

低轨巨星座多品类业务流低复杂度分段路由方法

王宁远^{1,2}, 刘 亮¹, 陈 东^{1,2}, 刘 欢³, 郝时光¹

(1. 中国空间技术研究院通信与导航卫星总体部, 北京 100094; 2. 鹏城实验室, 广东深圳 518052;
3. 北京空间机电研究所, 北京 100094)

摘 要: 含有星间链路的低轨巨星座网络在全球多种业务回传至有限地理区域场景下会产生严重的网络拥塞问题,集中式的流量规划可以在一定程度上实现负载均衡.然而大规模网络规划的计算时间开销无法满足低轨星座的动态性要求.为此,本文提出了低复杂度的多品类流分段路由(MCFSR)算法,将星座-地面网络依据负载情况划分为两个分区,并在分区内对规划算法的精度与复杂度之间进行权衡,以达到降低算法整体复杂度的目的.同时,对于规划算法,本文提出了复杂度可调的改进的完全多项式时间近似(IFPTA)算法,用于分区内的路由规划,在计算复杂度不变的情况下使算法吞吐量更接近最优值.仿真结果证明了本文提出的MCFSR算法在巨星座场景下可以使多业务回传的总吞吐量接近最优,且时间复杂度开销远低于其他同类算法.

关键词: 低轨巨星座; 多业务回传; 负载均衡; 多品类流; 分段路由; 时间复杂度

中图分类号: TN915.41; TP393.4 **文献标识码:** A **文章编号:** 0372-2112(2021)11-2124-09

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.12263/DZXB.20201335

A Low Time Complexity Segment Routing Approach for Multi-Commodity Traffic Flow in Mega LEO Constellation

WANG Ning-yuan^{1,2}, LIU Liang¹, CHEN Dong^{1,2}, LIU Huan³, HAO Shi-guang¹

(1. Institute of Telecommunication and Navigation Satellites, China Academy of Space Technology, Beijing 100094, China;

2. Peng Cheng Laboratory, Shenzhen, Guangdong 518052, China;

3. Beijing Institute of Space Mechanics & Electricity, Beijing 100094, China)

Abstract: The LEO mega-constellation network with inter-satellite links faces serious network congestion when multi-commodity traffics return to a limited geographical area from all over the world, which can be alleviated by centralized traffic planning. However, the computing time cost of large-scale network planning cannot meet the dynamic requirements of LEO constellation. Therefore, this paper proposes a low complexity Multi-Commodity Flow Segment Routing (MCFSR) algorithm, which divides the constellation-ground network into two zones according to different link payload, and balances between accuracy and complexity of the planning algorithm in each zone, so as to reduce the overall time complexity. Meanwhile, for the planning algorithm, this paper proposes an Improved Fully Polynomial Time Approximation (IFPTA) algorithm with adjustable precision and complexity for routing programming in each zone, which improves the planning accuracy without changing the computational complexity. Simulation results show that the total throughput of multi traffic backhaul programmed by proposed MCFSR algorithm approaches the optimal, with the time complexity far less than other similar algorithms.

Key words: LEO mega-constellation; multi-traffic backhaul; load balancing; multi-commodity flow; segment routing; time complexity

1 引言

低轨星座网络具有全球无缝覆盖、高鲁棒性、低时延等特点,可以作为地面通信系统的有效备份和良好

补充^[1,2],天地一体化的设计趋势更是给未来信息网络带来了新的可能^[3,4].近年来,国内外低轨星座建设浪潮如火如荼,以 Starlink、Lightspeed 等为代表的含有星

间链路的低轨巨星座网络工程已经进入实际部署阶段^[5,6]. 几百至上万颗卫星的巨大规模给星座带来了可接入用户数量和网络容量的大幅提升,但也使得网络拓扑结构更加复杂,且网络中星间、星地的切换由于卫星密度增大而更加频繁. 因此,地面成熟的网络路由技术难以直接应用^[7].

低轨星座具有优秀的全球覆盖能力,可用于物联网和移动通信等多种业务的数据回传^[8]. 考虑到低轨星座网络星间链路传输速率受限,当全球分布的业务回传到位于限定范围的接收站时,容易造成接收站附近星间链路的传输拥塞. 另一方面,空间星座的高动态性导致网络负载分布快速变化. 因此,为了确保巨型低轨星座网络多业务回传的服务质量,需要研究低复杂度的负载均衡路由算法,实现回传路由的快速规划.

低轨星座的多业务回传路由问题可以用多品类流(Multi-Commodity Flow, MCF)模型来描述^[9-13],其中,不同接收站回传的业务分别对应于不同的品类. 为了使流的吞吐量接近容量上界,有下列相关的解决方法. 基于Lagrange松弛和线性规划的文献^[10, 11]给出了在更短时间内解决任意多品类流的算法. Garg与Konemann提出了完全多项式时间近似(Fully-Polynomial-Time Approximation, FPTA)的低复杂度多品类流算法^[12],算法进行多次迭代,每次迭代通过最短路径进行路由并更新路径长度,长度随流量增加而增长. Karakosta进一步降低了FPTA算法的复杂度^[13],利用一次Dijkstra算法计算了所有共有源节点的路径长度,并给出了容量和各边流量的显式算法. 然而对于巨型LEO星座,上述算法时间复杂度大,路由运算耗时过长,无法满足实时计算要求,实用性大大降低. 另一方面,上述算法得到的最大吞吐量结果偏小,使得链路容量无法得到充分利用.

分段路由(Segment Routing, SR)是一种基于源路由的隧道技术,允许主机和边缘路由器通过某些分段进行网络通信,以达到路径间灵活组合的目的^[14-16]. 黄建洋等针对多路径间的流调度问题提出了基于最优化的动态路由算法^[17],但求解最优等价路径的算法复杂度本质上没有降低,无法适应卫星网络动态变化的特点. Liu等给出了一种低轨星座回传场景的分段路由算法^[18],然而算法只能解决单一业务的回传问题,无法适用于多品类业务回传到多个站点的问题.

针对上述的多业务低轨星座回传算法复杂度过高的问题,本文提出了低复杂度的多品类流分段路由算法(Multi-Commodity Flow Segment Routing algorithm, MCFSR). 首先,本文提出了改进的完全多项式时间近似(Improved Fully-Polynomial-Time Approximation, IFPTA)多品类流算法,使求解的吞吐量更逼近最优值,且

可以在精度和复杂度间进行折中. 其次,根据网络负载情况划分轻、重负载区. 综合以上两点,本文提出了MCFSR算法. 算法首先使用低精度的IFPTA算法将轻负载区的业务按比例向重负载区边缘路由,直至达到轻负载区的容量,得到两者负载映射关系及路由路径. 而后使用高精度IFPTA算法在重负载区内对边缘与内部的业务按比例进行路由规划,使得网络吞吐量接近最优值. 最后,根据优化得到的吞吐量以及轻负载区业务占比反推出轻负载区各路径的真实流量.

