

空天地一体化网络的无人机轨迹和计算卸载联合优化

陈奕天, 童英华

引用本文

陈奕天, 童英华. 空天地一体化网络的无人机轨迹和计算卸载联合优化[J]. 计算机科学, 2025, 52(4): 74-84.

CHEN Yitian, TONG Yinghua. [Joint Optimization of UAV Trajectories and Computational Offloading for Space-Air-Ground Integrated Networks](#) [J]. Computer Science, 2025, 52(4): 74-84.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[面向无人机航拍图像的YOLOv8目标检测改进算法](#)

An Improved YOLOv8 Object Detection Algorithm for UAV Aerial Images

计算机科学, 2025, 52(4): 202-211. <https://doi.org/10.11896/jsjcx.240500042>

[基于类脑脉冲神经网络的边缘联邦持续学习方法](#)

Edge-side Federated Continuous Learning Method Based on Brain-like Spiking Neural Networks

计算机科学, 2025, 52(3): 326-337. <https://doi.org/10.11896/jsjcx.240900070>

[基于数字孪生的卫星网络移动边缘计算研究综述](#)

Review of Digital Twin Based Satellite Network Mobile Edge Computing

计算机科学, 2024, 51(12): 12-19. <https://doi.org/10.11896/jsjcx.240700046>

[基于自供电无人机远距离中继通信与计算卸载策略优化研究](#)

Study on Optimization of Long-distance Relay Communication and Computational Offloading Strategy Based on Self-powered UAVs

计算机科学, 2024, 51(11A): 240300069-7. <https://doi.org/10.11896/jsjcx.240300069>

[基于深度强化学习的云边协同任务迁移与资源再分配优化研究](#)

Cloud-Edge Collaborative Task Transfer and Resource Reallocation Optimization Based on Deep Reinforcement Learning

计算机科学, 2024, 51(11A): 231100170-10. <https://doi.org/10.11896/jsjcx.231100170>

空天地一体化网络的无人机轨迹和计算卸载联合优化

陈奕天¹ 童英华^{1,2}

1 青海师范大学计算机学院 西宁 810008

2 青海师范大学省部共建藏语智能信息处理及应用国家重点实验室 西宁 810008

(1244339502@qq.com)

摘要 空天地一体化网络作为一种新兴的网络架构,近年来引起了广大研究者的关注,它能够很好地提高网络整体的服务质量。针对偏远地区的网络覆盖不全面,缺乏基本的网络基础设施的问题,提出了一种无人机和卫星共同收集任务的偏远地区空天地一体化网络框架,其中无人机与卫星为地面传感器提供边缘计算服务,云服务器为地面传感器提供云服务。由于无人机覆盖率、任务完成率和任务延迟都是影响系统性能的关键因素,因此对无人机轨迹和计算卸载进行联合优化,最大化无人机覆盖率和任务完成率,并且降低延迟。由于所提出的联合优化问题是一个混合非线性规划问题,因此设计了基于白鲸优化算法和沙猫群优化算法的双层优化算法,两层分别对无人机轨迹和计算卸载进行优化。实验结果表明,所提算法显著提高了多个无人机的覆盖率,且在计算卸载中有效提高了任务完成率,降低了任务的平均延迟。

关键词 空天地一体化网络;无人机;边缘计算;无人机轨迹;计算卸载

中图分类号 TP393

Joint Optimization of UAV Trajectories and Computational Offloading for Space-Air-Ground Integrated Networks

CHEN Yitian¹ and TONG Yinghua^{1,2}

1 School of Computer, Qinghai Normal University, Xining 810008, China

2 State Key Laboratory of Tibetan Intelligent Information Processing and Application, Qinghai Normal University, Xining 810008, China

Abstract As an emerging network architecture, space-air-ground integrated network has attracted significant attention from researchers in recent years, and it can greatly improve the overall quality of service. Addressing the challenges of insufficient network coverage and the lack of basic infrastructure in remote areas, a space-air-ground integrated network framework is proposed in which unmanned aerial vehicles (UAVs) and satellites collaboratively collect tasks. In this framework, UAVs and satellites provide edge computing services for ground sensors, while cloud servers deliver cloud services. Given that UAV coverage, task completion rate, and task latency are critical factors influencing system performance, this study jointly optimizes UAV trajectory and computation offloading to maximize UAV coverage and task completion rate while minimizing latency. The proposed joint optimization problem is formulated as a mixed-integer nonlinear programming problem, therefore, a dual-layer optimization algorithm based on the Beluga Whale Optimization and Sand Cat Swarm Optimization is developed, with the two layers separately optimizing UAV trajectory and computation offloading. Experimental results show that the proposed algorithm significantly improves the coverage rate of multiple UAVs, effectively enhances the task completion rate, and reduces average task latency in computation offloading.

Keywords SAGIN; UAV; Edge computing; UAV trajectories; Computational offloading

1 引言

空天地一体化网络 (Space-Air-Ground Integrated Network, SAGIN) 的出现为满足各种超密集异构地面传感器网络不断增加的应用需求迎来了转机。一般来说, SAGIN 可分为 3 个部分, 即空间网络、空中网络和地面网络。其可为大面积

区域提供无缝的、灵活的网络覆盖以及服务^[1], 因此可以在偏远地区利用 SAGIN 为大量的传感器设备提供网络通信服务。同时, 移动边缘计算 (Mobile-Edge Computing, MEC) 是提高服务质量 (QoS) 和网络性能的一种有效方法^[2], 可以在偏远地区提供可靠的通信服务, 因此在 SAGIN 的空中网络, 无人机 (Unmanned Aerial Vehicle, UAV) 可作为飞行边缘

到稿日期: 2024-10-20 返修日期: 2025-01-27

基金项目: 青海省应用基础研究项目 (2023-ZJ-713)

This work was supported by the Qinghai Province Application Basic Research Program (2023-ZJ-713).

通信作者: 童英华 (tyh_angel@126.com)

节点,为地面传感器提供远程无线接入和低延迟边缘计算服务^[3-6]。此外,空间网络中的卫星依靠无缝网络覆盖的优势,不仅可以为地面设备提供边缘计算服务也可以提供无处不在的云计算服务^[7]。与固定地面网络相比,SAGIN包括了卫星、无人机等空间和天空网络,具有通用性强、可机动部署、无缝覆盖等优势,成为新兴的研究热点^[8]。

考虑到在偏远地区还有许多恶劣的网络环境,研究如何提供低延迟、高可靠性的移动边缘计算系统至关重要^[9]。为获得更好的服务质量,首先要为用户提供完整的服务。目前的研究大多使用队列对计算任务进行处理,这样难免会造成队列满,新收集的任务无法入队,从而丢弃任务的情况^[10]。由于是被迫丢弃任务,一些重要的任务也有可能被丢弃,这极大地影响了SAGIN的服务质量,因此提高所有计算任务的整体完成率也是一个值得关注的问题。为了提高计算任务的完成率,首先要确保计算任务能够被收集,但由于偏远地区幅员辽阔,无人机的覆盖范围有限,因此必须考虑无人机轨迹的优化问题。

为了提高无人机的自主水平和运行效率,利用优化算法生成无人机的飞行轨迹已成为一项关键技术^[11]。一般来说,无人机的路径或轨迹规划可视为一个优化问题,其目的是在特定优化目标下,找到从起始位置到目的地位置的最优或接近最优的解决方案。主要挑战是避免与环境障碍物以及其他成员发生碰撞,并为跟踪控制系统提供有效的指导。提高无人机的覆盖范围,最终目标是提高无人机服务的地面传感器数量,使之能够更好地在计算卸载中提高任务完成率和降低任务延迟。

本文的研究目标是:在偏远地区,使用SAGIN技术来构建网络框架,在保证计算任务完成率的同时,尽可能降低计算任务的延迟。本文构建了联合优化无人机轨迹和计算卸载的优化模型,为解决此多目标问题,对问题进行了详细的分析,提出了基于白鲸优化(Beluga Whale Optimization, BWO)算法和沙猫群优化(Sand Cat Swarm Optimization, SCSO)算法的BW-SCS双层优化算法进行求解。本文的主要贡献如下:

1)设计了一种卫星和无人机共同收集计算任务的SAGIN框架,并且考虑了此框架中多设备的动态计算卸载。

2)提出了提高计算任务完成率和降低计算任务延迟的优化问题,将此问题拆解成无人机轨迹优化和计算卸载优化两个子问题。一是通过优化无人机轨迹来提高无人机覆盖率,增强无人机收集计算任务的能力,减轻卫星收集计算任务的压力;二是通过优化无人机的卸载决策,来提高任务完成率和降低任务延迟。

3)提出了BW-SCS双层优化算法,使用BWO算法对无人机轨迹进行优化求解,然后通过SCSO算法对卸载决策优化求解。对算法进行实验仿真,结果表明该算法极大地优化了无人机的轨迹,提高了覆盖率,并且对于提高任务完成率和降低任务延迟也有显著的成效。

