

文章编号: 2095-4980(2024)09-0933-11

面向高吞吐量的 NB-IoT 低轨卫星物联网资源调度

吉用华^{a,b}, 张晨^{*a}, 张更新^{a,b}

(南京邮电大学 a.通信与信息工程学院; b.通信与网络技术国家工程研究中心, 江苏 南京 210003)

摘要: 窄带物联网(NB-IoT)作为一种低功耗广域网技术, 专门设计用于连接大量低功耗设备。基于该技术的低轨卫星物联网有着较低的传输损耗和时延, 并且可以通过星座方式对地球实现无缝覆盖。然而, 低轨卫星具有高度动态性, 并且面临来自不同用户的 QoS 需求, 这些因素使得现有资源调度算法的吞吐量性能受到了极大的影响。针对这些挑战, 本文在海量物联网用户请求无线资源且时频资源有限的场景下, 综合考虑卫星信道特性、不同用户间的可靠性与时延要求以及卫星高动态性引起的差分多普勒等因素, 提出了一种面向高吞吐量的 NB-IoT 低轨卫星物联网资源调度算法。仿真结果表明, 相比现有方法, 提出的资源调度算法在系统吞吐量方面表现出显著的性能提升。

关键词: 窄带物联网; 低轨卫星; 物联网; 资源调度

中图分类号: TN927^{+.2}

文献标志码: A

doi: 10.11805/TKYDA2024112

NB-IoT low-orbit satellite IoT resource scheduling for high throughput

Ji Yonghua^{a,b}, Zhang Chen^{*a}, Zhang Gengxin^{a,b}

(a.College of Telecommunications and Information Engineering; b.National Engineering Research Center for Communications and Network Technology, Nanjing University of Posts and Telecommunications, Nanjing Jiangsu 210003)

Abstract: Narrow Band Internet of Things(NB-IoT), as a low-power wide area network technology, is specially designed to connect a large number of low-power devices. The low-orbit satellite IoT based on this technology has lower transmission loss and delay, and can achieve seamless coverage of the earth through constellations. However, low-orbit satellites are highly dynamic and face QoS requirements from different users. These factors greatly affect the throughput performance of existing resource scheduling algorithms. In response to these challenges, this paper proposes a high-throughput NB-IoT low-orbit satellite IoT resource scheduling algorithm by comprehensively considering satellite channel characteristics, reliability and delay requirements between different users, and differential Doppler caused by high satellite dynamics in a scenario where a large number of IoT users request wireless resources and time-frequency resources are limited. Simulation results show that compared with existing methods, the resource scheduling algorithm proposed in this paper shows significant performance improvement in system throughput.

Keywords: Narrow Band Internet of Things(NB-IoT); low-orbit satellites; Internet of Things; resource scheduling

随着物联网产业的不断发展, 物联网终端数量爆炸增长, 终端部署环境也日益复杂。为了满足上述需求同时推进社会的智能化、数字化的历程, LTE Rel-13 引入了一种称为 NB-IoT 的新技术, 以支持低速率机器类型通信。然而, 基于地面基础通信设施的物联网往往难以覆盖到偏远地区的边缘用户, 为其提供有效的接入服务。近年来, 作为地面通信网络的扩大与补充, 近地轨道(Low-Earth-Orbit, LEO)卫星通信系统得到了快速发展。低轨卫星物联网由于其较低的传输损耗和时延的特点, 且可以通过星座方式对地球实现无缝覆盖的优势, 为偏远

收稿日期: 2024-02-27; 修回日期: 2024-03-21

基金项目: 国家重点研发计划重点专项基金资助项目(2022YFB2902600); 国家自然科学基金资助项目(61901230; U21A20450)

*通信作者: 张晨 email:zhangchen@njupt.edu.cn

地区用户的接入提供了一个有前景的解决方案,成为辅助地面网络实现万物互联的可靠选择。

然而,低轨卫星物联网资源调度仍然面临着诸多挑战。低轨卫星的覆盖范围广阔,导致区域内用户对服务质量的需求以及信道条件的多样性显著增加^[1]。此外,低轨卫星的高速运动也引发了区域内用户间较大的差分多普勒频移。这些因素共同导致现有资源调度算法的吞吐量性能受到了极大的影响。

目前学者为突破 NB-IoT 的局限性和获取更好的网络性能进行了大量的相关工作。文献[2]以最小化系统的能耗为目标提出了一种超可靠通信的高能效上行资源单元调度,考虑不同用户间服务质量(Quality of Service, QoS)需求的差异。文献[3]对窄带物联网中的信道资源和时隙资源进行数学描述,以最大化系统服务设备数为目标,建立一定限制条件下的上行链路资源分配模型,设计了一种基于用户 QoS 需求的动态上行资源分配算法。文献[4]建立自适应分配模型来实现上行链路的调度和功率分配,并引入蜻蜓算法提高资源利用率与网络通信质量。文献[5]研究了 NB-IoT 下行链路调度问题,以满足每个设备的数据需求的同时最小化所使用的无线电资源,为目标提出了一个有效的启发式算法。文献[6]考虑控制信道开销、时间偏移和重复因子,提出了一种干扰感知的 NB-IoT 资源分配方法。除此以外,部分学者结合其他技术来提高系统性能。文献[7]提出了一种基于非正交多址(Non-Orthogonal Multiple Access, NOMA)的 NB-IoT 系统匹配博弈算法。使用匹配博弈算法来帮助选择 NB-IoT 设备之间的最佳配对,从而最大化整体系统数据速率。文献[8]提出了一种用于 NB-IoT 系统的功率域上行链路 NOMA 方案,其允许多个设备共享相同的子载波,将子载波和发射功率联合分配问题公式化,以最大化满足 QoS 和发射功率要求的设备的数量。然而,卫星信道的缓变特性和不明显的远近效应使得传播带来的功率差不足以实现功率域非正交条件。注意到,上述文献均针对地面场景,没有考虑低轨卫星的特点,因此不适合高速运动的低轨卫星物联网场景。针对卫星物联网场景,文献[9]考虑整个通信过程,包括信令、上下行链路消息和连接释放,分析了 GEO 卫星系统中 NB-IoT 的性能指数。文献[10]在 LEO 卫星场景下,提出了基于 NB-IoT 的高效的上行数据传输资源分配策略,考虑到用户间的差分多普勒频移的影响,然而没有考虑到不同用户间 QoS 需求的多样性,此外仅采用单子载波进行资源调度,没有充分利用 NB-IoT 的特性。

针对以上问题,本文综合考虑用户业务 QoS 需求差异、NB-IoT 的特性以及低轨卫星的特点,构建了系统吞吐量最大化问题,提出了面向高吞吐量的 NB-IoT 低轨卫星物联网资源调度算法。针对用户间的较大差分多普勒频移,以 3GPP 规范为依据对用户按距离进行分组,对于每组用户考虑 QoS 约束并力求吞吐量最大。鉴于该问题属于 NP 完全问题,传统的最优解方法需要海量计算资源。因此,本文提出一个低复杂度的方案来解决这个问题。首先,基于 QoS 约束以及最小资源占用准则确定用户的基本调度参数,然后基于最少浪费优先策略确定用户在整个资源池的位置,同时使用袋獾算法(Tasmanian Devil Optimization, TDO)进行随机局部搜索来提高解的质量,确定用户的最终调度顺序。仿真结果表明,相比现有方法,本文提出的资源调度算法在系统吞吐量方面有很大提高。