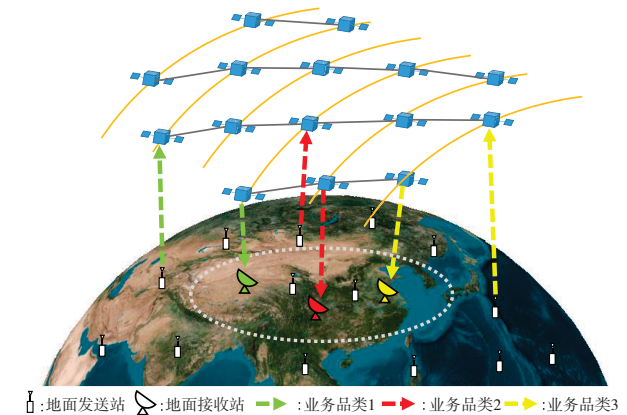


图1 低轨星座多点回传模型

2 巨星座多品类回传模型

2.1 网络拓扑的分块邻接矩阵模型

2.1.1 星座-地面网络的邻接矩阵描述

本文考虑多品类的业务通过低轨卫星星座从全球回传至国土多个地面站的场景,每种业务需要回传至各自的接收站,且所有接收站均分布在同一有限的区域内,站间无连接,如图1所示. 现有星座卫星的集合 $V_{\text{Sat}} = \{v_1, \dots, v_{N_s}\}$, 地面发送站集合 $V_T = \{v_1, \dots, v_{N_T}\}$, 接收地面站集合 $V_R = \{v_1, \dots, v_{N_R}\}$. 任意时刻,卫星-地面网络的拓扑结构可由分块的邻接矩阵 $A \in N \times N$ 表示:

$$A = \begin{bmatrix} A_{\text{Sat}} & \mathbf{0} & A_R \\ A_T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (1)$$

其中 $N = N_{\text{Sat}} + N_T + N_R$, 方阵 $A_{\text{Sat}} \in N_{\text{Sat}} \times N_{\text{Sat}}$ 为低轨星座的邻接矩阵,描述了低轨星座网络的拓扑连接关系, $A_T \in N_T \times N_{\text{Sat}}$ 与 $A_R \in N_{\text{Sat}} \times N_R$ 分别为发射站到星座、星座到接收站的连接关系矩阵. 由于本场景为回传模型,不考虑接收站到卫星或卫星到发送站的链路,且地面站间无连接,因此对应部分为全零矩阵. 由于地球自转与卫星运行,星地相对位置不断变化,因此这里取任一时刻的星地网络拓扑分析其路由问题.

2.1.2 星座场景

考虑包含 P_s 个轨道面,每个轨道面 Q_s 颗卫星的

Walker-Delta 星座, 卫星总数 $N_{\text{Sat}} = P_S Q_S$. 星座中的每颗卫星包含 4 条星间链路, 其中两条分别与相邻的两颗同轨卫星连接, 另外两条分别与两个相邻的同向轨道中的卫星连接. 由此, 构成邻接矩阵 \mathbf{A}_{Sat} 的元素 $a_{i,j}^{\text{Sat}} \in \mathbf{A}_{\text{Sat}}$ 满足:

$$a_{i,j}^{\text{Sat}} = \begin{cases} C_{\text{ISL}}, \exists (v_i, v_j), \forall v_i, v_j \in V_{\text{Sat}}, i \neq j \\ 0, \text{ 否则} \end{cases} \quad (2)$$

其中, (v_i, v_j) 表示第一项指向第二项的星间链路.

2.1.3 星地接入场景

回传场景中, 由于一般的小型终端无法直接向卫星发送数据, 通常将数据集中到临近的地面发送站, 再统一发送给卫星. 因此这里只考虑地面站 (包括发送站与接收站) 与卫星星座的通信.

星地连接关系建模如下: 将全球被星座覆盖的区域进行地理网格分区, 按经纬度等分成 $P_T \times Q_T$ 总计 N_T 个小区, 且 $N_T \leq N_{\text{Sat}}/2$, 使得每个小区中都至少包括一颗升轨星和一颗降轨星. 每个小区包含一个地面发送站, 且 N_R 个地面接收站位于不同小区内, $N_R \leq N_T$. 每个发送或接收站都可同时与区域上空的 N_{Connect} 颗卫星保持单向的链路.

对于 Walker-Delta 星座, 地面站同时可见升轨卫星和降轨卫星, 因此在业务发送时需要进行选择. 我们假设地面终端仅可以与 2 颗卫星保持星间链路, 即 $N_{\text{Connect}} = 2$. 假设地面站 $v_T \in V_T$ 所在小区上空升轨、降轨卫星集合分别为:

$$V_{\text{Visible}}^{\text{A}}(v_T) = \{v_1^{\text{A}}, v_2^{\text{A}}, \dots, v_{N_{\text{VA}}}^{\text{A}}\} \quad (3)$$

$$V_{\text{Visible}}^{\text{D}}(v_T) = \{v_1^{\text{D}}, v_2^{\text{D}}, \dots, v_{N_{\text{VD}}}^{\text{D}}\} \quad (4)$$

其中 $V_{\text{Visible}}^{\text{A}}(v_T) \subset V_{\text{Sat}}, V_{\text{Visible}}^{\text{D}}(v_T) \subset V_{\text{Sat}}$. 由于地面网格的划分过程保证了 v_T 所处小区内至少存在一颗升轨星和一颗降轨星, 因此 $2 \leq N_{\text{VA}} + N_{\text{VD}} < N_S$. 任意选择卫星 $v_{n_{\text{VA}}}^{\text{A}}(v_T) \in V_{\text{Visible}}^{\text{A}}(v_T), v_{n_{\text{VD}}}^{\text{D}}(v_T) \in V_{\text{Visible}}^{\text{D}}(v_T)$, 并分别根据收发需要建立单向星地链路, 上行、下行链路容量分别为 C_U 和 C_D . 通过这种接入方式, 路由算法会自动根据升降轨的负载情况选择新的流经由升轨还是降轨链路, 从而避免了整数规划的 NP-Hard 问题. 由此, 星地连接矩阵 \mathbf{A}_T 与 \mathbf{A}_R 可分别表示为:

$$a_{i,j}^{\text{T}} = \begin{cases} C_U, \exists (v_i, v_j), \forall v_i \in V_T, \forall v_j \in V_{\text{Sat}} \\ 0, \text{ 否则} \end{cases} \quad (5)$$

$$a_{i,j}^{\text{R}} = \begin{cases} C_D, \exists (v_i, v_j), \forall v_i \in V_{\text{Sat}}, \forall v_j \in V_R \\ 0, \text{ 否则} \end{cases} \quad (6)$$

其中 $a_{i,j}^{\text{T}} \in \mathbf{A}_T, a_{i,j}^{\text{R}} \in \mathbf{A}_R$.