本文第2章对近些年的无人机轨迹研究和SAGIN计算卸载相关的研究进行了分析;第3章提出了适用于偏远地区环境的SAGIN的MEC系统模型并对求解问题进行了分析;第4章对求解问题进行拆解,设计了BW-SCS双层优化算法

对问题进行求解;第5章对实验结果进行了分析;最后总结全文并展望未来。

2 相关研究

近年来,空天地一体化网络的研究逐渐进入研究人员的视野中,接下来对无人机轨迹和SAGIN中计算卸载的相关研究进行回顾。

2.1 无人机轨迹研究现状

Mao等^[12]通过联合无人机计算资源和部署位置、计算任务分配、关联控制、传出功率和带宽分配,最大限度地减少物联网设备之间的最大计算延迟;利用块坐标下降和连续凸近似,开发了一种保证收敛的交替优化算法,但优化重点是无人机位置而非轨迹。Xiong等^[13]考虑无人机在短时间内围绕该点周期性地小范围移动,并研究了物联网移动设备和无人机之间的计算卸载问题,对传输比特分配和无人机轨迹进行了联合优化,但研究局限在小范围环境。Tang等^[14]提出了混沌麻雀搜索算法对单无人机轨迹进行优化,并在一个战场环境对算法进行仿真实验,结果表明该方法能够快速得到一条满足约束条件且安全的飞行轨迹。Han等^[15]继续对无人机的轨迹规划问题进行研究,提出了螺旋自适应麻雀搜索算法。仿真结果表明,不管是对单无人机还是对多无人机进行轨迹优化,该算法均有效。Pervez等^[16]提出了一个用户关联、传输功率和无人机飞行轨迹的联合优化问题,通过分解这个迭代问题并依次求解,实现了用户平均吞吐量的优化。Li等^[17]探讨了无人机轨迹和发射功率的联合优化,通过问题分解和连续凸优化,提高了无人机系统的整体性能。Jia等^[18]针对6G空中接入网络,提出了基于无人机的SAGIN方案,重点关注在能源限制下降低无人机总能耗,提出了无人机轨迹和数据传输的联合优化问题,使用Dantzig-Wolfe分解法,并设计基于列生成的算法来解决问题,实验验证了低地轨道辅助无人机轨迹设计和数据传输的优势。

偏远地区地形复杂且面积广阔,为了提高无人机覆盖率,必须对无人机的飞行轨迹进行优化,而不仅仅是对无人机位置进行优化。然而在SAGIN中,无人机轨迹优化大多是与计算卸载优化放在同一个算法中一起联合优化,这显然会增加解的维度,降低解的精度。

2.2 SAGIN中计算卸载的研究现状

Zhou等^[19]设计了面向延迟的物联网任务调度(delay-Orientated IoT Task Scheduling, DOTS)的计算任务调度方案,在无人机上设置计算队列和卸载队列;用深度强化学习算法寻找最优卸载决策,以减少延迟和能耗。但是该方法没有考虑等待延迟、多无人机以及任务丢弃问题。Cheng等^[20]提出了一种联合资源分配和任务调度方法,使用基于行动者-评论家(Actor-Critic, AC)算法,从动态的环境中学习最优卸载策略,实现了低成本以及快速收敛。Li等^[21]研究了多无人机的计算卸载问题,采用多智能体强化学习算法优化延迟和能耗,但仍存在队列满会自动丢弃任务的问题。Xu等^[22]针对UAV部署、地面设备(GD)接入和计算卸载制定了一个联合优化问题,提出了一种基于粒子群优化(PSO)和贪婪策略(GS)的算法来优化系统平均响应延迟,但文中并没有考虑

SAGIN 边缘系统中的资源分配问题。Wei 等^[23]选择在计算能力更强的卫星物联网云节点集群上部署深度学习框架。它根据卫星的状态选择部分处理或卸载到附近的卫星上进行协同处理。Huang 等^[24]提出了一种基于深度强化学习 (Deep Reinforcement Learning, DRL) 的新型框架,用于优化 SAGINs 内混合云和多接入边缘计算环境中的计算卸载和资源分配。通过整合多个卫星、云服务器和无人机,该系统有效地降低了能耗和延迟,在动态和复杂的通信场景中展现出卓越的性能和实际应用潜力。Gao 等^[25]将无人机作为控制层的角色,利用毫米波雷达和视觉传感器收集多源数据,以减少 SAGIN 的不确定性并将任务分配给地面基站和卫星。但是无人机的计算资源有限,可能会影响其处理大规模任务的能力;同时,该方法没有优化无人机的飞行轨迹,这也会影响计算卸载的效率。

在进行计算卸载的优化时,之前的研究大多是在优化无人机的卸载决策,其优化目标是降低延迟和能耗,忽略了任务的完成率这一关键问题。虽然这些方法在延迟与能耗方面确有提升,但能否保证任务在计算周期内完成是未来值得思考的问题。

3 系统模型

3.1 SAGIN 模型

由于偏远地区环境中没有蜂窝网络覆盖,因此使用 SAGIN 为所有的地面传感器提供服务,如图 1 所示。

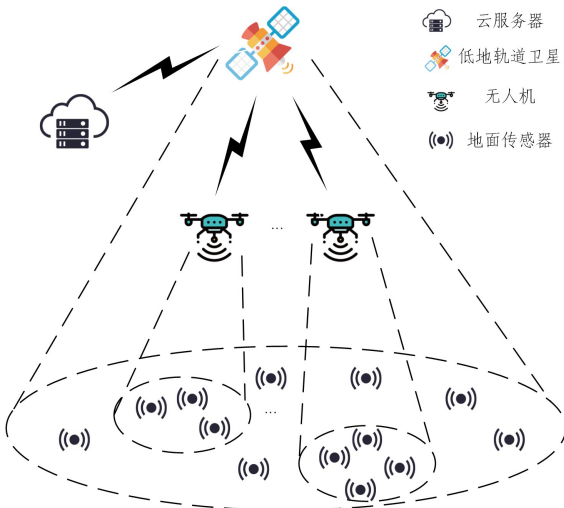


图1 SAGIN 框架

Fig. 1 SAGIN architecture

本文提出的无人机和卫星合作收集计算任务的 SAGIN 框架共分为 3 层:地面层、天空层和空间层。其中,地面层包含各种传感器设备和一个云服务器,要注意的是云服务器是设置在地面层的,传感器设备的集合用 S 表示, $S = \{0, 1, 2, \dots, S\}$, 传感器的位置固定不变,云服务器用 cl 表示;天空层中有若干个无人机,用 U 表示, $U = \{0, 1, 2, \dots, U\}$, 在无人机覆盖范围内的传感器直接与无人机进行通信,每个无人机在自己的轨迹路线上收集传感器产生的计算任务,由于无人机的计算能力有限,因此需要无人机选择合适的卸载策略将计算任务向卫星进行卸载;空间层包含一个近地轨道 (Low

Earth Orbit, LEO) 卫星,用 sa 表示,卫星的通信范围可以覆盖整个偏远地区中的无人机和传感器,不在无人机服务范围内的地面传感器由卫星提供通信服务,且卫星可与云服务器进行通信。在此框架中采用的是一个离散的时隙系统 T , T 代表无人机的整个飞行周期,将 T 分为 t 个时隙,每一个时隙时间相等,无人机、卫星和云服务器在每个时隙下都可以提供服务。

3.2 通信模型

由于无人机和卫星使用不同的频段进行通信,因此假设无人机和卫星之间不存在干扰^[26]。同时,由于无人机与传感器的距离足够近,因此忽略了从传感器到无人机的传播延迟^[27]。空中到地面的通信信道取决于高度、仰角和传播环境^[28]。根据文献[29],空中到地面信道的平均路径损耗可定义为:

$$P^{Loss}(r, h) = 20 \log \left(\frac{4\pi f_c (h^2 + r^2)^{\frac{1}{2}}}{c} \right) + P^{Los} \eta^{Los} + (1 - P^{Los}) \eta^{NLos} \quad (1)$$

其中, P^{Los} 表示传感器与无人机之间的视线 (Line of Sight, Los) 概率; $h, r, \eta^{Los}, \eta^{NLos}$ 分别表示无人机的飞行高度、无人机与传感器之间的水平距离以及 Los 链路和非 Los 链路在自由空间路径损耗基础上产生的附加损耗^[30]; f_c 表示载波频率; c 表示光速。根据文献[20],偏远地区的 $(\eta^{Los}, \eta^{NLos})$ 值为 $(0.1, 2.1)$ 。

由于卫星与无人机之间的距离远大于无人机移动的距离,因此无人机的移动速度不影响其与卫星的距离。基于 Weibull 信道模型的研究结论^[31],无人机和地面设备到卫星的信道增益计算式为:

$$G_{sa} = \frac{G_t G_r \lambda^2}{4\pi d^2} 10^{-\frac{F^{rain}}{10}} \quad (2)$$

其中, G_t 和 G_r 分别是发射器和接收器的天线增益, λ 是传播信号的波长, d 是卫星到地面设备的距离, F^{rain} 是遵循基于 Weibull 信道模型的雨衰减分布。