1 卫星物联网资源调度架构模型

1.1 系统框架

在基于 NB-IoT 的低轨卫星物联网中,具有透明有效载荷的 LEO 卫星向一定数量的地面用户设备(User Equipment, UE)提供 NB-IoT 覆盖。卫星充当中继器,通过网关(Gateway, GW)提供 UE 和地面服务基站(Base Station, BS)之间的链路。在该场景中,卫星通过使用相控阵天线来形成凝视波束,以在服务时间内实现固定的地面波位覆盖。所有 UE 按照 NB-IoT 标准进行通信,同时所有 UE 共享一个物理资源块(Physical Resource Block, PRB)带宽用于上行链路的数据传输。卫星物联网资源调度架构如图 1 所示。

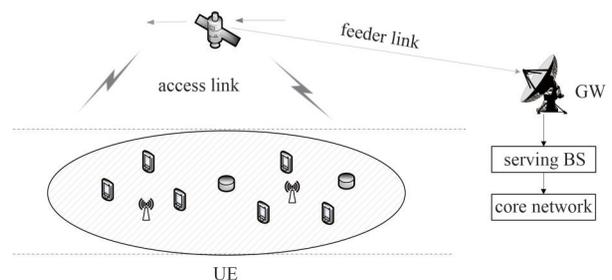


Fig.1 Resource scheduling architecture of satellite IoT

图 1 卫星物联网资源调度架构

1.2 业务建模

为实现更精确的资源调度和服务质量保证,有必要对不同业务类型的业务流量进行建模。3GPP 协议标准将物联网业务划分为 4 种主要类型,包括移动自主报告(Mobile Autonomous Reporting, MAR)异常业务、移动自主报告(MAR)周期业务、网络命令业务以及软件更新业务,同时对每种业务类型的业务特点及业务要求进行了说

明^[11]，如表 1 所示。考虑到相比异常业务，周期业务具有更为广泛的应用场景和大量的终端设备^[12]，本文以 MAR 周期业务作为流量模型。假设波束覆盖区域内包含 1 000 个用户，用户的业务类型为 MAR 周期业务，用户间的到达服从到达率为 30 的泊松分布，统计 20 天内每小时的发送数据包数以及数据量，如图 2 和图 3 所示。

表 1 不同种类的应用流量模型
Table1 Different types of application traffic models

application traffic models	application payload size distribution	periodic inter-arrival time
MAR exception reports	20 bytes	generally rare, typically occurring every few months or even years
MAR periodic reports	Pareto distribution with shape parameter alpha = 2.5 and minimum application payload size = 20 bytes with a cut off of 200 bytes	1 day (40%), 2 hours (40%), 1 hour (15%), and 30 minutes (5%)
network command	20 bytes	1 day (40%), 2 hours (40%), 1 hour (15%), and 30 minutes (5%)
software update/ reconfiguration model	Pareto distribution with shape parameter alpha = 1.5 and minimum application payload size = 200 bytes with a cut off of 2 000 bytes	(180 days)

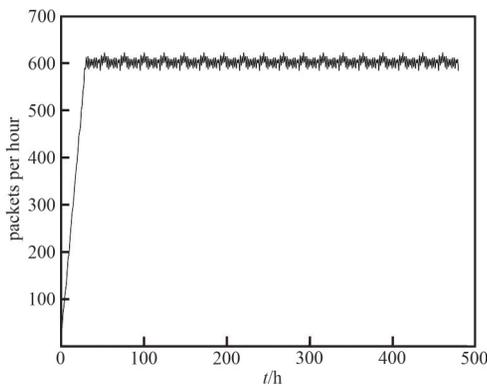


Fig.2 Statistics on the number of data packets sent per hour within 20 days

图 2 20 天内每小时发送数据包数统计

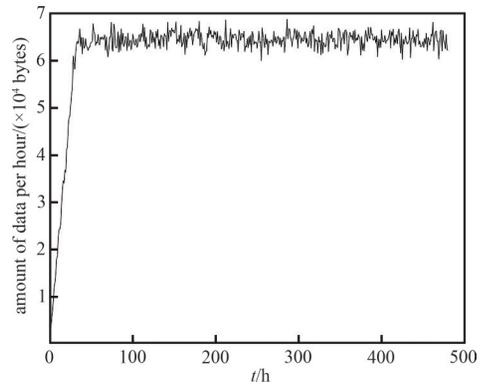


Fig.3 Statistics on the amount of data per hour within 20 days

图 3 20 天内每小时数据量统计

1.3 系统模型

如图 1 所示的场景中，设待发送数据的用户数为 N ，卫星的高度为 h_s ，用户在卫星覆盖区域内均匀分布。第 i 个用户的发射功率为 p_i ，其需要传输大小为 D_i (bytes) 的数据，并要求在 d_i (ms) 的时延内以可靠性 $R_i \in [0, 1]$ 完成上传到卫星。所有用户在频域上共享整个 PRB 带宽用于上行链路传输，总的可用子载波数为 B ，总的可用子帧数为 W 。

1) 卫星信道分析。卫星覆盖区域通常为开阔地区，直射波没有阻挡，接收信号可以看作直射分量和散射分量的叠加，因此使用莱斯信道进行建模。确定了覆盖区域内每个用户与卫星之间的链路距离后，可以根据链路预算公式来计算每个用户的链路信噪比^[13]：

$$\frac{C}{N} = P_{\text{EIR}} + \frac{G_r}{T} - L_{\text{FSP}} - A_{\text{loss}} - Ad_{\text{loss}} - k - 10 \lg(W_B) + H_{\text{Rice}} \quad (1)$$

式中： C 为接收信号功率； P_{EIR} 为发射天线的等效同向辐射功率； G_r/T 为具有天线增益 G_r 和等效系统温度 T 的接收机处的性能指数； L_{FSP} 为自由空间传播损耗； A_{loss} 和 Ad_{loss} 分别表示由于气体、雨衰等引起的大气损失和由于馈线链路引起的额外损失； k 为玻尔兹曼常数； W_B 为通信带宽； H_{Rice} 为莱斯信道中多径引起的小尺度衰落的叠加。

L_{FSP} 计算公式为：

$$L_{\text{FSP}} = 10 \lg \left(\frac{4\pi d}{c/f} \right)^2, d = -r_E \cdot \sin \alpha + \sqrt{r_E^2 (\sin \alpha)^2 + h_s^2 + 2r_E h_s} \quad (2)$$

式中: d 为倾斜距离; c 为光速; f 为载频; α 为用户仰角; r_E 为地球半径。

2) NB-IoT 技术分析。在 NB-IoT 中, 资源在时间上被划分为若干帧, 每个帧由 10 个子帧组成, 子帧的长度是 1 ms。窄带物理下行控制信道(Narrowband Physical Downlink Control Channel, NPDCCH)通过下行控制信息(Downlink Control Information, DCI)格式 N0 授权和控制上行链路调度。上行链路通过窄带上行链路共享信道(Narrowband Physical Uplink Share Channel, NPUSCH)传输数据, 资源单元(Resource Unit, RU)是在 180 kHz 的带宽中分配的基本传输资源单位。传输数据可以由一个或多个 RU 承载, 这取决于请求大小和调制编码方案(Modulation and Coding Scheme, MCS)。NB-IoT 基于子载波间隔支持多种类型的 RU, 同时为了提高传输可靠性, NB-IoT 采用重复发送机制。