2.2 业务权重分布模型

地面每个小区都会产生所有品类的业务, 同一品类的业务数据需要回传到同一地面站, 因此这里业务数量亦为 N_R . 对于同一品类的业务, 全球各小区在当前时刻按照固定的权重分布产生数据. 由于卫星自身不产生业务, 全部业务都来源于发送节点集 V_T , 而业务最终都流向接收节点集 V_R . 由此形成业务矩阵 $\mathbf{B} \in N_T \times N_R$:

$$\mathbf{B} = \begin{bmatrix} \beta_{1,1} & \cdots & \beta_{1,N_R} \\ \vdots & \ddots & \vdots \\ \beta_{N_T,1} & \cdots & \beta_{N_T,N_R} \end{bmatrix} \quad (7)$$

其中元素 $\beta_{i,j} \in \mathbf{B}$ 表示地面节点 $v_i \in V_T$ 发给地面节点 $v_j \in V_R$ 的业务比例, 第 i 行表示 v_i 发送的全部业务, 第 j 列表示 v_j 接收的全部业务. 且有

$$\sum_i \sum_j \beta_{i,j} = 1 \quad (8)$$

2.3 多品类流回传的吞吐量优化问题

基于低轨巨星座的多业务回传问题可用“多品类流”(Multi-Commodity Flow)问题描述. 将发送到同一个目的节点 t 的全部流定义为一个“品类”(Commodity)^[9], 所有品类流共用一张网络回传, 且不同品类之间的流量不能兼容. 更进一步, 在各品类业务权重分布固定的情况下, 该问题可以被归纳为多品类流中的最大并行流问题(Maximum Concurrent Flow Problem). 该问题的线性规划(Linear Programming, LP)形式为:

$$\text{maximize } \lambda_{f(P_k)} \quad (9)$$

$$\text{s.t. } \sum_{P_k: a \in P_k} f(P_k) \leq C(a), \forall a \in \mathbf{A} \quad (10)$$

$$\sum_{P \in \mathcal{P}_k} f(P_k) \geq \lambda \beta_k, \forall k \quad (11)$$

$$f(P_k) \geq 0, \forall P_k \in \mathcal{P}_k \quad (12)$$

其中 λ 为业务矩阵 \mathbf{B} 的比例系数, 由于矩阵中元素之和为 1, 因此 λ 的物理意义为整个承载网络的总吞吐量. $a \in \mathbf{A}$ 为网络中的边, \mathcal{P}_k 为品类 k 包含的所有路径集合, $P_k \in \mathcal{P}_k$ 为品类 k 中包含的某条路径, $f(P_k)$ 为该路径上路由的流量, β_k 为品类 k 包含的业务比例, $C(a)$ 为边 a 的容量. 线性规划算法可以求解多品类流问题, 然而其算法复杂度最低为 $O(N^{3.5})$ ^[19], 在星座规模较大时计算时间开销巨大, 难以有效适应网络负载分布的快速动态变化, 因此下面探讨低复杂度的规划算法.

3 基于多品类流的分段路由算法

网络吞吐量和时间复杂度是评价负载均衡算法的两个重要指标. 对于巨星座网络, 庞大的网络规模及更加频繁的切换需求大大提高了对负载均衡算法的时间

复杂度要求. 为了以较低的复杂度实现低轨巨星座回传场景的吞吐量优化, 本文提出了基于多品类流的分段路由 (Multi-Commodity Flow Segment Routing algorithm, MCFSR) 算法. 算法将网络划分为规模较小但资源紧张的“重负载区”和规模较大而资源富余的“轻负载区”. 轻负载区中产生的多品类业务总量为 λ_{LZ} , 首先路由至重负载区的边缘. 重负载区产生的业务总量为 λ_{HZ} , 与轻负载区传入的业务一起路由规划至对应的接收站. 由此可以对小范围的重负载区采用精度较高但开销较大的路由规划, 而对大范围轻负载区采用低精度却节约开销的路由规划, 从而达到平衡网络利用率和计算开销的目的. 如图 2 所示.

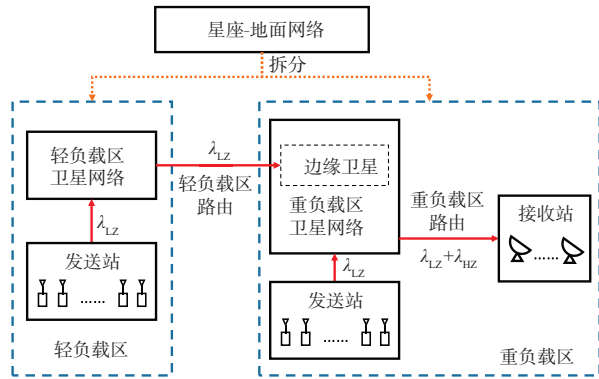


图2 基于多品类流的分段路由(MCFSR)算法

3.1 轻、重负载区的划分与子图的构建

根据 2 所述的场景模型和业务模型, 在回传场景中, 接收地面站分布于有限的地理区域内. 显然, 越靠近接收地面站区域的星座网络负载越重. 由此, 本文将卫星-地面网络划分为轻负载区和重负载区. 两部分区域在路由的计算上相互独立, 其中轻负载区负责将全球业务回传到重负载区的边缘, 重负载区负责将边缘的业务按类型分配到各个地面接收站. 我们将重负载区、轻负载区做出如下定义:

定义 1 以地面小区为单位, 在星座覆盖的区域内划分出某一经、纬度范围内的区域, 由 $P_H \times Q_H$ 个小区组成, 区域内包含全部地面接收站, 且 $P_H < P_G$, $Q_H < Q_G$. 区域内的全部节点及链路构成的网络称为重负载区, 区域外的全部节点及链路构成的网络称为轻负载区.

3.1.1 子图的拆分与构建

首先是讨论重负载区. 根据定义 1, 可得重负载区内的地面发送站的集合 $V_T^{HZ} \subset V_T$, 接收站的集合 V_R 保持不变. 而由单个小区可见卫星集合 (3)、(4), 可得重负载区升、降轨卫星集合分别为:

$$V_{HZ}^A = \bigcup V_{Visible}^A(v_T), \forall v_T \in V_T^{HZ} \quad (13)$$

$$V_{HZ}^D = \bigcup V_{Visible}^D(v_T), \forall v_T \in V_T^{HZ} \quad (14)$$

由于重负载区的升降轨卫星间无直接链路, 因此为两张网络. 重负载区内卫星-地面节点可表示为:

$$V_{HZ} = V_{HZ}^A \cup V_{HZ}^D \cup V_T^{HZ} \cup V_R \quad (15)$$

负载区子网络拓扑依旧采用邻接矩阵表示. 对于重负载区, 其邻接矩阵 $A_{HZ} \in N \times N$ 中的元素满足:

$$a_{ij}^{HZ} = \begin{cases} a_{ij}, v_i \in V_{HZ}, v_j \in V_{HZ}, a_{ij} \in A \\ 0, \text{否则} \end{cases} \quad (16)$$