由式(1)求出的平均路径损耗 P^{Loss} 和式(2)求出的信道增益 G_{sa} ,可以得到无人机的最大传输速率 R_u 和卫星的最大传输速率 R_{sa} :

$$R_u = W \log_2 \left(1 + \frac{P_t \cdot 10^{\frac{P^{Loss}}{10}}}{\sigma^2} \right) \quad (3)$$

$$R_{sa} = W \log_2 \left(1 + \frac{P_t \cdot |G_{sa}|^2}{\sigma^2} \right) \quad (4)$$

其中, P_t 是设备的发射功率, σ^2 是高斯噪声功率, W 是设备的网络带宽。

3.3 计算模型

传感器收集数据生成的计算任务 $M = \{0, 1, 2, \dots, M\}$ 可以建模为一个元组 (φ, γ) , 其中 $\varphi \in [\varphi^{\min}, \varphi^{\max}]$ 是任务的数据量,单位为 bit, $\gamma \in [\gamma^{\min}, \gamma^{\max}]$ 是任务的计算复杂度。一个任务从收集到计算完成,它的延迟一共分为 3 个部分,分别为传输延迟、计算延迟和等待延迟,其中传输延迟指数据从发送方全部传输到接收方的时间,计算延迟指开始对计算任务处理到处理结束的时间,等待延迟指计算任务在队列中排队等待传输或者排队等待计算的时间,第 m 个任务的总延迟为:

$$D_m = d_m^{\text{tra}} + d_m^{\text{com}} + d_m^{\text{wait}} \quad (5)$$

在每个时隙 t 下,卫星和各个无人机共同收集计算任务,如图 2 所示,无人机和卫星有两个队列,即计算队列、卸载队列,云服务器只有计算队列。无人机会将收集的计算任务放在计算队列中,然后依据时隙 t 下的无人机卸载率 $\alpha_u(t) \in [0, 1], t \in T$,依比率将任务卸载。若不卸载,则任务在计算队列中直接进行处理;若卸载,则将任务转移到卸载队列,卸载到卫星。而卫星将收集到的传感器计算任务和无人机卸载任务放入卫星的计算队列,并依据卫星在时隙 t 下的卸载率 $\alpha_s(t) \in [0, 1], t \in T$,选择任务是否卸载到云服务器。若卸载,则将任务转移到卫星的卸载队列,然后传输到云服务器的计算队列,由云服务器处理,否则将计算队列中的任务置于本地处理。

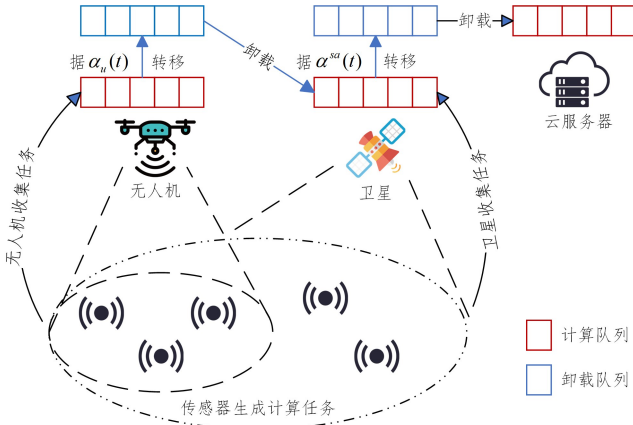


图 2 SAGIN 中的计算卸载示意图

Fig. 2 Schematic diagram of computational offloading in SAGIN

1) 传输延迟:由于无人机与传感器的距离非常近,所产生的传播延迟很小,可忽略不计^[32],因此第 m 个计算任务与无人机之间的传输延迟可以表示为:

$$d_m^{\text{tra}_u} = \frac{\varphi_m}{R_u} \quad (6)$$

但是,卫星与传感器、无人机还有云服务器的距离很远,产生的传播延迟无法忽略不计,因此传感器、无人机还有云服务器与卫星进行任务传输的传输延迟应该加上传播延迟^[33]。故第 m 个计算任务与卫星之间的传输延迟应表示为:

$$d_m^{\text{tra}_s} = \frac{\varphi_m}{R_{sa}} + d_{sa} \quad (7)$$

其中, d_{sa} 为卫星与 SAGIN 天空层以及地面层设备之间的传播延迟。

2) 计算延迟:无人机、卫星和云服务器都会将获得的计算任务按顺序放入计算队列中,按照先进先出的策略依次对任务进行处理。它们的计算延迟的计算式如下:

$$d_m^{\text{com}} = \frac{\varphi_m \gamma_m}{f} \quad (8)$$

其中, f 是计算设备 CPU 的计算能力。

3) 等待延迟:无人机、卫星和云服务器不管是对任务进行处理还是对任务进行卸载,都是在队列之中进行的,因此任务在队列中要进行排队,这不可避免地产生了任务的等待延迟。第 m 个任务在队列中的等待延迟可以表示为:

$$d_m^{\text{wait}} = \sum_{i=1}^{m-1} (\alpha \times d_i^{\text{tra}} + \beta \times d_i^{\text{com}}) \quad (9)$$

其中, α 与 β 是一对相反的布尔值,当 $\alpha=0$ 时 $\beta=1$,当 $\alpha=1$ 时 $\beta=0$ 。 d_i^{tra} 和 d_i^{com} 分别是第 i 个任务的传输延迟和计算延迟。

3.4 无人机轨迹模型

假设无人机在每个时隙 t 下的位置不变且无人机的飞行高度固定为 h ,此外每一架无人机都以匀速飞行。 $pos_u(t) = (x_u(t), y_u(t), h)$ 表示在 t 时隙第 u 个无人机的位置,其中 $t \in \{1, 2, \dots, T\}, u \in \{1, 2, \dots, U\}$, $pos_s = (x_s, y_s)$ 表示第 s 个传感器的坐标。无人机在每个时隙 t 后会前往下一个坐标,从而进入下一个时隙 t ,因为无人机在移动时是水平的,所以无人机的坐标更新公式为:

$$\begin{cases} x_u(t+1) = x_u(t) + dis_u^{\text{fly}} \times \cos \theta_u \\ y_u(t+1) = y_u(t) + dis_u^{\text{fly}} \times \sin \theta_u \end{cases} \quad (10)$$

其中, θ_u 是第 u 个无人机的飞行角度, dis_u^{fly} 是第 u 个无人机的飞行距离, dis_u^{fly} 的表达式为:

$$dis_u^{\text{fly}} = v_u \times t \quad (11)$$

其中, v_u 是第 u 个无人机的飞行速度。当无人机与传感器的距离小于无人机的最大覆盖半径时,即 $dis_{(s,u)} = \sqrt{(x_u - x_s)^2 + (y_u - y_s)^2 + h^2} \leq R^{\text{max}}$ (其中 R^{max} 是无人机的最大覆盖半径),无人机即可对此传感器进行通信服务。

考虑在真实环境中,难免会有一些高于无人机飞行高度的障碍物存在,而无人机是不允许与障碍物碰撞的,因此给出限制条件:无人机与障碍物的直线距离要小于障碍物的半径,即:

$$dis_{(u,obs)} < r_{obs} \quad (12)$$

其中, r_{obs} 是障碍物的半径。无人机与障碍物的直线距离的计算式如下:

$$dis_{(u,obs)} = \sqrt{(x_u - x_{obs})^2 + (y_u - y_{obs})^2} \quad (13)$$

其中, (x_{obs}, y_{obs}) 是障碍物 obs 的坐标。

为了保证无人机的覆盖率有切实的提高,各个无人机之间的覆盖范围不应该重叠,无人机与无人机之间的距离公式为:

$$dis_{(u_i, u_j)} = \sqrt{(x_{u_i} - x_{u_j})^2 + (y_{u_i} - y_{u_j})^2} \quad (14)$$

增加限制条件:无人机之间的覆盖范围不重叠。其计算式如下:

$$dis_{(u_i, u_j)} > r_{u_i} + r_{u_j} \quad (15)$$

当无人机经过整个飞行周期 T 的位置更新以后,会形成一个无人机的飞行轨迹,因此无人机的飞行轨迹就是由 t 个坐标点所组成的一个集合 $orbit = \{pos_u(1), pos_u(2), \dots, pos_u(t)\}$ 。在获得每个无人机的飞行轨迹之后,可以根据传感器的坐标计算出无人机在整个飞行周期中覆盖范围内的传感器个数 Cov^{num} ,从而求出所有无人机的总覆盖率,计算式如下:

$$Cov^{\text{rate}} = \frac{\sum_{i=1}^U \sum_{j=1}^t Cov^{\text{num}}}{S} \quad (16)$$