具体来说, DCI 格式 N0 中的子载波指示 $I_i^{sc} \in \{0 \sim 63\}$ 描述 RU 类型 N_i^{sc} (即频域上占用多少子载波)、单个 RU 的时隙数 N_i^{slot} 以及分配的子载波集合 S_i^{sc} , 见表 2^[14]。RU 重复数 N_i^{rep} 和 DCI 与上行传输间的延迟 k_0 则通过 I_{rep} 和 I_{Delay} 指示。 $N_i^{rep} = 2^l$, 其中 $l \in \{0 \dots 7\}$ 。 k_0 可能值为 8、16、32 和 64。3GPP 规定了在 NPUSCH 为 UE 编码的可能传输块大小, 系统根据缓冲区状态报告以及 MCS 查表获取为用户分配 RU 的数量, 即 N_i^{RU} 。

3) 接入及 QoS 分析。在用户接入后, 为确保卫星能够成功解码用户数据, 须满足条件:

$$\frac{C}{N}(i) \geq R_{SN}(S_{i,MC}, N_i^{sc}, R_{BLE}) \quad (3)$$

式中: $\frac{C}{N}(i)$ 为接收信号的信噪比; $R_{SN}(S_{i,MC}, N_i^{sc}, R_{BLE})$ 为在特定系统误块率 R_{BLE} 、调制和编码方案 $S_{i,MC}$ 以及 RU 类型 N_i^{sc} 下的解调门限。一般来说, 在 NPUSCH 中, R_{BLE} 为 0.1。

为确保用户的 QoS 需求, 须同时满足可靠性 $R_i \in [0, 1]$ 和严格延迟约束 d_i (ms)。即:

$$N_i^{RU} \times \frac{N_i^{slot}}{2} \times N_i^{rep} \leq d_i \quad (4)$$

$$1 - (1 - P_i^s)^{N_i^{rep}} \geq R_i \quad (5)$$

式中: $P_i^s = (1 - R_{BLE})^{N_i^{RU}}$ 为数据 D_i 传输一次的成功概率^[15]; $1 - (1 - P_i^s)^{N_i^{rep}}$ 为 N_i^{rep} 次重复后的成功概率。

4) 低轨卫星差分多普勒分析。在低轨卫星通信场景下, 卫星的高速运动会引发用户自身的高多普勒频移问题。除此以外, 覆盖区域内的不同用户之间还存在着差分多普勒频移问题。这一现象源于覆盖区域内的不同用户以不同的仰角与卫星进行通信, 即信道条件存在差异。覆盖区域所有用户的公共多普勒频移可以在网关处进行预补偿或后补偿, 则差分多普勒随时间变化的公式如下^[16]:

$$\Delta f_{dmax}^x(t) = \left| \frac{fw_s r_E}{c} \left(\frac{r \sin(w_s t) - D}{\sqrt{r_E^2 + r^2 - 2r_E r \cos(w_s t) + D^2 - 2Dr \sin(w_s t)}} - \frac{r \sin(w_s t)}{\sqrt{r_E^2 + r^2 - 2r_E r \cos(w_s t)}} \right) \right| \quad (6)$$

$$\Delta f_{dmax}^y(t) = \left| \frac{fw_s r_E r \sin(w_s t)}{c} \left(\frac{\eta \left[\arctan\left(\frac{h}{D/2}\right) \right]}{\sqrt{r_E^2 + r^2 - 2r_E r \cos(w_s t) \eta \left[\arctan\left(\frac{h}{D/2}\right) \right]}} - \frac{1}{\sqrt{r_E^2 + r^2 - 2r_E r \cos(w_s t)}} \right) \right| \quad (7)$$

式中: w_s 为卫星在地心惯性坐标系(Earth Centered Inertial, ECI)下的角速度; r 为卫星轨道半径; D 为区域直径。

需要注意的是, 3GPP 规范支持载频为 2 GHz, 最大速度在 500 km/h 时的高达 950 Hz 的子载波间多普勒频移^[17]。与卫星运动方向相比, 沿卫星运动的垂直方向的差分多普勒要小得多, 通常远远低于极限值, 因此可以在问题处理中忽略该方向的影响。然而, 在特定条件下, 沿卫星运动方向的差分多普勒可能超过设定的极限值,

表 2 子载波指示和相应的 RU 类型、单个 RU 时隙数以及子载波集合
Table 2 Subcarrier indication and corresponding RU type, number of slots, and subcarrier set

subcarrier indication	RU type(N_i^{sc})	number of slots	set of allocated subcarriers
field (I_i^{sc})		per RU(N_i^{slot})	(S_i^{sc})
0~11	1	16	I_i^{sc}
12~15	3	8	$3(I_i^{sc} - 12) + \{0, 1, 2\}$
16~17	6	4	$6(I_i^{sc} - 16) + \{0, 1, 2, 3, 4, 5\}$
18	12	2	$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$
19~63	reserved	reserved	reserved

且距离越大，差分多普勒越高。因此在进行资源调度时，需要保证同一子帧中调度用户沿卫星运动方向的距离不超过3GPP规范所推导出的距离极限值。否则，将导致发生子载波间的频谱重叠，进而在卫星接收端引发显著的信号失真问题。

5) 最大吞吐量资源调度目标函数。本文的设计目标是在海量低轨卫星物联网用户请求无线资源的场景下，基于NB-IoT机制设计资源调度算法最大化服务用户的吞吐量，同时满足调度用户QoS需求以及用户间的差分多普勒约束。

使用 $B \times W \times N$ 的矩阵 O 表示系统资源分配的状态，其中 $o_{b,w,i} = 1$ 表示子载波 b 下的子帧 w 已经被分配给了用户 i ，否则 $o_{b,w,i} = 0$ 。使用向量 $\mathbf{v} = \{v_1, v_2, \dots, v_N\}$ 表示用户是否成功接入NB-IoT网络服务，若通过调度算法确定某用户 i 被调度，则令 $v_i = 1$ ，否则 $v_i = 0$ 。

显然，对于所有被调度的用户，其需要满足基本的时延约束、可靠性约束以及差分多普勒约束，并且所占用的RU没有重叠。优化参数包括：a) 用户的RU类型 (N_i^{sc})；b) 用户所需RU数量 (N_i^{RU})；c) RU重复数 (N_i^{rep})；d) 用户的起始调度子帧 (T_i^{sc})；e) 用户RU的子载波集 (S_i^{sc})。