轻负载区则稍有不同, 其子网络由两部分构成: 轻负载区节点和虚节点. 首先根据定义, 轻负载区节点集为 V_{LZ} 的补集:

$$V_{LZ} = \complement_V V_{HZ} \quad (17)$$

子网络拓扑可表示为 $A_{LZ}^{Def} \in N \times N$, 其中元素满足:

$$a_{ij}^{LZ} = \begin{cases} a_{ij}, v_i \in V_{LZ}, v_j \in V_{LZ}, \forall i, j \\ 0, \text{否则} \end{cases} \quad (18)$$

由于业务的目的节点均在重负载区中, 除了定义中轻负载区的节点和链路, 我们建立与业务类型数量 N_R 数目相同的虚节点 $\{v_1^{Virtual}, \dots, v_{N_c}^{Virtual}\}$, 作为各个业务类型的目的节点. 在其中任取一节点, 以 $v_1^{Virtual}$ 为例, 与所有被负载区边界“斩断”链路的轻负载区节点相连. 其连接矩阵 $A_{LZ}^{Virtual} \in N \times N_R$ 为:

$$a_{ij}^{Virtual} = \begin{cases} C_{ISL}, a_{ik} \in (A - A_{HZ} - A_{LZ}^{Def}), \\ \forall i, k \in \{1, \dots, N\}, j = 1 \\ 0, \text{否则} \end{cases} \quad (19)$$

另外 $N_c - 1$ 个节点建立由 $v_1^{Virtual}$ 指向自身的链路:

$$A_{Inf}^{Virtual} = [0 \ \infty \ \dots \ \infty] \in 1 \times N_R \quad (20)$$

这种做法一方面使每个业务类型都保有一个目的节点, 另一方面确保所有业务均通过 $A_{LZ}^{Virtual}$ 中链路回传至目的节点. 综上所述, 轻负载区的子网络拓扑可以表示为 $(N+1) \times (N+N_R)$ 的邻接矩阵:

$$A_{LZ} = \begin{bmatrix} A_{LZ}^{Def} & A_{LZ}^{Virtual} \\ 0 & A_{Inf}^{Virtual} \end{bmatrix} \quad (21)$$

3.1.2 子业务的拆分与构建

轻负载区的业务由轻负载区内剩余的地面节点向虚拟目的节点 $v_1^{Virtual}, \dots, v_{N_c}^{Virtual}$ 发送的业务构成, 业务矩阵为 $B_{LZ} \in N_T \times N_R, \beta_{ij}^{LZ} \in B_{LZ}$:

$$\beta_{ij}^{LZ} = \begin{cases} \beta_{ij}, v_i \in V_{LZ}, \beta_{ij} \in B, \forall i, j \\ 0, \text{否则} \end{cases} \quad (22)$$

由图 2 可知, 重负载区的业务由两部分构成: 一为重负载区内产生的业务, 二为经由轻负载区回传到重负载区的业务. 在轻负载区子图中, 我们设置了虚拟目的节点, 并建立了全部边缘节点与其中一个虚拟目的节点间的链路 $A_{LZ}^{Virtual}$. 实际上, $A_{LZ}^{Virtual}$ 中链

路代表了全部轻负载区指向重负载区的链路,即 $a_{ij}^{LZ \rightarrow HZ} \in \mathbf{A}_{LZ \rightarrow HZ} \in N \times N$:

$$a_{ij}^{LZ \rightarrow HZ} = \begin{cases} a_{i,j}^{\text{Total}}, v_i \in V_{LZ}, v_j \in V_{HZ}, \forall i, j \\ 0, \text{否则} \end{cases} \quad (23)$$

其中分配的流量即为需要传递到重负载区的流量. 在完成轻负载区的流量分配后, $\mathbf{A}_{LZ \rightarrow HZ}$ 链路中承载的业务 k 流量为 $f_{i,j,k}^{LZ \rightarrow HZ} \in \mathbf{F}_{LZ \rightarrow HZ}$, $\mathbf{F}_{LZ \rightarrow HZ} \in N \times N \times N_R$. 从 (9)~(12) 可知, 该多品类流问题属于最大并行流问题, 因此 $f_{i,j,k}^{LZ \rightarrow HZ}$ 满足:

$$\sum_{i,j,k} f_{i,j,k}^{LZ \rightarrow HZ} = \lambda_{LZ} \sum_{m,n} \beta_{m,n}^{LZ}, \beta_{m,n}^{LZ} \in \mathbf{B}_{LZ} \quad (24)$$

因此, 将轻负载区输出流量除以 λ_{LZ} , 即可得到轻负载区输入业务比例. 在重负载区中, 我们将这部分业务看作由边缘卫星发送给对应地面接收节点. 这部分业务矩阵 $\mathbf{B}_{LZ \rightarrow HZ} \in N_T \times N_R$ 中元素 $\beta_{ij}^{LZ \rightarrow HZ}$ 可表示为:

$$\beta_{ij}^{LZ \rightarrow HZ} = \begin{cases} \sum_m f_{m,i,j}^{LZ \rightarrow HZ} / \lambda_{LZ}, v_i \in V_{HZ} \\ 0, \text{否则} \end{cases} \quad (25)$$

重负载区的另一部分业务为区内的固有业务, 其业务矩阵 $\mathbf{B}_{HZ \rightarrow HZ} \in N_T \times N_R$ 中元素 $\beta_{ij}^{HZ \rightarrow HZ}$ 可表示为:

$$\beta_{ij}^{HZ \rightarrow HZ} = \begin{cases} \beta_{i,j}, v_i \in V_{HZ}, v_j \in V_{HZ}, \forall i, j \\ 0, \text{否则} \end{cases} \quad (26)$$

其中 $\beta_{i,j} \in \mathbf{B}$. 重负载区总业务矩阵为:

$$\mathbf{B}_{HZ} = \mathbf{B}_{LZ \rightarrow HZ} + \mathbf{B}_{HZ \rightarrow HZ} \quad (27)$$

3.2 多品类流的容量优化算法

在低轨星座多业务回传场景中, 业务存在多个源节点和目的节点, 需要采用多品类流算法求解. 我们这里提出改进的完全多项式时间近似算法 (IFPTA), 主要改进在于将迭代后的不可行流变为可行流的比例因子的选取, 使结果更接近最优.

IFPTA 算法流程如算法 1 所示. 由于本文中定义发往同一目的节点 $t \in \{1, \dots, N_R\}$ 的流为同一品类, 下面利用符号 t 代表该品类流. 其中 $a \in \mathbf{A} \in N \times N$ 为图中的链路, 非负实数 $l(a)$ 为链路 a 的权重; λ_t 为接收节点 t 的总流量; $f_a^t \in \mathbf{F}_t$ 为品类 t 在链路 a 中所占流量, 满足 $\sum_a f_a^t \leq C_t$, $\mathbf{F}_t \in N \times N$ 为品类 t 的流量矩阵, $f_a^{s,t} \in \mathbf{F}$ 为 (s, t) 对在链路 a 中所占流量, $\mathbf{F} \in N \times N \times N_R$ 为多品类流量张量; 矩阵 $\mathbf{B}_{\text{Zoom}} = \sigma \mathbf{B}$ 为业务矩阵的放缩, $\sigma \in \mathbb{R}^+$, 目的是适当放大单次的派发流量, 以控制迭代次数; $\delta \in \mathbb{R}^+$ 为一个很小的正常数, 用于初始化各边权重; $u(a)$ 表示链路 a 的剩余容量.

