x}_i^0, \end{aligned} \quad (7)$$

其中, $\mathbf{x}_{i,\text{ref}}^k$ 和 $\mathbf{u}_{i,\text{ref}}^k$ 分别为UAV i 在第 k 时间步的参考位置向量和速度控制向量, N_{mpc} 为预测的时间步数. 在本文中, $\mathbf{u}_{i,\text{ref}}^k$ 设置为无人机匀速飞行时的期望速度, 而 $\mathbf{x}_{i,\text{ref}}^k$ 是依据 $\mathbf{u}_{i,\text{ref}}^{k-1}$ 和式(6)计算得出的. 与最优控制问题不同, 式(7)本质上为滚动优化问题, 可基于MPC理论对该问题展开求解. 具体过程为: 首先, 测量UAV的当前位置 $\mathbf{x}_i^0$, 并将其赋给位置向量 $\mathbf{x}_i^0$ 作为初始位置向量; 然后, 通过求解问题(7)可得到最优速度控制序列: $\mathbf{u}_i^0, \mathbf{u}_i^1, \dots, \mathbf{u}_i^{N_{\text{mpc}}-1}$; 最后, 取 $\mathbf{u}_i^0$ 作为当前时刻UAV i 的速度控制输入, 执行完毕后, 重复上述过程. 而针对问题(7)的求解, 当前已有较为成熟的求解方法, 本文使用开源求解器ACADO对问题(7)进行优化求解. 通过不断滚动优化, 可保证UAV能够精确地遵循预设轨迹行进.

6 实验验证及分析

为了验证本文提出方法的有效性, 本文选取单任务UAV和多中继UAV的场景进行仿真实验. 初始时刻, 所有中继UAV均位于基站周围. 当任务UAV起飞并飞向指定区域时, 为了保持通信的连通性, 需要部署中继UAV以提高链路的通信质量. 本节将验证两种中继策略: CMMP-AT和CMMP-OBO. 首先面向一个单阶段问题来验证算法的收敛性. 然后, 将通过实验对比CMMP-AT方法与CMMP-OBO方法的运行效果. 接着, 验证了在不同阈值距离设置下两种方法的性能. 最后, 对比了基于MPC控制方法和PID控制方法在轨迹追踪上的运行效果. 另外, 在本节的仿真实验中, 均设置任务UAV和中继UAV的飞行高度为5 m, 相关的信道和算法参数设置如表4所示.

6.1 单阶段模拟

为了验证基于动态规划算法的收敛性, 我们首先选择了一个单阶段场景进行模拟, 该场景只包含一个

表 4 信道和算法参数设置

Table 4 Parameter settings of the wireless channel and algorithms

信道参数	数值	算法参数	数值
B	20 MHz	T	8
β	2.11	N	2
N_0	-100 dBm/Hz	ε	10^{-3}
$P_{\text{tx},i}$	0.1 W	N_{mpc}	10
p_{sh}	0.2	Δt	0.5 s

任务UAV和一个中继UAV. 地面基站位于 $[0, 0]$ 处, 中继UAV和任务UAV的起飞地点均设置在 $[1, 1]$. 任务UAV遵循的运动轨迹为 $\{[3, 4], [5, 7], [8, 9], [11, 11], [14, 13]\}$. 使用CMMP-AT算法对该场景进行求解, 并记录了算法迭代的中间结果, 如图6所示. 图6(a)~(f)展示了每9次迭代中继UAV的轨迹优化结果, 从图中可以看到算法在收敛前会尝试多种运动策略, 直到找到最优的轨迹. 在本场景中, 算法可以在第52步收敛, 收敛过程相对稳定, 如图7所示.

6.2 CMMP-OBO与CMMP-AT方法比较

本节将考虑一个涉及多个阶段和多个中继UAV的较为复杂的场景. 为了验证不同中继策略的有效性, 分别基于CMMP-OBO方法和CMMP-AT方法对该场景下的中继UAV的最优轨迹展开求解. 同样假设在初始阶段, 所有UAV均位于 $[1, 1]$ 位置, 任务UAV的运动轨迹为 $\{[3, 4], [5, 7], [8, 9], [11, 11], [14, 13], [16, 16], [19, 14], [22, 16]\}$.