最终目标函数与约束条件表示如下：

$$\max_{N_i^{\text{sc}}, N_i^{\text{RU}}, N_i^{\text{rep}}, T_i^{\text{sc}}, S_i^{\text{sc}}} \sum_{i=1}^N D_i \cdot v_i \quad (8)$$

$$C_1: v_i \in \{0, 1\}, i = 1, 2, \dots, N \quad (9)$$

$$C_2: o_{b,w,i} \in \{0, 1\}, b = 1, 2, \dots, B, w = 1, 2, \dots, W, i = 1, 2, \dots, N \quad (10)$$

$$C_3: \sum_{i=1}^N o_{b,w,i} \leq 1, \quad b = 1, 2, \dots, B, w = 1, 2, \dots, W \quad (11)$$

$$C_4: (C/N)_i \geq R_{\text{SN}}(S_i, \text{MC}, N_i^{\text{sc}}, R_{\text{BLE}}) \times v_i \quad (12)$$

$$C_5: N_i^{\text{RU}} \times \frac{N_i^{\text{slot}}}{2} \times N_i^{\text{rep}} \times v_i \leq d_i \quad (13)$$

$$C_6: 1 - (1 - P_i^s)^{N_i^{\text{rep}}} \geq v_i \times R_i, P_i^s = (1 - R_{\text{BLE}})^{N_i^{\text{RU}}} \quad (14)$$

$$C_7: |T_{i_1}^{\text{sc}}, T_{i_1}^{\text{sc}} + N_{i_1}^{\text{RU}} \times \frac{N_{i_1}^{\text{slot}}}{2} \times N_{i_1}^{\text{rep}}| \cap |T_{i_2}^{\text{sc}}, T_{i_2}^{\text{sc}} + N_{i_2}^{\text{RU}} \times \frac{N_{i_2}^{\text{slot}}}{2} \times N_{i_2}^{\text{rep}}| = \emptyset, \text{ if } d_{i_1 \rightarrow i_2} > d_{\text{Doppler_lim it}} \quad (15)$$

式中：约束 C_3 表示同一子载波下的单个时隙只可以分配给单个用户，即时频资源无法被不同用户复用；约束 C_4 表示用户的解码约束； C_5 和 C_6 分别表示每个用户的时延约束和可靠性约束； C_7 中 $d_{i_1 \rightarrow i_2}$ 表示2个用户沿卫星运动方向的距离，该约束条件表示如果任意2个被调度用户之间的距离超过差分多普勒极限所对应的距离值，那么这2个用户在时域上子帧范围不能重叠。

该优化问题同时涉及对离散变量和连续变量的处理，具有较高的复杂度。为应对这一挑战，考虑对用户进行分组，并对每个用户组进行独立的优化来实现问题的松弛处理，分组的原则是确保同一组内用户不违反差分多普勒限制。然而，由于式(6)无法直接求得多普勒极限距离的闭式解，并且分组过少会导致同一组内用户间的距离超过极限值，分组过多又会提高系统复杂度。因此，本文择优选择将覆盖区域的用户沿卫星运动方向以20 km为单位进行分组，以确保既不违反差分多普勒约束，又不会增加过高的复杂度。如图4所示。

通过对用户分组，原优化问题的最后一个约束条件得到了满足。当然，这一过程需要确定不同用户组的时间资源。在这里可以合理假设每个用户组的时间资源是按照用户数量的比例进行分配，共分为 M 组。分组后，问题转变为一个复杂的整数规划问题，同时也属于0-1二维背包问题的范畴。需要注意的是，问题中用户(类似于物品)的时频资源(形状)是可变的。问题的核心挑战在于，在不超过系统可用资源(类似于背包容量)的前提下，最优地给 N 个用户分配不相交的时频资源，以最大化调度用户的吞吐量。该问题被证明为一个非确定多

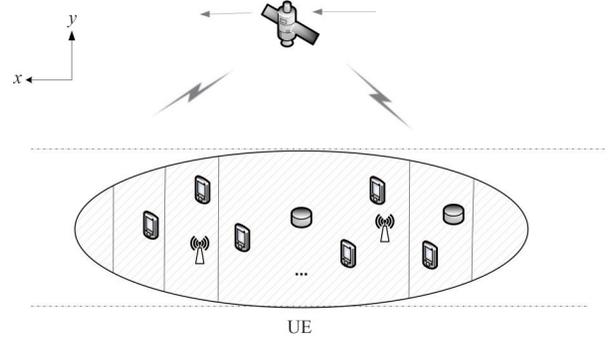


Fig.4 Schematic diagram of group scheduling
图4 分组调度示意图

项式(Non-deterministic Polynomial, NP)完全问题^[18]。

2 算法设计及实现

由于该问题是一个 NP 完全问题, 使用传统方法寻找最优解将耗费海量计算资源。因此, 本文提出一个低复杂度的方案来解决这个问题。首先, 基于 QoS 约束以及最小资源占用准则确定用户的基本调度参数, 包括 RU 类型 N_i^{sc} 、所需 RU 数 N_i^{RU} 以及重复数 N_i^{rep} ; 然后在调度顺序确定的条件下基于最少浪费优先策略确定用户的起始调度子帧 T_i^{sc} 以及子载波集 S_i^{sc} ; 考虑到调度顺序会影响解的结果, 使用袋獾算法(TDO)进行随机局部搜索来提高解的质量, 确定用户的最终调度顺序。

2.1 基于 QoS 约束以及最小资源占用准则的基本调度参数确定

参考文献[19], 对于分组中的每个用户, 根据信道条件以及待传数据量可以得到 RU 类型 N_i^{sc} 和所需 RU 数 N_i^{RU} 的可行组集合。然后对于每个用户, 考虑可行组合, 根据用户的时延约束以及可靠性约束计算得到允许的重复次数 N_i^{rep} 。最后根据最小资源占用准则, 获得最佳 RU 类型、RU 数量和重复次数作为用户的基本调度参数, 伪代码见表 3。

表 3 算法 1 基于 QoS 约束以及最小资源占用准则的基本调度参数确定

Table3 Determination of basic scheduling parameters based on QoS constraints and minimum resource occupancy criteria for Algorithm 1

input:	number of users N , satellite altitude h_s , total number of available subframes W , total number of available subcarriers B , delay constraints d_i and reliability constraints R_i for all users
output:	user's basic scheduling parameters, including RU type N_i^{sc} , number of RUs N_i^{RU} , and number of repetitions N_i^{rep}
1	function parameter-determination (N, h_s, W, B, d_i, R_i)
2	determine the number of user groups M (grouped in units of 20 km in this paper) based on differential Doppler constraints using Eq.(6), the number of users in the j th group is N_j , the time resource of the j th group is W_j
3	For $j = 1:M$
4	For $i = 1:N_j$
5	determine the link $R_{\text{SN}} C/N$ of each user based on channel conditions using Eq.(1)
6	obtain the set of feasible combinations of RU type and MCS, denoted as $A_i = \{(N_{i,j}^{\text{sc}}, MCS_{i,j})\}$ by traversing all RU types using Eq.(3)
7	obtain the number of RUs for each user based on 3GPP TS36.213, update set A_i to $A_i = \{(N_{i,j}^{\text{sc}}, N_{i,j}^{\text{RU}})\}$
8	calculate allowed repetitions based on the user's delay constraints and reliability constraints using Eq.(4-5), upgrade set A_i to set of triples $A_i = \{(N_{i,j}^{\text{sc}}, N_{i,j}^{\text{RU}}, N_{i,j}^{\text{rep}})\}$
9	select the best triplet of $A_i (N_{i,j}^{\text{sc}}, N_{i,j}^{\text{RU}}, N_{i,j}^{\text{rep}})$ according to the minimum resource occupation criterion
10	end for
11	end for
12	return $N_i^{\text{sc}}, N_i^{\text{RU}}, N_i^{\text{rep}}$
13	end function