算法 1 改进的 FPTA 多品类流算法 (IFPTA)

- 1 **输入:** 邻接矩阵 \mathbf{A} , 业务矩阵 \mathbf{B} , 迭代步长 ε , 业务矩阵缩放因子 σ , 权重初始化常数 δ
- 2 **输出:** 最大容量 λ , 多品类流量 $f_a^{s,t} \in \mathbf{F}$
- 3 **初始化:** $l(a) \leftarrow \delta / C_t, \forall a \in \mathbf{A}, \mathbf{B}_{\text{Zoom}} = \sigma \mathbf{B}, \Omega = 0$

4 算法开始

5 WHILE $\Omega < 1$ // 第一层循环

6 $\mathbf{B}_{\text{Zoom}}^* \leftarrow \mathbf{B}_{\text{Zoom}}$

7 FOR $t = 1: N$ // 第二层循环

8 WHILE $\mathbf{B}_{\text{Zoom}}^*$ 的第 t 列不全为零 // 第三层循环

9 $[P(:, t), d_t] \leftarrow$ 利用 Dijkstra 算法计算所有节点到 t 的最小权重路径 $P(:, t)$ 及权重 d_t

10 $u(a) \leftarrow C_t$

11 FOR $s = 1: N$ // 第四层循环

12 $C_b \leftarrow P(s, t)$ 在 $u(a)$ 下的瓶颈容量

13 IF $\mathbf{B}_{\text{Zoom}}^*(s, t) \leq C_b$

14 $f_a^{s,t} \leftarrow f_a^{s,t} + \mathbf{B}_{\text{Zoom}}^*(s, t), a \in P(s, t)$

15 $u(a) \leftarrow u(a) - \mathbf{B}_{\text{Zoom}}^*(s, t), a \in P(s, t)$

16 $f_s^{\text{temp}} \leftarrow \mathbf{B}_{\text{Zoom}}^*(s, t)$

17 $\mathbf{B}_{\text{Zoom}}^*(s, t) \leftarrow 0$

18 ELSE

19 $f_a^{s,t} \leftarrow f_a^{s,t} + C_b, \forall a \in P(s, t)$

20 $f_s^{\text{temp}} \leftarrow C_b$

21 $\mathbf{B}_{\text{Zoom}}^*(s, t) \leftarrow \mathbf{B}_{\text{Zoom}}^*(s, t) - C_b$

22 END IF

23 END FOR

24 $\Omega \leftarrow \Omega + \varepsilon \sum_s (f_s^{\text{temp}} \cdot d_t(s))$

25 $l(a) \leftarrow l(a)(1 + \varepsilon \cdot \sum_s f_s^{\text{temp}} / C_t), \forall a \in P(:, t)$

26 END WHILE

27 END FOR

28 END WHILE

29 $f_a^{s,t} \leftarrow C_t \cdot f_a^{s,t} / \max_{i,t} (f_a^{s,t}), \forall a \in \mathbf{A}, \forall s, t, s \neq t$

30 $\lambda_t \leftarrow \sum_a \sum_{s \in V} f_a^{s,t}, \forall t$ 与 t 相连的 a

31 $\lambda \leftarrow \sum_t \lambda_t, \forall t$

32 算法结束

该算法的核心思想是给各边设置权重, 每次利用 Dijkstra 算法计算最小权重和的路径, 根据业务矩阵进行路由, 并根据各边上路由的流量与容量的比值增大权重, 最后通过迭代实现多路径之间的负载均衡. 算法的精度由迭代步长 $\varepsilon \in (0, 1)$ 控制. ε 越小, 算法精度越高, 即越接近最优值. 算法共有四层循环. 第一层循环设置 $\Omega \geq 1$ 为跳出条件, 通过对第二层循环进行多次迭代使得业务分布均匀. 第二层循环利用了合并目的节点的思路来减少品类数量, 负责遍历每一个目的节点 t . 第三层循环在每一次循环中通过 Dijkstra 算法一次性得到所有源节点到该目的节点的最短路径, 并负责保证临时业务矩阵 $\mathbf{B}_{\text{Zoom}}^*$ 中的所有业务都进行了分配. 第四层循环负责遍历源节点 s , 并对每个 (s, t) 对中的业务按当前最短路径进行路由.

在该近似解法中,由于全部循环结束后,各边流量会超出边的容量,因此需要对流量进行等比例缩小. 在文献[12, 13]中所采用的方法为估计单边能够被分配的流量上界 $\log_{1+\varepsilon} C_l \cdot (1+\varepsilon)/\delta$, 再将分配流量缩小 $\log_{1+\varepsilon}(1+\varepsilon)/\delta$, 使得每条边的流量小于其容量. 然而该方法利用估计的上界作为比例因子, 在实际实施中往往使得缩小后的流量偏小, 各边的容量未得到充分利用. 因此, 我们给出一种新的比例因子计算方法. 作为一种数值解法, 每一条边被分配的流量是可以实际测得的, 可以计算出所有边中流量与容量的最大比值 $\max_{s,t} (f_a^{s,t})/C_l$, 作为比例因子进行缩小. 利用此方法, 一定存在至少一条边的流量达到其容量上界, 即在流量分配规则、计算量均不变的情况下, 比文献[12, 13]中方法更加接近最优值. 算法迭代次数越多、每次迭代步长 ε 越小, 则精度越高. 而由于算法的复杂度主要来源于循环内的部分, 因此比例因子的改进并不会影响算法复杂度.

3.3 全网业务回传的分段路由

基于以上的轻、重负载区子图和子业务的构建, 我们给出了多品类流分段回传路由算法(MCFSR)算法, 如算法2所示.

MCFSR算法通过两次调用IFPTA多品类流算法依次求解轻负载区和重负载区的业务流. 由于模型中全球业务的比例固定, 因此轻负载区传输到重负载区边缘的业务比例也是固定的. 由此, 这里可以把轻负载区的路由看作轻负载区业务到重负载区边缘的一种负载均衡的映射. 而重负载区的路由则是在较小的节点和链路范围内, 将全球所有业务按比例进行路由, 最终得到的总流量 λ_{Total} 即是网络能够承载的最大流量. 值得注意的是, 轻负载区的流量张量 $\mathbf{F}_{\text{LZ}}^{\text{Max}}$ 中存储着轻负载区业务从全球各个地面节点发送至重负载区边缘的最大可能流量, 而非最终的路由流量. 需要根据重负载区得出的总流量对轻负载区流量张量进行等比例缩小, 才能使重负载区与轻负载区的流量相匹配.