对于CMMP-OBO方法, 首先需确定阈值距离, 以实现阶段的划分. 阈值距离的选择需要综合考虑计算时间、平均信道容量、运动能耗等多个因素, 在多个指标中进行折中. 例如, 获得较高平均信道容量的同时将消耗较多的计算时间, 该部分将在第6.3节中进行详细讨论. 在本节实验中, 综合考虑计算时间和平均信道容量等因素, 设置阈值距离为20 m. 依据该设置, 任务UAV的轨迹可划分为两个阶段(第一阶段包含5个时间步, 第二阶段包含3个时间步). 当任务UAV和基站之间的距离超过20 m时, 新的中继UAV将被添加到系统中. 使用CMMP-OBO方法对该场景的多中继UAV的

图 6 (网络版彩图)算法迭代过程

Figure 6 (Color online) Algorithmic iterative process.

图7 (网络版彩图)单阶段模拟中值函数的收敛过程

Figure 7 (Color online) The convergence process of the value function in the single-stage simulation.

最优轨迹展开求解, 结果如图8所示. 从轨迹图中我们可以看到, 两个中继UAV随着任务UAV的运动不断调整位置, 以满足任务UAV与基站之间的通信质量要求. 第一阶段和第二阶段的价值函数收敛过程分别如图7和9所示.

在另一方面, CMMP-AT方法主张将所有中继UAV在初始阶段就全部部署到系统中, 因此, 不用考虑阶段的划分. 图10展示了使用CMMP-AT方法求出的中继UAV的最优轨迹. 该方法比较简便, 但是在初

图8 (网络版彩图)中继UAV的优化轨迹(CMMP-OBO方法)

Figure 8 (Color online) The optimized trajectories of the relay UAVs (CMMP-OBO).

图9 (网络版彩图)第二阶段值函数的收敛过程

Figure 9 (Color online) The convergence process of the value function in the second stage.

图10 (网络版彩图)中继UAV的优化轨迹(CMMP-AT方法)

Figure 10 (Color online) The optimized trajectories of the relay UAVs (CMMP-AT).

始阶段就将所有中继UAV部署到系统中也会面临一些缺点, 例如, 较高的运动能耗和计算时间复杂度等. 为了比较CMMP-AT方法和CMMP-OBO方法的运行效果, 我们对比了若干性能指标, 如计算时间、平均信道容量、运动能耗和收敛迭代次数, 对比结果如表5所示.

从表中可以看出, 由于采用状态空间压缩技术, CMMP-OBO方法仅消耗57.66 s和58次迭代(包括两个阶段迭代次数)就得到了解, 比CMMP-AT求解方法要快得多(40.03 h和117次迭代). 这是因为在本实验中,

表 5 CMMP-OBO方法与CMMP-AT方法的指标比较**Table 5** The metric comparison between CMMP-OBO and CMMP-AT

方法	计算时间	平均信道容量 (Mbps)	运动能耗	收敛 迭代次数
CMMP-OBO	57.66 s	6.65	$24.85 \text{ m} \times k_m$	58
CMMP-AT	40.03 h	6.49	$25.85 \text{ m} \times k_m$	117

未压缩前, CMMP-AT方法的每次迭代需遍历的状态空间大小为 $21 \times 15 \times 21 \times 15=99225$, 而在进行状态空间压缩后, CMMP-OBO方法在第一阶段和第二阶段每次迭代需遍历的状态空间大小分别为 $10 \times 10=100$ 和 $6 \times 6 \times 6 \times 6=1296$, 该数值远远小于CMMP-AT方法. 并且, 由于CMMP-AT方法状态空间的维度较大, 因此将会耗费更多的迭代次数才能够收敛, 如图11所示. 此外, 在运动能耗方面, 本文设UAV的运动能耗与运动距离成正比, 并设比例常数为 k_m ^[20]. 依据此设定可计算出, 采用CMMP-OBO方法, 中继UAV将比CMMP-AT方法少消耗3.87%的能量, 这将延长该多UAV系统的任务执行时间.