2.2 最少浪费优先策略

确定了用户基本的调度参数后, 接下来需要确定用户在整个资源池的位置, 即用户的子载波集和起始调度子帧。该问题是一个形状固定的二维背包问题, 文献[20]提出了针对矩形装箱问题的最少浪费优先的启发式算法。然而, 考虑到 NB-IoT 系统中用户的子载波集只能在有限范围内选择, 因此该方法无法直接应用。对此, 文献[2]定义一个函数表述用户的 RU 被分配在特定子载波集上的潜在资源浪费。本节中不妨假设分组中用户的调度顺序已确定, 对应用户列表为 UE_{list} , 从列表中依序调度, 并基于最少浪费优先策略确定用户的子载波集 S_i^{sc} 和初始调度子帧 T_i^{sc} 。定义 K 为所有子载波组成的集合, S_j 为第 j 个子载波的最早可用子帧。定义函数 $Waste(i, S_i^{\text{sc}})$ 来反映如果用户的 RUs 被分配在子载波集 S_i^{sc} 上的潜在资源浪费, 即:

$$Waste(i, S_i^{\text{sc}}) = \sum_{j \in K - S_i^{\text{sc}}} \left(\left(T_i^{\text{sc}} + \left(N_i^{\text{RU}} \times \frac{N_i^{\text{slot}}}{2} \times N_i^{\text{rep}} \right) \right) - S_j \right)^+ + \sum_{j \in S_i^{\text{sc}}} (T_i^{\text{sc}} - S_j) \quad (16)$$

式中 $(\cdot)^+ = \max\{\cdot, 0\}$ 为输出大于等于 0 的值, 可以看出如果用户的 RUs 被分配在子载波集 S_i^{sc} , 则式(16)对用户资源分配完成时间之前的未使用资源空间进行求和。基于式(16), 选择使用户具有最小 $Waste(i, S_i^{\text{sc}})$ 的最佳子载波集:

$$S_i^{\text{sc}*} = \arg \min_{S_i^{\text{sc}} \in \Theta(N_i^{\text{sc}})} Waste(i, S_i^{\text{sc}}) \quad (17)$$

式中 $\Theta(N_i^{\text{sc}})$ 是当使用 2.1 节确定的最佳 RU 类型 N_i^{sc} 时所有可用子载波集组成的集合。已知用户的最佳子载波集,

那么用户的起始调度时间为 S_i^{sc} 对应的 T_i^{sc} 。最少浪费优先策略如图 5 所示，其中 UE1~UE4 已调度，UE5 正在被调度。4 张子图分别代表使用 4 种子载波集合的调度结果，显然，在选择子载波集为 {3,4,5} 时资源浪费最小，因此在资源调度时选择(b)对应的方案。

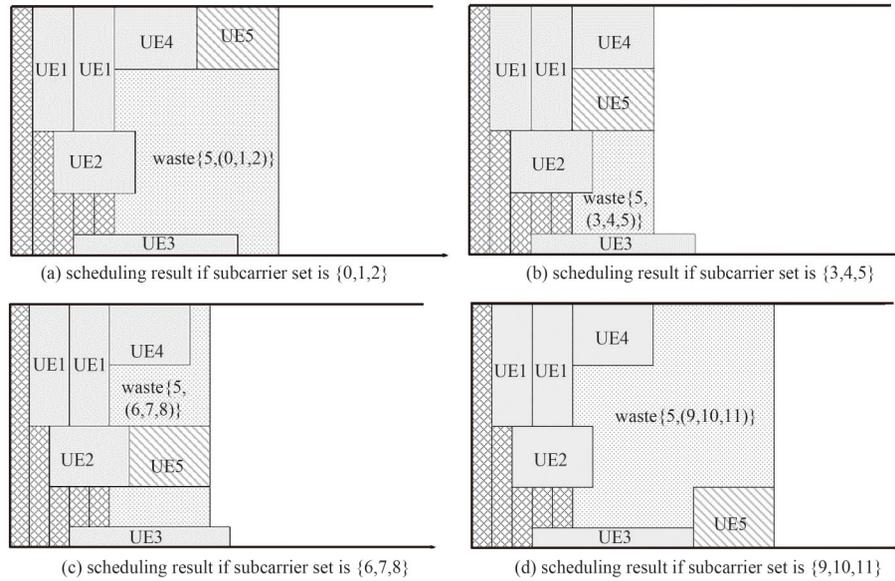


Fig.5 Schematic diagram of least waste first strategy

图 5 最少浪费优先策略示意图

每次调度后更新所有子载波的最早可用调度子帧，同时更新资源状态矩阵 O 和用户调度指示向量 v 。由此，基于浪费最少优先策略确定了所有用户的子载波集 S_i^{sc} 和初始调度子帧 T_i^{sc} 。伪代码见表 4。

表 4 算法 2 最少浪费优先策略

Table4 Least waste first strategy for Algorithm 2

input:	number of users N , the amount of data to be transferred by the user D_i , user's basic scheduling parameters, including RU type N_i^{sc} , number of Rus N_i^{RU} , and number of repetitions N_i^{rep} , total number of available subframes W , total number of available subcarriers B , scheduling order vector O_{order}
output:	total throughput $T_{throughput}$
1	function 2d-packing ($N, D_i, N_i^{sc}, N_i^{RU}, N_i^{rep}, W, B, O_{order}$)
2	resource status matrix $O \leftarrow 0$, user scheduling indication vector $v \leftarrow 0$ $T_{throughput} \leftarrow 0$
3	generate user list UE_{list} in O_{order}
4	for each user $UE_i \in UE_{list}$
5	determine the user's optional subcarrier set S_i^{sc} based on the user's RU type N_i^{sc}
6	for each subcarrier set $S_i^{sc} \in \Theta(N_i^{sc})$
7	calculate the resource waste of selecting a specific set of subcarriers S_i^{sc} using Eq.(16)
8	end for
9	select the subcarrier set with minimal resource waste S_i^{sc} and the corresponding starting scheduling subframe T_i^{sc} using Eq.(17)
10	update O and v
11	$T_{throughput} \leftarrow T_{throughput} + D_i$
12	If $T_i^{sc} \geq W$
13	break
14	end if
15	end for
16	return $T_{throughput}$
17	end function

2.3 基于 TDO 的随机局部搜索

由于 2.2 节的二维背包过程的结果受用户调度顺序影响，因此，本文引入了基于 TDO 的随机局部搜索来提高解的质量。袋獾优化算法(TDO)是一种新提出的智能算法，其机制是模拟袋獾的进食行为。其有 2 种策略：搜寻和捕食。假设袋獾选择任意一种进食策略的概率均为 50%，定义一个策略选择系数 Probability，并在 [0,1] 内均匀取值，当小于 0.5 时选择策略 1 搜寻，否则选择策略 2 捕食。在整个 TDO 过程中，以最大化系统的吞吐量为目标