由于轻负载区链路资源富余, 可承载的流往往大于重负载区实际能接收的流, 因此轻负载区路由时的算法吞吐量无需过高, 因此可以取较大的 ε_{LZ} 以降低运算复杂度. 另一方面, 重负载区中链路资源十分紧张, 需要在较小的网络范围内进行更加精细的资源分配, 因此重负载区路由应取较小的 ε_{HZ} . 而未采用分段路由的全局路由算法为保证运算吞吐量, 只能统一采用较小的 ε , 因此复杂度较高.

3.4 复杂度分析

本文中复杂度计算的标准为最恶劣情况下算法运算的次数. 由于本文中对FPTA算法的改进未改变

算法2 多品类流分段回传路由算法(MCFSR)

- 1 输入: $\mathbf{A}_{\text{Total}}, \mathbf{B}_{\text{expand}}$, 地面接收站位置
- 2 输出: $\lambda_{\text{Total}}, \mathbf{F}_{\text{Total}}$
- 3 算法开始
- 4 根据地面接收站位置, 划分轻、重负载区.
- 5 根据3.2节中方法构建轻负载区的子图邻接矩阵 \mathbf{A}_{LZ} 和子业务矩阵 \mathbf{B}_{LZ} .
- 6 利用 $\mathbf{A}_{\text{LZ}}, \mathbf{B}_{\text{LZ}}$, 运行**算法1**, 取较大的迭代步长 ε_{LZ} , 解得轻负载区网络能够承载的最大流量 $\lambda_{\text{LZ}}^{\text{Max}}$ 和流量张量 $\mathbf{F}_{\text{LZ}}^{\text{Max}}$.
- 7 从 $\mathbf{F}_{\text{LZ}}^{\text{Max}}$ 中得到 $\mathbf{F}_{\text{LZ} \rightarrow \text{HZ}}$, 并根据3.2节中方法构建重负载区的子图邻接矩阵 \mathbf{A}_{HZ} 和子业务矩阵 \mathbf{B}_{HZ} .
- 8 利用 $\mathbf{A}_{\text{HZ}}, \mathbf{B}_{\text{HZ}}$ 运行**算法1**, 取较小的迭代步长 ε_{HZ} , 其中 $\varepsilon_{\text{HZ}} < \varepsilon_{\text{LZ}}$, 解得网络能够承载的总最大流量 λ_{Total} 和流量张量 \mathbf{F}_{HZ} .
- 9 根据解得的最大流量 λ_{Total} 反推轻负载区业务的实际流量. 由轻负载区业务在全球业务中的占比 $\theta_{\text{LZ}} = \sum_{i,j} \beta_{i,j}^{\text{LZ}} / \sum_{i,j} \beta_{i,j}$, 可得轻负载区实际流量 $\lambda_{\text{LZ}} = \lambda_{\text{Total}} \cdot \theta_{\text{LZ}}$, 流量张量 $\mathbf{F}_{\text{LZ}} = \lambda_{\text{LZ}} / \lambda_{\text{LZ}}^{\text{Max}} \cdot \mathbf{F}_{\text{LZ}}^{\text{Max}}$.
- 10 求得全局总流量张量为 $\mathbf{F}_{\text{Total}} = \mathbf{F}_{\text{HZ}} + \mathbf{F}_{\text{LZ}}$.
- 11 算法结束

其复杂度, 根据文献[13], FPTA算法对于 N 节点 M 链路的复杂度为

$$O(\varepsilon_{\text{HZ}}^{-2} (M^2 + N_{\text{R}} N)) \quad (28)$$

其中 N_{R} 为流的品类数. 考虑重负载区子图中节点数为 N_{HZ} , 链路数为 M_{HZ} , 轻、重负载区算法的迭代步长分别为 ε_{LZ} 和 ε_{HZ} . 可得到本文提出的MCFSR算法的复杂度为:

$$O \left(\begin{array}{l} \varepsilon_{\text{LZ}}^{-2} \left[(M - M_{\text{HZ}})^2 + N_{\text{R}} (N - N_{\text{HZ}}) \right] \\ + \varepsilon_{\text{HZ}}^{-2} \left[M_{\text{HZ}}^2 + N_{\text{R}} N_{\text{HZ}} \right] \end{array} \right) \quad (29)$$

式中第一项和第二项分别表示轻负载区和重负载区, 在固定 ε_{HZ} 和 ε_{LZ} 的情况下, 两项的最高次均为2, 即平方复杂度. 相比而言, FPTA算法的复杂度也可写作类似(29)的形式:

$$O(\varepsilon_{\text{HZ}}^{-2} (M^2 + N_{\text{R}} N)) \left(\begin{array}{l} \varepsilon_{\text{LZ}}^{-2} \left[(M - M_{\text{HZ}})^2 + N_{\text{R}} (N - N_{\text{HZ}}) \right] \\ + \varepsilon_{\text{HZ}}^{-2} \left[M_{\text{HZ}}^2 + N_{\text{R}} N_{\text{HZ}} \right] \\ + (\varepsilon_{\text{HZ}}^{-2} - \varepsilon_{\text{LZ}}^{-2}) \left[(M - M_{\text{HZ}})^2 + N_{\text{R}} (N - N_{\text{HZ}}) \right] \\ + 2\varepsilon_{\text{HZ}}^{-2} M_{\text{HZ}} (M - M_{\text{HZ}}) \end{array} \right) \quad (30)$$

其中四项均非负, 第一项和第二项与MCFSR完全一致. 由于轻负载区路由算法的迭代步长 ε_{LZ} 取的较大也

可取的很好的效果,因此 $\epsilon_{\text{HZ}}^{-2} \gg \epsilon_{\text{LZ}}^{-2}$. 又因重负载区范围较小, $M \gg M_{\text{HZ}}$, 因此第三项在式(30)中占比例很大,同时第四项也不可忽略. 由此可知 MCFSR 算法对比 FPTA 算法大大减小了时间复杂度. 复杂度的数值仿真结果在 4.2.2 中呈现. 在所给场景下, MCFSR 的复杂度约为 FPTA 的 1/7. 特别的,在重负载区扩大至全球范围的极限情况下, $M = M_{\text{HZ}}, N = N_{\text{HZ}}$, 式(29)会退化为式(28),即 FPTA 算法可以看作 MCFSR 算法的一种特殊情况.

4 仿真与讨论

4.1 仿真场景

为了评估提出算法的性能,这里将 MCFSR 算法应用于不同规模的 Walker-delta 型星座. 网络模型如第 2 节中所描述,星座参数如表 1 所示. 需要强调的是,算法可适用于任意倾角及相位因子的 Walker-delta 星座. 本文考虑了 4 种不同的星座规模,用于比较不同星座规模下的算法性能.