3.5 问题定义

在 SAGIN 系统中,需要联合优化无人机轨迹和计算卸载策略,通过提高无人机覆盖率,达到提高计算任务完成率和降低计算任务延迟的目的。任务完成率指在整个周期 T 中所有完成的任务数量与总任务数的比值:

$$F^{\text{rate}} = \frac{F^{\text{num}}}{M} \quad (17)$$

其中, F^{num} 是已完成的任务数量。在该系统中并不对整个 SAGIN 系统中的能耗问题进行优化, 我们假设网络中所有设备分配固定的网络带宽资源和 CPU 频率。在整个时间周期 T 内, 联合无人机覆盖率和计算卸载提高任务完成率和降低任务延迟的最优化问题可以表述为:

$$\begin{aligned}
 & \text{P1: } \sum_t (\max_{(Cov^{\text{rate}}, a)} F^{\text{rate}}, \min_{(Cov^{\text{rate}}, a)} D) \\
 & \text{s. t. C1: } a_u(t), a^{\text{sd}}(t) \in [0, 1], t \in T \\
 & \text{C2: } \varphi \in [\varphi^{\text{min}}, \varphi^{\text{max}}] \\
 & \text{C3: } \gamma \in [\gamma^{\text{min}}, \gamma^{\text{max}}] \\
 & \text{C4: } e < E \\
 & \text{C5: } dis_{(u, obs)}(t) < r_{\text{obs}}, t \in T \\
 & \text{C6: } dis_{(u_i, u_j)}(t) > r_{u_i} + r_{u_j}, t \in T \\
 & \text{C7: } dis_{(s, u)}(t) \leq R^{\text{max}}, t \in T
 \end{aligned} \tag{18}$$

其中, C1 限制无人机和卫星的卸载率在 $0 \sim 1$ 之间; C2 和 C3 表示计算任务数据量和计算复杂度的取值范围分别在 $\varphi^{\text{min}} \sim \varphi^{\text{max}}$ 和 $\gamma^{\text{min}} \sim \gamma^{\text{max}}$ 之间; C4 中 e 表示网络的整体能耗, E 为网络的最大能耗容忍值, 网络的整体能耗要小于最大容忍度; C5 表示无人机不能与障碍物碰撞; C6 表示无人机之间的覆盖范围不能重叠; C7 表示无人机仅为服务范围内的传感器提供服务。

4 基于 BWO 和 SCSO 的双层优化算法

4.1 问题分析

为了提高任务完成率和降低所有任务的延迟, 若无人机覆盖率低, 则卫星会收集并处理过多的任务, 这会导致任务的排队延迟大大增加。而且, 卫星收集的任务只有云服务器一个卸载目的地, 而无人机收集的任务可以卸载到卫星, 也可以通过卫星再卸载到云服务器, 有两个卸载目的地, 这可以更好地均衡各个设备之间的计算资源和通信资源。因此, 需要对无人机轨迹和计算卸载联合优化。

本文中描述的优化问题是一个混合整数非线性规划问题, 具有整数和连续变量, 其中目标函数是非凸函数, 这使得直接对该问题进行求解变得困难。因此, 将所提出的问题进行分解是非常有必要的。其次, 问题 P1 实际上是将无人机轨迹优化问题和计算卸载问题这两个问题进行结合表述, 可以将其中两个问题分解出来。

将问题 P1 分解成两个部分。其一, 提高无人机覆盖率的无人机轨迹优化问题。在整个周期 T 中, 每一时刻由算法的上层确定无人机的飞行角度, 决定无人机在下一时刻的位置, 提高无人机的覆盖率。问题 P2 可表述为:

$$\begin{aligned}
 & \text{P2: } \sum_t \max Cov^{\text{rate}} \\
 & \text{s. t. C5-C7}
 \end{aligned} \tag{19}$$

$$X_{i,j}(t+1) = \begin{cases} X_{i,p_j}(t) + (X_{r,p_1}(t) - X_{i,p_j}(t)) \times (1+r_1) \sin(2\pi r_2), & j = \text{even} \\ X_{i,p_j}(t) + (X_{r,p_1}(t) - X_{i,p_j}(t)) \times (1+r_1) \cos(2\pi r_2), & j = \text{odd} \end{cases} \tag{23}$$

其中, $X_{i,j}(t+1)$ 是第 i 条白鲸在第 j 维上的新位置, $p_j (j=1, 2, \dots, d)$ 是从 d 维中随机选择的, $X_{i,j}(t)$ 是第 i 条白鲸在第 j 维上的旧位置, $X_{r,p_1}(t)$ 和 $X_{i,p_j}(t)$ 是第 i 条和第 r 条白鲸的当前位置, r 表示随机选择的白鲸, r_1 和 r_2 是范围在 $(0, 1)$ 之间

其二, 在整个周期 T 中, 每一时隙对无人机和卫星的计算卸载策略进行优化, 提高任务完成率和降低任务延迟。问题 P3 可以表述为:

$$\begin{aligned}
 & \text{P3: } \sum_t \max_a F^{\text{rate}}, \min_a D \\
 & \text{s. t. C1-C4}
 \end{aligned} \tag{20}$$

4.2 基于 BWO 的上层优化算法

对于问题 P2, 尽管将整个系统的时间划分为多个时隙引入了时间上的离散化, 但在每个时隙内进行的轨迹优化问题仍然是连续的。因此, 从整体上来看, 这种问题依然保留了连续优化问题的特性, 只是通过时间离散化的方式简化了求解的复杂性。而白鲸优化 (Beluga Whale Optimization, BWO) 算法^[34] 擅长处理复杂连续优化问题, 收敛速度快, 全局搜索能力强, 因此选用 BWO 算法对 P2 进行求解。

在无人机轨迹的优化问题中, 每个无人机的飞行高度相同, 则在同一时隙内的无人机坐标可由水平坐标表示。在所提模型中, 每个无人机下一时刻的坐标由当前时刻的飞行角度决定。在优化过程中, 对飞行角度进行优化, 每一个粒子都代表 5 个无人机最优飞行角度的解。本文中的模拟实验对 5 个无人机的轨迹进行优化, 每一个粒子都是 5 维的, 而 BWO 算法在处理高维、复杂问题时具有一定优势。

BWO 算法的流程如图 3 所示。BWO 首先对白鲸的种群进行初始化, 探索阶段随机选择白鲸, 保证了设计空间中的全局搜索能力, 开发阶段控制着设计空间中的局部搜索。为了模拟其行为, 白鲸被视为搜索的代理, 通过改变位置向量来实现在搜索空间中移动。最后, 在 BWO 中还考虑了鲸鱼坠落的概率。

1) 初始化: 由于 BWO 的种群机制, 白鲸被视为搜索代理, 每一条白鲸都是一个粒子, 在优化的过程中进行更新。搜索代理的位置矩阵为:

$$\mathbf{X} = \begin{pmatrix} x_{1,1} & \cdots & x_{1,d} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \cdots & x_{n,d} \end{pmatrix} \tag{21}$$

其中, n 和 d 分别表示白鲸种群的大小和算法中的变量维度。BWO 需要一个平衡因子判断在当前迭代次数 t 时, 白鲸处在探索阶段还是开发阶段, 平衡因子用 B_f 表示:

$$B_f = \frac{B_0(2T^{\text{max}} - t)}{2T^{\text{max}}} \tag{22}$$

其中, t 是当前迭代次数, T_{max} 是最大迭代次数, B_0 每次迭代时在 $(0, 1)$ 之间随机变化。探索阶段发生在平衡因子 $B_f > 0.5$ 时, 而开发阶段发生在 $B_f \leq 0.5$ 时。随着迭代次数 t 的增加, B_f 的波动范围从 $(0, 1)$ 减小到 $(0, 0.5)$ 。

2) 探索阶段: 搜索代理的位置是通过模拟一对白鲸的游泳行为来建立的。白鲸的位置更新公式如下:

用来增强探索阶段的随机算子, $\sin(2\pi r_2)$ 和 $\cos(2\pi r_2)$ 表示镜像的白鲸鱼鳍朝向水面, 根据奇数和偶数选择维度。更新后的位置反映了白鲸在游泳或潜水的镜像或同步行为。

3) 开发阶段: 白鲸可以根据相邻白鲸的位置, 分享彼此

的位置信息来捕捉猎物,从而选择最好的候选解。在BWO中引入 Levy 飞行策略捕捉猎物,白鲸捕食的计算式为:

$$X_i(t+1) = r_3 X^{\text{best}}(t) - r_4 X_i(t) + C_1 L_F(X_r(t) - X_i(t)) \quad (24)$$

其中, $X_i(t)$ 和 $X_r(t)$ 分别代表当前位置的第 i 条白鲸和一条随机选择的白鲸, r_3 和 r_4 为 $(0,1)$ 之间的随机数, C_1 表示测量出来的 Levy 飞行的随机跳跃强度, L_F 是 levy 的飞行函数。

4) 鲸落阶段:少数白鲸会坠落深海,滋养生物,形成“鲸落”现象。算法模拟白鲸坠落的行为,选择坠落概率来模拟白鲸群体的变化。为确保白鲸种群数量不变,利用白鲸位置和白鲸坠落的步长更新白鲸位置。

$$X_i(t+1) = r_5 X_i(t) - r_6 X_r(t) + r_7 X^{\text{step}} \quad (25)$$

其中, r_5 , r_6 和 r_7 为 $(0,1)$ 之间的随机数; X^{step} 是白鲸坠落的步长,其表达式为:

$$X^{\text{step}} = (u_b - l_b) \exp\left(-\frac{C_2}{T_{\text{max}}}\right) \quad (26)$$