函数, 如式(8)所示。为此使用向量 O_{order} 即所有用户的调度顺序表示每个袋獾个体, 并利用 2.2 节最少浪费优先策略获得系统的吞吐量作为适应度值。通过不断迭代以获得最优的调度顺序, 从而实现系统吞吐量的最大化。使用 B_{best} 保存迭代过程的最优值。

在搜寻策略中, 袋獾以栖息范围内的腐肉进食, 每个袋獾将种群中的其他个体所在位置视为腐肉位置。若选定腐肉的目标函数值较好, 则袋獾向腐肉移动, 否则远离腐肉。在袋獾移动后, 如果目标值在新位置则更好选择更新位置, 否则袋獾保持在原来的位置。向腐肉方向移动可以视为在腐肉对应的调度顺序上随机打乱若干元素, 对位置的更新可以视为向量 O_{order} 的更新。在捕食策略中, 袋獾进行主动捕猎并进食, 该策略分为 2 个阶段。在第 1 个阶段中, 通过对目标区域进行扫描, 搜寻到猎物并开始追踪, 该阶段的建模类似于搜寻策略; 在第 2 个阶段中, 袋獾开始追逐进而完成进食, 该阶段以袋獾的位置为圆心, 以追踪猎物的范围为半径, 在这个范围内为袋獾计算一个新的位置, 如果新位置下的目标函数比之前更优, 则进行位置更新^[21]。以袋獾的位置为圆心计算新的位置可以视为在原有顺序的基础上打乱特定数量的连续元素的顺序。

为了加快收敛速度, 首先对用户按照单位时频资源的待传数据量进行降序排序, $Area_i$ 表示用户所占用的时频资源面积。

$$\frac{D_1}{Area_1} \geq \frac{D_2}{Area_2} \geq \dots, \frac{D_i}{Area_i}, \dots, \geq \frac{D_{\text{end}}}{Area_{\text{end}}}, \quad Area_i = N_i^{\text{RU}} \times \frac{N_i^{\text{slot}}}{2} \times N_i^{\text{rep}} \times N_i^{\text{sc}} \quad (18)$$

在上述顺序的基础上, 随机选择用户进行打乱, 不断调用 2.2 节函数获得系统的吞吐量作为适应度值以初始化种群。接着对于种群中的每个个体, 以等概率选择 2 种策略, 检查结果是否得到改善, 如果是, 接受该顺序改变, 否则保持原状。如果调用次数超过最大迭代次数, 结束该过程。开发的算法伪代码见表 5。

表 5 算法 3 基于 TDO 的随机局部搜索
Table 5 Random local search based on TDO for Algorithm 3

input:	number of users N , the amount of data to be transferred by the user D_i , user's basic scheduling parameters, including RU type N_i^{sc} , number of RUs N_i^{RU} , and number of repetitions N_i^{rep} , total number of available subframes W , total number of available subcarriers B , number of iterations $I_{\text{iterations}}$, number of population members P_{popsize}
output:	best result so far B_{best}
1	function TDO-search ($N, D_i, N_i^{\text{sc}}, N_i^{\text{RU}}, N_i^{\text{rep}}, W, B, I_{\text{iterations}}, P_{\text{popsize}}$)
2	sort users in descending order to get a list using Eq.(18)
3	initialize the population: randomly shuffle the order and call 2d-packing to obtain different fitness values
4	$B_{\text{best}} \leftarrow 0$
5	for $t = 1 : I_{\text{iterations}}$
6	for $p = 1 : P_{\text{popsize}}$
7	if $P_{\text{probability}} < 0.5$, $P_{\text{probability}} = \text{rand}$
8	strategy 1: feeding by eating carrion (exploration phase)
9	select carrion for the i th Tasmanian Devil. If the fitness value of the carrion is better, then it moves towards the carrion (that is, randomly disrupting several elements based on the carrion), otherwise it moves away from the carrion. calculate the fitness value at the new position, update the position if it is better (update vector order), otherwise keep the original position
10	else
11	strategy 2: feeding by eating prey (exploitation phase)
12	stage 1: prey selection and attacking
13	similar to strategy 1
14	stage 2: prey chasing
15	calculate a new position with the Tasmanian devil's position as the center of the circle (that is, disrupt the order of a specific number of consecutive elements based on the original order). if the objective function at the new position is better than before, then update the position
16	end if
17	end for
18	save the best proposed solution O_{order} so far
19	end for
20	return B_{best}
21	end function

2.4 算法复杂度分析

在算法的基本调度参数确定过程中, 总用户数为 N , 时间复杂度为 $O(N)$ 。后续执行最少浪费优先策略对用户计算资源浪费函数, 假设该阶段确定调度的用户数为 $N_{\text{scheduled}}$, 时间复杂度为 $O(N_{\text{scheduled}} \cdot B)$ 。使用袋獾优化算法进行最优调度顺序的迭代过程中, 种群中的所有个体都要进行迭代搜索, 直至达到最大迭代次数为止。当种群规

模为 P_{popsize} ，最大迭代次数为 $I_{\text{iterations}}$ ，时间复杂度为 $O(P_{\text{popsize}} \cdot I_{\text{iterations}} \cdot N_{\text{scheduled}} \cdot B)$ 。由此可见，总的算法复杂度为 $O(P_{\text{popsize}} \cdot I_{\text{iterations}} \cdot N_{\text{scheduled}} \cdot B + N)$ 。

3 仿真结果及分析

本节通过数据分析软件对提出的算法进行仿真验证，此外，在仿真时为进一步证明算法的有效性，将已有文献资源调度算法与本算法进行比较。通过仿真结果，可以看出本文提出的面向高吞吐量的NB-IoT低轨卫星物联网资源调度算法可以很大程度地提高系统的性能。

3.1 仿真场景描述

网络场景中包含单个卫星以及在卫星覆盖区域内均匀分布的用户，用户的数据包大小基于1.2节MAR周期业务服从形状参数为2.5的帕累托分布，最小值为20 bytes，最大值为200 bytes。主要仿真参数如表6所示。

为了进一步验证算法的有效性，将提出的算法与单子载波轮询调度算法、文献[10]提出的单子载波多背包贪心算法以及文献[2]提出的高效上行多子载波调度算法进行比较。

3.2 仿真结果分析

图6~图7给出了低轨卫星物联网系统中对卫星覆盖区域内用户进行分组后各组用户平均吞吐量以及各组调度用户比率的曲线，如前所述各组按照用户比例划分时域资源。

表6 主要仿真参数

Table6 Main simulation parameters

parameter	value
carrier frequency f /GHz	2
user number N	15 000
satellite altitude h_s /km	1 000
satellite beam diameter D /km	400
transmit power P_t /dBm	23
request data size D_i /bytes	[20,200]
delay constraint d_i /ms	[50,100]
required reliability R_i	[90%~99%]
total number of available subcarriers B	12
total number of available subframes W	36 000