在平均信道容量方面, CMMP-OBO方法同样要优于CMMP-AT方法. 这是因为依据式(3), 单步信道容量除了和每跳信道容量相关外, 还与中继UAV的架数有关. 当任务UAV距离基站不远时, 此时两种方法的单跳信道容量的差距并不大, 而多引入一架中继UAV, 将会使式(3)的分母变大, 从而降低单步信道容量. 从

图 11 (网络版彩图)CMMP-AT方法的价值函数收敛过程**Figure 11** (Color online) The convergence process of the value function of CMMP-AT. **图 12** (网络版彩图)两种方法的单步信道容量比较**Figure 12** (Color online) The single-step channel capacity comparison of the two methods.

图12的单步信道容量对比也可以看到, 在前两个时间步, 部署2架中继UAV(CMMP-AT方法)获得的单步信道容量反而比部署单架中继UAV(CMMP-OBO方法)还要低.

6.3 阈值距离选择

对于CMMP-OBO方法, 如何定义触发新阶段的阈值距离是一个开放性的问题. 分别设置阈值距离 $r_{th}=0, 18, 20, 23, 27 \text{ m}$, 并比较所对应的计算时间、平均信道容量、运动能耗和收敛迭代次数, 结果如表6所示. 其中, 收敛迭代次数包括了第一阶段和第二阶段两个部分. 考虑到第二阶段会引入新的中继UAV, 因此, 第二阶段每次迭代的时间均会长于第一阶段, 这就是为什么阈值距离小的情况下, 迭代次数较少, 但计算时间还较长. 随着阈值距离的减小, 第二阶段会涉及更多时间步的优化, 也会使每次迭代的计算时间更长. 从表中也可以看出, 随着阈值距离的增加, 平均信道容量和运动能耗会先增加后减小, 对应的计算时间则会持续减小. 因此, 在解决实际问题时, 我们往往需要在获取较高的信道容量与节约额外的计算和运动开销之间做出权衡.

6.4 基于V-REP的轨迹追踪实现

为了验证MPC方法在实际系统中的运行效果, 本

表 6 不同阈值距离配置下的各指标比较

Table 6 The metric comparison in different threshold distance settings

阈值距离 (m)	计算时间	平均信道容量 (Mbps)	运动能耗	收敛 迭代次数
0	40.03 h	6.49	$25.85 \text{ m} \times k_m$	117
18	679.65 s	6.74	$27.09 \text{ m} \times k_m$	60
20	57.66 s	6.65	$24.85 \text{ m} \times k_m$	58
23	7.73 s	6.61	$22.02 \text{ m} \times k_m$	68
27	1.19 s	6.51	$17.78 \text{ m} \times k_m$	80

节将基于V-REP仿真器进行系统实现,并以图8和10所示的场景为例进行演示验证。另外,为了对比效果,本文选取PID控制方法作为对比案例。在每次仿真中,所有加入到系统的UAV均先上升到指定高度(5 m)后,再向指定位点行进。由于每个时间步中,不同UAV的飞行距离不同,因此可能出现先后到达指定位点的情况。为此,规定中继UAV到达指定位点后,在通信链路上向任务UAV发布到达位点消息,待任务UAV收到所有中继UAV的消息后,则确定所有UAV均达到指定位点。接着,由任务UAV向所有中继UAV发送向下一位点行进的指令,重复该过程直到最后一个位点。

图13和14分别展示了使用PID控制方法和MPC控制方法对图8所示场景进行的仿真过程。其中折线表示UAV的规划轨迹和飞行轨迹。从图中可以看到,相比PID控制方法,MPC控制方法能够更精确地追踪规划的轨迹。这是因为,MPC控制方法可通过模型预测转弯点,从而提前实现减速,进而实现平稳转弯。而PID

图 13 (网络版彩图)使用PID控制方法获得的UAV的飞行轨迹(图8场景)

Figure 13 (Color online) The flight trajectories calculated by PID control method (the scenario in Figure 8).

图 14 (网络版彩图)使用MPC控制方法获得的UAV的飞行轨迹(图8场景)

Figure 14 (Color online) The flight trajectories calculated by MPC control method (the scenario in Figure 8).