其中, C_2 是步长因子, $C_2 = 2 W_f \times n$, 与鲸落概率和种群大小有关; u_b 和 l_b 分别表示搜索空间中的上下界。白鲸坠落的概率 W_f 是一个线性函数,即:

$$W_f = 0.1 - \frac{0.05t}{T_{\text{max}}} \quad (27)$$

白鲸坠落的概率从刚开始迭代的 0.1 下降到最后一次迭代的 0.05,说明当白鲸在优化过程中更接近食物源时,白鲸的危险会降低。

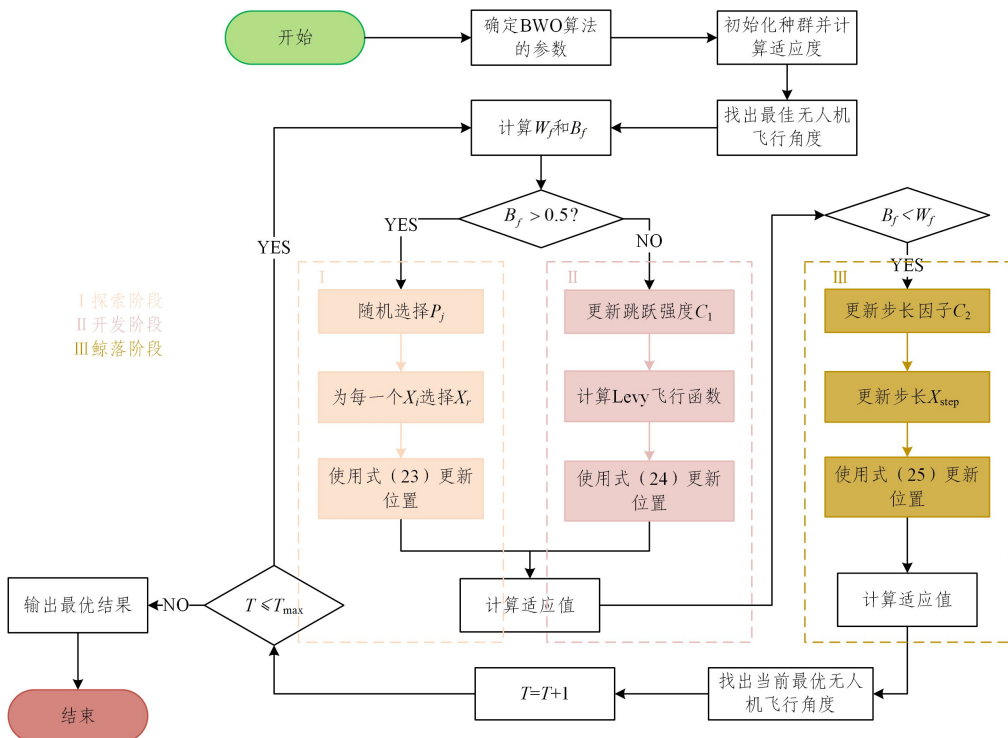


图3 BWO算法的流程图

Fig. 3 Flowchart of BWO algorithm

BWO算法的适应函数采用奖惩的方式进行设计。首先计算无人机覆盖范围内的传感器个数,以被无人机覆盖的传感器的个数为整个算法的适应度奖励,每当无人机的轨迹不符合约束条件时,会给予惩罚。基于BWO的无人机轨迹优化算法如算法1所示。

算法1 BW-SCS上层算法

输入:设置种群数量 N 和最大迭代次数 T ,初始化白鲸种群

输出:下一时刻的无人机位置坐标

1. while $t \leq T$
2. 由式(27)得出白鲸的坠落概率 W_f ,由式(22)得出平衡因子 B_f
3. for i to N
4. if $B_f(i) > 0.5$
5. 使用式(23)更新第 i 个白鲸的位置
6. else if $B_f(i) \leq 0.5$
7. 更新随机跳跃强度 C_1 ,计算 Levy 飞行函数

8. 使用式(24)更新第 i 个白鲸的位置
9. end
10. 检查新位置边界,评估适应度
11. end
12. for i to N
13. if $B_f(i) \leq W_f$
14. 使用式(25)更新第 i 个白鲸的位置
15. 检查新位置边界,评估适应度
16. end
17. end
18. 根据最优飞行角度计算下一时刻的无人机坐标
19. $t = t + 1$
20. end

4.3 基于SCSO的下层优化算法

在计算卸载优化中,由问题 P3 可知,此优化问题也是非

凸的 NP 难问题,且要对多个无人机中的计算卸载策略进行同时优化,各设备之间既存在竞争关系,也存在合作关系。本文中网络的计算卸载环境是动态变化的,使用传统的数学方法,如分支定界法,并不能有效地适应环境的变化。故采用沙猫群优化(Sand Cat Swarm Optimization, SCSO)算法来寻找可行解^[35]。

SCSO 能够在动态变化的环境中进行有效搜索,适应环境的变化,及时对策略进行调整。最重要的一点是,SCSO 算法天然具有并行计算的优势,每个沙猫个体在群体中独立更新位置、计算适应度,这使得 SCSO 可以通过并行计算加速搜索过程,尤其是在处理大规模问题时表现优异。

1) 初始化:SCSO 是一种基于种群的算法,其将相关结构定义为向量。在 d 维优化问题中每一个沙猫是一个 $1 \times d$ 数组,代表问题的解,在本文模型中无人机和卫星都会选择卸载率对计算任务进行卸载,因此每一只沙猫均是 1×6 的数组,其卸载率遵循问题 P3 中的限制条件 C1 和 C2,其定义如图 4 所示。在算法初始化中,根据问题的大小创建一个候选矩阵,并计算出每一个沙猫的适应值。

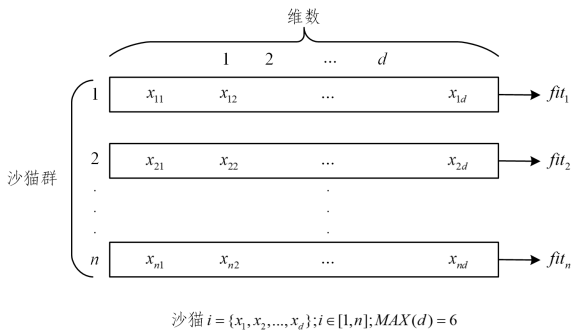


图 4 沙猫初始化过程

Fig. 4 Sand cat initialisation process

2) 探索阶段(搜索猎物):在探索阶段和开发阶段有一个重要的参数 R ,用于控制探索与开发阶段的过渡。当 $|R| > 1$ 时,沙猫搜索猎物。沙猫对猎物的搜索依赖于低频噪声的释放,假设沙猫的听觉范围 r^G 为 0 到 2 kHz,听觉范围的表达式为:

$$r^G = s^M \left(\frac{s^M \times iter_c}{iter_{max}} \right) \quad (28)$$

s^M 灵感来自沙猫的听觉特征,假设其值为 2。 $iter_c$ 为当前迭代次数, $iter_{max}$ 为最大迭代次数。参数 R 和 r 可以表示为:

$$\begin{cases} R = 2 \times r^G \times rand(0, 1) - r^G \\ r = r^G \times rand(0, 1) \end{cases} \quad (29)$$

其中, $rand(0, 1)$ 表示 0 到 1 的随机数。由以上参数,可得位置更新公式为:

$$X(t+1) = r \times (X^{best}(t) - rand(0, 1) \times X(t)) \quad (30)$$

其中, $X^{best}(t)$ 表示当前种群最优个体的所在位置。

3) 开发阶段(攻击猎物):当 $|R| \leq 1$ 时,沙猫攻击猎物。首先利用最佳位置和当前位置生成一个随机位置,随机位置可以确保沙猫靠近猎物。

$$X^{ran} = |(rand(0, 1) \times G^{best} - X(t))| \quad (31)$$

其中, X^{ran} 表示随机的位置向量, G^{best} 为当前种群中的最佳个体的位置向量。假设沙猫的灵敏度范围是一个圆,利用轮盘赌法给每一只沙猫随机选择一个角度 θ ,随机角度可以避免算法陷入局部最优。最后通过式(36)实现攻击猎物。

$$X(t+1) = (G^{best} - r \times X^{ran} \times \cos\theta) \quad (32)$$

其中, θ 为沙猫灵敏度范围圆上的随机角度。沙猫通过计算各自的适应值,来选择最优个体,从而得到最优解。将任务完成率和任务延迟共同作为计算适应值的一部分,这样即可完成共同优化任务完成率和任务延迟的目标。本文采用两个指标加权的方式计算适应值,适应函数设计如下:

$$fit = k^{1-F^{rate}} \times F^{rate} - (1 - k^{1-F^{rate}}) \times d^{avg} \quad (33)$$