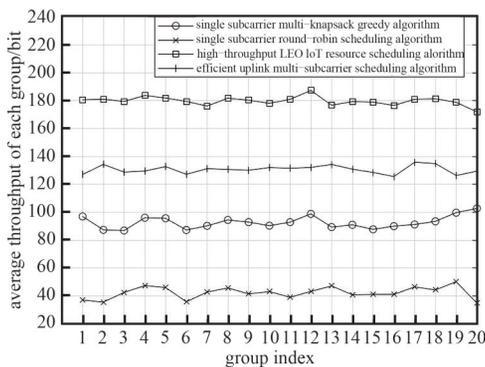


Fig.6 Relationship between the average throughput of each group and the group index
图6 各分组用户平均吞吐量与分组索引关系

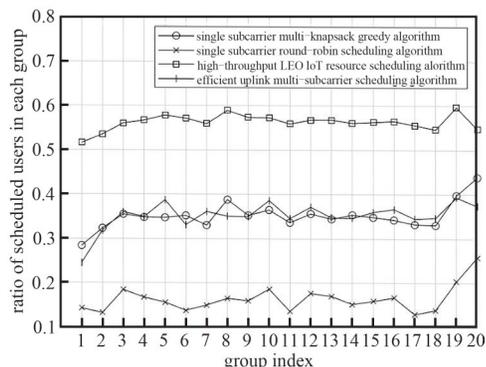


Fig.7 Relationship between ratio of scheduled users for each group and the group index
图7 各分组调度用户比率与分组索引关系

从曲线中可以看出本文提出的算法相比文献提出的算法在各组用户平均吞吐量以及各组调度用户比率上均有很大提升。同时，通过各组按比例分配时域资源，使得各组获得近似的性能，从而保证了用户之间公平性。

为了进一步研究系统性能，在同样的延迟约束以及可靠性约束下改变系统用户总数。对各组用户数加和，统计系统总吞吐量以及调度用户总数与系统总用户数的关系，如图8~图9所示。

从图8和图9中可以看出，本文提出的算法在吞吐量以及调度用户数方面的趋势具有显著特点。首先，曲线图呈现出线性增长的趋势，这是由于系统可用无线资源远远大于所有用户请求的资源，因此系统能够满足所有用户的需求。随着请求用户数逐渐趋于系统的容量，增长速率逐渐平缓，系统已经达到了资源利用的极限，然而该增长趋势仍然快于现有文献的算法。这种趋势的出现体现了本文算法的优点，即在资源利用效率方面具有良好的性能表现。

对比文献[2]提出的高效上行多子载波调度算法，在用户数小于7 000时，本文提出的算法性能略优，这是因为文献[2]的算法没有考虑差分多普勒的约束，使得部分被调度的用户无法成功解码数据，在用户数大于7 000时，即到达拐点时，本文提出的算法对应的曲线仍然继续增长，这是因为此时系统的时频资源已经充分利用，

而本文基于 TDO 算法确定了最优调度顺序，从而显著提升了算法性能。

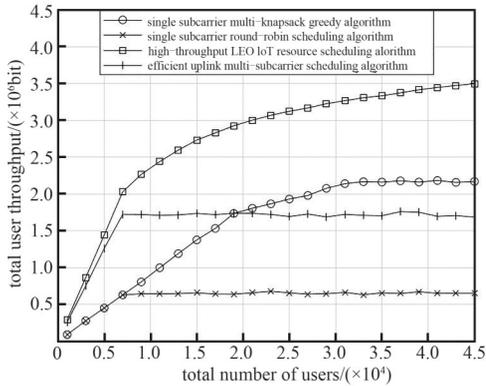


Fig.8 Total throughput versus total number of users graph
图 8 总吞吐量随用户总数变化曲线图

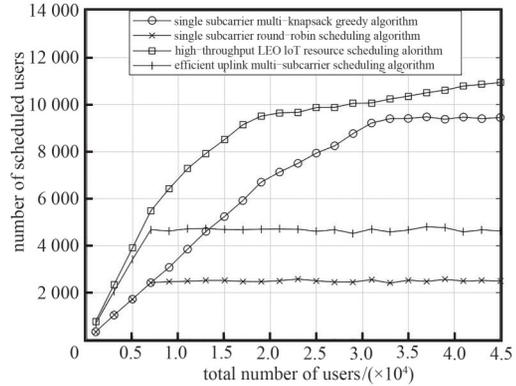


Fig.9 Number of scheduled users versus total number of users
图 9 调度用户数随用户总数变化曲线图

对比单子载波轮询调度算法以及文献[10]提出的单子载波多背包贪心算法，本文算法具有显著优势，增长趋势显然更快，这是因为上述 2 种算法均没有考虑用户的 QoS 约束，并且没有利用 NB-IoT 的重复特性。因为没有通过动态改变 RU 重复数来提高可靠性，所以在存在可靠性约束时，性能会大幅降低。通过仿真可以看出本文所提出的算法比其他算法表现更优秀，系统总吞吐量以及调度用户数比其他算法有较大提高。

图 10 展示了延迟约束与系统总吞吐量之间的关系。在仿真中，延迟约束下界统一设定为 50 ms，而图中的横坐标的值表示延迟约束的上界，其值为 100，表示用户延迟约束在 [50,100]ms 内均匀分布，以此类推。随着延迟约束的增大，本文提出的算法系统总吞吐量呈现缓慢上升的趋势。这是因为随着延迟约束的增加，原本受限于延迟约束的用户得以重新获得调度。此外，曲线增长缓慢的原因在于影响系统性能最大的 QoS 约束是可靠性约束而非时延约束。

图 11 展示了可靠性约束与系统总吞吐量之间的关系，图中横坐标的值表示可靠性约束的下界，其值为 60，表示仿真中用户的可靠性在 [60%，69%] 内均匀分布，以此类推。假设用户总数为 35 000，可以看出随着可靠性约束减小，所有算法的系统总吞吐量都呈现较大提升。这是因为随着可靠性约束减小，系统中用户可以通过较低的 RU 重复数满足可靠性约束，从而系统有限时频资源可以容纳更多用户，系统吞吐量得以提升。

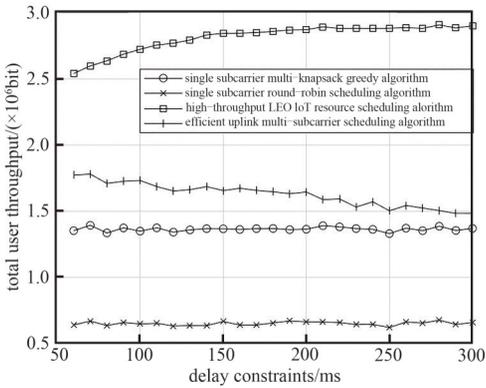


Fig.10 Total throughput versus delay constraints
图 10 总吞吐量随延迟约束变化曲线图

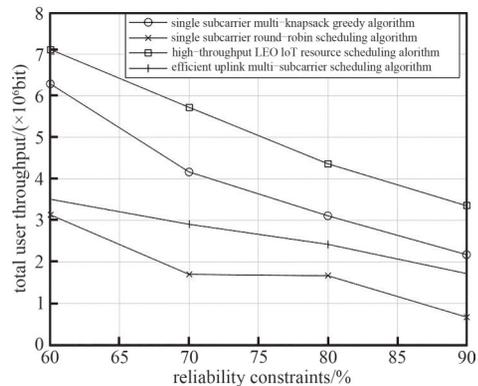


Fig.11 Total throughput versus reliability constraints
图 11 总吞吐量随可靠性约束变化曲线图

4 结论

本文构建了基于 NB-IoT 的低轨卫星物联网的资源调度问题，提出了一种面向高吞吐量的资源调度算法，尽可能最大化系统的吞吐量，提高系统的资源利用率。综合考虑用户业务 QoS 需求差异、NB-IoT 的特性以及低轨卫星的特点，针对用户间的较大差分多普勒频移，对用户进行分组，对于每组用户考虑 QoS 约束。基于最少浪费优先策略确定用户在整个资源池的位置，同时使用仿生算法迭代确定用户的最优调度顺序。仿真结果表明，相比现有文献的资源调度方案，本文提出的资源调度算法能够显著提高系统的吞吐量。