控制方法是依据当前位置反馈,当UAV到达转弯点时无法瞬间减速,从而出现“冲出”转弯点的情况。

表7给出了两种控制方法的轨迹追踪误差。可以看到,使用MPC控制方法,任务UAV、中继UAV1和中继UAV2的轨迹追踪的误差都小于PID控制方法。并且,MPC控制方法的平均轨迹追踪误差只有PID控制方法的43.97%。

针对图10所示的场景,同样使用两种控制方法进行仿真,结果如图15和16所示。其中,各折线的意义与图13和14相同。从图中可以看出,MPC控制方法相比PID控制方法,同样可以获得更精确的追踪轨迹。从表4也可看出,使用MPC控制方法,各UAV的轨迹追踪误差都小于PID控制方法,且各UAV的平均追踪误差为PID控制方法的41.42%。

7 结论

本文针对任务无人机与地面基站的通信连通性保持问题展开研究,基于动态规划理论,提出了两种多无人机中继规划方法:CMMP-AT和CMMP-OBO。两种规划方法都以系统的长期信道容量为优化目标,CMMP-AT方法考虑在初始时刻就部署全部中继无人机,该方法部署简便,但计算时间较长;CMMP-OBO方法则从计算的实时性和节约能耗角度出发,采用增量式部署的方式规划中继无人机的行动。实验结果表明,与CMMP-AT方法相比,CMMP-OBO方法可有效提升规划的效率,并且节约中继无人机的能耗。为了保证规划的轨迹能够被精确执行,本文基于模型预测控制方

表 7 PID控制方法和MPC控制方法的轨迹追踪误差对比(单位: m)

Table 7 The tracking error comparison between PID and MPC (unit: m)

控制方法	图8场景仿真				图10场景仿真			
	任务UAV	中继UAV1	中继UAV2	平均误差	任务UAV	中继UAV1	中继UAV2	平均误差
PID	0.0917	0.0770	0.0502	0.0730	0.0784	0.1204	0.1301	0.1096
MPC	0.0321	0.0307	0.0335	0.0321	0.0285	0.0581	0.0497	0.0454

图 15 (网络版彩图)使用PID控制方法获得的UAV的飞行轨迹(图10场景)

Figure 15 (Color online) The flight trajectories calculated by PID control method (the scenario in Figure 10).

图 16 (网络版彩图)使用MPC控制方法获得的UAV的飞行轨迹(图10场景)

Figure 16 (Color online) The flight trajectories calculated by MPC control method (the scenario in Figure 10).

法实现中继无人机对规划轨迹的精确追踪, 并基于V-REP仿真环境进行了系统实现. 实验结果表明, 相比于

PID控制方法, 模型预测控制方法可更为精确地实现轨迹的追踪.

参考文献

- 1 Oosedo A, Abiko S, Narasaki S, et al. Flight control systems of a quad tilt rotor unmanned aerial vehicle for a large attitude change. In: 2015 IEEE International Conference on Robotics and Automation (ICRA). Seattle: IEEE, 2015. 2326–2331
- 2 Ponda S S, Johnson L B, Kopeikin A N, et al. Distributed planning strategies to ensure network connectivity for dynamic heterogeneous teams. *IEEE J Sel Areas Commun*, 2012, 30: 861–869
- 3 Cesare K, Skeeel R, Yoo S H, et al. Multi-UAV exploration with limited communication and battery. In: 2015 IEEE International Conference on Robotics and Automation (ICRA). Seattle: IEEE, 2015. 2326–2331
- 4 Faigl J, Vána P. Surveillance planning with bezier curves. *IEEE Robot Autom Lett*, 2018, 3: 750–757
- 5 Omidshafiei S, Agha-Mohammadi A, Amato C, et al. Decentralized control of multi-robot partially observable Markov decision processes using belief space macro-actions. *Int J Robotics Res*, 2017, 36: 231–258
- 6 Malmirchegini M, Mostofi Y. On the spatial predictability of communication channels. *IEEE Trans Wireless Commun*, 2012, 11: 964–978
- 7 Yan Y, Mostofi Y. Co-optimization of communication and motion planning of a robotic operation under resource constraints and in fading environments. *IEEE Trans Wireless Commun*, 2013, 12: 1562–1572
- 8 Yan Y, Mostofi Y. Robotic router formation in realistic communication environments. *IEEE Trans Robot*, 2012, 28: 810–827
- 9 Rooker M N, Birk A. Multi-robot exploration under the constraints of wireless networking. *Control Eng Practice*, 2007, 15: 435–445
- 10 Lv S, Wang X, Zhou X. On the rate adaptation for IEEE 802.11 wireless networks. *Comput Networks*, 2010, 54: 3173–3186
- 11 Zavlanos M M, Egerstedt M B, Pappas G J. Graph-theoretic connectivity control of mobile robot networks. *Proc IEEE*, 2011, 99: 1525–1540
- 12 Ji M, Egerstedt M. Distributed coordination control of multiagent systems while preserving connectedness. *IEEE Trans Robot*, 2007, 23: 693–703
- 13 Zavlanos M M, Pappas G J. Distributed connectivity control of mobile networks. *IEEE Trans Robot*, 2008, 24: 1416–1428