其中, d^{avg} 是已完成任务的平均延迟。 k 是任务完成率和任务平均延迟的平衡系数,令 k 与任务完成率相关联,当任务完成率低时, $k^{1-F^{rate}}$ 的值较大,完成率对适应值的影响较大,算法会优先提升任务完成率,反之, $k^{1-F^{rate}}$ 的值较小,则算法会优先优化延迟,这样就平衡了任务完成率和任务延迟对适应值的影响。

基于 SCSO 的计算卸载优化算法如算法 2 所示。

算法 2 BW-SCS 下层算法

输入:通过算法 1 获取无人机的轨迹坐标,根据无人机轨迹坐标计算

无人机的覆盖范围

输出:最优卸载策略

1. 初始化:设置沙猫群数量和最大迭代次数,初始化种群
2. while $t \leq T$
3. for i to S
4. for j to U
5. 判断传感器是否在无人机覆盖范围内
6. 无人机收集计算任务
7. 卫星收集计算任务
8. 更新 r, r_c, R
9. 检查是否有沙猫超出搜索空间并进行修改
10. 使用式(33)计算每只沙猫的适应度,并找到最优个体
11. for i to N
12. 利用轮盘法选择随机角度
13. if $|R| \leq 1$
14. 使用式(32)更新沙猫搜索位置
15. else if $|R| > 1$
16. 使用式(30)更新沙猫搜索位置
17. end
18. end
19. $t = t + 1$
20. end
21. 输出每个无人机的最优卸载比例,即最优卸载策略

4.4 BW-SCS 双层优化算法

上文介绍了 BW-SCS 上层算法和 BW-SCS 下层算法,本节详细阐述 BW-SCS 上层算法的具体流程。

如图 5 所示,首先使用 BW-SCS 上层算法对每一个无人机的飞行角度进行优化;其次根据每一个无人机的飞行角度,即可计算出在下一时隙下的无人机位置,同时在当前时隙下,无人机和卫星在当前位置收集计算任务,使用

BW-SCS 下层算法对计算卸载策略进行优化;然后使用最优卸载策略处理计算任务直至当前时隙结束;最后无人机飞行至下一时隙的坐标,系统进入下一时隙;当时隙来到最终时间,算法结束。

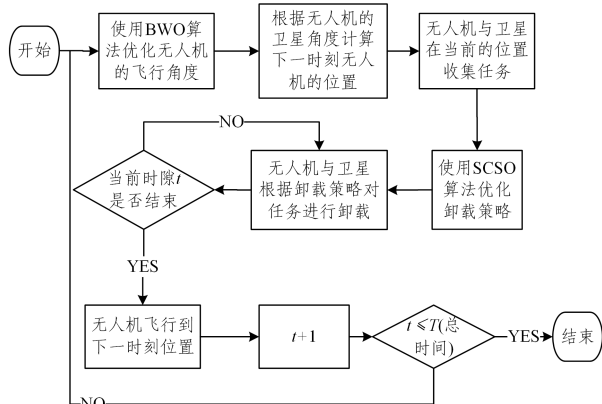


图5 BW-SCS 双层算法流程图

Fig. 5 BW-SCS two-layer algorithm flow chart

4.5 BW-SCS 双层优化算法时间复杂度分析

BW-SCS 双层算法由两部分组成:上层 BWO 算法和下层 BWO 算法。

上层 BWO 算法在初始化时生成一个 $n \times nd$ 的位置矩阵,时间复杂度为 $O(n \times nd)$ 。在算法主循环中位置更新、边界处理以及鲸落阶段的时间复杂度均为 $O(n \times nd)$ 。适应度计算的时间复杂度为 $O(n)$ 。由于主循环进行 t 次,因此上层 BWO 算法总的时间复杂度为 $O(t \times (n \times nd + n))$ 。

下层 SCSO 算法初始化 N 个 d 维的沙猫群,时间复杂度为 $O(n \times d)$ 。在主循环中,每次位置更新的时间复杂度也是 $O(n \times d)$ 。最优解更新的时间复杂度则是 $O(n)$ 。主循环循环 t 次,下层 SCOS 算法的总时间复杂度为 $O(t \times (nd + n))$ 。

综上所述,BW-SCS 双层算法的时间复杂度为 $O(t \times (n \times nd + nd + n))$ 。

5 仿真实验

为了验证文中提出的 BW-SCS 优化算法的可行性和有效性,本章针对以下研究问题进行仿真实验。

Q1:分别验证 BW-SCS 上层算法和下层算法的收敛性。

Q2:验证 BW-SCS 上层算法对无人机轨迹优化以提升无人机对地面传感器覆盖率的有效性。

Q3:使用 BW-SCS 上层算法优化无人机轨迹,然后进行计算卸载优化,验证 BW-SCS 算法的有效性。

5.1 实验环境与参数

本实验在 Windows11 系统上使用 pyCharm 编程软件和 Python3.8 进行离线的模拟实验。假设模拟实验的实验场地是一个 $300\text{m} \times 300\text{m}$ 的野外。传感器的位置固定,服从均匀分布。无人机的初始位置固定,由优化算法来确定之后所有时隙下的无人机轨迹。假设传感器生成的都是 $[3.5, 4]$ MB 的数值计算任务,任务到达无人机和卫星的数量服从泊松分布^[36]。本文详细的环境参数设置如表 1 所列。

表 1 参数设置

Table 1 Parameter settings

参数	意义	值
S	传感器数量/个	200
U	无人机数量/个	5
Sa	障碍物数量/个	4
r_{obs}	障碍物半径/m	[42,32,36,46]
r_u	无人机覆盖半径/m	23
v^u	无人机最大飞行速度/(m/s)	1
W	带宽/MHz	1
f_{UAV}	无人机 CPU 频率/GC/s	2.8
f_{sate}	卫星 CPU 频率	4.8
f_{cloud}	云服务器 CPU 频率	9
d_{sate}	卫星传播延迟/ms	6.44
φ	计算任务大小/MB	[3.5,4]

5.2 BW-SCS 上层 BWO 算法的有效性

将所提方法与粒子群优化算法(PSO)、遗传算法(GA)、正弦余弦算法(SCA)以及随机算法进行对比实验。

5.2.1 BW-SCS 上层 BWO 算法的收敛性

首先,如图 6 所示,取每个时隙下的 BWO 算法适应值的平均值进行分析。可见 BWO 算法在不断地收敛,直至第 46 轮时完全收敛,适应值达到最低点,证明了 BWO 算法在本文实验环境中的收敛性。

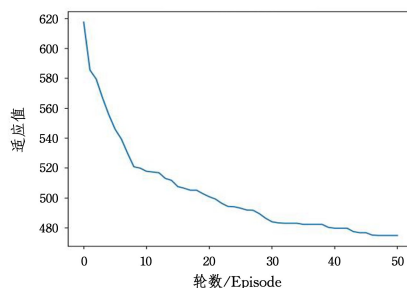


图6 BWO 算法收敛情况

Fig. 6 BWO algorithm convergence

5.2.2 不同算法下的无人机飞行轨迹

如图 7 所示,其中黑色圆圈代表障碍物,蓝色圆点代表传感器,5 种不同颜色的线条分别表示 5 架无人机的飞行轨迹。图 7(a)是基于本文算法的无人机轨迹,可以看出每一个无人机都可以完全避开障碍物,并且每个无人机之间的覆盖范围也没有相交,5 架无人机在飞行过程中的轨迹分布比较均匀,基本实现了区域全覆盖,除布置在障碍物上的传感器外,无人机对其他传感器基本都提供了服务,实现了服务的公平性,提高了服务的效率。

图 7(b)是基于 PSO 算法的无人机轨迹,可以看出无人机也基本实现了区域全覆盖,但是 PSO 并不能保证所有的无人机都能满足限制条件。在图中有两架无人机的覆盖范围发生了重合现象,还有一架无人机与障碍物发生了碰撞,这是因为部分无人机陷入了局部最优,导致算法不能推导出无人机下一时刻的最优位置。

图 7(c)是基于 GA 算法的无人机轨迹,图中 5 架无人机都能很好地满足限制条件,没有与障碍物发生碰撞,无人机之间也没有发生覆盖范围相交的情况。但是 GA 算法下的无人机并不能实现对区域的基本覆盖,虽然它没有 PSO 算法不满足限制条件的情况,但是其训练效果不稳定。

图 7(d)是基于 SCA 算法的效果图。SCA 算法在复杂的

优化问题中,尤其是在带有约束的优化问题中可能陷入局部最优解难以跳出,特别是存在多个局部最优解的情况下,算法的全局搜索能力会下降。图中,无人机的轨迹并没有很好地

服从限制条件,也没有明显扩大覆盖范围。图 7(e)使用的是随机算法,随机算法每一个时隙下的无人机轨迹都是随机的,因此 5 个无人机并没有满足限制条件,覆盖范围也很小。