对于地面小区划分及业务模型,这里只给出星座为 24×24 时的地面业务分布模型,如图 3,其中每个小区内的数字代表某一品类业务的权重,这里取所有业务的权重分布都相同,且每个小区发送等量的各品类业务. 图中标号①~④为 4 个可能的地面站位置,其余星座规模更小的情况均为该模型的合并或插值. 在星座有 24 个轨道平面,每个轨道平面 24 颗卫星的极轨星座情况下,全球被分为 12×24 个小区,每个小区内都刚好包含一颗升轨卫星和一颗降轨卫星. 重负载区的划分方式不唯一,可根据星座规模、地面布站情况、地面分区情况等进行调整,图中的黑色框线区域给出了其中一种 6×8 的重负载区划分方式. 对于 MCFSR 算法,取轻负载区迭代步长 $\epsilon_{\text{LZ}} = 0.5$,重负载区迭代步长 $\epsilon_{\text{HZ}} = 0.2$. FPTA 算法全局迭代步长 $\epsilon = 0.2$.

表 1 星座参数

参数	数值			
轨道面数	20	24	28	32
每轨卫星数	20	24	28	32
轨道倾角	90°			
相位因子	0			
星间链路带宽(Gbps)	$C_{\text{ISL}} = 1$			
星地链路带宽(Gbps)	$C_{\text{U}} = C_{\text{D}} = 4$			

4.2 结果与讨论

4.2.1 算法最大吞吐量

在这部分中,本文就低轨巨星座多点回传问题,对 FPTA 算法以及本文提出的 MCFSR 算法进行最大吞吐量和时间复杂度的比较. 重负载区划分按照图 3 中的地理位置进行划分,划分中不满整个小区的向上取整.

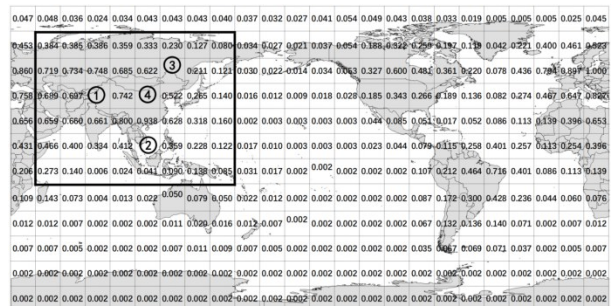


图 3 全球业务分部及接收站分布

为了不失一般性,这里考虑表 1 中的 4 种星座规模,且考虑①②、①~③、①~④三种地面站分布情况. 仿真结果如图 4 所示. 其中虚线为吞吐量的上界,由卫星回传地面站的总带宽决定.

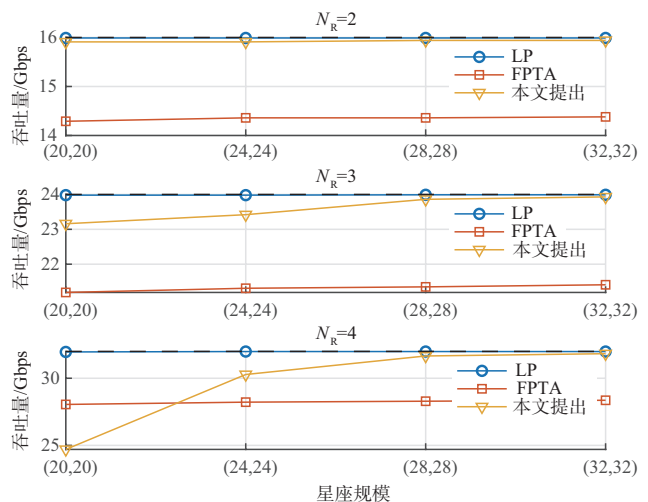


图 4 不同地面站分布下的算法最大容量对比

可以观察到,本文提出的 MCFSR 算法可达到的最大吞吐量随着星座规模的扩大不断接近吞吐量的上界,这是由于在重负载区地理大小不变的情况下,星座规模的增大使得重负载区内部可供业务交换的卫星和星间链路增多,因此总吞吐量得以不断接近最优值. 而当重负载区内接收站密度较大时,例如图 4 中 $N_r = 4$ 时星座规模为 (20,20) 的情况,重负载区内的负载压力较大,无法完全满足边缘到接收站的回传,因此吞吐量达不到极限值. 对比文献[13]中的 FPTA 算法,MCFSR 算法在图中所示的大部分情况都具有更优的吞吐量,仅在星座规模较小、且接收节点较多的情况下表现欠佳,这也说明了该算法对巨星座的适用性.

4.2.2 时间复杂度

本文对线性规划、FPTA 以及提出的 MCFSR 三种算法的时间复杂度进行对比. 取 $N_r = 4$,其余仿真条件与 4.2.1 中一致. 仿真显示了星座规模的变化带来的算法复杂度的变化趋势,纵坐标为对数表示,如图 5 所示.

可见,MCFSR算法的复杂度为三种算法中最低的,在当前仿真条件下约为线性规划算法的 $1/10^7$,为FPTA算法的 $1/7$.而MCFSR算法在最大吞吐量领先的情况下,运行时间仍然全面低于FPTA算法.并且随着接收站数量的增加,MCFSR算法的优势更加明显.

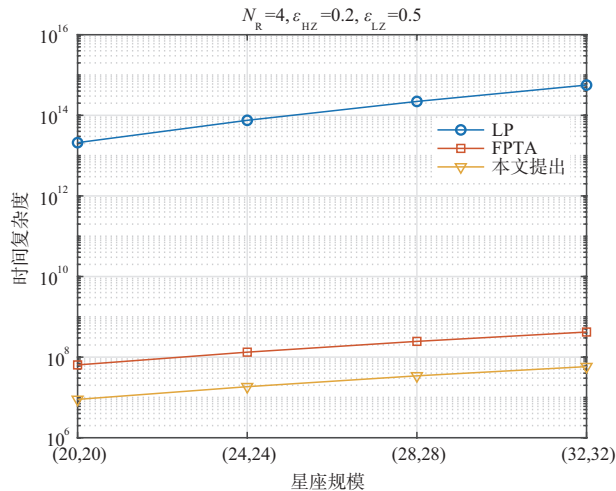


图5 不同算法的时间复杂度对比

4.2.3 地面站位置与重负载区大小的影响

在4.2.1中提到,重负载区内接收站的密度对总吞吐量有较大影响.因此在本节中分别探讨重负载区与地面站相对位置及重负载区的大小对吞吐量的影响.

在重负载区大小的仿真中,固定星座规模为 24×24 ,将地面站所在区域置于负载区的中心位置,依次改变重负载区的大小,仿真结果如图6(a)所示.当重负载区仅有 4×4 时,即区域刚好能覆盖所有地面站,每一种地面站分布下的吞吐量仅有约上界的 $1/2$.而随着重负载区的扩大,吞吐量基本恢复到接近上界的水平.值得注意的是,三种接收站分布的吞吐量在重负载区规模增大到 6×6 时同时增长到最大,而非图4中依次达到最大.这是由于此处的仿真只在同一个星座规模下扩大了重负载区,而几个接收站之间的节点和链路数量并没有增加.因此在接收站相对集中的区域,例如重负载区的中心区域,接收站的局部密度并没有随着负载区的扩大而减小.当负载区扩大时,吞吐量的瓶颈由外围链路转至接收站之间的链路,因此会同时增长至最大.