- 14 Olfati-Saber R, Fax J A, Murray R M. Consensus and cooperation in networked multi-agent systems. *Proc IEEE*, 2007, 95: 215–233
- 15 Yan Y, Mostofi Y. Communication and path planning strategies of a robotic coverage operation. In: 2013 American Control Conference (ACC). Washington DC: IEEE, 2013. 860–866
- 16 Ali U, Yan Y, Mostofi Y. An optimal control approach for communication and motion co-optimization in realistic fading environments. In: 2015 American Control Conference (ACC). Chicago: IEEE, 2015. 2930–2935
- 17 McGrew J S, How J P, Williams B, et al. Air-combat strategy using approximate dynamic programming. *J Guid Control Dyn*, 2010, 33: 1641–1654
- 18 Mozaffari M, Saad W, Bennis M, et al. Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs. *IEEE Trans Wireless Commun*, 2016, 15: 3949–3963
- 19 Goldsmith A. *Wireless Communications*. Cambridge: Cambridge University Press, 2005
- 20 Wei M, Isler V. Coverage path planning under the energy constraint. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). Brisbane: IEEE, 2018. 368–373

Research on multi-UAV communication connectivity maintenance based on dynamic programming

WU YunLong^{1,2}, ZHANG Bo¹, REN XiaoGuang^{1,2}, WANG YanZhen^{1,2} & YI XiaoDong¹

¹ National Innovation Institute of Defense Technology (NIIDT), People's Liberation Army (PLA) Academy of Military Science, Beijing 100071, China;

² Tianjin Artificial Intelligence Innovation Center (TAIIC), Tianjin 300457, China

With the continuous development of unmanned aerial vehicle (UAV) technology and its falling cost, UAV has been widely used in important fields, such as agriculture, industry, security, and military. In operation, UAVs often need to maintain communication connectivity with ground stations. However, in the case of long distance or shadowing, the quality of the wireless channels will be seriously degraded, which will affect the communication connectivity. To solve this problem, this paper considers deploying multiple UAVs as communication relays. In this paper, we select the long-term channel capacity as the optimization objective and propose two planning methods for UAV relays based on dynamic programming (DP): CMMP-AT and CMMP-OBO. The CMMP-AT method emphasizes the deployment of all relay UAVs at the beginning of the task. This method is simpler to implement, but the computational complexity is intolerable. The CMMP-OBO method proposes to deploy relay UAVs one by one according to the task requirements. Although the CMMP-OBO method may lose performance, it has a lower computational complexity and better scalability and can save more motion energy. The experimental results reveal that the CMMP-AT method requires 40.03 h to obtain the planning results for a two-relay scenario, whereas the CMMP-OBO method only needs 57.66 s and can save 3.87% motion energy. In addition, to precisely control the relay UAVs to follow the planned trajectory, this paper implements trajectory tracking, based on model predictive control (MPC), and realizes the simulation of the multi-relay UAV scenario in the VREP environment. The simulation results reveal that compared with the PID control method, the MPC method can track the planned trajectory more accurately. In two simulation scenarios, the tracking error of MPC is only 43.97% and 41.42% of the PID control method.

communication connectivity, dynamic programming, multi-UAV system, communication relay, model predictive control

doi: [10.1360/SST-2019-0291](https://doi.org/10.1360/SST-2019-0291)