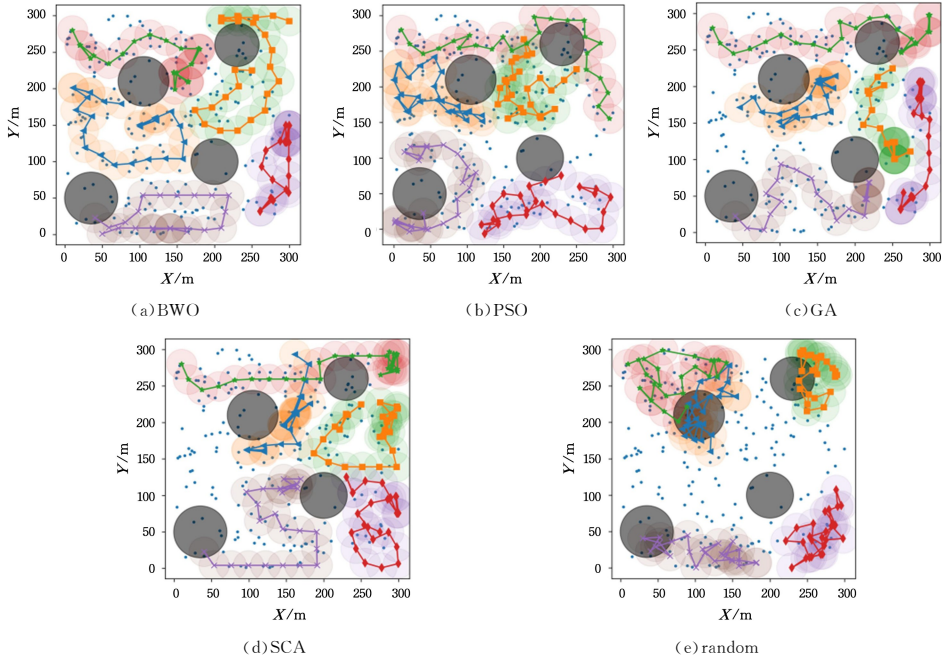


图 7 BWO,PSO,GA,SCA 和 random 算法优化的无人机轨迹效果图(电子版为彩图)

Fig. 7 Effect of UAV trajectories optimised by BWO,PSO,GA,SCA and random algorithms

5.2.3 不同算法下的无人机覆盖率对比

图 8 给出了 BWO,PSO,GA,SCA 和 random 优化无人机覆盖率的对比图。随机算法下的覆盖率只有 54.5%,有近 50%的传感器不在无人机的服务范围之内,而 SCA 算法把覆盖率提高到了 74%,但仍满足不了无人机要对大部分传感器进行服务的需求。在 GA 算法和 PSO 算法的优化下,覆盖率分别达到了 80.5%和 87.5%,但是 GA 算法下的无人机并不能实现对区域的基本覆盖,PSO 并不能保证所有的无人机都能满足限制条件,因此 GA 算法和 PSO 算法仍无法满足优化需求。采用 BWO 算法优化下的无人机轨迹的覆盖率最高达到了 90.5%,优于其他 4 种算法,满足了服务的公平性,提高了无人机的利用效率。

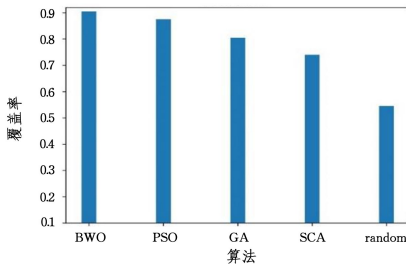


图 8 BWO,PSO,GA,SCA 和 random 算法优化无人机覆盖率对比图

Fig. 8 Comparison of BWO,PSO,GA,SCA and random algorithms for optimising UAV coverage

实验结果表明,BWO 算法在提升无人机覆盖率以及对各个限制条件的满足下均优于其他 4 种算法。

5.3 BW-SCS 双层算法对计算卸载的有效性

为验证 BW-SCS 算法对计算卸载的有效性,引入了对比

算法 BW-PSO,BW-GA,BW-SCA 和 BW-local。这些算法在上层均采用 BWO 算法对无人机轨迹进行优化,在下层分别使用粒子群优化算法(PSO)、遗传算法(GA)、正弦余弦算法(SCA)和全本地计算(local)对无人机和卫星的卸载策略进行优化。在每次优化时,5 种算法均在相同的实验环境下进行。

5.3.1 各算法的收敛情况对比

由于全本地计算在同一环境下,每次的优化结果都是相同的,收敛性没有意义,因此在这里不对全本地计算算法进行比较。由图 9 可知,本文提出的 BW-SCS 双层优化算法和其他 3 种对比算法都可以收敛,其中 BW-SCS,BW-GA 和 BW-PSO 收敛的整体趋势相同,BW-SCA 在 22 轮才趋近收敛。此外,BW-SCS 的适应度从始至终都优于其他 3 种算法,并且曲线很平滑,收敛也很快。由此可见,SCSO 算法的收敛速度快,适应性强。

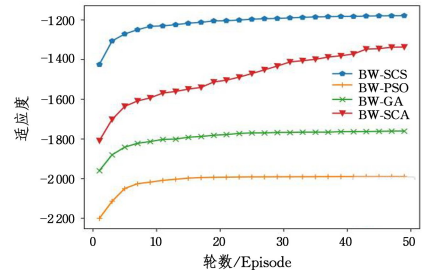


图 9 BW-SCS,BW-PSO,BW-GA 和 BW-SCA 算法的适应度对比

Fig. 9 Comparison of adaptation of BW-SCS,BW-PSO, BW-GA and BW-SCA algorithms

5.3.2 不同算法下的任务完成率对比

如图 10 所示,BW-PSO 算法优化后的任务完成率达到

82.5%, BW-GA 算法优化后的任务完成率达到了 85.9%, BW-SCA 算法对任务完成率的优化效果只达到了 76.5%, 这也是 BW-SCA 算法的适应度比 BW-PSO, BW-GA 和 BW-SCS 低的主要原因。在 BW-local 算法下的任务完成率达到 63%, 也就是说无人机和卫星收集到的任务只在本地计算的情况下只能完成 63% 的任务。BW-SCS 算法的完成率达到 87.5%, 优于其他对比算法。

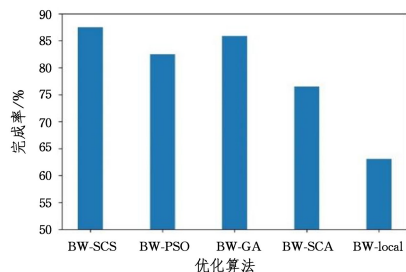


图 10 BW-SCS, BW-PSO, BW-GA, BW-SCA 和 BW-local 算法所优化的任务完成率对比

Fig. 10 Comparison of task completion rates optimised by BW-SCS, BW-PSO, BW-GA, BW-SCA and BW-local algorithms

5.3.3 不同算法下的任务延迟对比

如图 11 所示, BW-local 算法的平均延迟是最高的。BW-SCA 算法的平均延迟最低, 这也是它的适应度相对较高的主要原因, 但是 BW-SCA 算法的任务完成率相较于 BW-PSO, BW-GA 和 BW-SCS 最低。BW-GA 算法虽然在任务完成率的优化中起到了不错的效果, 但是除了 BW-local 算法外, 它的平均延迟明显高于其他 3 种算法。BW-PSO 算法在任务延迟的优化上同任务完成率的优化一样, 并没有明显的效果。BW-SCS 算法的平均延迟达到了 311.5 ms, 仅比平均延迟最低的 BW-SCA 算法高出 5.6 ms, 这是因为任务的完成率对任务的延迟有着严重的影响, 完成率越高, 所排队的任务就会增加, 其延迟就会增加。BW-SCS 算法的完成率比 BW-SCA 算法的完成率高出 9%, 因此 BW-SCS 算法在平均延迟优化上的性能仍优于 BW-SCA 算法。经过以上分析可知, BW-SCS 优化算法在任务延迟的优化上优于其他对比算法, 达到了很好的效果。

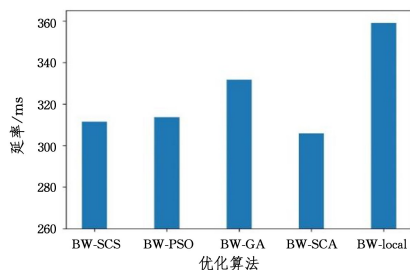


图 11 BW-SCS, BW-PSO, BW-GA, BW-SCA 和 BW-local 算法的任务延迟对比

Fig. 11 Comparison of task latency for BW-SCS, BW-PSO, BW-GA, BW-SCA and BW-local algorithms

经过以上分析, 本文提出的 BW-SCS 优化算法不管是在上层中对无人机轨迹进行优化, 还是在下层中对任务完成率和任务延迟进行优化, 与其他算法相比, 都取得了最好的优化效果。