在相对位置的仿真中,星座规模不变,固定重负载区为 6×8 的矩形,与图3情况一致.在重负载区包含全部地面站的前提下,将重负载区由东向西依次移动,仿真结果如图6(b)所示.结果显示,无论是何种地面站分布,地面站距离重负载区边缘越远,最大吞吐量越接近上界.这同样是由于重负载区边框的存在影响了接收站周围的局部链路密度导致的.由此可见,在重负载区

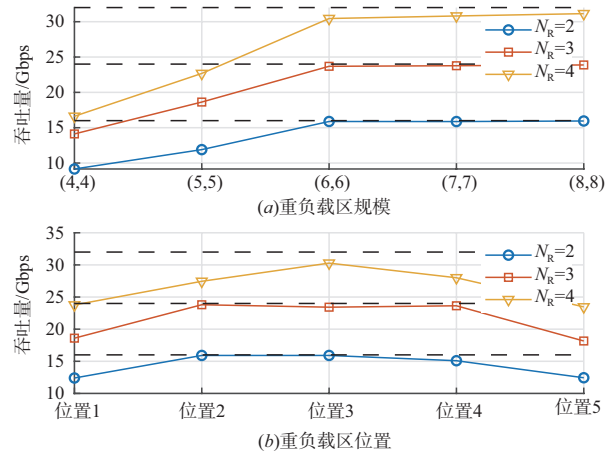


图6 重负载区大小与位置对吞吐量的影响

划分时应控制区域内地面站密度,并尽量使地面站位于区域中央,以达到最优的网络性能.

5 结论

本文研究了低轨巨星座多业务回传国土场景的负载均衡路由问题.针对现有的集中规划算法复杂度高的问题,本文提出了MCFSR算法.算法将网络分为轻、重负载区,并施以不同复杂度的多品类流算法进行规划,以降低算法复杂度.同时,对于MCFSR所采用的多品类流算法,本文提出了IFPTA算法,在计算复杂度保持不变的情况下使网络吞吐量更逼近最优值.仿真表明,MCFSR算法在不同地面站分布下,随着星座规模扩大,多业务回传的总吞吐量接近最优,且时间复杂度远低于其他同类算法.本文同时分析了重负载区的大小、与地面站的相对位置对吞吐量的影响,并给出了重负载区的划分建议.

参考文献

- [1] Di B, Song L, Li Y, et al. Ultra-dense LEO: Integration of satellite access networks into 5G and beyond[J]. IEEE Wireless Communications, 2019, 26(2): 62 – 69.
- [2] 宋海丰. 国外新兴低轨通信星座发展态势分析[J]. 国际太空, 2018,(5):17 – 22.
Song Hai-feng. Analysis on situation of foreign emerging LEO communication satellite constellations[J]. Space International, 2018,(5): 17 – 22. (in Chinese)
- [3] 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.
- [4] Zhang N, Zhang S, Yang P, et al. Software defined space-air-ground integrated vehicular networks: Challenges and solutions[J]. IEEE Communications Magazine, 2017, 55 (7): 101 – 109.

- [5] Albulet M. SpaceX non-geostationary satellite system: Technical information to supplement schedules-Attachment to fcc application sat-loa-20161115-00118[R]. Federal Commun Commission, Washington DC, USA, Tech Rep SAT-LOA-20161115-00118, 2016.
- [6] Del Portillo I, Cameron B G, Crawley E F. A technical comparison of three low earth orbit satellite constellation systems to provide global broadband[J]. Acta Astronautica, 2019, 159: 123 – 135.
- [7] Jun X, Lu W, Zhang G. Traffic modeling and simulation of broadband LEO satellite communication system[A]. IOP Conference Series: Materials Science and Engineering[C]. IOP Publishing, 2018, 452(4): 042082.
- [8] Jia Z, Sheng M, Li J, et al. Joint HAP access and LEO satellite backhaul in 6G: Matching game-based approaches[J]. IEEE Journal on Selected Areas in Communications, 2020, 39(4): 1147 – 1159.
- [9] Shmoys D B. Cut problems and their application to divide-and-conquer[J]. Approximation Algorithms for NP-Hard Problems, 1997: 192 – 235.
- [10] Shahrokhi F, Matula D W. The maximum concurrent flow problem[J]. Journal of the ACM (JACM), 1990, 37(2): 318 – 334.
- [11] Leighton T, Makedon F, Plotkin S, et al. Fast approximation algorithms for multicommodity flow problems[J]. Journal of Computer and System Sciences, 1995, 50(2): 228 – 243.
- [12] Garg N, Koenemann J. Faster and simpler algorithms for multicommodity flow and other fractional packing problems[J]. SIAM Journal on Computing, 2007, 37(2): 630 – 652.
- [13] Karakostas G. Faster approximation schemes for fractional multicommodity flow problems[J]. ACM Transactions on Algorithms (TALG), 2008, 4(1): 1 – 17.
- [14] Filsfil C, Nainar N K, Pignataro C, et al. The segment routing architecture[A]. 2015 IEEE Global Communications Conference (GLOBECOM)[C]. USA: IEEE, 2015. 1 – 6.
- [15] Davoli L, Veltri L, Ventre P L, et al. Traffic engineering with segment routing: SDN-based architectural design and open source implementation[A]. 2015 Fourth European Workshop on Software Defined Networks[C]. USA: IEEE, 2015. 111 – 112.
- [16] Bhatia R, Hao F, Kodialam M, et al. Optimized network traffic engineering using segment routing[A]. 2015 IEEE Conference on Computer Communications (INFOCOM) [C]. USA: IEEE, 2015. 657 – 665.
- [17] 黄建洋, 兰巨龙, 胡宇翔, 等. 一种基于分段路由的多路径流传输机制[J]. 电子学报, 2018, 46(6): 1488 – 1495.
Huang Jian-yang, Lan Ju-long, Hu Yu-xiang, et al. A segment routing based multipath flow transmission mechanism[J]. Acta Electronica Sinica, 2018, 46(6): 1488 – 1495. (in Chinese)
- [18] Liu W, Tao Y, Liu L. Load-balancing routing algorithm based on segment routing for traffic return in LEO satellite networks[J]. IEEE Access, 2019, 7: 112044 – 112053.
- [19] Boyd Stephen, Vandenberghe Lieven. Convex optimization[J]. IEEE Transactions on Automatic Control, 2006, 51(11): 1859 – 1859.

作者简介



王宁远 男, 1993年生, 安徽安庆人, 博士研究生. 2018年毕业于北京航空航天大学航空宇航科学与技术专业, 同年进入中国空间技术研究院攻读博士学位, 研究方向为通信卫星星座网络、星座资源管理等.

E-mail: ningyuan.wang@foxmail.com



刘亮(通信作者) 男, 1986年生, 陕西西安人, 博士, 高级工程师. 现任职于中国空间技术研究院通信与导航卫星总体部, 主要研究方向为网络协议、编码理论及卫星通信.

E-mail: liuliang1945@buaa.edu.cn