参考文献：

- [1] 彭明阳,张晨,张更新,等. 低轨星座的跳波束资源调度策略[J]. 太赫兹科学与电子信息学报, 2023,21(12):1429-1439. (PENG Mingyang, ZHANG Chen, ZHANG Gengxin, et al. Beam-hopping resource scheduling strategy of Leo constellation[J]. Journal of Terahertz Science and Electronic Information Technology, 2023,21(12):1429-1439.) doi:10.11805/TKYDA2021396.
- [2] LIANG Jiaming, WU Kunru, CHEN J J, et al. Energy-efficient uplink resource units scheduling for ultra-reliable communications in NB-IoT networks[J]. Wireless Communications and Mobile Computing, 2018(3):1-17. doi:10.1155/2018/4079017.
- [3] 陈玮. 窄带物联网中资源分配算法研究[D]. 北京:北京邮电大学, 2019. (CHEN Wei. Research on resource allocation algorithm in narrowband internet of things[D]. Beijing:Beijing University of Posts and Telecommunications, 2019.)
- [4] 张旺林. NB-IoT上下行链路资源调度算法研究[D]. 重庆:重庆邮电大学, 2022. (ZHANG Wanglin. Research on NB-IoT uplink and downlink resource scheduling algorithm[D]. Chongqing, China:Chongqing University of Posts and Telecommunications, 2022.) doi:10.27675/d.cnki.gcydx.2022.000045.
- [5] YU Yaju, TSENG S C. Downlink scheduling for narrowband Internet of Things(NB-IoT) systems[C]// 2018 IEEE 87th Vehicular Technology Conference(VTC Spring). Porto, Portugal:IEEE, 2018:1-5. doi:10.1109/VTCSpring.2018.8417666.
- [6] MALIK H, PERVAIZ H, MAHTAB-ALAM M, et al. Radio resource management scheme in NB-IoT systems[J]. IEEE Access, 2018(6):15051-15064. doi:10.1109/ACCESS.2018.2812299.
- [7] METWALY S S, AHMED M, El-Haleem A, et al. NOMA based matching game algorithm for narrowband Internet of Things(NB-IoT) system[J]. Ingénierie des Systèmes d' Inf, 2020(25):345-350.
- [8] MOSTAFA A E, ZHOU Yong, WONG V W S. Connectivity maximization for narrowband IoT systems with NOMA[C]// 2017 IEEE International Conference on Communications (ICC). Paris, France:IEEE, 2017:1-6. doi:10.1109/ICC.2017.7996362.
- [9] BARBAU R, DESLANDES V, JAKLLARI G, et al. NB-IoT over GEO satellite:performance analysis[C]// 2020 10th Advanced Satellite Multimedia Systems Conference and the 16th Signal Processing for Space Communications Workshop(ASMS/SPSC). Graz, Austria:IEEE, 2020:1-8. doi:10.1109/ASMS/SPSC48805.2020.9268829.
- [10] KODHELI O, MATURO S, CHATZINOTAS S, et al. NB-IoT via LEO satellites:an efficient resource allocation strategy for uplink data transmission[J]. IEEE Internet of Things Journal, 2021(7):5094-5107.
- [11] ANON. Cellular system support for ultra-low complexity and low throughput Internet of Things(CIoT):3GPP TR 45.820[R/OL]. [2024-02-27]. <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2719#>.
- [12] 周晨曦. 基于多业务场景的窄带物联网资源调度算法研究[D]. 北京:北京邮电大学, 2021. (ZHOU Chenxi. Research on NB-IOT resource scheduling algorithm based on multi service scenarios[D]. Beijing:Beijing University of Posts and Telecommunications, 2021.) doi:10.26969/d.cnki.gbydu.2021.001177.
- [13] KODHELI O, MATURO N, ANDRENACCI S, et al. Link budget analysis for satellite-based narrowband IoT systems[C]// Ad-Hoc, Mobile, and Wireless Networks. Berlin, Germany:Springer International Publishing, 2019:259-271. doi:10.1007/978-3-030-31831-4_18.
- [14] Technical Specification Group Radio Access Network. Physical layer procedures: 3GPP TS 36.213[R/OL]. [2024-02-27]. https://ptabdata.blob.core.windows.net/files/2017/IPR2017-01508/v4_Ex.%201004%20-%20TS%2036.213%20v8.2.0.pdf.
- [15] JACOBSSON M, ROHNER C. Estimating packet delivery ratio for arbitrary packet sizes over wireless links[J]. IEEE Communications Letters, 2015,19(4):609-612. doi:10.1109/LCOMM.2015.2398443.
- [16] KODHELI O, ANDRENACCI S, MATURO N, et al. Resource allocation approach for differential Doppler reduction in NB-IoT over LEO satellite[C]// 9th Advanced Satellite Multimedia Systems Conference and the 15th Signal Processing for Space Communications Workshop. Berlin, Germany:IEEE, 2018:1-8. doi:10.1109/ASMS-SPSC.2018.8510724.
- [17] ANON. Requirements for Evolved UTRA(E-UTRA) and Evolved UTRAN(E-UTRAN):3GPP TR 25.913[R/OI]. [2024-02-27]. <https://itcspec.com/archive/3gpp-specification-tr-25-913/>.
- [18] KELLERER H, PFERSCHY U, PISINGER D. Knapsack problems[M]. Berlin, Heidelberg:Springer, 2004.
- [19] 王泽昊. NB-IoT调度算法的优化设计及性能评估[D]. 兰州:兰州大学, 2020. (WANG Zehao. Optimization design and performance evaluation of NB-IoT scheduling algorithm[D]. Lanzhou, China:Lanzhou University, 2020.) doi:10.27204/d.cnki.glzhu.2020.000445.
- [20] WEI Lijun, ZHANG Defu, CHEN Qingshan. A least wasted first heuristic algorithm for the rectangular packing problem[J]. Computers & Operations Research, 2009,36(5):1608-1614. doi:10.1016/j.cor.2008.03.004.
- [21] DEGHANI M, HUBÁLOVSKÝ Š, TROJOVSKÝ P. Tasmanian devil optimization: a new bio-inspired optimization algorithm for solving optimization algorithm[J]. IEEE Access, 2022(10):19599-19620. doi:10.1109/ACCESS.2022.3151641.