结束语 本文研究了适用于偏远地区环境的空天地一体化网络, 在网络中部署多台无人机和一个卫星作为边缘设备, 在地面部署一个云服务器, 提出了联合无人机轨迹和计算卸载的优化问题, 以提高计算任务完成率, 并降低计算任务延迟。为此, 提出了 BW-SCS 双层算法, 该算法在上层使用 BWO 算法对每一架无人机的轨迹进行优化, 在下层使用 SC-SO 算法寻找最优的卸载策略。实验结果表明, 本文提出的算法与其他基准算法相比, 在无人机轨迹的覆盖率和任务的完成率方面都得到了显著的提升, 同时任务的延迟显著降低。

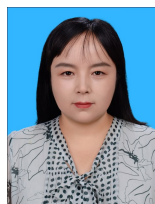
参考文献

- [1] LIU J, SHI Y, FADLULLAH Z M, et al. Space-air-ground integrated network: A survey[J]. IEEE Communications Surveys & Tutorials, 2018, 20(4): 2714-2741.
- [2] ALI S S D, ZHAO H P, KIM H. Mobile edge computing: A promising paradigm for future communication systems[C] // TENCON 2018 - 2018 IEEE Region 10 Conference. 2018: 28-31.
- [3] ZHOU Z, ZHANG C, XU C, et al. Energy-efficient industrial internet of UAVs for power line inspection in smart grid[J]. IEEE Transactions on Industrial Informatics, 2018, 14(6): 2705-2714.
- [4] WU Q, ZENG Y, ZHANG R. Joint trajectory and communication design for multi-UAV enabled wireless networks[J]. IEEE Transactions on Wireless Communications, 2018, 17(3): 2109-2121.
- [5] LI X, YAO H, WANG J, et al. A near-optimal UAV-aided radio coverage strategy for dense urban areas[J]. IEEE Transactions on Vehicular Technology, 2019, 68(9): 9098-9109.
- [6] LI X, YAO H, WANG J, et al. Rechargeable multi-UAV aided seamless coverage for QoS-guaranteed IoT networks[J]. IEEE Internet of Things Journal, 2019, 6(6): 10902-10914.
- [7] CHEN Q, GUO Z, MENG W, et al. A Survey on Resource Management in Joint Communication and Computing-Embedded SAGIN[J]. arXiv: 2403.17400, 2024.
- [8] ZHANG Z, XIAO Y, MA Z, et al. 6G wireless networks: Vision, requirements, architecture, and key technologies[J]. IEEE Vehicular Technology Magazine, 2019, 14(3): 28-41.
- [9] HOU X, REN Z, WANG J, et al. Latency and Reliability Oriented Collaborative Optimization for Multi-UAV Aided Mobile Edge Computing System[C] // IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). 2020: 150-156.
- [10] ZHOU C, WU W, HE H, et al. Delay-Aware IoT Task Scheduling in Space-Air-Ground Integrated Network[C] // 2019 IEEE Global Communications Conference (GLOBECOM). 2019: 1-6.
- [11] SHAO S, HE C, ZHAO Y, et al. Efficient trajectory planning for UAVs using hierarchical optimization[J]. IEEE Access, 2021, 9: 60668-60681.
- [12] MAO S, HE S, WU J. Joint UAV Position Optimization and Resource Scheduling in Space-Air-Ground Integrated Networks With Mixed Cloud-Edge Computing[J]. IEEE Systems Journal, 2021, 15(3): 3992-4002.
- [13] XIONG J, GUO H, LIU J. Task Offloading in UAV-Aided Edge

- Computing: Bit Allocation and Trajectory Optimization [J]. IEEE Communications Letters, 2019, 23(3): 538-541.
- [14] TANG A D, HAN T, XU D W, et al. Path planning method of unmanned aerial vehicle based on chaos sparrow search algorithm[J]. Journal of Computer Applications, 2021, 41(7): 2128-2136.
- [15] HAN T, TANG A D, ZHOU H, et al. Multiple UAV cooperative path planning based on LASSA method [J]. Systems Engineering and Electronics, 2022, 44(1): 233-241.
- [16] PERVEZ F, ZHAO L, YANG C. Joint User Association, Power Optimization and Trajectory Control in an Integrated Satellite-Aerial-Terrestrial Network[J]. IEEE Transactions on Wireless Communications, 2022, 21(5): 3279-3290.
- [17] LI X, FENG W, CHEN Y, et al. Maritime Coverage Enhancement Using UAVs Coordinated with Hybrid Satellite-Terrestrial Networks[J]. IEEE Transactions on Electron Devices, 2020, 68(1): 2355-2369.
- [18] JIA Z, SHENG M, LI J, et al. LEO-Satellite-Assisted UAV: Joint Trajectory and Data Collection for Internet of Remote Things in 6G Aerial Access Networks[J]. IEEE Internet of Things Journal, 2021, 8(12): 9814-9826.
- [19] ZHOU C, WU W, HE H, et al. Deep reinforcement learning for delay-oriented IoT task scheduling in SAGIN [J]. IEEE Transactions on Wireless Communications, 2020, 20(2): 911-925.
- [20] CHENG X, LYU F, QUAN W, et al. Space/aerial-assisted computing offloading for IoT applications: A learning-based approach[J]. IEEE Journal on Selected Areas in Communications, 2019, 37(5): 1117-1129.
- [21] LI Y, LIANG L, FU J, et al. Multiagent reinforcement learning for task offloading of space/aerial-assisted edge computing [J/OL]. <https://onlinelibrary.wiley.com/doi/epdf/10.1155/2022/4193365>.
- [22] XU Y, DENG F, ZHANG J. UDCO-SAGIMEC: Joint UAV Deployment and Computation Offloading for Space-Air-Ground Integrated Mobile Edge Computing [J]. Mathematics, 2023, 11(18): 4014.
- [23] WEI J, HAN J, CAO S. Satellite IoT Edge Intelligent Computing: A Research on Architecture[J]. Electronics, 2019, 8(11): 1247.
- [24] HUANG C, CHEN G, XIAO P, et al. Joint Offloading and Resource Allocation for Hybrid Cloud and Edge Computing in SAGINs: A Decision Assisted Hybrid Action Space Deep Reinforcement Learning Approach[J]. IEEE Journal on Selected Areas in Communications, 2024, 42(5): 1029-1043.
- [25] GAO Y, YE Z, YU H. Cost-Efficient Computation Offloading in SAGIN: A Deep Reinforcement Learning and Perception-Aided Approach[J]. arXiv: 2407.05571, 2024.
- [26] ZHANG N, LIANG H, CHENG N, et al. Dynamic spectrum access in multi-channel cognitive radio networks[J]. IEEE Journal on Selected Areas in Communications, 2014, 32(11): 2053-2064.
- [27] SHI W, LI J, XU W, et al. Multiple Drone-Cell Deployment Analyses and Optimization in Drone Assisted Radio Access Networks[J]. IEEE Access, 2018, 6: 12518-12529.
- [28] CHANDRASEKHARAN S, GOMEZ K, AL-HOURANI A, et al. Designing and implementing future aerial communication networks[J]. IEEE Communications Magazine, 2016, 54(5): 26-34.
- [29] SEID A M, BOATENG G O, MARERI B, et al. Multi-Agent DRL for Task Offloading and Resource Allocation in Multi-UAV Enabled IoT Edge Network[J]. IEEE Transactions on Network and Service Management, 2021, 18(4): 4531-4547.
- [30] AL-HOURANI A, KANDEEPAN S, LARDNER S. Optimal LAP Altitude for Maximum Coverage[J]. IEEE Wireless Communications Letters, 2014, 3(6): 569-572.
- [31] KANELLOPOULOS S A, KOUROGIORGAS C I, PANAGOPOULOS A D, et al. Channel model for satellite communication links above 10GHz based on weibull distribution[J]. IEEE Communications Letters, 2014, 18(4): 568-571.
- [32] HOU X, WANG J, FANG Z, et al. Edge Intelligence for Mission-Critical 6G Services in Space-Air-Ground Integrated Networks[J]. IEEE Network, 2022, 36(2): 181-189.
- [33] CUI J, NG S X, LIU D, et al. Multiobjective Optimization for Integrated Ground-Air-Space Networks: Current Research and Future Challenges [J]. IEEE Vehicular Technology Magazine, 2021, 16(3): 88-98.
- [34] ZHONG C, LI G. Beluga whale optimization: a novel nature-inspired metaheuristic algorithm [J]. Knowledge-Based Systems, 2022, 251: 109215.
- [35] SEYYEDABBASI A, KIANI F. Sand cat swarm optimization: a nature-inspired algorithm to solve global optimization problems [J]. Engineering with Computers, 2023, 39: 2627-2651.
- [36] SHI W, LI J, CHENG N, et al. Multi-Drone 3D Trajectory Planning and Scheduling in Drone Assisted Radio Access Networks [J]. IEEE Transactions on Vehicular Technology, 2019, 68(8): 8145-8158.



CHEN Yitian, born in 1999, postgraduate. His main research interests include edge computing and air-heaven-ground integrated networks.



TONG Yinghua, born in 1982, Ph.D., associate professor. Her main research interests include embedded system optimization and IoT system reliability.